Sextant: Towards Ubiquitous Indoor Localization Service by Photo-Taking of the Environment

Ruipeng Gao, Yang Tian, Fan Ye, Guojie Luo, *Member, IEEE*, Kaigui Bian, *Member, IEEE*, Yizhou Wang, Tao Wang, *Senior Member, IEEE*, and Xiaoming Li, *Senior Member, IEEE*

Abstract—Mainstream indoor localization technologies rely on RF signatures that require extensive human efforts to measure and periodically recalibrate signatures. The progress to ubiquitous localization remains slow. In this study, we explore Sextant, an alternative approach that leverages environmental reference objects such as store logos. A user uses a smartphone to obtain relative position measurements to such static reference objects for the system to triangulate the user location. Sextant leverages image matching algorithms to automatically identify the chosen reference objects by photo-taking, and we propose two methods to systematically address image matching mistakes that cause large localization errors. We formulate the benchmark image selection problem, prove its NP-completeness, and propose a heuristic algorithm to solve it. We also propose a couple of geographical constraints to further infer unknown reference objects. To enable fast deployment, we propose a lightweight site survey method for service providers to quickly estimate the coordinates of reference objects. Extensive experiments have shown that Sextant prototype achieves 2-5 m accuracy at 80-percentile, comparable to the industry state-of-the-art, while covering a 150×75 m mall and 300×200 m train station requires a one time investment of only 2-3 man-hours from service providers.

Index Terms—Smartphone indoor localization, triangulation method, lightweight site survey, benchmark image selection

1 Introduction

Indoor localization [1], [2], [3] is the basis for novel features in various location based applications. Despite more than a decade of research, localization service is not yet pervasive indoors. The latest industry state-of-the-art, Google Indoor Maps [4], covers about 10,000 locations in 18 countries, which are only a fraction of the millions of shopping centers, airports, train stations, museums, hospitals and retail stores on the planet.

One major obstacle behind the sporadic availability, is that current mainstream indoor localization technologies largely rely on radio frequency (RF) signatures from certain IT infrastructure (e.g., WiFi access points [1], [2] and cellular towers [5]). Additionally, obtaining the *signature map* usually requires dedicated labor efforts to measure the signal parameters at fine grained grid points. Because they are susceptible to intrinsic fluctuations and external disturbances, the signatures have to be re-calibrated periodically to ensure accuracy. Some recent research [6], [7], [8] has started to leverage crowd-sourcing to reduce site survey efforts, but incentives are still lacking for wide user adoption. Thus the progress is inevitably slow.

Localization also requires more than mere network connectivity. For example, six strongest towers are usually needed [5] for GSM localization, but the obstruction of walls may deprive many places signals from enough number of towers. WiFi localization also requires enough number of access points in signatures to effectively distinguish different locations. Thus places with network connectivity may not always be conducive to localization.

In this paper, we explore an alternative approach that has comparable performance but without relying on the RF signature. Specifically, we leverage environmental *physical features*, such as logos of stores or paintings on the walls, as *reference objects*. Users use the smartphone to measure their relative positions to physical features, and the coordinates of these reference objects are used to compute user locations. This has a few advantages: 1) Physical features are part of and abundant in the environment; they do not require dedicated deployment and maintenance efforts like IT infrastructure; 2) They seldom move and usually remain static over long periods of time. They are not affected by and thus impervious to electromagnetic disturbances from microwaves, cordless phones or wireless cameras. Once measured, their coordinates do not change, thus eliminating the need for periodic re-calibration.

The realization of such benefits, however, turns out to be a non-trivial journey. First, we need to identify a suitable form of relative position that can be effectively measured by smartphones with accuracies favorable for localization. Second, the abundance of physical features is not always a blessing: users need some guidelines to decide which ones to measure for smaller localization errors. Third, to enable fast deployment, service providers have to obtain the coordinates of reference objects in a new environment with low human efforts. Finally, the system has to know which reference objects are selected by users. Relying on explicit user input can be a nonstarter. Ideally,

- *R. Gao, Y. Tian, G. Luo, K. Bian, Y. Wang, T. Wang, and X. Li are with the EECS School, Peking University, Beijing 100871, China. E-mail: {gaoruipeng, tianyangty, gluo, bkg, yizhou.wang, wangtao, lxm}@pku.edu.cn.*
- *F. Ye is with the ECE Department, Stony Brook University, Stony Brook, NY 11794. E-mail: fan.ye@stonybrook.edu.*

Manuscript received 23 Sept. 2014; accepted 19 Mar. 2015. Date of publication 31 Mar. 2015; date of current version 4 Jan. 2016.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TMC.2015.2418205

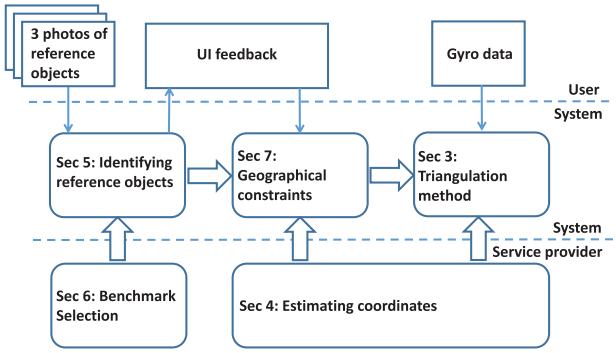


Fig. 1. Relationship among different components in Sextant prototype. Each component uses images or gyro data and output from the previous component.

the system should gain such input with as little efforts from users as possible.

Our investigation leads us to the localization method of *Sextant*.[1] In the prototype we build on smartphones [9], the user takes a picture for each of three nearby reference objects one by one. The photos are matched against the benchmark images of each reference object, to identify which reference objects are selected; thus their coordinates, together with relative position measurements, are used to triangulate the user's location.

The main source of inaccuracy in Sextant is the imperfection of image matching algorithms. Their accuracy is affected significantly by which images are used as benchmark for reference objects. An image taken from extreme angles or distances may lead to significant matching errors. Then the system does not have the coordinates of the correct reference objects for localization. Had the matching been perfect, Sextant could have achieved much higher accuracy.

Thus we also propose two methods to systematically address the image matching errors [10]. First, we study how to select the best images as the benchmark when multiple are available for each reference object. The purpose is to minimize "cross matching" where one reference object's photo is incorrectly matched to the benchmark of another. Second, we impose additional constraints to correct wrong matching results. This is based on the observation that those reference objects chosen by the user are usually close to each other. Given even one correct match, the unknown reference objects can be inferred with much higher probability from a nearby range. Prototype experiments in large indoor environments have shown promising results, with 80-percentile accuracy at 2-5 m, comparable to Google Indoor Maps($\sim$7 m).

Our Sextant prototype is described in Fig. 1. Service provider selects certain benchmark images for each reference object, and estimates their coordinates; photos and gyro data from user smartphones are used as inputs in Sextant system. Sextant leverages image matching techniques to identify those chosen reference objects, infers the unknown ones after user feedback, and then computes user location using a triangulation method.

1. Sextant is commonly used by sailors to determine their longitude/latitude by measuring the angle between visible objects, usually celestial ones like the Sun.

We make the following contributions in this work:

- We identify a form of relative position measurement and its respective triangulation method suitable for modern smartphone hardware. We also analyze the localization errors and devise a simple rule of reference object selection to minimize errors.
- We leverage image matching techniques to identify the chosen reference objects, formulate their benchmark selection as a combinatorial optimization problem and prove its NP-completeness. We then propose and evaluate a heuristic algorithm based on iterative perturbation for realistic solutions.
- We propose a lightweight site survey method such that a service provider can quickly obtain the coordinates of reference objects in a previously unmapped environment with reasonable accuracy ($\sim$1 m at 80-percentile). Our experiments find that it takes a *one time* investment of 2-3 man-hours to survey a 150 $\times$ 75 m shopping mall or a 300 $\times$ 200 m train station.
- We propose several geographical constraints that help make much informed decision about the identities of incorrectly matched reference objects. Together they greatly improve the inference accuracy of the system.
- We build a Sextant prototype, and conduct extensive experiments in large complex indoor environments that shows 2-5 m accuracy at 80-percentile using the estimated coordinates, which are comparable to the industry state-of-the-art.

In the rest of the paper, we study the forms of relative positions and the accuracies of suitable sensors (Section 2). We then describe the localization operations, study the optimal reference object selection and demonstrate the feasibility of the operations as a localization primitive (Section 3). We propose a lightweight approach for estimating the coordinates in an unmapped environment (Section 4), describe the automatic recognition of chosen reference objects using image matching algorithm (Section 5), address the benchmark selection problem for reference objects (Section 6), and use geographical constraints to lower localization errors caused by image matching mistakes (Section 7). We discuss our limits (Section 8) and review related work (Section 9), then conclude the paper (Section 10).

2 Relative Position Measurement

Relative positions include the distance and orientation between the user and the reference object. Although smartphones can measure their pairwise distance easily [11], they are not equipped with a sensor to directly measure the distance to a physical object. While orientation can take two forms, *absolute and relative angles*, both of which can be used to triangulate the user.

Absolute angle based localization. As shown in Fig. 5, given the coordinates of two reference objects R_1, R_2 and the absolute angle α, β (w.r.t. an axis in the coordinate system), the user P is at the intersection of two rays from R_1, R_2.

Relative angle based localization. Given the coordinates of two reference objects R_2, R_3 and the relative angle α (i.e., $\angle R_2PR_3$) between them, the edge R_2R_3 and α can uniquely determine a circle where R_2R_3 is the subtense and α is the

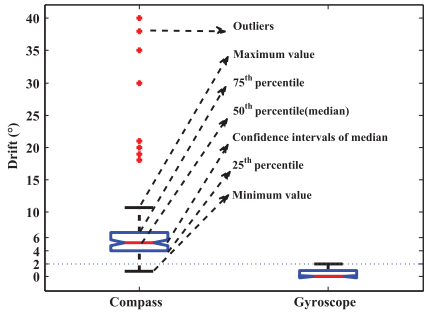


Fig. 2. Compass/gyroscope drifts (in degrees °) when moving the phone along a straight line.

