

Towards Accurate Object Localization with Smartphones

Longfei Shangguan, *Student Member, IEEE*, Zimu Zhou, *Student Member, IEEE*, Zheng Yang, *Member, IEEE*, Kebin Liu, *Member, IEEE*, Zhenjiang Li, *Member, IEEE*, Xibin Zhao, *Member, IEEE*, and Yunhao Liu, *Senior Member, IEEE*

Abstract—In this study, we explore the possibility of locating remote objects via cameras together with built-in inertial sensors of off-the-shelf smartphones. Our solution, *CamLoc*, enables a user taking two photos of an object using a smartphone at a fixed location and immediately knowing the location of the object in global coordinates, thus facilitating myriad location-based services. Such usage is user-friendly but error prone. We devise several techniques to mitigate the errors caused by cheap and noisy sensors, upgrading the positioning accuracy to an applicable level. We prototype *CamLoc* on Android OS, and evaluate its performance across different scenarios with various building densities. Experiment results show that our system achieves 89 percent and 72 percent physical location mapping accuracy in rural and downtown areas, respectively, which is competitive with existing solutions.

Index Terms—Object localization, photograph-based ranging, smartphone

1 INTRODUCTION

THE prevalence of smartphones with built-in positioning sensors (e.g., Global Positioning System module) has fostered a variety of location-based services, such as context-aware advertisement [1], driving navigation [2], geo-tagging [19], recommendation of restaurants, hotels and entertainment [3], [4], [5], etc.

In its basic form, location-based services refer to a process in which a user sends a query including her current location and a specific request related to the location (e.g., “restaurants around me”, “driving directions from here to the airport”). When a service provider receives such a query, it responds with proper guidance. We term this form as *Self-location based query*. Such basic form requires the user to expose his/her location to the untrusted service provider, which may lead to privacy leakage. Apart from this basic form, its counterpart, *Target-location based query* can be entirely natural and acceptable, such as “how expensive are rooms in that nice hotel far away” or “the close time of a shopping mall within view”. Instead of walking or driving up to the object’s location, the ability to immediately look up a hotel’s price based on hotel’s location is finely desirable. It not only

satisfies users’ specific requirement, but also protects their location privacy. Many efforts have been made for such “object localization” [10], [11], [12], [19] and we focus on it in this study.

The majority of previous solutions for object localization are based on computer vision (CV) techniques. Specifically, by analyzing a large amount of pictures of the target object, the location of this object can be computed [18], [19], [20], [21]. Manweiler *et al.* [19] release such data-richness method and only require taking pictures of an object at several different sites. They use CV techniques to create an approximate 3D structure of the object through camera, and apply mobile phone sensors to scale and rotate the structure to its absolute configuration. By solving (nonlinear) optimizations on the residual (scaling and rotation) error, the location of the object is figured out.

However, their solution requires photos taken from notably various angles w.r.t the target, so that the target’s 3D structure can be easily built. In practice, it is sometimes cumbersome, if not impossible, to shoot at different many-footstep-apart sites when a user is driving or when a user can only walk along a straight street. In addition, constructing the spatial relationship between objects and cameras, as well as solving optimization problems, induces significant computation cost, response delay, and consequent issues about energy consumption and service quality. Such inconveniences impede object localization becoming mainstream.

In this study, we explore the possibility of an easy yet effective object localization scheme. We propose **CamLoc**, a range-based object localization system. Our solution enables a user shooting two pictures of an object using a smartphone at a fixed location to immediately know the location of the object in global coordinates, based on which vast location-based services become available for the object. The positioning is real-time and implemented in off-the-shelf smartphones.

The key idea of *CamLoc* is to leverage both sensor readings and photos to calculate the user-to-target distance.

- L. Shangguan was a research associate at School of Software, Tsinghua University during this study, and now a PhD student at CSE Department, Hong Kong University of Science and Technology, Kowloon, Hong Kong. E-mail: longfei@greenorbs.com.
- Z. Zhou is with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Kowloon, Hong Kong. E-mail: zhouzimu@greenorbs.com.
- Z. Yang, K. Liu, X. Zhao and Y. Liu are with the School of Software and TNLIST, Tsinghua University, Beijing 100084, China. E-mail: {zheng, kebin, yunhao}@greenorbs.com, zxb@tsinghua.edu.cn.
- Z. Li is with the School of Computer Engineering, Nanyang Technological University, Singapore 639798. E-mail: lzjiang@ntu.edu.sg.

Manuscript received 11 May 2013; revised 2 Aug. 2013; accepted 3 Sept. 2013. Date of publication 16 Sept. 2013; date of current version 17 Sept. 2014. Recommended for acceptance by K. Wu.

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/TPDS.2013.236

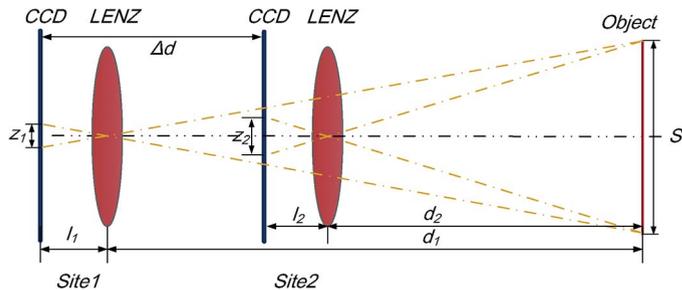


Fig. 1. Illustration of photograph-based ranging methodology.

Specifically, a mobile user launches CamLoc system on his smartphone and shoots a photo of the object of interest. Afterwards, he stretches his arm towards the target, and shoots another photo of the target. CamLoc processes these images and calculates the scaling ratio of target. Parallel to this, the sampled acceleration records during the arm's movement are employed to compute the phone's displacement. The user-to-object distance could be computed based on the well-known lens formula [7].

However, codifying such a simple idea into a practical system incurs plenty of challenges. First, the arm's displacement is particularly difficult to calculate. The built-in accelerometers are cheap and noisy. As a consequence, a small error of sensor readings would cause severe deviation with respect to the total length of arm stretching. Second, accurately measuring the scaling ratio of the object in two photos is non-trivial, which requires specialized image processing techniques. Third, mapping the physical object into an online map (e.g., Google Map) with annotated virtual objects also needs meticulous calibration.

CamLoc addresses such issues by introducing a novel data processing framework. Specifically, a three-phase filtering process is performed on the accelerometer data, followed by an overall displacement optimization to precisely compute the phone displacement. Parallel with phone displacement computation, CamLoc leverages the state-of-the-art boundary detection algorithm with a novel object recognition method to accurately measure the scaling ratio of the object in two photos. After that, the phone displacement and the scaling ratio parameter, along with the user's location information are combined and translated into the object's physical location, e.g., $\langle \text{latitude}; \text{longitude} \rangle$. CamLoc correlates the environmental context of the observer and generates heuristic constraints for physical mapping process, which finally assist in mapping the object into physical coordinates marked on the E-map. The contributions of this paper are summarized as follows.

