Smartphone Indoor Localization by Photo-taking of the Environment

Ruipeng Gao, Fan Ye, Tao Wang
School of Electronics Engineering and Computer Sciences
Peking University, Beijing, China
Email: {gaoruipeng,yefan,wangtao}@pku.edu.cn

***Abstract*—Existing mainstream indoor localization technologies mainly rely on RF signatures and thus incur significant and recurring labor cost to measure the time-varying signature map. We have proposed a smartphone localization system using the embedded gyroscope for triangulation from nearby physical features (e.g., store logos) recognized from photo-taking. It requires a much reduced and one-time measurement, while incurs uncertain localization errors. In this paper, we propose two methods to systematically address image matching errors that cause unrecognized physical features and large errors in our system. We formulate the optimal benchmark image selection problem and propose a heuristic algorithm that finds the best benchmark images for high matching accuracy. We propose a couple of geographical constraints to further infer unknown physical features based on the observation that the features chosen by the user are close together. Experiments in a $150\times75m$ shopping mall, $300\times200m$ train station show that dramatically we cut down both maximum and general localization errors, and achieve $2-8m$ accuracy at 80-percentile even with only one benchmark image on the phone.**

***Keywords*—smartphone indoor localization; environmental photos; benchmark selection; geographical constraints**

I. Introduction

Indoor localization is the basis for many novel location based services in buildings. Despite more than a decade of research and development, it is still not yet pervasive enough. The latest industry state-of-the-art, Google Indoor Maps [1], covers about 10,000 locations in 18 countries, which are only a fraction of the millions of indoor environments on the planet, e.g. shopping centers, airports, train stations, museums and hospitals. One major obstacle is that mainstream indoor localization technologies rely on RF (Radio Frequency) signature maps that require significant human efforts to build and periodic efforts to calibrate. Although a number of proposals [2], [3] have started to leverage crowdsourcing to reduce the efforts, incentives and deployment are lacking.

In a recent work [4], we propose an alternative approach that leverages static environmental physical features such as store logos. It requires only a one-time human effort that is a fraction of that of building a WiFi signature map. Specifically, users use a smartphone to capture a picture of each of three nearby physical features (called reference points) one by one. The photos are matched against the benchmark images of known reference points (e.g., store logos) to identify which physical features they are. Their coordinates, together with angle measurements from the gyroscope, are used to triangulate the user's location. Our prototype "Sextant" achieves comparable accuracy to Google Indoor Maps ($\sim 7m$). The main advantages are: 1) Physical features usually remain static over long periods of time. Once measured, there is little need for periodic re-calibration. 2) They are abundant in the environment and do not require dedicated deployment and maintenance like IT infrastructure.

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

In this paper, we propose two methods to systematically address the image matching errors. First, we study how to select the best images as the benchmark when multiple are available for each reference point. The purpose is to minimize "cross matching" where one reference point'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 the three reference points chosen by the user are usually close to each other. Given even one correct match, the unknown reference points can be inferred with much higher probability from a nearby range.

We make the following contributions in this work:

- We formulate optimal benchmark selection as a combinatorial optimization problem and prove it NP-complete. We then propose and evaluate a heuristic algorithm based on iterative perturbation for realistic solutions.
- We propose several geographical constraints that help make much informed decision about the identities of incorrectly matched reference points. Together they greatly improve the inference accuracy of the system for both offline and online users.
- We have conducted extensive evaluation in two real environments (a $150\times75m$ mall and $300\times200m$ train station). We find that the two methods improve image matching accuracy to 91.7% in the mall and 87.6% in the station for offline mode, and 99.4% in the mall and 97.9% in the station for online mode. Hence the localization errors are reduced to around $3m$ in mall and $8m$ in station at 80% percentile for offline mode, and around $2m$ in mall and $5m$ in station at 80% percentile for online mode, which are comparable to the industry state-of-the-art.

II. Background on Sextant

We describe how Sextant works and explain problems due to image matching imperfections in detail.

A. Sextant Localization Principle

In Sextant, the user stands at one location P, spins his arm and body to capture one photo (called test image) for each of three nearby reference points R_1, R_2, R_3 one by one (shown in Figure 1). The gyroscope can measure the two angles β, α rotated between two consecutive reference points. Given the coordinates ((x_1, y_1) to (x_3, y_3)) of the three reference points, the user location (x, y) is triangulated as the intersection P of the two circles. We also have a lightweight method to measure the coordinates of reference points [4].