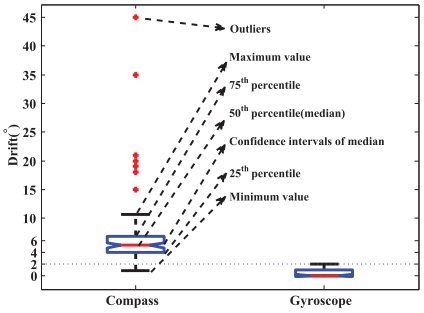


Fig. 3. Compass/gyroscope drifts (in degree °) when the phone is placed on radial lines.

interior angle (see Fig. 6). The user is located along the arc of the circle. With three such reference objects (R_1, R_2, R_3) and two relative angles (α, β), two circles are determined and the user P is at the intersection of the circles.

Modern smartphones are usually equipped with a digital compass that gives the absolute angle with respect to geographic north, and a gyroscope that measures the rotated angle of the phone between two positions.[2] Although there has been some reports [8] on the error of the compass, it is not immediately clear to us whether the accuracies of the compass and gyroscope are consistent under various factors. To this end, we conduct an experiment using an iPhone 4 in a $20.4\,\mathrm{m} \times 6.6\,\mathrm{m}$ office area where 50 test locations are evenly distributed.

Moving the phone along a straight line. When the phone is moved along a one-meter straight line at 25 cm steplengths (shown in Fig. 7a), the compass or gyroscope readings are expected to remain the same. Thus the *drift*, the difference of two consecutive sensor readings, should be close to zero. From Fig. 2, we can see that the compass has quite significant drifts (e.g., 6 degree at 75-percentile); it also has large outliers (e.g., 18-40 degree) due to electromagnetic disturbances such as nearby electric wires. However, the gyroscope has consistently small (e.g., maximum at 2 degree) drifts.

Rotating the phone on radial lines. Next we align the phone along radial lines separated by 30 degree in a semi-circle (shown in Fig. 7b). We define the *measured angle* (expected to be close to 30 degree) between two adjacent radial lines as the difference between two respective sensor readings. The *drift* is how much the measured angle deviates from 30 degree. From Fig. 3, we make similar observations to those of Fig. 2. The gyroscope still has consistently small drifts while the compass is unsuitable for accurate angle measurements.

Time, building, orientation and rotation speed. We repeat the second experiment for the gyroscope at 10 AM, 2 PM and 10 PM, and in rooms of three buildings (classroom, lab, indoor stadium). From Fig. 4, we find similar small drifts (~ 1 degree). We place the phone at a test location, and point the phone to four vertically-intersected directions, east, south, west, and north (as shown in Fig. 7a). Then we rotate the phone by $\pm d$ degree where $-d$ degree is a clockwise and $+d$ degree a counter-clockwise rotation, and $d = 15, 30, 45$. This is repeated three times. We find that the error is at most 1 degree and more than half of them have less than 1 degree errors. We place the phone at a fixed location, and rotate the phone at two different speeds, finishing a 10 degree rotation in 2 and 5 seconds. This is intended to see how it behaves under different user operations. Again we find consistently small drift in both cases.

From the above study, we find that the compass has quite large drifts caused by ferromagnetic materials (e.g., magnets, floor tiles, decorative marble) and electrical objects (e.g., electric appliances, electric wires under the floor), whose disturbances are impossible to eliminate. Thus we conclude that the gyroscope has consistently high level of accuracy, and decide to use the relative angle based localization as shown in Fig. 6.

3 Triangulation Method

3.1 User Operations and Location Computation

Given the triangulation method in Sextant, the user needs to measure two relative angles between three reference objects. He can stand at his current location, spin his body and arm to point the phone to these reference objects one by one (as illustrated in Fig. 8). Given the two angles α, β and the coordinates of the three reference objects (as illustrated in Fig. 6), the user location can be computed as:[3]

$$\begin{cases} x = x_0 \frac{x_3 - x_2}{a} - y_0 \frac{y_3 - y_2}{a} + x_2, \\ y = x_0 \frac{y_3 - y_2}{a} + y_0 \frac{x_3 - x_2}{a} + y_2, \end{cases} \tag{1}$$

where

$$\begin{cases} a &= \sqrt{(x_3 - x_2)^2 + (y_3 - y_2)^2}, \\ b &= \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}, \\ x_0 &= \frac{ab[\sin(\beta+\theta)\cot\alpha + \cos(\beta+\theta)][a\sin\beta\cot\alpha + b\cos(\beta+\theta)]}{[b\sin(\beta+\theta) - a\sin\beta]^2 + [b\cos(\beta+\theta) + a\sin\beta\cot\alpha]^2}, \\ y_0 &= \frac{ab[\sin(\beta+\theta)\cot\alpha + \cos(\beta+\theta)][b\sin(\beta+\theta) - a\sin\beta]}{[b\sin(\beta+\theta) - a\sin\beta]^2 + [b\cos(\beta+\theta) + a\sin\beta\cot\alpha]^2}, \\ \theta &= \arccos\left[\frac{(x_3 - x_2)(x_1 - x_2) + (y_3 - y_2)(x_3 - x_2)}{ab}\right]. \end{cases} \tag{2}$$

For the above operations to become a reliable localization primitive, we need to address localization errors from

2. To be exact, the gyroscope measures the rotation rates of the phone in radian/sec around its x-, y-, and z-axes. The angle is obtained by integrating the rotation rate against time between the two positions.

3. Because an object (e.g., a door) might be large, pointing to different parts (e.g., left versus right edge) can incur different angle readings. We impose a *default convention* of always pointing to the horizontal center of an object.

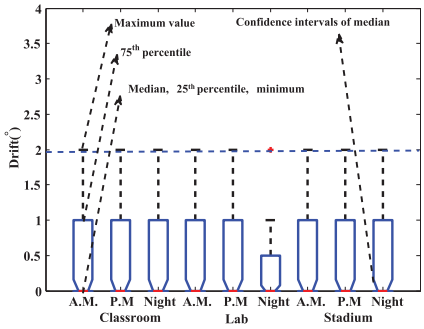


Fig. 4. Gyroscope drifts (in degree °) versus time of the day and building types.

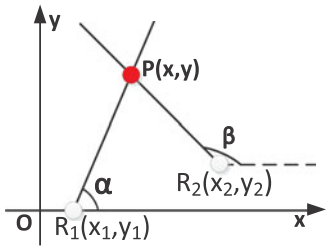


Fig. 5. Absolute angle based.

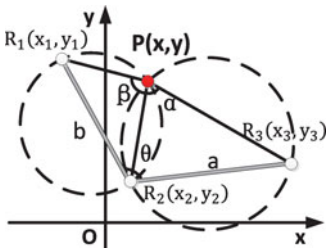


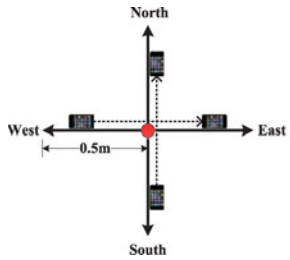
Fig. 6. Relative angle based.

two more sources other than angle measurements (studied in Section 2): 1) We use obvious environmental features such as store logos as reference points. In a complex environment most locations have multiple of them around. The user needs to select three that lead to smaller localization errors. 2) The error introduced by imperfections in user pointing (e.g., various wrist/arm/foot gestures) and device hardware. We study these two issues in the next two sections.

3.2 Criteria for Users to Choose Reference Objects

Impact of rotated angle drift. To understand the impact of the drift on localization errors, we conduct a numerical simulation for an a m$\times b$ m rectangle area with four corners as reference objects. We repeat the localization computation at a grid of test locations at $(m\delta, n\delta)$ where δ is the grid cell size, and $m \in [1, a/\delta]$, $n \in [1, b/\delta]$. Although this is a rather simplified case, we want to find guidelines for combinations of reference objects that lead to higher localization accuracy.

We use Skewness/Kurtosis tests (a.k.a. SK-test) [12] on the gyroscope readings and find that the drift conforms to normal distribution. The mean is close to zero, and the 95 percent confidence interval is about ± 6 degree. Thus we use ± 6 degree to evaluate worst-case localization errors in the following simulation.



(a) The phone is moved along a straight line, and the dot represents a test location;

(b) The phone is placed on the radial lines of a semi-circle, and the dot at the center represents a test location.

Fig. 7. Two experimental scenarios for angle measurements using smartphones.

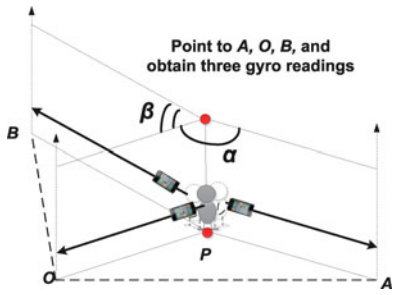
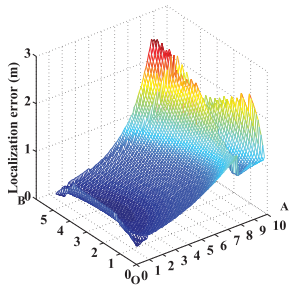
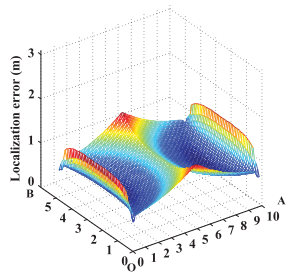


Fig. 8. The main steps of user operations: three reference objects are chosen by the user; two rotated angles α and β are measured by the phone gyroscope. Assuming the coordinates of O, A and B are known, the user's location can be calculated.



(a) When three reference objects A, O, B are always chosen;



(b) When the closest reference object rule is used.

Fig. 9. Average localization error when $\Delta\alpha, \Delta\beta = 0, \pm 6$ degree.