1. **Identifying the possibility of locating remote object via the calibration of built-in sensors on off-the-shelf smartphones.** Camera, GPS, accelerometer, compass and gyroscope on Samsung Nexus S are used to sense the environmental context, and we further codify them into an object location.
2. **Investigating the root causes for sensor noise based on comprehensive benchmark experiments.** We take a scrutiny on the raw sensor traces, identify the root cause of noise for each sensor, and handle it with rigid mathematical models.

3. **Proposing a novel data processing framework to deal with sensor noise.** The rigid mathematical models on sensor noise motivate us to develop an efficient data processing framework that successfully eliminates the sensor noise, further reducing the observer-target ranging error.

To validate this design, we prototype CamLoc on Android 2.3.3 Gingerbread platform, and evaluate its performance on a Samsung Nexus S smartphone. We conduct extensive experiments in scenarios with various building densities from rural airport to downtown areas. The user-to-target distance spans from 40 m ~ 250 m. The experimental result shows that the average absolute ranging error is bounded in 30 m. Further, our system achieves a physical location mapping accuracy of 89 percent in rural areas, and 72 percent in downtown areas, which is competitive and promising.

2 PRELIMINARY

In this section, we first review the *principle of photograph-based ranging*, which serves as the basis for our CamLoc localization system. Then we highlight the main challenges to consolidate the principles into a practical system.

As Fig. 1 shows, suppose an object is projected onto a CCD plane via a camera. Let d represent the distance between the object and the LENS of the camera. We denote s and z as the physical size, and the projected size of the target object, respectively. Let f and l be the focal length of the camera and the distance between the LENS and CCD plain. According to the lens formula [7], we have: $\frac{1}{l} + \frac{1}{d} = \frac{1}{f}$, $z \cdot d = s \cdot l$. Suppose d_i is the distance between the object and the observing point i (denoted as Site i in Fig. 1). Similarly, we define z_i as the size of the object on the image photographed at observing point i . Then we have the following equation:

$$d_2 - k \cdot d_1 - f \cdot (1 - k) = 0 \quad (1)$$

where k stands for the ratio of z_1 to z_2 , i.e., $\frac{z_1}{z_2}$. If we obtain k and Δd ($\Delta d = d_1 - d_2$), then d_1 and d_2 are calculated by the following equations:

$$d_1 = f + \frac{1}{1-k} \Delta d \quad (2)$$

$$d_2 = f + \frac{k}{1-k} \Delta d. \quad (3)$$

In essence, the photograph-based ranging aims to extract accurate relative distance w.r.t the observer and the target. As illustrated in Eqs. (2) and (3), three parameters are crucial for precise ranging: k , Δd and f . Although f is the camera's built-in parameter which can be obtained directly from the camera specification, estimating k and Δd poses a range of challenges. The main hurdle lies in the noisy embedded sensors and the unpredictable user actions. More specifically, computing the scaling ratio k in realtime requires tactful algorithm design, while the displacement of the built-in camera, Δd , is envisioned to be derived by embedded sensors on the phone to minimize user interactions. For instance, if the user shoots one photo,

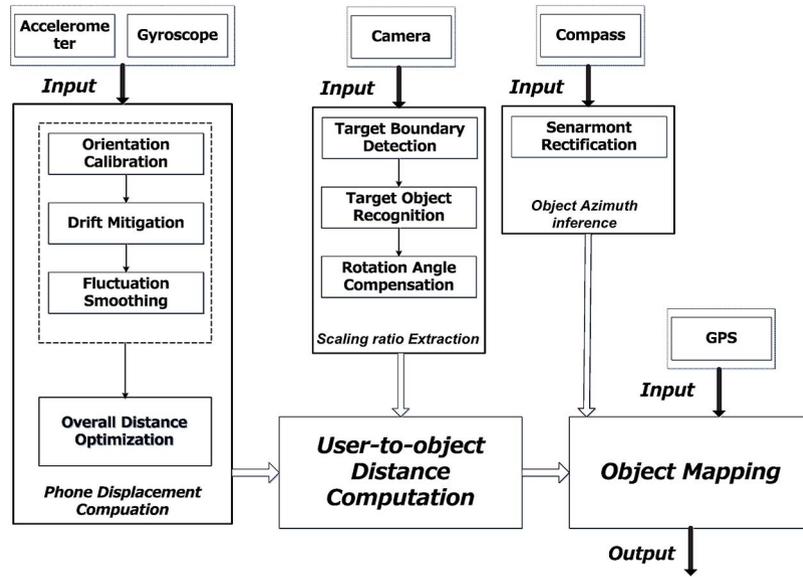


Fig. 2. System architecture of CamLoc.

and stretches his arm to shoot another one, the distance of arm stretching can be employed as a reference phone displacement Δd , which is obtained from acceleration readings. Once Δd and k are achieved, either d_1 or d_2 can be computed and regarded as the distance between the observer and the target since arm's stretching distance is negligible compared to the relative distance w.r.t the observer and target.

The computation process involves near zero user interactions, yet poses considerable challenges to derive such short displacement from noisy sensors. Besides, beyond simply ranging the observer-to-target distance, we would like to precisely locate the target with the assistant of digital maps such as Google Map. Therefore, we also have to estimate the accurate azimuth of the target, and then efficiently map the object into physical coordinates marked on a map.

3 SYSTEM OVERVIEW

Despite the convenient operations of CamLoc from the user's perspective, the underlying processes at the system level are no easy task. Fig. 2 portrays an overview of CamLoc's architecture. We introduce the basic components in the system, and their work-flow. We expect it to help the transition from operation to technical details.

CamLoc initializes with the first shooting of the desired object and waits for successive movement of the phone, which is envisioned to be conducted by the stretching of the user's arm. On detecting effective phone displacement, CamLoc triggers accelerometer and gyroscope to track the motion trail of the arm. Raw accelerometer trace is then processed via three-step filter, namely, orientation calibration, drift mitigation and fluctuation smoothing. Finally, sanitized accelerometer readings and gyroscope samples are utilized to estimate the phone's displacement, and we get the final result via a distance optimization model.

Parallel to this, CamLoc calculates the scaling ratio of the target building (within two photos) via light-weighted CV techniques. Specifically, CamLoc first detects bound-

aries of each building that appears in the photo. Then it automatically recognizes the target boundary via advanced target recognition technique. On distinguishing the target building, CamLoc rectifies its size and calculates its scaling ratio. Taking one step further, these intermediate parameters are transformed into relative distance between the object and the observer.

Upon computing the relative distance w.r.t the object and the observer, CamLoc records the azimuth of the target object via the phone's compass. Along with the GPS at the shooting spot, CamLoc is capable of computing the object's universal location. Finally, such universal location is mapped into the corresponding physical building marked on the E-map.

4 SYSTEM DESIGN

As aforementioned, CamLoc consists of four key components:

1. phone displacement computation,
2. scaling ratio extraction,
3. object azimuth inference and
4. physical location mapping.

In this section, we detail the design challenges and implementation of each module.

4.1 Phone Displacement Computation

Precise phone displacement computation is crucial for accurate user-to-target relative distance estimation, whereas raw data provided by built-in sensors tend to be noisy. We initiate this subsection with a scrutiny on the acceleration readings to derive the root cause for both data drift and fluctuation, after which we detail the acceleration integral based displacement computation scheme.