The image matching algorithm, e.g. SURF (Speeded Up Robust Features) [5], is used to identify which are the three chosen reference points, so their coordinates are used for location computation. Feature vectors are extracted from a test image and compared with those from the benchmark images. The benchmarks are sorted in descending order of the number of matching feature vectors. The top one is regarded as the best match.

Before the system conducts localization computation, users can provide feedbacks to help correct mismatches. As shown in Figure 2, the top 3 matched reference points are shown in a column below each photo taken by the user. By default the top one is considered the correct match. If the correct match is the second or third best match, the user can click on it to indicate it is the correct match. Thus the system can still obtain the correct reference point if it is among the top 3. Otherwise, the user clicks on the photo he took, indicating the correct match is not in top 3.

B. Image Matching Problems in Sextant

The Sextant system achieves $4-5m$ accuracy at 80-percentile when each reference point has 3 benchmark images. This is comparable with the industry state-of-the-art Google Indoor Maps [6] ($\sim 7m$). However, there are still extreme localization errors for some test locations, which could reach as large as $20m$. Those errors are essentially caused by image matching mistakes and thus incorrect reference points.

In the next two sections, we propose two complimentary means to deal with matching mistakes. First, we describe how to select the best benchmark when multiple candidate images are available for each reference point. The goal is to minimize the number of incorrect matches. Second, if the correct match is not in top three, the system has to make educated guesses. Thus we present a couple of geographical constraints that can be used to narrow the scope of potential candidates for unknown reference points.

III. Optimal Benchmark Selection

Image matching algorithms inevitably make mistakes. Multiple benchmark images taken from different angles of a reference point improves accuracy significantly. However, more benchmarks lead to higher computing overhead. Thus in Sextant the number of benchmarks for each reference point 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.

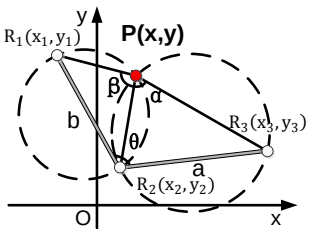

Fig. 1. Localization principle of Sextant. By measuring the two rotated angles α, β, the system computes user location (x, y) from the coordinates (x_1, y_1) to (x_3, y_3) of the reference points.

Fig. 2. 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.

A. Online and Offline Modes

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 points. 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 points. In reality we found this is not a problem. An 800×600 benchmark image is only about 30KB, while 50-60 reference points are sufficient for a large mall or train station. Thus the total storage is less than 2MB. 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 point 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 the offline mode in this section.

TABLE I
Notations

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

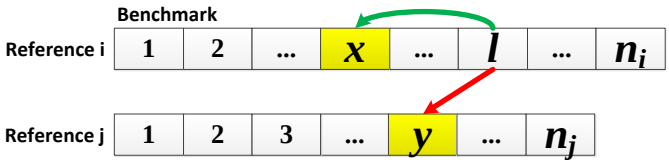

Fig. 3. An example of where x and y denote the chosen benchmark (marked yellow) of reference point i and j respectively, and l denotes an image of i being matched to both x and y.

B. Benchmark Selection Problem

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

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

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

Given the candidate images, we could profile the number of incorrectly matched images for each reference point, as Figure 3 shows. When reference point i and j choose x and y as its benchmark respectively, an incorrect match from i to j means an image l for reference point i is incorrectly matched to reference point j. We use $p_{i,x,j,y}$ to denote the number of images for reference point i incorrectly matched to reference point j, and $P = (p_{i,x,j,y})$ as the model given input.

Objective: 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:

$$C_{obj} = \sum_{i=1}^{m} \sum_{j=1}^{m} p_{i,b_i,j,b_j} \tag{2}$$

Thus our objective is formulated as:

$$\min_{B} C_{obj} \tag{3}$$

The benchmark selection problem belongs to NP, since its results can be verified in polynomial time if P,B and an objective target are given. Furthermore, the Hamiltonian-cycle problem, which is known to be NP-complete [7], is reducible in polynomial time to our benchmark selection problem (detailed proof is omitted due to space limitation). Thus the benchmark selection problem is NP-complete, and a heuristic is needed for problem-solving strategy, which is described as below.

C. A Heuristic Algorithm

We propose a heuristic algorithm consisting of three stages, namely, profiling, benchmark initialization, and random perturbation.

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

According to [5], 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 point i and image y for reference point j can be computed beforehand and denoted as:

$$K = \{k_{i,x,j,y} | 1 \leq i, j \leq m, 1 \leq x \leq n_i, 1 \leq y \leq n_j\} \tag{4}$$

As Figure 3 shows, an image l for reference point i is incorrectly matched to reference point 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:

$$p_{i,x,j,y} = \begin{cases} \sum_{l=1}^{n_i} u(k_{i,l,j,y} - k_{i,l,i,x}), \ if \ i \neq j \\ 0, \ if \ i = j \end{cases} \tag{5}$$

With the computed beforehand P, we then present how to choose the best benchmark set B to obtain minimum objective value C_{obj}, which is calculated according to Equation 2.