Choose a fixed set of reference objects. We first study a simple rule: always choose a fixed set of three reference objects (e.g., corner A, O, B). We set the area size $a = 10, b = 5$, grid size $\delta = 0.2$, then vary the drift as $\Delta\alpha = 0, \pm 6$ degree, $\Delta\beta = 0, \pm 6$ degree, and show the average localization error of the eight combinations of $\Delta\alpha$ and $\Delta\beta$ (except $\Delta\alpha = \Delta\beta = 0$) in Fig. 9a as a 3d plot. We observe that the localization error is small (e.g., <1 m) when the test location is close to the center reference object O; it becomes much larger when the location moves farther away from object O. We observe similar patterns with areas of other sizes and drifts of other values.

Small acute angles lead to larger errors. Intuitively, a distant test location tends to have a small acute angle between two reference objects. The distant location can have a larger displacement while still incurring a small angle drift. As illustrated in Fig. 10, the same error δ is added to two angle measurements β_1 and β_2. The localization error is roughly how much the user location P can

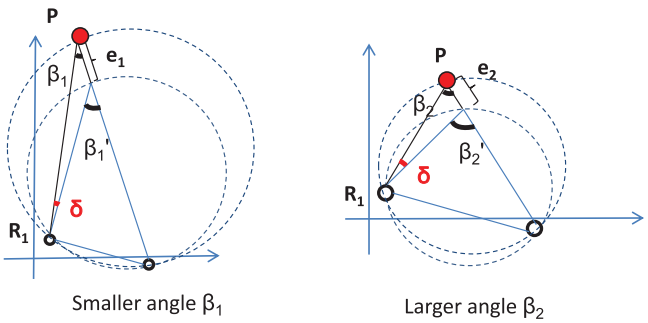


Fig. 10. The same angle drift δ on a smaller angle β_1 causes a larger localization error e_1 than that on a larger angle β_2, because the longer $\overline{R_1P}$ distance leads to more displacement.

move when the radial line R_1P rotates angle δ around center R_1. Over the same rotated angle δ, a larger radius leads to longer displacement of P, thus larger localization error. We have conducted further tests and validated the intuition. This is similar to GDOP in GPS localization [13].

Closest reference object rule. From the above observation, we come up with a simple rule: choose the closest reference object and its left, right adjacent ones as three reference objects. Such closer objects lead to larger angles, thus avoiding the small acute angles that cause large localization errors. We repeat the simulation using this simple rule in the same rectangle area. Fig. 9b shows that the average localization error is no more than 1 m at all test locations. This clearly demonstrates the effectiveness of this simple rule. Simulations of other area sizes also confirm our discovery.

3.3 Robustness of the Localization Primitive

We further investigate the impact of a number of practical factors on localization error. We find that all of them can be addressed and the operations described in Section 3.1 can be made a robust primitive for localization.

Impact of pointing gestures. To study the error caused by various user pointing gestures, we recruit ten volunteers to point using three types of gestures with an iPhone4. The first two types require a user to stand still and only spin his arm or wrist to point to objects; the third requires a user to spin his body and arm together.

Fig. 11a shows the angle drift from each type of gesture. By only twisting the wrist, users make relatively large errors ($\sim$8 degree), while spinning body and arm leads to the least error ($\sim$2 degree). Thus we recommend the third gesture for pointing.

Impact of the phone's altitude. While spinning the arm, a user may not be able to keep the phone in a horizontal

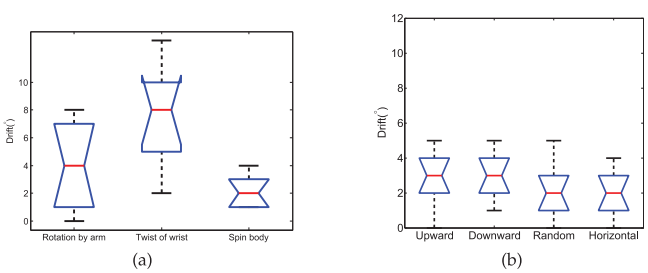


Fig. 11. The rotated angle drift under: (a) various types of users' pointing gestures, and (b) different pointing altitude.

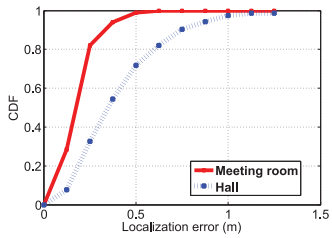


Fig. 12. The CDF of error distribution for two rectangle rooms.

plane. He may unwittingly raise or lower the phone. Thus the difference between two gyroscope readings may not accurately reflect the horizontal rotated angle. To avoid such inaccuracies, we use the horizontal component of the gyroscope readings to accurately measure the angle in the horizontal plane.

We recruit four test groups of users to point the phone with different altitude trajectories: 1/2) raise/lower the phone with a random upwards/downwards altitude; 3) randomly raise and lower the phone during rotation; and 4) absolutely horizontal using a water level device. From Fig. 11b, we observe that the average angle drift in the two groups of "upwards" and "downwards" is just 1 degree more than those in the other two groups, owing to our method of calculation using the horizontal component. In the following experiments we also find that the pointing altitude trajectories have little impact on localization errors. We ask the same four test groups of users to repeat the experiments in the meeting room mentioned in Fig. 12, the 90-percentile accuracy is below 0.5 m for all groups.

Impact of the area size, shape and reference object width. We conduct experiments in two rectangle areas (a 6.6 m $\times$ 4.2 m meeting room and a 14.4 m $\times$ 13.2 m hospital hall). We use the closest reference object rule, repeat the experiment three times at each test location on a grid of $\sim$1 m cell size. The CDFs of average errors are shown in Fig. 12. We find that the 80-percentile accuracy is around 0.2 and 0.6 m, respectively. Due to the linear scaling, the larger hall has slightly larger errors.

We test in a polygon room (roughly 7.6 m $\times$ 5.7 m) and find similar results (e.g., 0.7 m at 90-percentile). We also test in two large outdoor areas of 30 m$\times$30 m and 20 m $\times$ 40 m sizes. The 80-percentile error is $\sim$1 m and maximum at 1.5 m, slightly larger than that of indoor environments because it scales to the area size. Finally we try reference objects of some widths (e.g., 1 m wide posters), and find that when users aim at the center of reference objects during pointing, the accuracy is not affected much ($\sim$0.5 m for 90-percentile). The above shows that the pointing primitive's accuracies are not affected much by the size, shape of the enclosing area and widths of reference objects.

Impact of user efforts. How carefully the user points to reference objects inevitably influences the accuracy of angle measurements. We employ three groups of users to evaluate the impact of user efforts: "normal" users use the closest reference object rule and point with certain care; "savvy" users pay more attention to measure the angles very carefully; while "impatient" users tends to finish the operations quickly and cursorily.

Fig. 13 shows the CDF results in the meeting room. We make several observations: a savvy user obtains the best

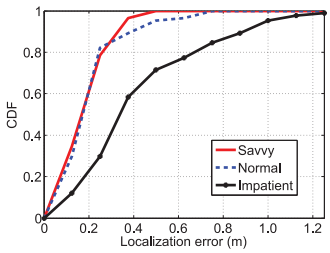


Fig. 13. The CDF of error distribution by different types of users.

accuracy (e.g., $\sim$0.3 m for 90-percentile); a normal user can achieve comparable accuracy; and an impatient user has lower but still reasonable accuracy with the closest reference object rule (e.g., 0.9 m at 90-percentile). These show that: 1) The pointing primitive can achieve reasonable accuracy with various degrees of use efforts; and 2) the closest reference object is an effective rule-of-thumb. We repeat the same experiments in the hall and have similar observations with that in the meeting room.

Impact of mobile device hardware. Gyroscope in different phones have varying qualities. We pick four popular devices (iPhone4, iTouch4, Samsung i9100, Samsung i9100g) to compare their performance. Fig. 14a shows that iPhone4, iTouch4 and i9100g almost have the same expected performance at a high level of accuracy (e.g., ~ 0.4 m at 90-percentile). However, i9100 shows the worst results (over 1.2 m).

We place the i9100 phone at a static location and record the readings once the gyroscope is turned on (at time 0 in Fig. 14b). We find the value declines at the very beginning, and then starts increasing (as shown in Fig. 14b). This is caused by the relatively lower quality of the STMicroelectroinics K3G gyroscope in i9100. To compensate for such intrinsic drifts, we use curve fitting methods to derive equations that characterize the variations over time to calibrate the gyroscope reading. We then repeat the experiments and the results ("Adjusted i9100" curve in Fig. 14a) show that after calibration it has accuracy comparable to the other three devices. For the other devices i9100g, iPhone4, iTouch4, same experiments are repeated and the curves tend to be flat horizontal lines, showing little drift over time.

From the above study, we conclude that the pointing operations can be made a robust localization primitive provided that the user follows the guidelines with certain care. In the next two sections, we investigate how a service provider can quickly obtain the coordinates of reference objects, and how the system can gain input of which reference objects the user has chosen.

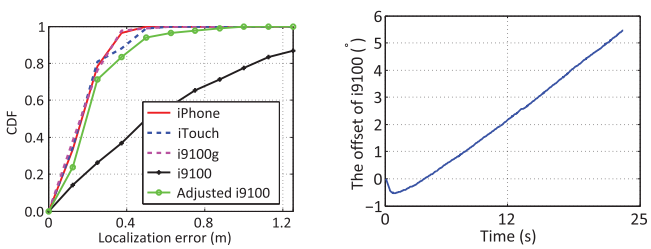


Fig. 14. Experiments using different types of devices in the meeting room. (a) The CDF of error distribution, and (b) Angle drift vs. time for i9100.

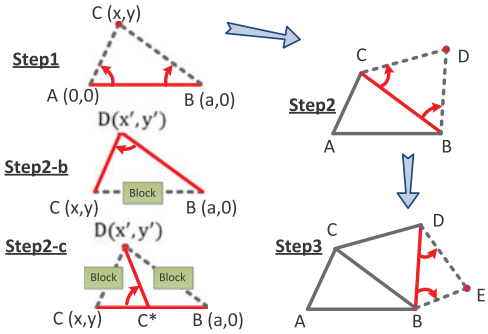


Fig. 15. Procedure to estimate the coordinates of reference objects.

