Towards Ubiquitous Indoor Localization Service Leveraging Environmental Physical Features*

Yang Tian, Ruipeng Gao, Kaigui Bian, Fan Ye, Tao Wang, Yizhou Wang, Xiaoming Li
School of Electronics Engineering and Computer Science
Peking University, Beijing, China
Email: {tianyangty,gaoruipeng,bkg,yefan,wangtao,yizhou.wang,lxm}@pku.edu.cn

***Abstract*—Mainstream indoor localization technologies rely on RF signatures that require extensive human efforts to measure and periodically re-calibrate. Although recent crowdsourcing based work has started to address the issue, incentives are still lacking for wide user adoption. Thus the progress to ubiquitous localization remains slow. In this paper, we explore an alternative approach that leverages environmental physical features such as store logos or wall posters. A user uses a smartphone to obtain relative position measurements to such static reference points for the system to triangulate the user location. We study the principle of such localization, determine the suitable sensor, and devise guidelines for the user to choose reference points for better accuracy. To enable fast deployment, we propose a lightweight site survey method for service providers to quickly estimate the coordinates of reference points. We incorporate and enhance image matching algorithms with a heuristic technique to automatically identify chosen reference points at high accuracy. Extensive experiments have shown that the prototype achieves $4-5m$ accuracy at 80-percentile, comparable to the industry state-of-the-art, while covering a $150\times75m$ mall and $300\times200m$ train station requires a one time investment of only $2-3$ man-hours from service providers.**

I. Introduction

Localization [1]–[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 RF (Radio Frequency) signatures from certain IT infrastructure (e.g., WiFi access points [1], [2] and cellular towers [5]).

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]–[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, 6 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, paintings on the walls. Users use the smartphone to measure their relative positions to physical features, and the coordinates of these reference points 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 points in a new environment with low human efforts. Finally, the system has to know which reference points are selected by users. Relying on explicit user input can be a nonstarter. Ideally, 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, the user takes a picture for each of three nearby reference points one by one. The photos are sent to a server to identify which reference points are selected, thus their coordinates, together with relative position measurements, are used to triangulate the user's location. Prototype experiments in large indoor environments have shown promising results, with 80-percentile accuracy at 4-5m, comparable to Google Indoor Maps.

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 caused by inaccuracies in such position measurements, and devise a simple rule of reference point selection to minimize errors.

*The first two authors contribute equally and this work is supported partially by China NSFC-61201245, NSFC-61231010 and NSFC-61073155.

[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 propose a lightweight site survey method such that a service provider can quickly obtain the coordinates of reference points in a previously unmapped environment with reasonable accuracy ($\sim$ $1m$ at 80-percentile). Our experiments find that it takes a *one time* investment of 2-3 man-hours to survey a 150×75m shopping mall or a 300×200m train station.
- We enhance image matching algorithms [9], [10] with a spatial constraint based heuristic to automatically identify selected reference points at high accuracy, thus reducing the users' cognitive efforts.
- We build a Sextant prototype consisting of a phone and a backend, and conduct extensive experiments in large complex indoor environments that shows 4-5 m accuracy at 80-percentile, using estimated coordinates.
- We also share the tips and lessons we have learned correcting image matching mistakes, and hope such insights can help further refine this approach.

In the rest of the paper, we study the forms of relative positions and the accuracies of suitable sensors (Section II). We then describe the localization operations, study the optimal reference object selection and demonstrate the feasibility of the operations as a localization primitive (Section III). We propose a lightweight approach for estimating the coordinates in an unmapped environment (Section IV), describe the automatic recognition of chosen reference points using image matching algorithm (Section V). We discuss our limits (Section VI) and review related work (Section VII), then conclude the paper (Section VIII).

II. Localization based on Relative Positions

Relative positions include the distance and orientation between the user and the reference point. 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 Figure 1, given the coordinates of two reference points 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 points 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 interior angle (see Figure 2). The user is located along the arc of the circle. With three such reference points (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.

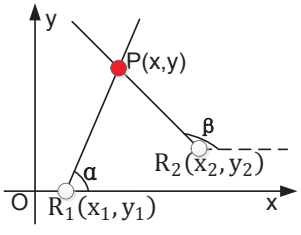


Fig. 1. Absolute angle based.