Benchmark initialization. Initially, each reference point 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 points. Thus its own images match well while other reference points' images do not match this chosen benchmark. We use two metrics below to measure its similarity to its own reference point's images and its interference to images of other reference points.

For a chosen benchmark x of reference point i, we use the number of images for i correctly matched to x, rather than chosen benchmark t of reference point j, as the similarity metric. The metric is summed over all possible combinations of $\{j, t\}$ pairs, shown in Equation 6.

$$S^{+}_{i,x} = \frac{1}{n_i} \sum_{s=1}^{n_i} \sum_{j=1}^{m\&j\neq i} \sum_{t=1}^{n_j} u(k_{i,s,i,x} - k_{i,s,j,t}) \tag{6}$$

Similarly, we use the number of images for another reference point 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 7.

$$S^{-}_{i,x} = \frac{1}{\sum_{j=1}^{m\&j\neq i} n_j} \sum_{j=1}^{m\&j\neq i} \sum_{t=1}^{n_j} \sum_{s=1}^{n_j} u(k_{j,s,i,x} - k_{j,s,j,t}) \tag{7}$$

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

$$Score_{i,x} = S^{+}_{i,x} / S^{-}_{i,x} \tag{8}$$

For benchmark initialization, we select the image with highest score as the chosen benchmark for reference point 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 point, 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

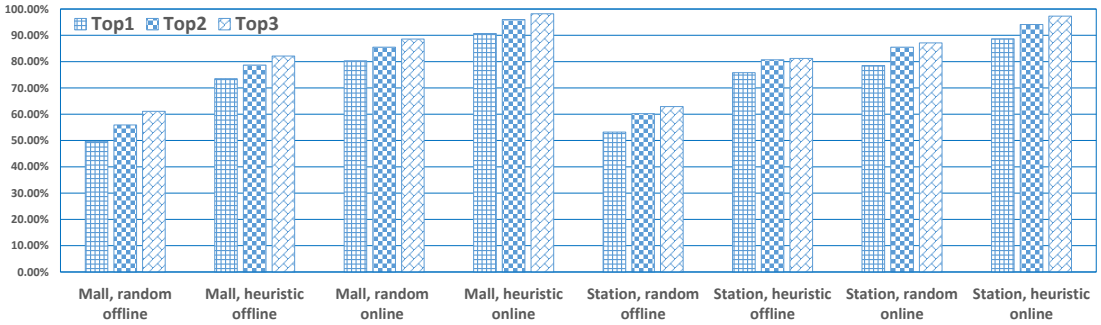

Fig. 4. Image matching accuracy where the correct match is contained in top 1-3 results, between random benchmark selection vs. our heuristic, for the mall and train station, both online and offline modes.

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 $X = 100$.

Algorithm 1 Benchmark selection heuristic algorithm

compute $k_{i,x,j,y}$ for each two candidate images;
for each reference point i **do**
  for each benchmark x **do**
    compute $S^+_{i,x}$ according to Equation (6);
    compute $S^-_{i,x}$ according to Equation (7);
    compute $Score_{i,x}$ according to Equation (8);
  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 point;
  compute objective value C_{obj} according to Equation (2);
  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 point 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

D. 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 references at different places, while in the station we captured 441 photos of 53 references, and each reference has $4 \sim 9$ photos taken at different places. We use these photos as candidate images. In online and offline modes each reference should select 3 and 1 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.

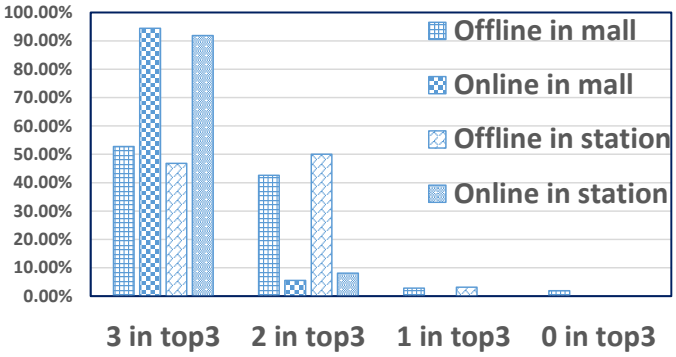

Fig. 5. Fraction of test locations with top 3 correctly matched test images

Image matching accuracy. Compared with random benchmark selection, our heuristic improves image matching accuracy by more than 20% in offline and 10% in online, both in the mall and station (shown in Figure 4). The chance that the correct match is in top 3 results in offline mode can reach 82.1% in the mall and 81.2% in the station, and that in top 3 in online mode can reach 98.2% and 97.3%, respectively.