4 Site Survey for Reference Objects Coordinates

Sextant needs the coordinates of reference objects to compute user location. The most straightforward method is to manually measure the distances, thus coordinates directly. Although this is a one-time investment because reference objects do not move, it still consumes time when there are many of them. In this section, we present a method for a service provider to significantly reduce the human effort.

4.1 Location Estimation in Unmapped Environments

In an unmapped environment, two workers[4] of a service provider first choose two *pair-wise visible* reference objects, say A and B, called *starting pairs* (step1 in Fig. 15). They each stand at A and B, then measure the distance a between them (e.g., by counting floor tiles, using a tape measure or techniques such as BeepBeep [11]). We can set a coordinate system with A at the origin $(0, 0)$ and B at $(a, 0)$. We call objects A and B as *positioned* objects.

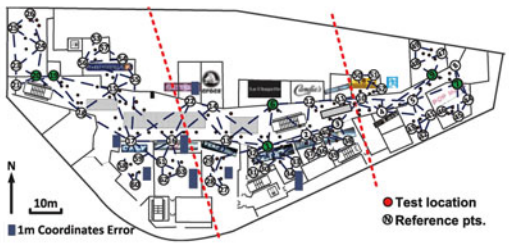
Then, the workers select a third *un-positioned* object C and determine its coordinates (x, y). When C is visible from A and B , the worker at A points the phone to B, and then C to measure $\angle BAC$. Similarly, the other worker can measure $\angle ABC$. The two angles $\alpha = \angle BAC$ and $\beta = \angle ABC$ can be used to calculate the coordinates of C: $x = (a \tan \beta)/(\tan \alpha + \tan \beta)$, and $y = (a \tan \alpha \tan \beta)/(\tan \alpha + \tan \beta)$.

The positioned object C together with A and B form a triangle, and the distance $\overline{AC}$ (or $\overline{BC}$) can be easily derived using the estimated coordinates of C. The worker at A can then move to C, and repeat similar processes to locate additional objects D, E (steps 2 and 3 in Fig. 15), and so on. The coordinates of each additional positioned object can be uniquely determined in this coordinate system.

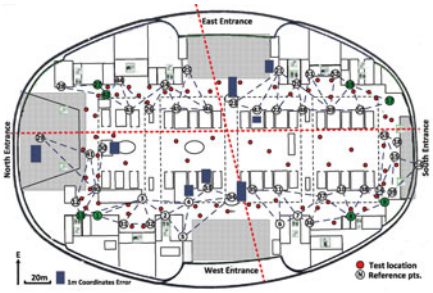
Blocked positioned objects. During the process when the direct line of sight between B and C is blocked (step2-b in Fig. 15), one of $\overline{BD}$, $\overline{CD}$ plus angle $\angle BDC$ are measured, together with $\overline{BC}$ (known already), the coordinates of D can be determined by the law of sines.

Blocked unpositioned objects. When an unpositioned object D is blocked from both B and C (step2-c in Fig. 15), one worker has to move along the line between C and B to find an appropriate location C^* where object D is visible. They measure distance $\overline{CC^*}$, the angle $\gamma = \angle CC^*D$, and $\overline{C^*D}$ to locate D relative

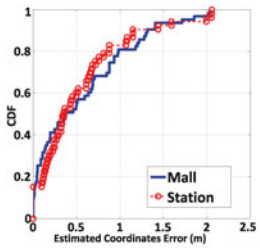
4. The procedure can be conducted by one worker with more walking, or multiple workers in parallel.



(a) The floor map of a mall dissected in three sections each with a starting pair, in total 63 reference objects and 108 test locations.



(b) The floor map of a train station dissected in four sections each with a starting pair, in total 53 reference objects and 46 test locations.



(c) Errors in estimated coordinates for a mall and train station.

Fig. 16. Floor map of a mall (a) and station (b), as well as their estimated coordinates errors (c). The vertical bars on (a) and (b) show the errors in estimated coordinates; those <1 m are not shown.

to C, thus eventually its coordinates. We omit the case when D is blocked from only one of B, C, which is similar to step2-b.

New starting pairs to control the error accumulation. One problem arises from such hop-by-hop estimation: the coordinates of a new object may contain error; when they are used to position another object, the error may grow. To control such accumulation, a simple method is to use a new starting pair after a few hops to reset the error back to zero.

4.2 Experiments on Site Survey

We conduct experiments in two large indoor environments, a 150×75 m shopping mall (Fig. 16a) and a 300×200 m train station (Fig. 16b).

Accuracy. When only one starting pair is used (reference points {① ⑥} in Fig. 16a and {① ⑬} in Fig. 16b, shown in green or slightly darker color), errors are small (<2 m) up to $4 \sim 6$ hops away, beyond which they quickly grow to more than 12 m. Obviously such large errors are not acceptable. After we add 2, 3 more starting pairs in these two environments ({⑦ ⑨} and {⑲ ⑳} in the mall, {㉗ ㊷}, {⑰ ⑲} and {⑧ ⑨} in the station), the 80-percentile errors are within 1 m, while the maximum about 2 m (Fig. 16c). They eventually lead to satisfactory localization accuracy (Section 5.2).

Human efforts. In the mall each of the 63 reference objects takes about 2 minutes to measure the angle(s) and/or distance(s); in the station each of the 53 objects takes about 3 minutes due to longer walking distances. In total they cost 2, 2.6 man-hours. Assuming WiFi signatures are measured 2 m apart and each location takes 10s, excluding inaccessible areas 5,200 m^2 and 23,700 m^2 areas need to be covered, resulting in 3.6, 16.5 man-hours. Thus the cost is roughly 16-55 percent that of WiFi. Note that over long time WiFi incurs periodic re-calibration costs each of similar amounts, while we pay only a one-time effort.

If brute-force measurements are used, each reference point takes 50 percent more time when regular floor tiles are available to count the coordinates; otherwise using a tape measure can triple the time. Although the quantifications are quite rough, they show that our site survey method can significantly reduce the human efforts compared to those of brute-force or WiFi.

5 Identifying Chosen Reference Objects

The Sextant system has to know which reference objects are selected by the user. However, it is impractical to require every user to explicitly tell the system about her/his choice. Thus how to identify chosen reference objects with less user efforts becomes a quite challenging problem in a complex environment with many reference objects.

We explore image matching algorithms to handle this issue. The user takes one photo (i.e., test image) for each of the three chosen reference objects one by one, which are matched with benchmark images to identify the corresponding reference objects. Nevertheless, we find that the matching algorithms make wrong identifications in certain situations. Next we will explain how we use the algorithms and classify error situations in this section. We also address matching errors with some heuristic algorithms in Section 6 and Section 7.

5.1 System Architecture and Work Flow

We have prototyped our Sextant system consisting of a smartphone for gyroscope data and image acquisition, a back-end server for image matching against a collection of benchmark images of reference objects (taken by a service provider during site survey).

Image capture via finger taps. To accommodate test images taken from different angles, we take three benchmark images for each reference object. The user uses the same spin operations. He taps the phone's screen to take a test image when a chosen reference object is centered on the camera. The tapping also triggers the capture of gyroscope readings. The test image is immediately sent to the server as the user continues for the next reference object.

Image matching and ranking. We examine two most popular image feature vector extraction algorithms, Scale Invariant Feature Transform (SIFT) [14] and Speeded Up Robust Features (SURF) [15]. Comparison [15] has shown that SURF is much faster while achieving comparable accuracy to SIFT. Thus we decide to use SURF in the prototype. Meanwhile, we use the same procedure used in [15] to rank benchmark images based on the number of matched feature vectors. We apply RANdom SAmple Consensus (RANSAC) [16] that uses the relative position constraints among feature vectors to detect and filter wrong feature vector matches.

For each test image, the server ranks the reference objects in descending order of the matching metric, the number of matched feature vectors, then returns this ranked list of [ID: matching metric value] tuples to the phone. The phone presents the results as a 4×3 thumbnail matrix (Fig. 17), with the top row showing the three

Fig. 17. The UI presented to the user for correction of image matching results. The top row are the three test images taken by the user, below each are the top 3 matched reference objects. The user can denote the correct match by tapping the thumbnail images.

test images, below each is a column of three best matched reference objects. By default the top match is highlighted. The user can tap the correct one if the top match is wrong. Then the user taps the 'confirm' button, and the phone computes the user location based on the corrected matching results and the angles. If none of the top 3 match is correct, the user taps the test image before proceeding with 'confirm'. The phone applies a heuristic that takes the feedbacks and the ranked list to search for a better match, and displays the final localization result.

Online and offline modes. Image matching algorithms inevitably make mistakes. Multiple benchmark images taken from different angles of a reference object improves accuracy significantly. However, more benchmarks lead to higher computing overhead. Thus in Sextant the number of benchmark images for each reference object is limited to a small number. When many candidate images are available, which images to select as benchmarks to match incoming test images greatly impact the matching accuracy. Thus we should select the subset of images leading to the best matching accuracy.

In Sextant depending on whether there is network connectivity, the phone can work in online or offline modes. Due to the complexity in image matching algorithms, the preferred location for matching computation is on a backend server. This is when the phone has network connectivity and can upload test images to the server to identify chosen reference objects. This is the online mode.

It is not uncommon that many locations do not have network connectivity due to the lack of WiFi APs or strong enough cellular signals. Sextant can still work if the computation is done locally. A couple of challenges have to be addressed: 1) The phone must have enough storage to store the benchmark images of reference objects. In reality we found this is not a problem, and they can be downloaded on demand before the user enters the environment while there is still network connectivity. 2) To reduce the latency, each reference object has to use less, ideally only one benchmark image. Nevertheless we have to provide enough matching accuracy. Thus the benchmark must be selected carefully to maximize correctness. This is what we address in the offline mode in Sextant.