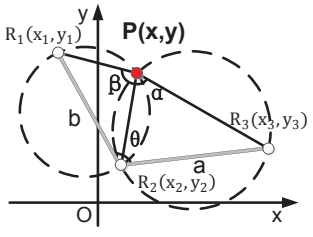
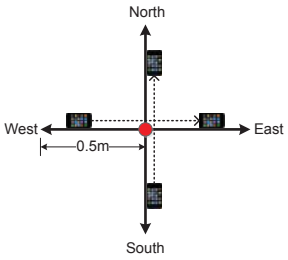
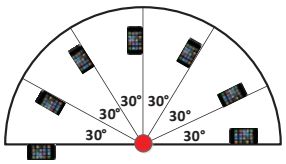


Fig. 2. Relative angle based.

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 m×6.6 m office area where 50 *test locations* are evenly distributed.



(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. 3. Two experiments for angle measurements using smartphones.

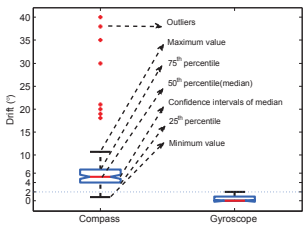


Fig. 4. Compass/gyroscope drifts (in degrees °) when moving the phone along a straight line.

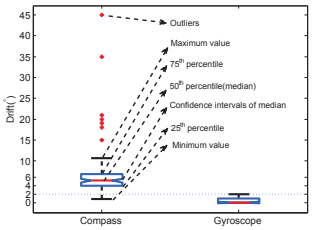


Fig. 5. Compass/gyroscope drifts (in degree °) when the phone is placed on radial lines.

Moving the phone along a straight line. When the phone is moved along a one-meter straight line at 25 cm step-lengths (shown in Figure 3(a)), 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 Figure 4, we can see that the compass has quite significant drifts (e.g., 6° at 75-percentile); it also has large outliers (e.g., 18-40°) due to electromagnetic disturbances such as nearby electric wires. However, the gyroscope has consistently small (e.g., maximum at 2°) drifts.

Rotating the phone on radial lines. Next we align the phone along radial lines separated by 30° in a semi-circle (shown in Figure 3(b)). We define the *measured angle* (expected to be close to 30°) between two adjacent radial lines as the difference between two respective sensor readings. The *drift* is how much the measured angle deviates from 30°. From Figure 5, we make similar observations to those of Figure 4. 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). We find similar small drifts ($\sim$ 1°). We

[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.

place the phone at a test location, and point the phone to four vertically-intersected directions, east, south, west, and north (as shown in Figure 3(a)). Then we rotate the phone by $\pm d^\circ$ where $-d^\circ$ is a clockwise and $+d^\circ$ a counter-clockwise rotation, and $d = 15, 30, 45$. This is repeated three times. We find that the error is at most 1° and more than half of them have less than 1° errors. We place the phone at a fixed location, and rotate the phone at two different speeds, finishing a 10° 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 conclude that the gyroscope has consistently high level of accuracy. Thus we decide to use the relative angle based localization as shown in Figure 2.

III. Pointing as a Localization Primitive

A. User Operations and Location Computation

Given the triangulation method, 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 Figure 6). Given the two angles α, β and the coordinates of the three reference points (as illustrated in Figure 2), the user location can be computed as: [3]

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

where

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

For the above operations to become a reliable localization primitive, we need to address localization errors from two more sources other than angle measurements (studied in Section II): 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 subsections.

B. Criteria for Users to Choose Reference Objects

Impact of 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 4 corners as reference points. 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 points 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% confidence interval is about $\pm 6^\circ$. Thus we use $\pm 6^\circ$ to evaluate worst-case localization errors in the following simulation.

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

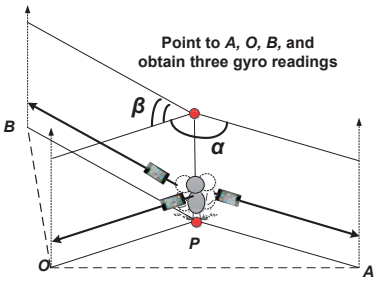
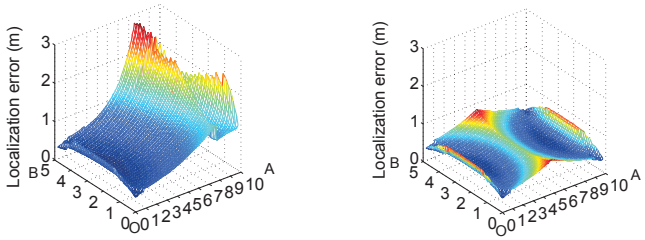


Fig. 6. The main steps of user operations: three reference objects are chosen by the user; two rotated angles α and β are measured by the phone gyroscope. Assuming the coordinates of O, A and B are known, the user's location can be calculated.