4.1.1 Integral-Based Displacement Computation Principle

In principle, the phone's displacement is deduced from two successive integrations on acceleration within the time

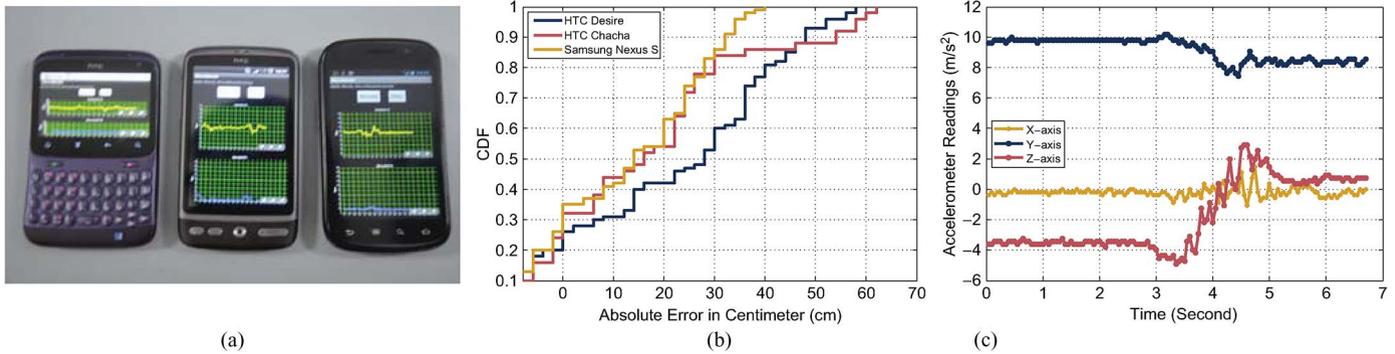


Fig. 3. Benchmark experiment. (a) Three kinds of mobile platforms for accelerometer trace collection. (b) CDF of displacement computation error. (c) Accelerometer trace along three axis.

interval of interest

$$\int \left(\int a dt \right) dt. \quad (4)$$

Where a represents the instantaneous accelerate value. Albeit light-weighted, the built-in sensors often accompany with massive *drift and noise* [9], resulting in unacceptable deviation of computed displacement due to the *drift and noise magnification effects* of two successive integration operations.

To verify the severeness of displacement error caused by acceleration noise, we record 50 traces of accelerometer readings, each capturing the arm stretching movement of a user with a mobile phone in his hand. The raw acceleration readings are then exploited to derive the phone displacement based on Eq. (4). Fig. 3b depicts the CDF of absolute displacement errors on three different mobile phones (Illustrated in Fig. 3a). Consistent with our envision, only 5 percent accelerations traces achieve reasonable accuracy (given the acceptable absolute displacement error is within 10 cm), thus making the localization system infeasible.

4.1.2 Root Cause for Drift and Fluctuation

Fig. 3c depicts the typical acceleration traces sampled along three axes, which consist of both a stationary state (0 ~ 3 second), and an arm stretching process (3 ~ 7 second). We make three critical observations from the acceleration readings. 1) Acceleration fluctuation is inevitable whether in static position or during phone movement due to inherent sensor noise. 2) The readings suffer sharp vibrations over the initial stage of arm stretching. We term the sawtooth-shaped large disturbance as acceleration drift. 3) Even when the phone stays motionless, acceleration samples in the horizontal axes (x and z) are not necessarily zero, since the phone might be held in arbitrary orientation with respect to the physical coordinates. This coordinate inconsistency introduces extra source for acceleration deviation, which stems from the unstable phone orientation during arm stretching. Therefore, CamLoc strives to mitigate errors induced by all the three root causes to achieve accurate integration-based displacement computation.

4.1.3 Orientation Calibration

The first phase to eliminate acceleration errors is to recalibrate phone orientation. For simplicity of illustration,

we define the 3-Dimension Cartesian coordinate system of the mobile phone as $(x; y; z)$, where x , y and z represent the orthogonal coordinate. Correspondingly, we denote $(X; Y; Z)$ as the physical Cartesian coordinate system. In general, the two coordinate systems are not well-aligned. Let θ_x , θ_y and θ_z represent the angles between each pair of axes, respectively. Since these angles always change during the arm's movement, we record them on each time unit $(\langle \theta_x, \theta_y, \theta_z, t \rangle)$ by integrating angular velocity captured by gyroscope. Let $\langle a_x, a_y, a_z, t \rangle^1$ represent the accelerometer readings at the same time index. If, ideally, $(x; y; z)$ is well-aligned with $(X; Y; Z)$,² then $a_x = 0$, $a_y = 0$, and $a_z = g$.

Our calibration scheme is based on the fact that in the normal course of photo shooting, y axis usually points directly to the target, which aligns with Y axis in the world coordinate system. The integrations are therefore performed upon the acceleration values along Y axis. When jointly considering phone orientation, we have:

$$a_Y = a_y \cdot \cos(\theta_y) + a_x \cdot \sin(\theta_x) \cdot \cos(\theta_z) + a_z \cdot \sin(\theta_z) \cdot \cos(\theta_x). \quad (5)$$

To timely rectify the acceleration errors caused by coordinate deviation, we set the sampling rate of gyroscope consistent with the accelerometer. At each time interval, acceleration reading and the corresponding rotation angle are substituted according to Eq. (5). Finally, the acceleration trace along Y -axis is calibrated.

4.1.4 Drift Mitigation

Assuming the acceleration fluctuation and drift are pairwise independent along each axis, it is reasonable to decompose the acceleration readings along arbitrary orientations, while each component along individual axis follows similar error patterns. This way, we are able to mitigate drift and fluctuation on each direction separately. We first deal with acceleration drift before tackling fluctuation.

The sawtooth-shaped drift, as previously discussed, occurs over the initial stage of arm stretching. This disturbance is mainly caused by the irregular arm stretching patterns. A sudden halt of arm, for instance, potentially incurs a short negative impulse in the accelerometer reading,

1. Both of the gyroscope readings and accelerometer readings are treated as discrete sequence of signal samples.

2. Note that the lower case letter coordinate (x, y, z) is used to represent the coordinate system of mobile phone and the upper case letter coordinate (X, Y, Z) is used to represent the planet's frame of reference.

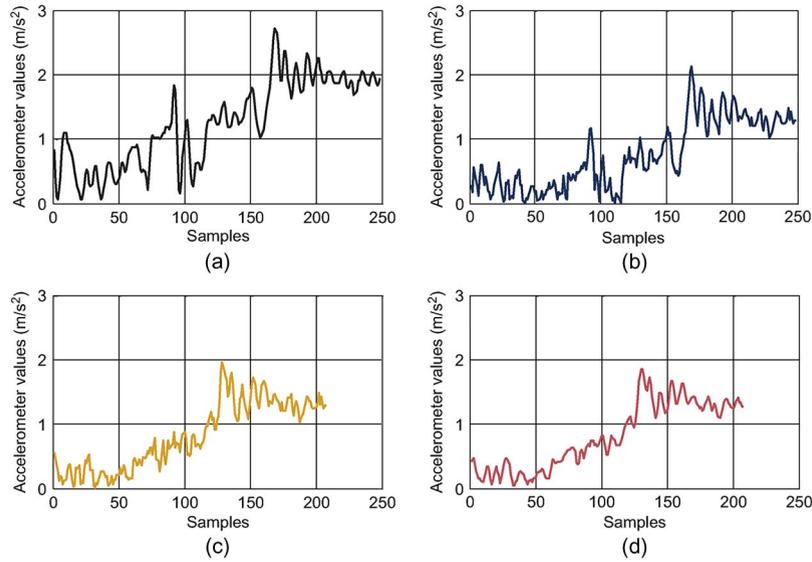


Fig. 4. Acceleration trace. (a) Illustrates the raw data. (b) Shows the trace after orientation calibration. (c) Presents the data after drift mitigation. (d) Shows the trace after performing kalman filter.

contributing to discontinuous acceleration traces. Unless occasionally compensated by a corresponding positive one, the negative impulse is doomed to explode displacement errors after integrating twice on the acceleration readings.