TABLE 1
Image Matching Accuracy in Offline Mode

Top M Results	Mall	Station
Top 1	90.3%	88.2%
Top 2	95.4%	94.1%
Top 3	97.2%	96.8%
Top 4	97.8%	96.8%
Top 5	97.8%	96.8%
Top 6	97.8%	96.8%

Data stored on the phone. The implementation requires the phone to store the IDs, coordinates, small image icons and benchmark images of reference objects. Since each icon is about 3 KB, it takes about 200 and 150 KB for 63, 53 reference objects in the mall and train station. . An 800 $\times$ 600 benchmark image is only about 30 KB, while 50-60 reference objects are sufficient for a large mall or train station. Thus the total storage is less than 2 MB. Such data can be downloaded on demand before the user enters the building. Having the phone doing the localization computation avoids a second interaction to send the corrected results to the server for final results, thus reducing the latency.

5.2 System Performance

We conduct experiments with the prototype in both the mall (63 reference objects, 41 in stores and 22 outside) and train station (53 reference objects), with 108 and 46 test locations scattered around the environment (see Figs. 16a and 16b).

Image quality versus accuracy. First we examine the impact of image resolution on the matching accuracy. A higher resolution has better accuracy but larger size as well. The original JPEG image has about 3,200 $\times$ 2,400 resolution at 3 MB. JPEG images have a "quality" parameter that can be tuned, which affects the resolution and size. We vary the "quality" parameter from 0 to 100 in steps of 10, and see how image size and matching accuracy change for the 22 reference objects outside stores in the mall. We find that quality 40 achieves a desirable balance: the image size is only 30 KB (about 800 $\times$ 600 resolution), while the accuracy is about 88 percent. Thus we set the metric at 40 for images uploaded by the phone.

Image matching accuracy. Table 1 shows the probability that the top M matched reference objects contain the correct one in offline mode (each reference object with three benchmark images). We find that there is certain increase up to top 3, beyond which the improvements are minimal. That is why the UI presents the top 3 matches for the user: it achieves a balance between users' correction needs and cognitive efforts.

When a test image's correct match is in top 3, the system knows the correct reference object after user feedback (i.e., tapping the correct thumbnail from top 3). We call such a test image *"correctable"*. Next we examine (in Table 2) the fraction of test locations having 3, 2, 1 or 0 correctable test images. We find that 92.7 and 90.3 percent of the test locations in the mall and station have three correctable test images. The system knows all the three reference objects after user feedback. Less than 10 percent of test locations have two correctable test images. For the uncorrectable test

TABLE 2
Fraction of Test Locations Whose Test Images' Correct Matches in Top 3

Environment	3 in top 3	2 in top 3	1 in top 3	0 in top 3
Mall	91.7%	8.3%	0%	0%
Station	90.3%	9.7%	0%	0%

image, the phone has to rely on the heuristic (Section 7) to "guess" a better match. Luckily we have not found test locations with only one or zero correctable test images. This means the phone has to make at most one guess for a test location.

Latency. The latency includes three components: user operation, transmission delay and image matching time. It takes a user a few seconds to take photos of three reference objects. The transmission delay for a 30 KB photo is less than a second. Latest image retrieval [17] can match a photo against a million images in about 0.5 s. Thus the localization takes only a few seconds.

Initial localization results. We examine the localization results in offline mode using the correct match when it is in top 3, and the top 1 (incorrect) match if it is not. Fig. 20 shows the CDF of the localization accuracy for both environments (the portion of $0-6$ m enlarged in the small embedded figure), using both real and estimated coordinates of reference objects.

We make several observations: 1) The 80-percentile errors are around 2 and 4.5 m for the mall and train station, which is comparable to the industry state-of-the-art Google Indoor Maps [18] ($\sim$7 m). The larger errors in the train station are due to larger distances between the user location and reference objects: the distances are around 10 and 30 m at 80-percentile for the mall and station, as illustrated in Fig. 19. 2) The tails of the curves are long, reaching 40m for both the station and mall. These are because the correct match is not in top 3, which we further classify and address using the heuristic. 3) The differences between the results using real and estimated coordinates are not that much. This means that our coordinate estimation method can achieve reasonable localization performance while cutting down human efforts.

The last observation is further confirmed by the ideal localization error (shown in Fig. 18) assuming perfect image matching. Fig. 18 also shows that 80-percentile errors similar to those in Fig. 20, which is because the majority of test locations already have three correct matches in top 3. It

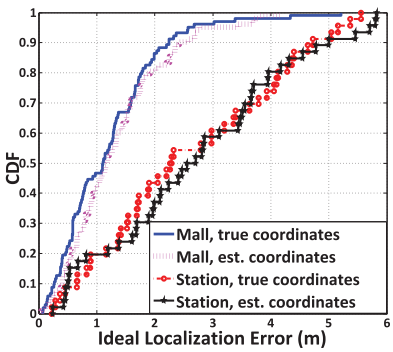


Fig. 18. Ideal localization error with perfect image matching.

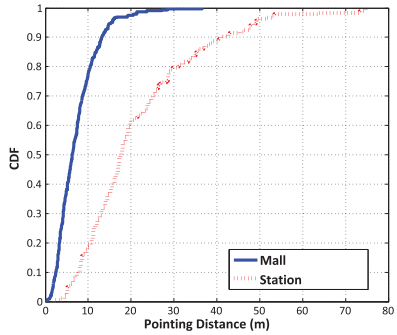


Fig. 19. Pointing distance between user location and reference objects.

shows how much improvements we may gain by further correcting image matching errors: the maximum error can be reduced to 5-6 m.

Matching error classification. We examine the test locations with large localization errors (i.e., those >6 m) one by one and classify them into several categories based on the causes, with the worst case shown in Table 3.

Extreme angle or distance. We find that in eight cases, some chosen reference objects can be very far (e.g., >50 m), or the test image taken from extreme angles (e.g., <30 degree or almost completely from the side). Although SURF descriptors are rotation-invariant, test images from such distances or angles exceed their limit and lead to wrong matching results.

User error or obstruction. In one case the chosen reference object is not at the center of its test image, leading to both incorrect match and large angle errors. In another case obstacles (e.g., people) obstruct the view to a reference object, resulting in wrong match.

Reference objects of similar appearances. We also find that some reference objects (e.g., two information desks in the train station) may have similar appearances. The benchmark images of them are inevitably difficult to distinguish even to the human eye.

Multiple reference objects in one test image. Sometimes due to the proximity and angle of photo taking, a test image may include two reference objects. The best match may be the unintended one, while the true match is ranked out of top 3.

6 Benchmark Selection of Reference Objects

In Sextant, the user takes one photo (i.e., test image) for each of the three chosen reference objects, which are then matched against benchmark images to identify the corresponding reference objects. We find that the selected benchmarks are

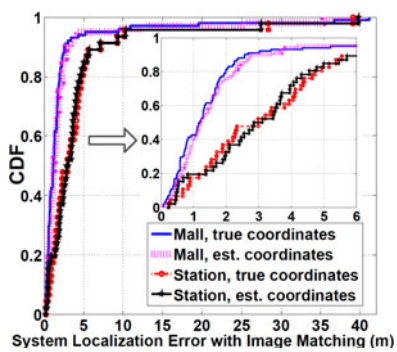


Fig. 20. Initial system localization error with image matching.

TABLE 3
Large Error Classification in Offline Mode

Cause	Number of cases	Worst loc error
Extreme angle	4	36.7 m
Extreme distance	4	39.1 m
Not centered	1	9.3 m
Obstructions	1	41.2 m
Similar appearance	1	10.1 m
Multiple points	3	19.6 m

crucial in improving image matching accuracy. In this section, we model the benchmark selection problem, prove its NP-completeness, and propose heuristic algorithms to solve it.

6.1 Benchmark Selection Problem

We formally define the problem of benchmark selection (notations in Table 4): Given m reference objects $\{1, \ldots, m\}$, and a set of n_i candidate images for reference object i, find one image for each reference object such that the total number of matching errors is minimized.

We denote the decision variables, the labels of the chosen benchmark for each reference object as

$$B = \{b_i | 1 \leq i \leq m, 1 \leq b_i \leq n_i\}. \tag{3}$$

Given the candidate images, we could profile the number of incorrectly matched images for each reference object, as Fig. 21 shows. When reference object i and j choose x and y as its benchmark respectively, an incorrect match from i to j means an image l for reference object i is incorrectly matched to reference object j. We use $p_{i,x,j,y}$ to denote the number of images for reference object i incorrectly matched to reference object j, and $P = (p_{i,x,j,y})$ as the model given input.

The objective is to find the label selection B that minimizes the number of total incorrectly matched images. We denote C_{obj} as the objective value. Given $P = (p_{i,x,j,y})$, the C_{obj} could be computed by each $B = (b_i)$, as:

$$C_{obj}(B) = \sum_{i \neq j} p_{i,b_i,j,b_j}. \tag{4}$$

And $i, j \in \{1, 2, \ldots, m\}$. Thus our objective is formulated as:

$$\min_B C_{obj}(B). \tag{5}$$

TABLE 4
Notations

$M = \{1, \ldots, m\}$, $i \in M, j \in M$	Reference objects
$N_i = \{1, ..., n_i\}$	Candidate images of reference object i
$B = \{b_i\}$	Label of selected benchmark images for reference object i
$P = \{p_{i,x,j,y}\}$, $x \in N_i, y \in N_j$	Number of images for i incorrectly matched to j, when x and y are selected benchmark images for i and j respectively.
$K = \{k_{i,x,j,y}\}$	Number of matched feature vectors between image x for i and image y for j
$u(.)$	Unit step function, equals 0 when input is less than 0, and equals 1 otherwise

Fig. 21. An example of where x and y denote the chosen benchmark (marked yellow) of reference object i and j respectively, and l denotes an image of i being matched to both x and y.

6.2 NP-Completeness Proof

We formulate its corresponding decision problem, called Benchmark Selection Decision (BSD) problem: Given $P = (p_{i,x,j,y})$ and integer c, determine whether there exists $B = (b_i)$, such that

$$\sum_{i \neq j} p_{i,b_i,j,b_j} \leq c. \tag{6}$$

And we denote its instance as $BSD(P, B, c)$.