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

(b) When the closest referent point rule is used.

Fig. 7. Average localization error when $\Delta\alpha, \Delta\beta = 0, \pm 6^\circ$.

Choose a fixed set of reference points. We first study a simple rule: always choose a fixed set of three reference points (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 $\Delta\alpha = 0, \pm 6^\circ$, $\Delta\beta = 0, \pm 6^\circ$, and show the average localization error of the eight combinations of $\Delta\alpha$ and $\Delta\beta$ (except $\Delta\alpha = \Delta\beta = 0$) in Figure 7(a) as a 3-d plot. We observe that the localization error is small (e.g., $< 1m$) when the test location is close to the center reference point 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 points. The distant location can have a larger displacement while still incurring a small angle drift. As illustrated in Figure 8, 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 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 Point Rule. From the above observation, we come up with a simple rule: choose the closest reference point and its left, right adjacent ones as three reference points. Such closer points 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. Figures 7(b) 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.

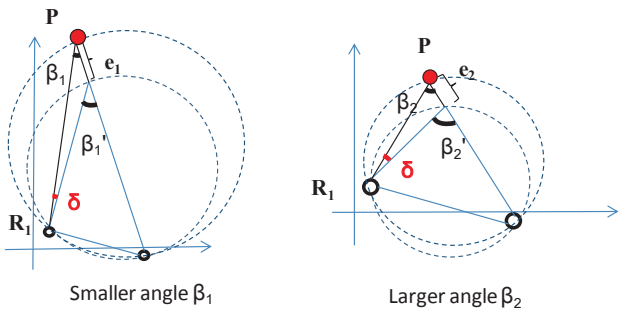


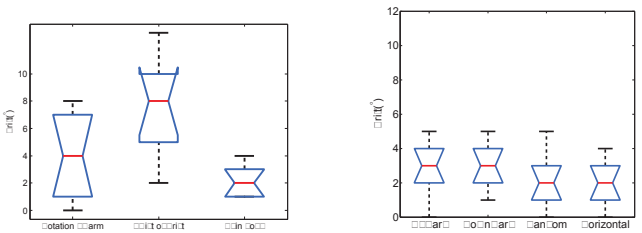
Fig. 8. The same angle drift δ on a smaller angle β_1 causes a larger localization error e_1 than that on a larger angle β_2, because the longer $\overline{R_1P}$ distance leads to more displacement.

C. 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 III-A 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.

Figure 9(a) shows the angle drift from each type of gesture. By only twisting the wrist, users make relatively large errors ($\sim 8°$), while spinning body and arm leads to the least error ($\sim 2°$). Thus we recommend the third gesture for pointing.



(a) (b)

Fig. 9. The rotated angle drift under: (a) various types of users' pointing gestures, and (b) different pointing altitude.

Impact of the phone's altitude. While spinning the arm, a user may not be able to keep the phone in a horizontal 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 Figure 9(b), we observe that the average angle drift in the two groups of "upwards" and "downwards" is just 1° 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 Figure 10, the 90-percentile accuracy is below 0.5 m for all groups.

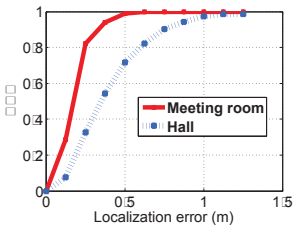


Fig. 10. The CDF of error distribution for two rectangle rooms.

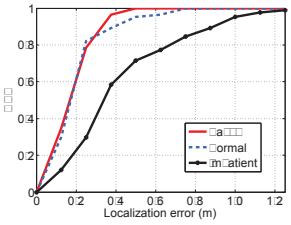


Fig. 11. The CDF of error distribution by different types of users.