Since the drastic data drifts are caused by irregular phone movements, they are regarded as outliers in the acceleration traces accordingly. Therefore, we leverage Random Sample Consensus (RANSAC) [8], an iterative method to efficiently discard data drifts from acceleration traces. As accelerometer readings follow “increase first down later” pattern during arm’s stretching, thus we adopt quadratic function to depict its variance. The drift mitigation algorithm runs in an iterative fashion. Specifically, in each iteration, a random subset of the acceleration data is selected for model fitting. These data are hypothetical inliers and this hypothesis is then tested as follows:

1. All of the unselected data are tested against the fitted model and considered as inliers if they fit the fitted model.
2. The parameters of the fitted model are updated from all hypothetical inliers.
3. The fitted model is regarded accurate if sufficient data samples (t) have been regarded as inliers.

This procedure is repeated k times. For each procedure, the algorithm outputs either a model which is rejected because too few data samples are classified as inliers or a refined model together with a batch of inlier data samples. Finally, we keep the refined model if its error is lower than the last saved model, and record the inlier acceleration samples for phone’s displacement computation.

4.1.5 Fluctuation Smoothing

The accelerometer fluctuations can be regarded as system noise, which complies with Gaussian White Noisy Distribution ($N(0, \sigma^2)$). We further assume that the acceleration readings are just the first order due to the randomness of the system noise, and adopt first order Gauss-Markov

process on the time-varying acceleration readings. This means that the acceleration data ζ_Y^k satisfies the autoregressive(AR) relation:

$$\zeta_Y^k = \mu \cdot \zeta_Y^{k-1} + \varepsilon^k \quad (6)$$

where μ is a positive parameter, and ε^k is system noise with zero mean and variance σ^2 . Similarly, the observation model of our acceleration readings can be written as:

$$a_Y^k = \eta \cdot a_Y^{k-1} + w^k \quad (7)$$

where η is a positive parameter, and w^k is a Gaussian random variable. Given the AR model in Eq. (6) and the observation model in Eq. (7), we apply *Kalman filter* to eliminate the variation of acceleration readings. Due to the page limitation, we omit the details of this algorithm here.

Algorithm 1: Drift mitigation algorithm

Input : Model $M: y = ax^2 + bx + c$;

 Data samples: ζ_Y ;

Output: Model coefficient: a, b, c ;

 Inlier data samples: ζ_Y^- ;

```

1 while iterations ≤ k do
2   randomly selecting a subset  $\zeta_Y^-$  from  $\zeta_Y$ , where
   |  $|\zeta_Y^-| = n$ ;
3   estimating parameters  $a, b, c$ ;
4   for each data sample  $i$  in  $\zeta_Y - \zeta_Y^-$  do
5     if  $i$  fits current model then
6       | insert  $i$  into  $\zeta_Y^-$ ;
7   if  $|\zeta_Y^-| \geq t$  then
8     | return  $a, b, c, |\zeta_Y^-|$ ;
9   else
10    | continue;
11 return false;
```

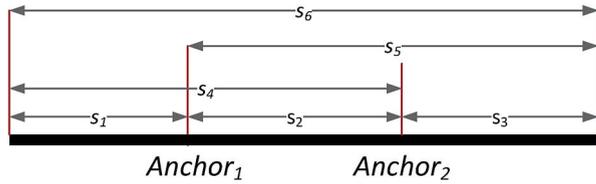


Fig. 5. Decomposition of phone's movement.

Fig. 4 illustrates the processed acceleration trace after the three-phase filtering.

4.1.6 Overall Displacement Optimization

Previous subsections primarily focus on error inflation control on the acceleration level. In this subsection, we propose an overall optimization model for the computed displacement.

To facilitate interpretation, we first present some relevant notations here. As illustrated in Fig. 5, arm's displacement is partitioned into three consecutive intervals by two randomly picked time indices. Let d_i be the integrity computation result (displacement) during time period i , and s_i be the true displacement in this period. For example, d_6 represents the displacement of arm's stretching calculated by Formula (4), while s_6 is the true displacement of arm's movement.

Ideally, the computed displacement over each period equals to the true ones, i.e., $d_i = s_i$, whereas displacement offset always occurs due to sensor imperfection. The optimization model, therefore, searches for the most-likely displacement by correcting the computed values with the following constraints:

$$\text{Min} \quad \sum_{i=1}^6 |\Delta d_i| \quad (8)$$

$$s_1 + s_2 = s_4 \quad (9)$$

$$s_2 + s_3 = s_5 \quad (10)$$

$$s_1 + s_2 + s_3 = s_6 \quad (11)$$

$$\forall i \quad d_i + \Delta d_i = s_i \quad (12)$$

where Δd_i represents the displacement offset of d_i . The underlying rationale is twofold. On the one hand, the long integration interval is divided into short ones, which effectively alleviates accumulative noise propagation. On the other hand, the model leverages the constraints among the partial displacements within each time intervals to derive the whole displacement, which bears the optimal resemblance to the true value. We employ the standard lagrangian multiplier method to solve this linear optimization model.

4.2 Scaling Ratio Extraction

Parallel with the phone displacement computation, CamLoc derives the scaling ratio of the target object from two successive photos taken before and after the phone movement. The scaling ratio extraction process consists of two stages, target boundary detection and size measurement.