First we show that BSD problem belongs to NP. Given its instance, it's easy to verify whether $C_{obj} = \sum_{i \neq j} p_{i,b_i,j,b_j} \leq c$ is satisfied in polynomial time.

To prove BSD problem is NP-complete, we prove that the quadratic assignment problem (QAP), which is known to be NP-complete [19], is reducible in polynomial time to BSD problem. The decision problem of QAP describes that there are two sets: $M = \{0, 1, \ldots, m-1\}$ and $N = \{0, 1, \ldots, m-1\}$, two functions: $w(i, j) : M \times M \rightarrow R^+$, and $d(x, y) : N \times N \rightarrow R^+$, and a constant c; it determines whether there exists an one-to-one mapping $f : M \rightarrow N$, such that $\sum_{i \neq j} w(i, j) \cdot d(f(i), f(j)) \leq c$.

Based on the following BSD construction, we could reduce the decision problem of QAP to our BSD problem

$$M = \{0, 1, \ldots, m-1\},\ N = \{0, 1, \ldots, m-1\}, \tag{7}$$

$$i, j \in M,\ x, y \in N, \tag{8}$$

$$p_{i,x,j,y} = \begin{cases} w(i, j) \cdot d(x, y), if\ x \neq y, \\ \eta, if\ x = y. \end{cases} \tag{9}$$

We set η as a sufficient large constant, e.g. $\eta = \sum_{i \neq j} \sum_{x,y} w(i, j) \cdot d(x, y)$, to guarantee the solution of BSD problem is an one-to-one mapping from M to N. Its solution also corresponds to the solution of decision problem of QAP.

Thus the BSD problem is NP-complete, and we will present our heuristic algorithm to find an approximated solution in reasonable time.

6.3 A Heuristic Algorithm

We here propose a heuristic algorithm to select the best benchmark image for each reference object, the intuition is that the selected image should be similar to other images of its own reference object, and distinct to images of other reference objects.

Profiling. This part is the preparation for our algorithm, aiming to measure the distinction between two candidate images.

Algorithm 1. Benchmark Selection Heuristic Algorithm

compute $k_{i,x,j,y}$ for each two candidate images;
for each reference object i **do**
  for each benchmark x **do**
    compute $S^+_{i,x}$ according to Equation (12);
    compute $S^-_{i,x}$ according to Equation (13);
    compute $Score_{i,x}$ according to Equation (14);
  end for
  $b_i = arg\max Score_{i,x}$;
end for
$time = 0$;
while $time \leq X$ **do**
  randomly select several chosen benchmarks in B, replace each with a random image of its same reference object;
  compute objective value C_{obj} according to Equation (4);
  if C_{obj} is decreased **then**
    update B based on the random benchmarks;
    update C_{obj};
    $time = 0$;
  else
    $time++$;
  end if
end while
if more benchmark is used **then**
  for each reference object i, j and benchmark x, y **do**
    $k_{i,x,j,y} = \max\{k_{i,x,j,y},\ k_{i,x,j,b_j}\}$;
  end for
  remove B from candidate image set;
  go to Step 2 to find the second best benchmark;
end if

According to [15], we first extract feature vectors on each candidate image, and calculate distance between two feature vectors to measure their similarity. The number of matched feature vectors between image x for reference object i and image y for reference object j can be computed beforehand and denoted as:

$$K = \{k_{i,x,j,y} | 1 \leq i, j \leq m, 1 \leq x \leq n_i, 1 \leq y \leq n_j\}. \quad (10)$$

As Fig. 21 shows, an image l for i is incorrectly matched to j when it has more matched feature vector with j's benchmark y than with i's benchmark x. Thus $p_{i,x,j,y}$, how many i's images are incorrectly matched to j, can be computed as:

$$p_{i,x,j,y} = \begin{cases} \sum_{l=1}^{n_i} u(k_{i,l,j,y} - k_{i,l,i,x}), if\ i \neq j, \\ 0, if\ i = j. \end{cases} \quad (11)$$

Thus we could compute P beforehand. Next we will present how to choose the best benchmark set B, aiming at the minimum objective value C_{obj} which is calculated from Equation (4).

Benchmark initialization. Initially, each reference object is assigned the "best matching" image as its benchmark, meaning its other images are very similar to the benchmark, and the benchmark is very distinct to images of other reference objects. Thus its own images match well while other reference objects' images do not match this chosen benchmark. We use two metrics below to measure its similarity to its own reference object's images and its interference to images of other reference objects.

For a chosen benchmark x of reference object i, we use the number of images for i correctly matched to x, rather than chosen benchmark t of reference object j, as the similarity metric. The metric is summed over all possible combinations of $\{j, t\}$ pairs, shown in Equation (12):

$$S^+_{i,x} = \frac{1}{n_i} \sum_{s=1}^{n_i} \sum_{j=1}^{m\&j\neq i} \sum_{t=1}^{n_j} u(k_{i,s,i,x} - k_{i,s,j,t}). \quad (12)$$

Similarly, we use the number of images for another reference object j incorrectly matched to x, rather than the chosen benchmark t of j, to measure the interference of x to j. This is also summed over all possible combinations of $\{j, t\}$ pairs, shown in Equation (13):

$$S^-_{i,x} = \frac{1}{\sum_{j=1}^{m\&j\neq i} n_j} \sum_{j=1}^{m\&j\neq i} \sum_{t=1}^{n_j} \sum_{s=1}^{n_j} u(k_{j,s,i,x} - k_{j,s,j,t}). \quad (13)$$

Then, we could score the efficiency of each benchmark x for reference object i, calculated as:

$$Score_{i,x} = S^+_{i,x} / S^-_{i,x}. \quad (14)$$

For benchmark initialization, we select the image with highest score as chosen benchmark for reference object i.

Random perturbation. Since the initialization does not necessarily give the overall optimal solution, we use random perturbation to continue improve the solution. Each time we randomly replace a chosen benchmark with an unchosen image of the same reference object, and check if the objective value decreases. If so, we update both the chosen benchmark set and objective value. This is repeated until the objective value decreases less than a threshold C_{th} after X times of continuous replacements. Then we stop the random perturbation and output the chosen benchmark set. In our implementation we use $C_{th} = 0$ and $X = 100$.

6.4 Image Matching Accuracy

We compare the image matching accuracy of our heuristic and random selection of benchmark images.

Benchmark and test image dataset. In the mall we captured 362 photos of 63 reference objects at different places, while in the station we captured 441 photos of 53 reference objects, and each reference object has four $\sim$ nine photos taken at different places. We use these photos as candidate images. In online and offline modes each reference object should select three and one benchmark, respectively. We also have 324 test images taken at 108 locations in the mall, and 138 test images taken at 46 locations in the station. We match the test images against benchmarks and measure the fraction of test images whose correct match shows in top 1-3 matching results.

Image matching accuracy. Compared with random benchmark selection, our heuristic improves image matching accuracy by more than 20 percent in offline and 10 percent in online, both in the mall and station (shown in Fig. 22). The chance that the correct match is in top 3 results in offline mode can reach 82.1 percent in the mall and 81.2 percent in the station, and that in top 3 in online mode can reach 98.2 and 97.3 percent, respectively.

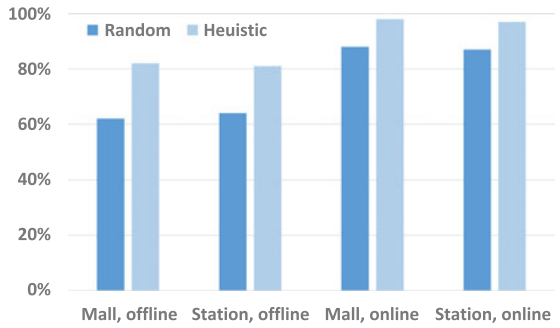


Fig. 22. Image matching accuracy where the correct match is contained in top 3 results, between random benchmark selection versus our heuristic, for the mall and train station, both online and offline modes.

Next we examine the fraction of test locations having 3,2,1 or 0 "correctable" test images. A test image is "correctable" if its correct match is within the top 3 results, where a user can click and indicate to the system. After the user feedback the system knows the true match.

According to Fig. 23, we find that in offline mode, 52.8 and 46.8 percent of test locations in mall and station have three correctable test images, while in online mode the number is 94.4 percent in mall and 91.9 percent in station. The system then knows all the 3 reference objects after user feedback. For those with 2 or 1 correctable test images, the system uses additional constraints (Section 7) to guess a better match for the unknown reference object(s). Only less than 2 percent of offline test locations in mall suffer from 0 correctable test images, where users may need to take photos of another set of reference objects.

7 Improve Localization with Geographical Constraints

Even with optimized benchmarks, there are still test images whose correct match does not show up in the top 3 results. For such images, we make informed guesses based on an observation: the three chosen reference objects by a user are usually close to each other. We propose two heuristics to estimate unknown reference objects when there are one and two "uncorrectable" test images.

7.1 Experiments and Problems in Early Prototype

We conduct experiments in two large indoor environments, a 150×75 m shopping mall and a 300×200 m train station. We test our system in both online and offline modes.

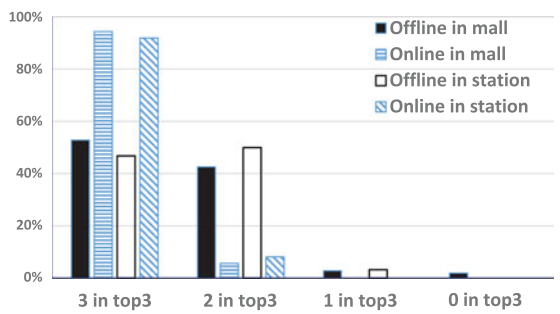


Fig. 23. Fraction of test locations with top 3 correctly matched test images.