Impact of the area size, shape and reference object width. We conduct experiments in two rectangle areas (a 6.6 m×4.2 m meeting room and a 14.4 m×13.2 m hospital hall). We use the closest reference point rule, repeat the experiment three times at each test location on a grid of ∼1m cell size. The CDFs of average errors are shown in Figure 10. We find that the 80-percentile accuracy is around 0.2 m 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×5.7m) and find similar results (e.g., $0.7m$ at 90-percentile). We also test in two large outdoor areas of 30 m×30 m and 20 m×40 m sizes. The 80-percentile error is ∼1 m and maximum at $1.5m$, slightly larger than that of indoor environments because it scales to the area size. Finally we try reference points of some widths (e.g., $1m$ wide posters) and find that when the center convention is followed, the accuracy is not affected much (∼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 points.

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 point 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.

Figure 11 shows the CDF results in the meeting room. We make several observations: a savvy user obtains the best accuracy (e.g., ∼ 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 point 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 point 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. Figure 12(a) shows that iPhone4, iTouch4 and i9100g almost have the same expected performance at a high level of accuracy (e.g., ∼ $0.4m$ at 90-percentile). However, i9100 shows the worst results (over $1.2m$).

We place the i9100 phone at a static location and record the readings once the gyroscope is turned on (at time 0 in

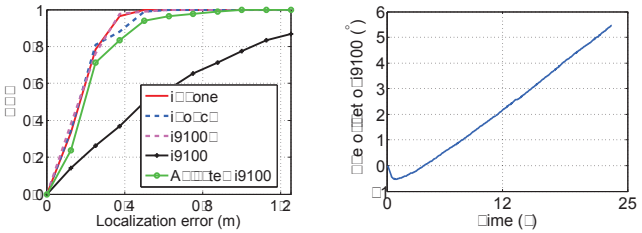


Fig. 12. Experiments using different types of devices in the meeting room. (a) The CDF of error distribution, and (b) Angle drift vs. time for i9100.

Figure 12(b)). We find the value declines at the very beginning, and then starts increasing (as shown in Figure 12(b)). 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 Figure 12(a)) 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 points, and how the system can gain input of which reference points the user has chosen.

IV. Site Survey for Reference Point Coordinates

Sextant needs the coordinates of reference points 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 points 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.

In an unmapped environment, two workers [4] of a service provider first choose two *pair-wise visible* reference points, say A and B, called *starting pairs* (step1 in Figure 13). 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 Figure 13), and so on. The coordinates of each additional positioned object can be uniquely determined in this coordinate system.

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

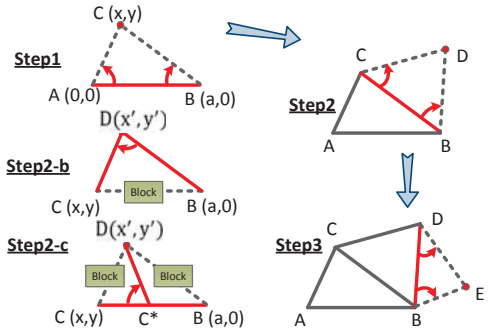


Fig. 13. Procedure to estimate the coordinates of reference points.

Blocked positioned objects. During the process when the direct line of sight between B and C is blocked (step2-b in Figure 13), 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 Figure 13), 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 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.

Accuracy. We conduct experiments in two large indoor environments, a $150 \times 75m$ shopping mall (Figure 14) and a $300 \times 200m$ train station (Figure 15). When only one starting pair is used (reference points [1,6] in Figure 14 and [1,13] in Figure 15, shown in green or slightly darker color), errors are small ($< 2m$) up to $4 \sim 6$ hops away, beyond which they quickly grow to more than 12m. 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 $1m$, while the maximum about $2m$ (Figure 16). They eventually lead to satisfactory localization accuracy (Section V-C).