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 Figure 5, we find that in offline mode, 52.8% and 46.8% of test locations in mall and station have 3 correctable test images, while in online mode the number is 94.4% in mall and 91.9% in station. The system then knows all the 3 reference points after user feedback. For those with 2 or 1 correctable test images, the system uses additional constraints (Section IV) to guess a better match for the unknown reference point(s). Only less than 2% of offline test locations in mall suffer from 0 correctable test images, where users may need

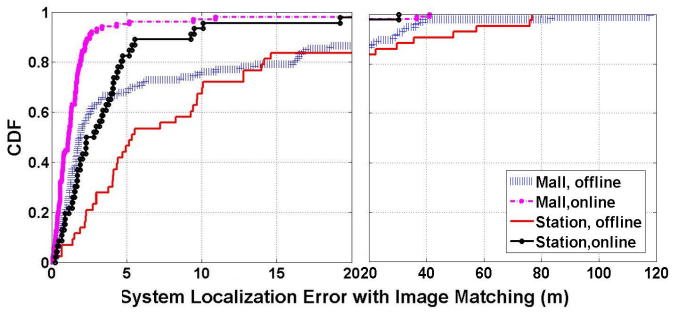

Fig. 6. System localization error after the benchmark selection heuristic.

to take photos of another set of reference points.

IV. Improve Localization with Geographical Constraints

Even with optimized benchmark, 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 points by a user are usually close to each other. We propose two heuristics to estimate unknown reference points when there are 1 and 2 "uncorrectable" test images.

A. Experiment Results And Problems in Early Prototype

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

Figure 6 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 $2m$ in mall and $5m$ in station. This is because 3 benchmarks are used for each reference point, leading to very high image matching accuracy (e.g., more than 97%). However, the offline mode has 80-percentile error of $14m$ 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% according to Figure 4). Had all test images been perfectly matched, there would be less than $2m$ localization error at 80-percentile in mall and $4m$ in station; while the maximum error would be around $5m$ both in mall and station. Thus there is quite some space for improvements.

B. Geographical Constraints

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

1) Cluster partition: 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 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 points based on geographical layout, e.g. wall obstruction. Thus all points 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.

2) Distance metric measurement: When 2 test images are matched to their correct reference points (denoted as A and B), we find the unknown reference point by computing a metric for each possible reference point X in the same cluster with A and B.

$$D_X = (|AX| + |BX|)/2 \tag{9}$$

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

$$D_{X,Y} = (|AX| + |AY| + |XY|)/3 \tag{10}$$

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:

$$score_X = \frac{K_{X,1}}{D_X^2}, \text{for 1 unknown reference point} \tag{11}$$

$$score_{X,Y} = \frac{K_{X,1} + K_{Y,2}}{D_{X,Y}^2}, \text{for 2 unknown reference points} \tag{12}$$