4.2.1 Target Boundary Detection

CamLoc utilizes the prevalent *Sobel operator* for boundary detection. Sobel operator significantly improves detecting accuracy by exploiting weight accumulation method to

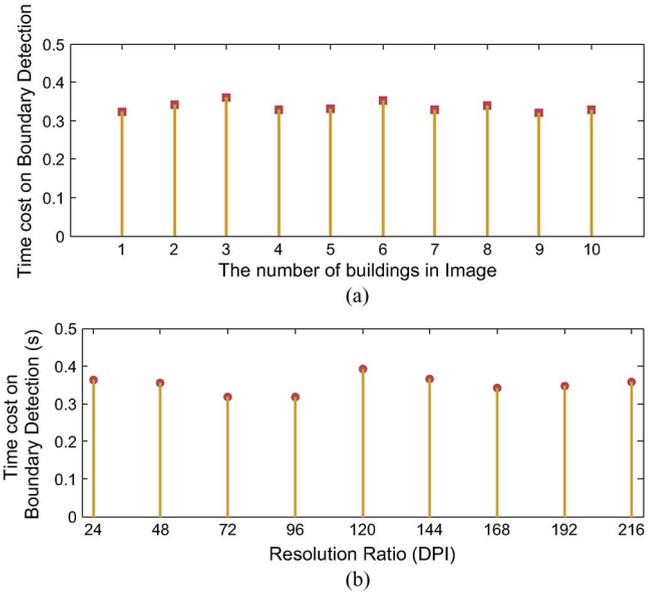


Fig. 6. Sensitivity testing on Sobel operator w.r.t. image complexity and image quality. (a) Image complexity vs. boundary detection cost. (b) Image quality vs. boundary detection cost.

reduce the edge blur effect, while retaining relatively low computation complexity as it convolves the image with a small, separable and integer valued filter in both horizontal and vertical directions [6]. To validate it, we conduct experiments under different settings, including the variance of difference density of buildings, as well as different image resolution. Fig. 6 gives the experiment results. As this group of figures shows, the computation cost of Sobel operator is insensitive to both the complexity and the resolution ratio of the photo, making it suitable for resource-constrained mobile platforms.

4.2.2 Target Object Recognition

It is common to capture multiple irrelevant background objects in the photo except for the target one. Consequently, recognizing the target building for localization is non-trivial, especially in situations where the target is surrounded by buildings with similar structure. As a result, picking out the target from boundary line collection is no easy task. CamLoc intelligently distinguishes the boundaries of the desired object based on a practical observation: when taking photos, the object of interest is always locate roughly at the center of the CCD plane. Motivated by this observation, CamLoc recursively detects the boundary lines and picks those with the shortest distance to the center of the CCD plane. As shown in Fig. 7, these lines are regarded as approximate boundaries of the target object. Recall that we have assumed the target object is always within view and is not blocked by other buildings. Therefore, the boundary lines of the target commonly appear in the vicinity of the CCD center, which theoretically justifies the correctness of our object recognition scheme.

4.2.3 Rotation Angle Compensation

With relatively low resolution of built-in camera, it is necessary to compensate for the rotation of mobile phone when measuring the size of the target object in the photo, as

a change in the angle of the phone would lead to the movement of the captured object, as well as considerable projected variation of its size in the photo.

To quantify the relationship between the phone angle change and the scaling variation of the image in the camera CCD, we build an optical model based on camera image-forming principles. As illustrated in Fig. 8, when the camera rotates by γ , the new projection of the object locates at the distance of h_2 from the optical axis. This image location movement can be written as:

$$\begin{aligned} \Delta h &= h_2 - h_1 \\ &= f \cdot \tan(\beta + \gamma) - \tan \beta \\ &\approx f \cdot \gamma \cdot \sec^2(\beta) \\ &\approx f \cdot \gamma. \end{aligned} \quad (13)$$

As shown in (13), the object movement can be estimated via the calibration of focal length f and phone's rotation angle γ . The focal length is an intrinsic parameter which can be obtained from the camera, while the rotation angle γ can be captured by the embedded gyroscope.

4.3 Object Azimuth Inference

After calculating the distance between the observer and the target, the next step is to determine the corresponding azimuth to uniquely locate the target. It is natural to infer the azimuth from the compass on the phone, as the phone is pointing towards the target while taking photos. Nevertheless, the built-in compass suffers both internal interference due to the magnetic field of the bypassing current, and external perturbation induced by surrounding high-power electronic devices. We experimentally analyze the deviations of the estimated directions and propose our azimuth rectification method.

4.3.1 Constant Direction Estimation Bias

To analyze the compass deviation pattern, we collect compass readings along with the ground truth orientations of 10 randomly chosen directions at 10 different outdoor locations. Fig. 9 depicts the PDF of the compass deviations against the ground truth. As illustrated in Fig. 9, the deviation is around 8° with 90 percent probability. Therefore, the azimuth estimation bias is approximated as a environment-related constant offset. The constant offset phenomenon indicates that internal electromagnetic interference due to bypassing current dominates the direction

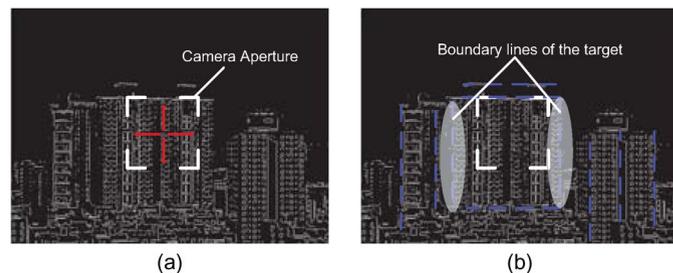


Fig. 7. Illustration of target object recognition. (a) Boundary searching process. (b) Searching result illustration.

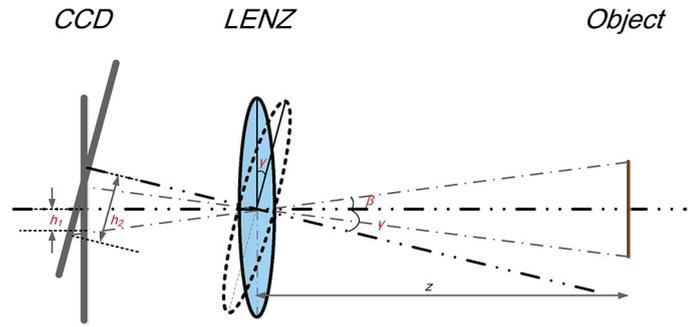


Fig. 8. Illustration of rotation effect.

estimation bias, while external perturbation contributes to quite limited random offsets.

4.3.2 Senarmont Rectification

Inspired by the experiment analysis, we employ a simple yet effective senarmont method to rectify the compass readings. In particular, we first align the mobile phone with the physical coordinates, with its screen facing upwards. We term the stable compass reading as ω' . Then we invert the phone with its screen facing downwards, and record another stable reading ω'' . Recall the angle deviation is approximately constant. ω' is hence rewritten as $\omega + \Delta\omega$, where ω is the ground truth, and $\Delta\omega$ represents the unknown direction bias due to electromagnetic interferences. Ideally, ω'' equals $\omega - \Delta\omega$. The physical underpinning is that the inverse rotation of mobile phone (and thus an inverse directional current) leads to an internal inverse electromagnetic field with the same intensity. Finally, we derive a more precise target azimuth ω via the following formula:

$$\omega = \frac{\omega' + \omega''}{2}. \quad (14)$$

4.4 Physical Location Mapping

The last stage in CamLoc is simply mapping the target object into physical plane, i.e., $\langle \text{latitude}, \text{longitude} \rangle$ based on the observer-target distance and azimuth, along with the observer's GPS. In practice, there might be multiple

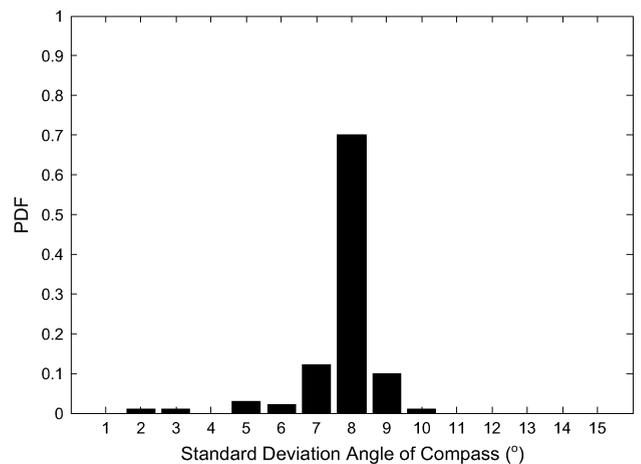


Fig. 9. PDF of standard deviation angle on compass.