Human efforts. In the mall each of the 63 reference points takes about 2 minutes to measure the angle(s) and/or distance(s); in the station each of the 53 points takes about 3 minutes due to longer walking distances. In total they cost 2, 2.6 man-hours. Assuming WiFi signatures are measured $2m$ apart and each location takes 10s, excluding inaccessible areas $5,200m^2$ and $23,700m^2$ areas need to be covered, resulting in 3.6, 16.5 man-hours. Thus the cost is roughly $16 - 55\%$ 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% 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,

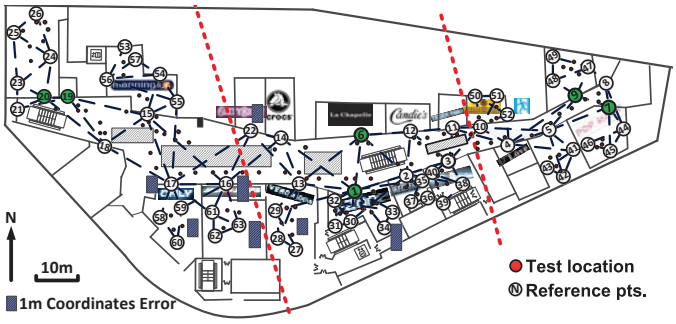


Fig. 14. The floor map of a mall dissected in three sections each with a starting pair, in total 63 reference points and 108 test locations. The vertical bar shows the error in estimated coordinates; those $< 1m$ are not shown.

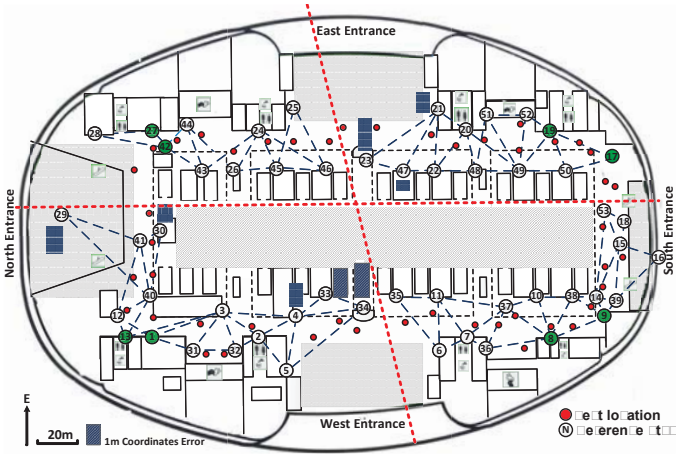


Fig. 15. The floor map of a train station dissected in four sections each with a starting pair, in total 53 reference points and 46 test locations. The vertical bars show $> 1m$ errors in estimated coordinates.

they show that our site survey method can significantly reduce the human efforts compared to those of brute-force or WiFi.

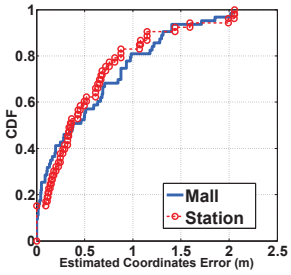


Fig. 16. Errors in estimated coordinates for a mall and train station.

V. Identifying Chosen Reference Points

The Sextant system has to know which reference points 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 points with less user efforts becomes a quite challenging problem in a complex environment with many reference points.

We explore image matching algorithms: the user takes one photo (i.e., test image) for each of the 3 chosen reference points, which are sent to a server to identify the corresponding reference points. Nevertheless, we find that the matching algorithms make wrong identifications in many situations. Next we will explain how we use the algorithms, classify error situations and address them with a simple heuristic.

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

A. System Architecture and Work Flow

We have prototyped a 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 points (taken by a service provider).

Image capture via finger taps. To accommodate test images taken from different angles, we take 3 benchmark images for each reference point, from the front, 45° left and right sides at medium distances (e.g., $\sim$5m). The user uses the same spin operations. He taps the phone's screen to take a test image when a chosen reference point 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 point.

Image matching and ranking. We examine two most popular image feature vector extraction algorithms, SIFT (Scale Invariant Feature Transform) [14] and SURF (Speeded Up Robust Features) [9]. Comparison [9] 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 [9] to rank benchmark images based on the number of matched feature vectors. We apply RANSAC (RANdom SAmple Consensus) [10] 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 points 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 (Figure 17), with the top row showing the 3 test images, below each is a column of 3 best matched reference points. 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.

Data stored on the phone. The implementation requires the phone to store the IDs, coordinates and small image icons of reference points. Since each icon is about 3KB, it takes about 200 and 150 KB for 63, 53 reference points in the mall and train station. 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.

B. Experiment Results

We conduct experiments with the prototype in both the mall (63 reference points, 41 in stores and 22 outside) and train station (53 reference points), with 108 and 46 test locations scattered around the environment (see Figure 14 and 15).