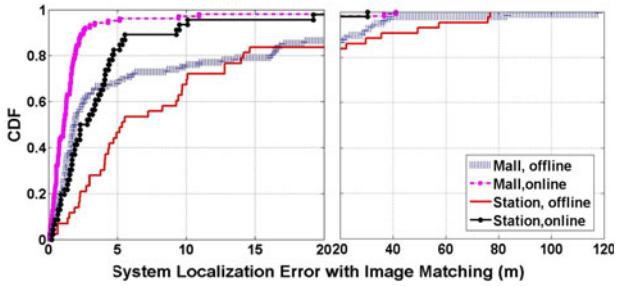


Fig. 24. System localization error after benchmark selection heuristic.

Fig. 24 shows the CDF of localization errors in offline and online modes for both the mall and station. We make a couple observations. First, the online mode has much smaller errors, with 80-percentile localization error within 2 m in mall and 5 m in station. This is because three benchmarks are used for each reference object, leading to very high image matching accuracy (e.g., more than 97 percent). However, the offline mode has 80-percentile error of 14 m in both the mall and station, with large errors reaching tens of meters. This is simply because a single benchmark has much lower matching accuracy even after user feedback (e.g., ~81 percent according to Fig. 22). Had all test images been perfectly matched, there would be less than 2 m localization error at 80-percentile in mall and 4 m in station; while the maximum error would be around 5 m both in mall and station. Thus there is quite some space for improvements.

7.2 Geographical Constraints

To better infer the identify of unknown reference objects whose correct match does not appear in top 3 results, we propose a couple geographical constraints, including cluster partition, distance metric measurement and scoring.

7.2.1 Cluster Partition

Due to the obstructions of walls, some reference objects are unlikely visible to and chosen by the user at the same time. For example, a user in a store can only see reference objects inside but not those outside. If the system knows any correctly matched image inside, the unknown ones must be inside as well.

Accordingly we cluster reference objects based on geographical layout, e.g. wall obstruction. Thus all objects inside the same store are in one cluster, those outside are in another cluster. Given any correctly matched image, we search the unknown ones within the same cluster using the following two measurements.

7.2.2 Distance Metric Measurement

When two test images are matched to their correct reference objects (denoted as A and B), we find the unknown reference object by computing a metric for each possible reference object X in the same cluster with A and B

$$D_X = (|AX| + |BX|)/2. \tag{15}$$

When only one image is matched correctly to its reference object A, we compute the following metric for each possible reference object pair X and Y in the same cluster as A

$$D_{X,Y} = (|AX| + |AY| + |XY|)/3. \tag{16}$$

7.2.3 Scoring

Then we score the possible candidate(s) according to both their image matching degree and distance metric. The score is defined as follows:

$$score_X = \frac{K_{X,1}}{D_X^2}, \text{for 1 unknown reference object,} \quad (17)$$

$$score_{X,Y} = \frac{K_{X,1} + K_{Y,2}}{D_{X,Y}^2}, \text{for 2 unknown reference objects,} \quad (18)$$

where $K_{i,j}$ is the number of matched feature vectors between the benchmark image(s) of reference object i and the test image of label j ($j = 1$ or 2). The candidate(s) with the highest score is chosen as the unknown reference object (s). The detailed description of the algorithm is shown in Algorithm 2. Note that when there are two unknown reference objects, $score_{X,Y}$ and $score_{Y,X}$ are different due to different pairings between X, Y and test image $1, 2$.

Algorithm 2. Heuristic Algorithm for Geographical Constraints

```
cluster reference objects based on geographical layout;
find the cluster T of correctly matched reference object(s) after user feedback;
if number of unknown reference objects =1 then
    for each reference object X in T do
        compute D_X according to Equation (15);
        compute score_X according to Equation (17);
    end for
    X_Est = arg max score_X;
else if number of unknown reference objects = 2 then
    for each reference object pair X, Y in T do
        compute D_{X,Y} according to Equation (16);
        compute score_{X,Y} according to Equation (18);
    end for
    {X_Est, Y_Est} = arg max score_{X,Y};
end if
```

7.3 System Localization Performance

We find that the geographical constraints improve our image matching accuracy to 91.7 percent (from 82.1 percent) in the mall and 87.6 percent (from 81.2 percent) in the station in offline mode, while 99.4 percent (from 98.2 percent) in the mall and 97.9 percent (from 97.3 percent) in the station for online use.

Fig. 25 shows the CDF of localization errors after the constraints. Compared with earlier system without geographical constraints (Fig. 20), localization error is reduced to around 3 m in mall and 8 m in station (both from 14 m) at 80 percent percentile in offline mode; the maximum error is cut to 20 m (from 118 m) in the mall and 36 m (from 76 m) in the station for offline use. For online use, the 80 percent error does not reduce much (around 2 m in the mall and 5 m in the station), but the maximum error is lowered to about 7 m (from 41 m) in the mall and 19 m (from 30 m) in the station. These show that the geographical constraints are effective in greatly cutting down maximum error, and improves the general case for offline mode significantly.

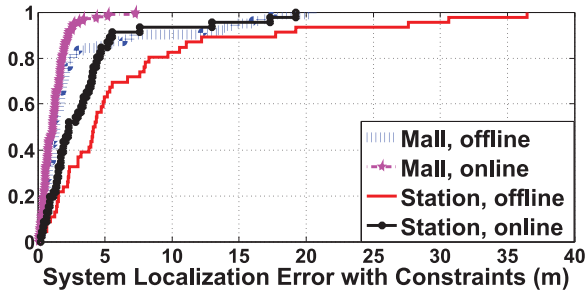


Fig. 25. System localization error with geographical constraints.

8 Discussion

Physical features selection for reference objects. Users need to understand which physical features are likely reference objects included by the system. We choose obvious ones such as store logos, information desks and find 50-60 reference objects can cover the mall and train station. However, users may still occasionally pick an object not in the reference object set. Even after the heuristic the system cannot obtain the correct coordinates. We plan to investigate methods to add such objects into the set incrementally.

Continuous localization. Sextant provides localization after a user completes the operations. It does not yet provide continuous localization when the user is in continuous motion. We plan to investigate how to combine other techniques (e.g., dead-reckoning [20]) to infer user locations in moving.

Energy consumption. The collection of image and inertial data costs some energy. According to specifications of mainstream smartphones, the power consumption for gyroscope sampling is quite low (about 30 mW), and users customarily take three photos. Sextant uses downsized images of 800×600 resolution, each about 30 kB. Based on WiFi radio transmission power around 700 mW and practical speed of 2 MB/s, uploading three photos takes about 0.035 Joule. Compared to the battery capacity of 20 k Joules, we believe the collecting and uploading of three images with inertial data do not constitute any signification energy consumption for a smartphone.

9 Related Work

Smartphone localization has attracted lots attention due to the explosive growth of location based phone applications. We describe those most relevant to Sextant and provide a comparison that is far from exhaustive.

Signature-based localization. A vast majority of existing research efforts depend on RF signatures from certain IT infrastructure. Following earlier studies that utilize WiFi signals [1], [2] for indoor localization, Liu et al. [3] leverages accurate acoustic ranging estimates among peer phones to aid the WiFi localization for meter level accuracy. Accurate GSM indoor localization is feasible in large multi-floor buildings by using wide signal-strength fingerprints that include signal readings from more than six-strongest cells [5]. Sextant does not rely on such signatures for localization. It uses network connectivity only for computation offloading.

Some work takes advantage of other smartphone sensing modalities for different signatures. SurroundSense [21] combines optical, acoustic, and motion sensors to fingerprint and identify the logical environment (e.g., stores). UnLoc

[22] proposes an unsupervised indoor localization scheme that leverages WiFi, acceleromter, compass, gyroscope and GPS to identify signature landmarks. Sextant does not use such signatures but static environmental reference objects for triangulating user locations.

Building the signature map. Some recent work has focused on methods for reducing the laborious efforts for building and maintaining signature maps. LiFS [6] leverages the user motion to construct the signature map and crowdsources its calibration to users. EZ [7] proposes genetic-based algorithms to derive the constraints in wireless propagation for configuration-free indoor localization. Zee [8] tracks inertial sensors in mobile devices carried by users while simultaneously performing WiFi scans.

However, since most of those signals are susceptible to intrinsic fluctuations and external disturbances, they must re-calibrate the signature map periodically to ensure accuracy. This incurs periodic labor efforts to measure the signal parameters at fine grained grid points. Compared to these signatures, the physical features (e.g., store logos) we use are static. Jigsaw [23] uses crowdsensed images to reconstruct the floor plan, while Sextant only requires a one-time effort to estimate the coordinates of reference objects, significantly reducing the measurement efforts.

Computer vision based work. OPS [24] allows users to locate remote objects such as buildings by taking a few photos from different known locations. It uses computer vision algorithms to extract the 3D model of the object and maps it to ground locations. Structure from Motion [25] is a mature technique in computer vision to build the 3D model of an object, it relies on large numbers of images and heavy optimization method. We use image matching algorithms for identifying chosen reference objects, not 3D models. We also propose a lightweight site survey method to quickly estimate the coordinates of reference objects.

Simultaneous localization and mapping (SLAM) [26] is a technique for robots to build the model of a new map and localize themselves within that map simultaneously. For localization the robots' kinematics information is needed. Although smartphones carried by people can provide sensory data, accurate kinematics information remains a challenge. In computer vision, extracting 3D models could estimate locations based on captured images. OPS [24] allows users to locate remote objects such as buildings by taking a few photos from different known locations. Compared to them, our localization is based on triangulation from angle measurements by the gyroscope. We use image matching algorithms only for identifying which reference objects are chosen by the user.

User efforts. Explicit user effort such as body rotation has been adopted for different purposes recently. Zhang et al. [27] show that the rotation of a user's body causes dips in received signal strength of a phone, thus providing directions to the location of an access point. SpinLoc [28] leverages similar phenomena to provide user localization at accuracies of several meters.