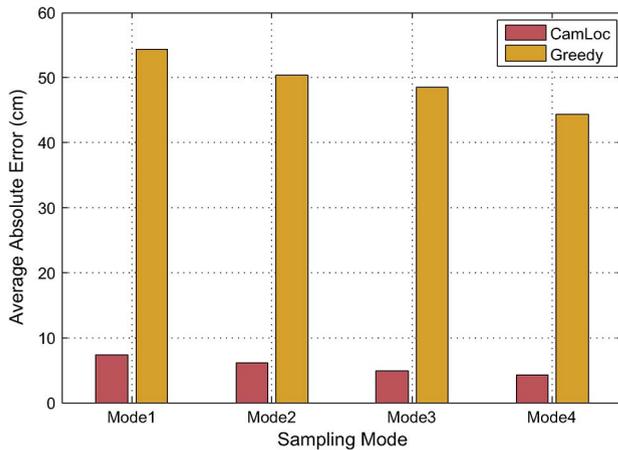


Fig. 10. Sampling rate vs. average absolute errors of displacement computation.

candidate buildings in the vicinity based on the estimated coordinate. This makes the straightforward mapping process rather error-prone, especially in tall building serried streets.

Definition 1. *Image Sparsity* is defined as $\frac{\sum_{i \in \chi, j \in \chi, i \neq j} d_{i,j}}{N}$, where $d_{i,j}$ stands for the distance between boundary i and j in pixels, χ is the boundary set, N is the number of detected boundary.

To avoid the potential false matching, a user-assisted method is further exploited to refine the mapping result. Here we employ *image sparsity* ρ to characterize the distribution of objects. Generally, a high sparsity value states that few objects are captured in an image or the gap between each object is apparent and distinguishable, while a low sparsity value indicates that multiple objects appear in this image, which impede the mapping accuracy of our system. Our mapping result refinement technique can be described as follows: once the boundary detection is performed, CamLoc automatically computes the sparsity ρ of this image. Then this value is compared with a predefined threshold δ . If $\rho < \delta$, CamLoc identifies that few objects are captured therefore delivers the object location to service provider directly. Otherwise, it requires the mobile user to simply pinpoint the category of the target object. e.g., academic building, shopping mall.

5 EXPERIMENT AND EVALUATION

In this section, we evaluate displacement computation, scaling ratio extraction and azimuth inference individually, before measuring the overall localization performance of CamLoc.

5.1 Experiment Settings

We prototype CamLoc on Android 2.3.3 Gingerbread platform, and evaluate its performance on a Samsung Nexus S smartphone with 1 GHz CPU and 512 MB RAM. Specifically, the CamLoc prototype is implemented in Java and we leverage Android NDK to cross compile OpenCV [28] source code to Android, which is then invoked for image processing. In our system, the phone accelerometer was programmed to obtain acceleration trace in four default

sampling rate. For each experiment, we cache the compass readings within 3 seconds, and average them as the sensor output.

To give a comprehensive evaluation of our system, we collect data in scenarios with various building densities, ranging from sparse rural airport, gas stations and campus, up to clustered hotels and restaurants in downtown areas, with the observer-to-target distance spanning from 40 m to 250 m. Over 30 locations across three cities (Hong Kong, Beijing, and Wuxi) are evaluated in total. In what follows, we first evaluate the performance of each module in our system, and then measure the overall localization accuracy and physical mapping performance.

5.2 Micro-Benchmarks

5.2.1 Displacement Computation

As previously discussed, CamLoc aims at mitigating acceleration error propagation to derive accurate displacement computation. Intuitively, though, a high sensor sampling rate naturally captures fine-grained motion features, thereby controlling the error propagation on integration operations. To fairly demonstrate the effectiveness of CamLoc compared with the raw data integral method (termed as Greedy), we conduct evaluations with all four accessible sampling rates provided by Android, as illustrated in Fig. 10. The four modes represent the four accessible sampling rates defined by Android API in an increasing order. The vertical axis denotes the average absolute displacement error by CamLoc and Greedy. As the bar-chart indicates, the accuracy of both CamLoc and Greedy improves with the increasing sampling rate, yet CamLoc considerably surpasses Greedy, achieving less than 5 cm average absolute error with the highest sampling rate mode. Taking this displacement error into account, the average localization error is smaller than 10 m for object within 100 m in our view, 17 m for object within 150 m, and 33 m for object within 250 m, which can be effectively amended via the collaboration of logical location refinement.

The error distributions of displacement computation for CamLoc and Greedy are shown in Fig. 11. Although Greedy achieves acceptable errors occasionally, its overall performance makes it infeasible for localization. The absolute errors of CamLoc, in contrast, remains within small ranges in all the four sampling modes. The overall improvements in displacement computation stem from its three-phase filtering process, which effectively wipes off the impact of phone orientation and outlier samples, and smoothes the system noise at the same time.

5.2.2 Scaling Ratio Extraction

In this trail of experiments, we investigate the robustness of scaling ratio extraction module. Fig. 12 reveals the relationship between resolution ratio and absolute size measurement error of the target. Through our experiment, we find that the trend of absolute size measurement error remains stable in most of the resolution ratio settings. For the last two extreme settings in which objects are fuzzy and indistinguishable, the measurement error grows too high for practical displacement computation. Nevertheless, the figure still demonstrates that our scaling ratio extraction

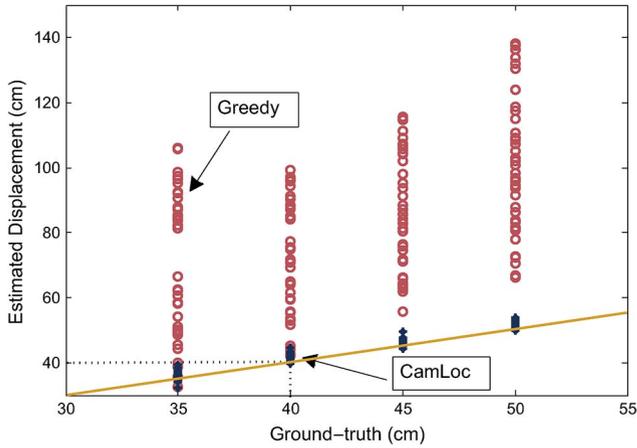


Fig. 11. Distribution of displacement computation value.

module is robust to the image quality, therefore applicable in most shooting scenarios.