Image quality vs. 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 3200x2400 resolution at 3MB. 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 points outside stores in the mall. We find that quality 40 achieves a desirable balance: the image size is only 30KB (about 800x600 resolution), while the accuracy is about 88%. Thus we set the metric at 40 for images uploaded by the phone.

Image matching accuracy. Table I shows the probability that the top M matched reference points contain the correct one. 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.

TABLE I. IMAGE MATCHING ACCURACY

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%

TABLE II. 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	none in top 3
Mall	91.7%	8.3%	0%	0%
Station	90.3%	9.7%	0%	0%

When a test image's correct match is in top 3, the system knows the correct reference point after user feedback (i.e., tapping the correct thumbnail from top 3). We call such a test image *"correctable"*. Next we examine (in Table II) the fraction of test locations having 3, 2, 1 or 0 correctable test images. We find that 92.7% and 90.3% of the test locations in the mall and station have 3 correctable test images. The system knows all the 3 reference points after user feedback. Less than 10% of test locations have 2 correctable test images. For the uncorrectable test image, the phone has to rely on the heuristic (Section V-C) 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 30KB photo is less than a second. Latest image retrieval [15] can match a photo against a million images in about $0.5s$. Thus the localization takes only a few seconds.

Initial localization results. We examine the localization results using the correct match when it is in top 3, and the top 1 (incorrect) match if it is not. Figure 18 shows the CDF of the localization accuracy for both environments (the portion of $0-6m$ enlarged in the small embedded figure), using both real and estimated coordinates of reference points.

We make several observations: 1) The 80-percentile errors are around 2m and 4.5m for the mall and train station, which is comparable to the industry state-of-the-art Google Indoor Maps [16] ($\sim 7m$). The larger errors in the train station are due to larger distances between the user location and reference points: the distances are around 10m and 30m at 80-percentile for the mall and station. 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 Figure 19) assuming perfect image matching. Figure 19 also shows that 80-percentile errors similar to those in Figure 18, which is because the majority of test locations already have 3 correct matches in top 3. It shows how much improvements we may gain by further correcting image matching errors: the maximum error can be reduced to $5-6m$.

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 III.

TABLE III. LARGE ERROR CLASSIFICATION.

Cause	Number of cases	Worst example	Chosen point	Top match	Loc error
Extreme angle	4	mall 9-2	15	31	36.7m
Extreme distance	4	station 46-1	28	27	39.1m
Not centered	1	station 55-3	19	41	9.3m
Obstructions	1	mall 10-1	20	46	41.2m
Similar appearance	1	station 52-2	23	34	10.1m
Multiple points	3	mall 1-1	20	21	19.6m

Extreme angle or distance. We find that in 8 cases, some chosen reference points can be very far (e.g., $>$ 50m), or the test image taken from extreme angles (e.g., $< 30^\circ$ 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 ("station 55-3" meaning the third test image for location 55) the chosen reference point 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 point, resulting in wrong match.

Reference points of similar appearances. We also find that some reference points (e.g., two information desks in the train station, "station 52-2") may have similar appearances. The benchmark images of them are inevitably difficult to distinguish even to the human eye.

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

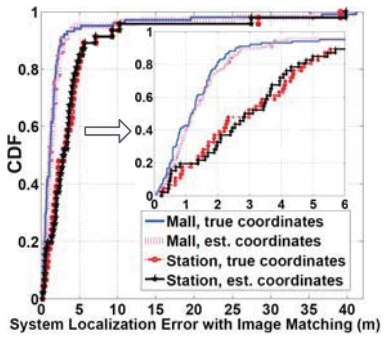


Fig. 18. Initial system localization error with image matching

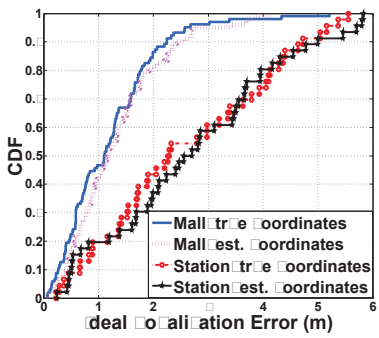


Fig. 19. Ideal localization error with perfect image matching

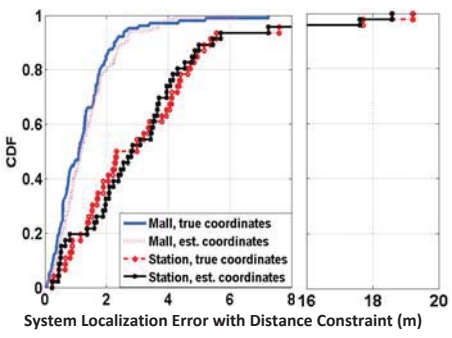


Fig. 20. System location error after applying the error correction heuristic.