10 Conclusion

In this paper, we explore a new approach that leverages environmental reference objects to triangulate user locations using relative position measurements from smartphones. Because the reference objects seldom move, it avoids extensive human efforts in obtaining and maintaining RF signatures in mainstream indoor localization technologies. We have described the triangulation principle, guidelines for reference object selection and shown the feasibility of pointing operations as a localization primitive. Then we propose a lightweight site survey method to quickly estimate the coordinates of reference objects in unmapped environments. We also adopt image matching algorithms to automatically identify the selected reference objects by users.

Finally we study two issues: image matching mistakes and inferring unknown reference objects. We formulate the benchmark selection problem, prove its NP-completeness and devise a heuristic algorithm that selects benchmark images of reference objects for high image matching accuracy. We also propose a couple of geographical constraints to infer the identities of unknown reference objects that cannot be corrected by user feedback. Extensive experiments conducted in two large indoor environments, a 150×75 m shopping mall and a 300×200 m train station, have demonstrated that Sextant achieves *comparable* performance to the industry state-of-the-art, while requiring only a one-time investment of 2-3 man-hours to survey complex indoor environments hundreds of meters in size.

Acknowledgments

This work was supported in part by China NSFC Grants 61231010, 61272027, 61421062, 61210005, 61201245, and Beijing National Science Foundation (NSF) 4142022. F. Ye and G. Luo are corresponding authors.

References

[1] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *Proc. IEEE INFOCOM*, 2000, pp. 775–784.

[2] M. Youssef and A. Agrawala, "The horus WLAN location determination system," in *Proc. ACM 3rd Int. Conf. Mobile Syst., Appl. Services*, 2005, pp. 205–218.

[3] H. Liu, Y. Gan, J. Yang, S. Sidhom, Y. Wang, Y. Chen, and F. Ye, "Push the limit of wifi based localization for smartphones," in *Proc. ACM 18th Annu. Int. Conf. Mobile Comput. Netw.*, 2012, pp. 305–316.

[4] Google indoor maps availability [Online]. Available: https://support.google.com/gmm/answer/1685827?hl=en, 2014.

[5] V. Otsason, A. Varshavsky, A. LaMarca, and E. de Lara, "Accurate GSM indoor localization," in *Proc. ACM 7th Int. Conf. Ubiquitous Comput.*, 2005, pp. 141–158.

[6] Z. Yang, C. Wu, and Y. Liu, "Locating in fingerprint space: Wireless indoor localization with little human intervention," in *Proc. ACM 18th Annu. Int. Conf. Mobile Comput. Netw.*, 2012, pp. 269–280.

[7] K. Chintalapudi, A. Padmanabha Iyer, and V. N. Padmanabhan, "Indoor localization without the pain," in *Proc. ACM 16th Annu. Int. Conf. Mobile Comput. Netw.*, 2010, pp. 173–184.

[8] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen, "Zee: Zero-effort crowdsourcing for indoor localization," in *Proc. ACM 18th Annu. Int. Conf. Mobile Comput. Netw.*, 2012, pp. 293–304.

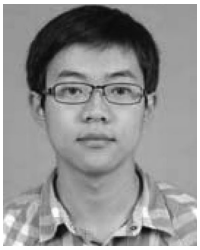
[9] Y. Tian, R. Gao, K. Bian, F. Ye, T. Wang, Y. Wang, and X. Li, "Towards ubiquitous indoor localization service leveraging environmental physical features," in *Proc. IEEE INFOCOM*, 2014, pp. 55–63.

[10] R. Gao, F. Ye, and T. Wang, "Smartphone indoor localization by photo-taking of the environment," in *Proc. IEEE Int. Conf. Commun.*, 2014, pp. 2599–2604.

[11] C. Peng, G. Shen, Y. Zhang, Y. Li, and K. Tan, "BeepBeep: A high accuracy acoustic ranging system using cots mobile devices," in *Proc. ACM 5th Int. Conf. Embedded Netw. Sensor Syst.*, 2007, pp. 1–14.

[12] N. L. Johnson, S. Kotz, and N. Balakrishnan, *Continuous Univariate Distributions*, 2nd ed. New York, NY, USA: Wiley, 1994.
[13] Dilution of precision in GPS. [Online]. Available: http://en.wikipedia.org/wiki/Dilution_of_precision_(GPS), 2014.
[14] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. IEEE 7th Int. Conf. Comput. Vis.*, 1999, pp. 1150–1157.
[15] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," *Comput. Vis. Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.
[16] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.
[17] J. Lin, L. Duan, T. Huang, and W. Gao, "Robust fisher codes for large scale image retrieval," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2013, pp. 1513–1517.
[18] Google i/o 2013—The next frontier: Indoor maps [Online]. Available: http://www.youtube.com/watch?v=oLOUXNEcAJk, 2013.
[19] S. Sahni and T. Gonzalez, "P-complete approximation problems," *J. ACM*, vol. 23, no. 3, pp. 555–565, Jul. 1976.
[20] I. Constandache, X. Bao, M. Azizyan, and R. R. Choudhury, "Did you see bob?: Human localization using mobile phones," in *Proc. ACM 16th Annu. Int. Conf. Mobile Comput. Netw.*, 2010, pp. 149–160.
[21] M. Azizyan, I. Constandache, and R. R. Choudhury, "SurroundSense: Mobile phone localization via ambience fingerprinting," in *Proc. ACM 15th Annu. Int. Conf. Mobile Comput. Netw.*, 2009, pp. 261–272.
[22] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury, "No need to war-drive: Unsupervised indoor localization," in *Proc. ACM 10th Int. Conf. Mobile Syst., Appl. Services*, 2012, pp. 197–210.
[23] 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 *Proc. ACM 20th Annu. Int. Conf. Mobile Comput. Netw.*, 2014, pp. 249–260.
[24] J. Manweiler, P. Jain, and R. R. Choudhury, "Satellites in our pockets: An object positioning system using smartphones," in *Proc. ACM ACM 10th Int. Conf. Mobile Syst., Appl. Services*, 2012, pp. 197–210, 2012, pp. 455–456.
[25] I. S. S. M. S. Sameer Agarwal, Noah Snavely, and R. Szeliski, "Building Rome in a day," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, 2009, pp. 72–79.
[26] P. Robertson, M. Angermann, and B. Krach, "Simultaneous localization and mapping for pedestrians using only foot-mounted inertial sensors," in *Proc. ACM 11th Int. Conf. Ubiquitous Comput.*, 2009, pp. 93–96.
[27] Z. Zhang, X. Zhou, W. Zhang, Y. Zhang, G. Wang, B. Y. Zhao, and H. Zheng, "I am the antenna: Accurate outdoor ap location using smartphones," in *Proc. ACM 17th Annu. Int. Conf. Mobile Comput. Netw.*, 2011, pp. 109–120.
[28] S. Sen, R. R. Choudhury, and S. Nelakuditi, "SpinLoc: Spin once to know your location," in *Proc. 12th Workshop Mobile Comput. Syst. Appl.*, 2012, pp. 12:1–12:6.

Ruipeng Gao received the BE degree in communication engineering from the Beijing University of Posts and Telecommunications, Beijing, China, in 2010. He is currently working toward the PhD degree in computer science from Peking University, Beijing, China. His research interests include wireless communication and mobile computing.



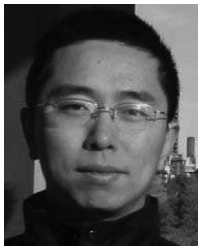
Yang Tian received the BE degree in network engineering from Beijing University of Posts and Telecommunications, in 2012. He is currently working toward the master's degree in computer science from Peking University, Beijing, China. His research interests include mobile computing, mobile sensing, and data analysis of mobile world.

Fan Ye received the BE and MS degrees in automation and computer science from Tsinghua University, Beijing, China. After receiving the PhD degree in computer science from UCLA, he joined IBM T.J. Watson Research Center as a research staff member. He was with CECA at Peking University, after which he joined the ECE Department, Stony Brook University, as an assistant professor. His research interests include mobile computing and applications, mobile cloud, sensor networks, and Internet of Things.

Guojie Luo (M'12) received the BS degree in computer science from Peking University, Beijing, China, in 2005, and the MS and PhD degrees in computer science from the University of California at Los Angeles, in 2008 and 2011, respectively. He received the 2013 ACM SIGDA Outstanding PhD Dissertation Award in Electronic Design Automation. He is currently an assistant professor with the Center for Energy-Efficient Computing and Applications, EECS School, Peking University. His research interests include physical design automation, scalable EDA algorithms, and advanced design technologies for 3D ICs. He is a member of the IEEE.

Kaigui Bian (M'11) received the PhD degree in computer engineering from Virginia Tech, Blacksburg, VA, in 2011. He is currently an associate professor with the Institute of Network Computing and Information Systems, School of Electronics Engineering and Computer Science, Peking University, Beijing, China. His research interests include mobile computing, cognitive radio networks, and network security and privacy. He is a member of the IEEE.

Yizhou Wang received the bachelor's degree in electrical engineering from Tsinghua University in 1996, and the PhD degree in computer science from the University of California at Los Angeles (UCLA) in 2005. After receiving the PhD degree, he joined Xerox Palo Alto Research Center (Xerox PARC) as a research staff from 2005 to 2007. He is a professor in the Computer Science Department, Peking University, Beijing, China. He is a vice director in the Institute of Digital Media, Peking University, and the director in New Media Lab, National Engineering Lab of Video Technology. He was at Hewlett-Packard as a system and network consultant from 1996 to 1998. His research interests include computational vision, statistical modeling and learning, pattern analysis, and digital visual arts.



Tao Wang (SM'11) received the PhD degree in computer science from Peking University, Beijing, China, in 2006. He is currently an associate professor with Peking University, Beijing, China. His research interests include computer architecture, reconfigurable logic, wireless network, and parallel computing. He is a senior member of the IEEE.

Xiaoming Li (SM'03) received the PhD degree in computer science from Stevens Institute of Technology, Hoboken, NJ, in 1986. He is currently a professor with Peking University, Beijing, China. His research interests include web search and mining and online social network analysis. He is an editor of both *Concurrency and Computation and Networking Science*. He is a senior member of the IEEE.