Further, we evaluate the effectiveness of this module under different illumination intensities (the bottom subfigure in Fig. 12). The experimental images are captured at different times of a day via a mobile phone fixed upon the window of our lab. According to the result, Sobel operator is able to precisely capture the framework of the desired object in weak illumination (6:00 ~ 7:00), ordinary illumination (7:00 ~ 9:00, 17:00 ~ 19:00), and strong illumination (9:00 ~ 17:00) environment. Only under extremely dim illumination conditions e.g., 20:00 pm would Sobel operator fail to extract the border of the object. However, we believe that with the assistance of built-in flashlight, CamLoc is still viable for localization in caliginous environment.

5.2.3 Object Azimuth Inference

Fig. 13 visualizes the compass readings before and after the proposed rectification. The black solid arrow represents the ground-truth collected via high-precision Damping oil compass. Consistent with our key analysis in devising the rectification scheme, the unpolished compass readings deviate from the ground-truth with a wide, constant angle, i.e., around 8° in our experiment. Such constant offset therefore is effectively eliminated by simply inverting the

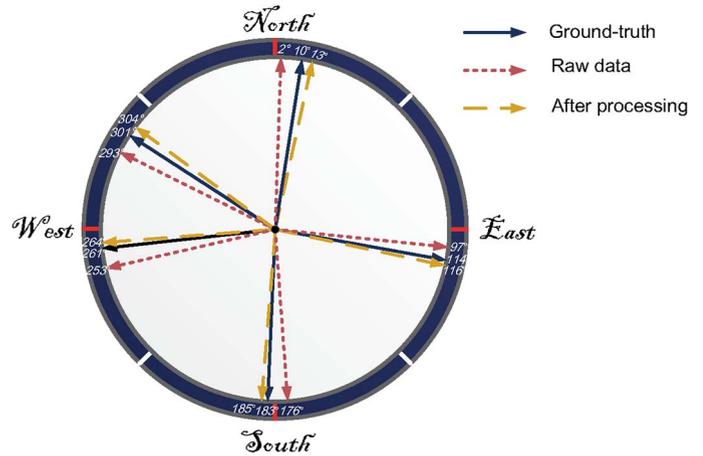


Fig. 13. Compass reading before and after processing.

orientation of mobile phones to create an inverse interference magnetic field. As demonstrated in Fig. 13, the deviation angle is decompressed by 3 to 4 times after rectification and remains within 3°, which is acceptable to achieve reasonable localization accuracy.

5.3 Localization Performance

Fig. 14 shows the object ranging accuracy with varied resolution ratio of photos. For this experiment, three desired objects are photographed in different levels of resolution ratio, along with the ground-truth distance obtained from Google Map. As illustrated in Fig. 14, the absolute ranging errors drop dramatically with the increasing resolution ratios, before staying almost the same once the resolution is above 72 dpi, which is common for mainstream built-in cameras. This insensitivity to resolution indicates that current built-in cameras are sufficient for our localization scheme, since high resolution does not necessarily contributes to considerable ranging improvement.

Fig. 15 presents the estimated relative distance respect to the ground-truth for 30 different buildings, where the real observer-to-target distances are computed by Google Map. As observed from the figure, the ranging error is pretty small for relatively close-by buildings. This is mainly due to the fact that the scaling ratios of nearby objects are larger and easier to capture, whereas the scaling ratio grows faint

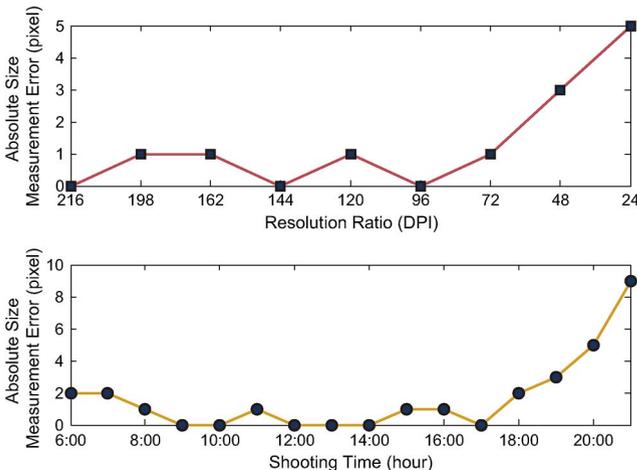


Fig. 12. Absolute size measurement error in different shooting scenarios.

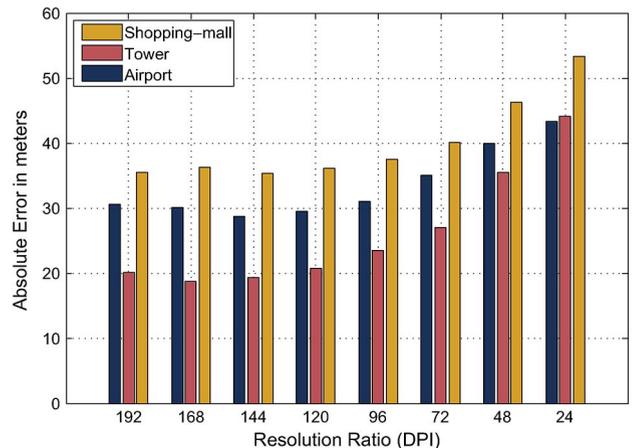


Fig. 14. Image resolution vs. absolute ranging errors.

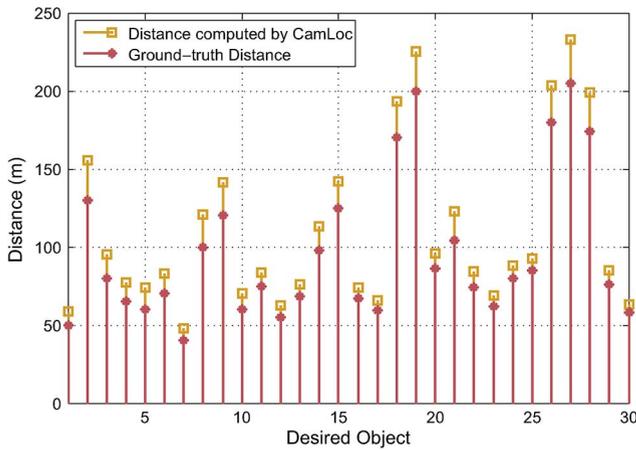


Fig. 15. Absolute ranging errors at 30 locations.

and fuzzy with the increasing distance s . The inaccurate scaling ratio, along with the erroneous phone displacement computation, finally leads to large ranging deviations for remote object localization.

Table 1 details the localization result after mounting location refinement module. According to the mapping result, CamLoc achieves around 89 percent object mapping accuracy in sparse environment. The accuracy stems from two aspects. On the one hand, the received GPS is more reliable and accurate in sparse areas, which contributes to precise physical coordinate estimation. On the other hand, the sparse distribution of buildings also assists in pruning the mapping candidates. In all, the precise physical coordinate estimation and the fewer mapping candidates considerably improve the mapping accuracy. Although CamLoc might occasionally misidentify the desired target in clustered scenarios, we still envision a significant mapping performance gain with advanced built-in sensors.

6 RELATED WORK

In this section, we broadly review two categories of research directly related to our work.