C. A Heuristic Correcting Matching Mistakes

When the correct match is not in top 3, the user can only inform the system of the mistake but not the true match. We devise a heuristic to make educated guesses based on two ideas: distance and clustering constraints. 1) Usually the three chosen ones will not be too far from each other. If the system finds that a top matched one is far from the other two, it is likely a false match. 2) Due to the obstructions of walls, some reference points are unlikely visible to and chosen by the user at the same time. For example, a user in a store can only see reference points inside; the walls obstruct his line of sights to those outside the store. If the system knows the two correctly matched ones are inside, the unknown one must be inside as well.

We cluster the reference points in the mall based on wall obstruction: all points inside the same store are in one cluster, those outside are in another cluster. For the train station all points are in one cluster. For each triplet of reference points k, i, j in the same cluster, we define a "closeness" metric $D(k,i,j) = d(k,i) + d(k,j)$ where $d(k,i)$ is the distance between k and i, $d(k,j)$ that of k, j. The metric can be computed/downloaded beforehand and stored on the phone for later lookup. We find that this adds 15KB for the mall and 150KB for the station (because all points are in one cluster).

Given the true match i, j for two test images, a score c_k is computed for each point k in the same cluster as i, j

$$c_k = n(k)/D(k,i,j)^2 \tag{3}$$

where $n(k)$ is the matching metric value between the test image and the best matched benchmark image of reference point k (returned by the server as a ranked list of [ID: matching metric value] tuples), $D(k,i,j)$ is the closeness metric. The score is higher for better match or shorter distances. The square gives those closer to i,j much stronger preference. The point with the highest c_k is chosen as the correction.

TABLE IV. RESULTS OF HEURISTIC CORRECTION

Cause	#Cases	#Corrected	#Improved
Extreme angles	4	2	2 station ($7.2 \rightarrow 4.8m$) ($28.4 \rightarrow 19.2m$)
Extreme distance	4	2	2 station ($4.9 \rightarrow 4.9m$) ($39.1 \rightarrow 17.7m$)
Not centered	1	0	1 station ($9.3 \rightarrow 7.6m$)
Obstructions	1	0	1 mall ($41.2 \rightarrow 7.3m$)
Similar appearances	1	1	0
Multiple points	3	3	0

Figure 20 shows the error after applying the heuristic. The maximum errors of the mall are reduced to about $7.5m$, slightly larger than those ($\sim 5.5m$) with perfect matching (Figure 19); the 80 and 90-percentile errors are similar. For the train station except 2 test locations, all errors are less than $7.5m$, while 80 and 90-percentile errors are similar. This demonstrates the effectiveness of the heuristic in correcting image matching errors.

Table IV shows the number of cases corrected to the true match, and those not true match but with improved localization errors. We find the heuristic is effective dealing with similar appearances and multiple reference points. All 4 are corrected to the true match. It corrects half of the extreme angle/distance cases, and reduces errors in the other half significantly. It is not able to correct un-centered objects or obstructions, nevertheless the errors are reduced.

VI. DISCUSSION

Which physical features are reference points. Users need to understand which physical features are likely reference points included by the system. We choose obvious ones such as store logos, information desks and find $50-60$ reference points can cover the mall and train station. However, users may still occasionally pick an object not in the reference point 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.

Disturbances from moving people. Occasionally, a test image may have many customers or passengers getting between the user and reference points. Some of the feature vectors extracted may come from such dynamic objects. Even though the store logos are not blocked, they can disturb the matching algorithm and lead to false match.

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 [17]) to infer user locations in moving.

Appropriate benchmark images. Ideally, benchmark images should be taken at likely user locations around a reference point. Then the feature vectors in test images are more likely to match those in benchmarks. Our benchmark set includes 3 images taken from the front, $\sim 45^\circ$ to the left and right of each reference point at medium distances (e.g., $\sim$5m). Although they have the correct match in top 3 for about 95% test images, they fail for test images taken from extreme angles or distances. We plan to further investigate the proper locations for benchmarks dealing with such cases.