where $K_{i,j}$ is the number of matched feature vectors between the benchmark image(s) of reference point i and the test image of unknown reference j. The candidate(s) with the highest score is chosen as the unknown reference point(s). The detailed description of the algorithm is shown in Algorithm 2. Note that when there are two unknown reference points, $score_{X,Y}$ 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 points based on geographical layout;
find the cluster T of correctly matched reference point(s) after user feedback;
if number of unknown reference points=1 then
    for each reference point X in T do
        compute D_X according to Equation (9);
        compute score_X according to Equation (11);
    end for
    X_Est = arg max score_X;
else if number of unknown reference points=2 then
    for each two benchmarks X,Y in T do
        compute D_{X,Y} according to Equation (10);
        compute score_{X,Y} according to Equation (12);
    end for
    {X_Est, Y_Est} = arg max score_{X,Y};
end if
```

C. System Localization Performance

We find that the geographical constraints improve our image matching accuracy to 91.7% (from 82.1%) in the mall and 87.6% (from 81.2%) in the station in offline mode, while

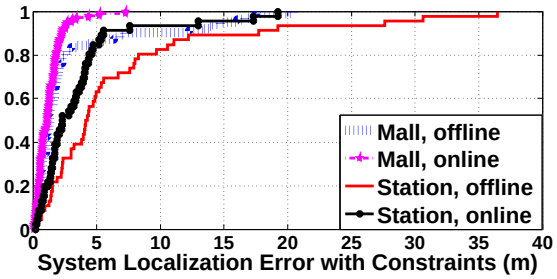

Fig. 7. System localization error with geographical constraints.

99.4% (from 98.2%) in the mall and 97.9% (from 97.3%) in the station for online use.

Figure 7 shows the CDF of localization errors after the constraints. Compared with earlier system without geographical constraints (Figure 6), localization error is reduced to around $3m$ in mall and $8m$ in station (both from $14m$) at 80% percentile in offline mode; the maximum error is cut to $20m$ (from $118m$) in the mall and $36m$ (from $76m$) in the station for offline use. For online use, the 80% error does not reduce much (around $2m$ in the mall and $5m$ in the station), but the maximum error is lowered to about $7m$ (from $41m$) in the mall and $19m$ (from $30m$) 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.

V. Related Work

A vast majority of existing research efforts depend on RF signatures from certain IT infrastructure. Mostly they leverage WiFi access points [8], [9] and cellular towers [10], and measure signature map of certain RF signals from different infrastructures for localization. Following earlier studies some work takes advantage of other smartphone sensing modalities for different signatures. Liu *et al.* [11] uses accurate acoustic ranging estimates among peer phones to aid the WiFi localization for meter level accuracy. UnLoc [12] proposes an unsupervised indoor localization scheme that leverages WiFi, acceleromter, compass, gyroscope and GPS to identify signature landmarks.

However, since most of those signals are susceptible to intrinsic fluctuations and external disturbances, they must recalibrate the signature map periodically to ensure accuracy. This incurs periodic labor efforts to measure the signal parameters at fine grained grid points. Some recent researches [2], [3] aim to leverage crowd-sourcing to reduce dedicated site survey costs. But the deployment to large user base is still slow due to the lack of strong incentives.

Compared to these signatures, the physical features (e.g., store logos) we use are static. Thus a one time measurement is sufficient for long term service. We also have an algorithm [4] to reduce the measurement effort to a fraction of that of a single WiFi site survey.

SLAM (simultaneous localization and mapping) [13] 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 [14] 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 points are chosen by the user.

VI. Conclusion

In this paper, we study two issues in our smartphone indoor localization system: image matching accuracy and inferring unknown reference points. We formulate the optimal benchmark selection problem, prove it is NP-complete and design a heuristic algorithm that selects optimal benchmark images for high image matching accuracy. We propose a couple of geographical constraints to infer the identities of unknown reference points that cannot be corrected by user feedback. The experiments conducted in two large indoor environments, a 150×75m shopping mall and a 300×200m train station, show that the benchmark selection algorithm improves image matching accuracy significantly, and the geographical constraints dramatically cut down the maximum and general errors, especially for offline mode where only one benchmark image is available on the phone. As future work, we plan to investigate methods to build and maintain the system's reference point set incrementally from user inputs, which is necessary if users choose objects unknown to the system as reference points.

References

[1] "Google indoor maps availability." [Online]. Available: http://support.google.com/gmm/bin/answer.py?hl=en&answer=1685827
[2] Z. Yang, C. Wu, and Y. Liu, "Locating in fingerprint space: Wireless indoor localization with little human intervention," in *Mobicom*, 2012, pp. 269–280.
[3] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen, "Zee: Zero-effort crowdsourcing for indoor localization," in *Mobicom*, 2012, pp. 293–304.
[4] 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 *IEEE INFOCOM*, 2014.
[5] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," in *CVIU*, 2008.
[6] "Google i/o 2013 - the next frontier: Indoor maps." [Online]. Available: http://www.youtube.com/watch?v=oLOUXNEcAJk
[7] S. Sahni and T. Gonzalez, "P-complete approximation problems," *J. ACM*, vol. 23, no. 3, pp. 555–565, Jul. 1976.
[8] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *INFOCOM*, 2000.
[9] M. Youssef and A. Agrawala, "The horus wlan location determination system," in *MobiSys*, 2005.
[10] V. Otsason, A. Varshavsky, A. LaMarca, and E. de Lara, "Accurate gsm indoor localization," in *UbiComp*, 2005, pp. 141–158.
[11] 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 *Mobicom 2012*.
[12] 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.
[13] P. Robertson, M. Angermann, and B. Krach, "Simultaneous localization and mapping for pedestrians using only foot-mounted inertial sensors," in *Ubicomp*, 2009.
[14] J. Manweiler, P. Jain, and R. R. Choudhury, "Satellites in our pockets: An object positioning system using smartphones," in *MobiSys*, 2012, pp. 211–224.