Localizing using more than 3 reference points. In principle, more reference points add more constraints and improve the localization accuracy. It also increases the chances of user localization when one picked point is not in the benchmark

set. The costs are more user efforts taking photos and overhead matching images. We will investigate the tradeoff to determine if potential gains outweigh costs.

VII. 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 6-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 [18] combines optical, acoustic, and motion sensors to fingerprint and identify the logical environment (e.g., stores). UnLoc [19] 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 physical features 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. Sextant does not need periodic re-calibration and requires only a one-time effort to estimate the coordinates of reference points.

Computer vision based work. OPS [20] 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. We use image matching algorithms for identifying chosen reference points, not 3D models. We also propose a lightweight site survey method to quickly estimate the coordinates of reference points.

User efforts. Explicit user effort such as body rotation has been adopted for different purposes recently. Zhang *et al.* [21] 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 [22] leverages similar phenomena to provide user localization at accuracies of several meters.

VIII. Conclusions

In this paper, we explore a new approach that leverages environmental physical features to triangulate user locations using relative position measurements from smartphones. Because the physical features 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 point 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. Finally we adopt image matching algorithms to automatically identify chosen reference points, and devise a heuristic to correct matching mistakes. Extensive experiments have demonstrated that it 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.

References

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

[2] M. Youssef and A. Agrawala, "The horus wlan location determination system," in *ACM MobiSys*, 2005.

[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 *ACM Mobicom 2012*.

[4] "Google indoor maps availability." [Online]. Available: http://support.google.com/gmm/bin/answer.py?hl=en&answer=1685827

[5] V. Otsason, A. Varshavsky, A. LaMarca, and E. de Lara, "Accurate gsm indoor localization," in *UbiComp*, 2005, pp. 141–158.

[6] Z. Yang, C. Wu, and Y. Liu, "Locating in fingerprint space: Wireless indoor localization with little human intervention," in *Mobicom*, 2012, pp. 269–280.

[7] K. Chintalapudi, A. Padmanabha Iyer, and V. N. Padmanabhan, "Indoor localization without the pain," in *ACM MobiCom*, 2010.

[8] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen, "Zee: Zero-effort crowdsourcing for indoor localization," in *Mobicom*, 2012, pp. 293–304.

[9] H. Bay and A. Ess and T. Tuytelaars and L. Van Gool, "Surf: Speeded up robust features," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[10] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[11] C. Peng, G. Shen, Y. Zhang, Y. Li, and K. Tan, "Beepbeep: A high accuracy acoustic ranging system using cots mobile devices," in *ACM SenSys*, 2007.

[12] N. L. Johnson, S. Kotz, and N. Balakrishnan, *Continuous Univariate Distributions.*, 2nd ed. Wiley, 1994.

[13] "Dilution of precision in gps." [Online]. Available: http://en.wikipedia.org/wiki/Dilution_of_precision_%28GPS%29

[14] D. G. Lowe, "Object recognition from local scale-invariant features," in *ICCV*, 1999.

[15] J. Lin, L. Duan, T. Huang, and W. Gao, "Robust fisher codes for large scale image retrieval," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013.

[16] "Google I/O 2013 - The Next Frontier: Indoor Maps." [Online]. Available: http://www.youtube.com/watch?v=oLOUXNEcAJk

[17] I. Constandache, X. Bao, M. Azizyan, and R. R. Choudhury, "Did you see bob?: human localization using mobile phones," in *ACM MobiCom*, 2010.

[18] M. Azizyan, I. Constandache, and R. Roy Choudhury, "Surroundsense: mobile phone localization via ambience fingerprinting," in *ACM MobiCom*, 2009.

[19] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury, "No need to war-drive: Unsupervised indoor localization," in *MobiSys*, 2012, pp. 197–210.

[20] J. Manweiler, P. Jain, and R. R. Choudhury, "Satellites in our pockets: An object positioning system using smartphones," in *MobiSys*, 2012, pp. 211–224.

[21] 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 *ACM MobiCom*, 2011.

[22] S. Sen, R. R. Choudhury, and S. Nelakuditi, "Spinloc: Spin once to know your location," in *HotMobile*, 2012.