6.1 Image-Based Localization

There has been extensive research that employ cameras for object localization. Generally, these works can be divided into

two categories. The first is fingerprint-based localization [10], [11], [12], [13], [14]. Li *et al.* [10] proposed a system to describe outdoor places with local descriptors extracted from images taken around geographical places. In [11], the author proposed LandMarker, an image-based localization system which leverages the content information of the image to assist landmark recognition. Schindler *et al.* [12] introduced a generalization of the traditional vocabulary tree search algorithm which could improve the performance of landmark matching process. In [13], the author proposed a location recognition method which exploits the appearance and geometric structure of an environment to achieve both high efficiency and good generalization. Hays *et al.* [14] designed an algorithm to estimate image location as a probability distribution over the earth's surface based on a data-driven scene matching approach. Finally, a location with the highest matching ranking is sent back to the mobile user. Such method has been successfully applied in tourism guidance [15], landmark recognition [16] and geography tagging [17]. However, the high deployment cost, along with cockamamie updating process, together impedes such method becoming mainstream. Different from this approach, CamLoc is a light-weighted object localization system. It eliminates the need for database construction and database maintenance, which paves a way for ubiquitous object localization services.

The second category leverages 3D reconstruction techniques (STERO and SFM) for object localization [18], [19], [20], [21]. Sattler *et al.* [18] proposed a 2D-to-3D matching framework based on visual vocabulary quantization and a prioritized correspondence search for real time localization. In [20], the authors designed a two-step algorithm to estimate where a given photo or video was taken based on the tags that a user has assigned to it. Different from [20], the authors in [21] proposed a new method to determine the location of the recording place of Flickr videos based on both the textual data and visual cues. OPS [19] is the most representative system that is similar to CamLoc. It attempts to localize a distant object via the calibration of CV techniques and phone-equipped sensors. Similar to SFM technique, OPS requires users to photograph the object from various sites and send these photos to the computational center for processing. After that, the distance between the user and the object will be computed via CV techniques. However, the complex photograph survey and high latency query

TABLE 1
Illustration of Physical Location Mapping Results over 30 Buildings

No.	Description	Environment	Mapping Result	No.	Description	Environment	Mapping Result
01	Shopping-mall	Dense	Right	16	Airport	Sparse	Right
02	School	Sparse	Right	17	Shopping-mall	Dense	Right
03	School	Sparse	Right	18	Shopping-mall	Dense	Right
04	Gas station	Sparse	Right	19	School	Sparse	Right
05	Shopping-mall	Dense	Wrong	20	School	Sparse	Right
06	Airport	Sparse	Right	21	Civic buildings	Dense	Wrong
07	Hotel	Dense	Wrong	22	Civic buildings	Dense	Right
08	Airport	Sparse	Right	23	Historical buildings	Dense	Wrong
09	Hotel	Dense	Wrong	24	Civic buildings	Dense	Right
10	Hotel	Dense	Right	25	Shopping-mall	Dense	Right
11	Shopping-mall	Dense	Right	26	Shopping-mall	Dense	Right
12	School	Sparse	Wrong	27	Overline bridge	Dense	Right
13	Restaurant	Dense	Wrong	28	Mall	Dense	Right
14	Restaurant	Dense	Right	29	School	Dense	Right
15	Hotel	Dense	Right	30	School	Dense	Right

process impede its prevalence in delay sensitive applications like parking space searching. Conversely, there is no need for CamLoc to photograph the object at various sites or utilize complex image processing techniques for reconstructing the 3D structure of the target. This, we believe, makes it much more suitable for location-based services.

6.2 Phone-Based Localization

There are plenty of works employing mobile phones for localization. For indoor cases, mainstream researches are focusing on employing various sensors on mobile phone for localization purpose. SurroundSense [22] utilizes various sensors on the mobile platform to collect fingerprints signals for logical localization. WILL [23] proposes an indoor localization technique by utilizing user movement patterns and WiFi signal trace matching. UnLoc [24] bypasses the need for war-driving by leveraging identifiable signature on varied sensing dimensions. For outdoor environment, avoiding the over-dependence on GPS system remains an active problem. Escort [25] obtains cues from social encounters by using an audio beacon infrastructure. ALoc [26] dynamically trade-offs location accuracy and energy consumption of mobile phone based on the probabilistic model of sensing errors and user location. CAPS [27] uses a cell-ID sequence matching technique to estimate the real-time position of mobile phone on the history of cell-ID and GPS position sequences. These systems focus solely on one part of localization, e.g., receiving the estimation of self-location, while CamLoc aims at addressing localizing remote object in real-time manner, which complements the mainstream research.

7 CONCLUSION

In this paper, we present the design, implementation and evaluation of CamLoc, a simple yet practical system to localize remote object via the collaboration of phone sensors and photos. Although the core idea of CamLoc has been known for centuries, its adoption to pervasive object localization service is previously lacking. Our system enables a user shooting an object twice using a smartphone at a fixed location to immediately know the location of the object in global coordinates, based on which location-based services become available for the object. Blocked by sixpenny phone sensors, the localization accuracy of CamLoc is still undesirable. However, we believe that CamLoc explores the possibility of making object localization as easy as possible, taking a significant step towards pervasive object localization services.

ACKNOWLEDGMENTS

This work is supported in part by the NSFC Major Program 61190110, National Basic Research Program of China (973) under grant No. 2012CB316200, NSFC Distinguished Young Scholars Program under Grant 61125202, NSFC grants under No. 61171067, 61133016 and 61373175.

REFERENCES

- [1] POIdo. [Online]. Available: <http://www.poido.com/>
- [2] Mapquest. [Online]. Available: <http://www.mapquest.com/>
- [3] Where. [Online]. Available: <http://www.where.com/>

- [4] Urbanspoon. [Online]. Available: <http://www.urbanspoon.com/>
- [5] Dianping. [Online]. Available: <http://www.dianping.com/>
- [6] [Online]. Available: http://en.wikipedia.org/wiki/Sobel_operator
- [7] [Online]. Available: <http://graphics.stanford.edu/courses/cs178/applets/gaussian.html>
- [8] M. Fischler and R. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *ACM Commun.*, vol. 24, no. 6, pp. 381-395, June 1981.
- [9] M. Beckler, "Accelerometer-Based Inertial Navigation," *Elect. Comput. Eng. Dept., Univ. Minnesota, Minneapolis, MN, USA, Tech. Rep.*, 2008.
- [10] Y. Li and J.H. Lim, "Outdoor Place Recognition Using Compact Local Descriptors and Multiple Queries with User Verification," in *Proc. Multimedia*, 2007, pp. 549-552.
- [11] A. Qamra and E.Y. Chang, "Scalable Landmark Recognition Using EXTENT J. *Multimedia Tools Appl.*, vol. 38, no. 2, pp. 187-208, Jun. 2008.
- [12] G. Schindler, M. Brown, and R. Szeliski, "City-Scale Location Recognition," in *Proc. CVPR*, 2007, pp. 1-7.
- [13] K. Ni, A. Kannan, A. Criminisi, and J.M. Winn, "Epitomic Location Recognition," in *Proc. CVPR*, 2008, pp. 1-8.
- [14] J. Hays and A. Efros, "IM2GPS: Estimating Geographic Information from a Single Image," in *Proc. CVPR*, 2008, pp. 1-8.
- [15] Y. Zheng, M. Zhao, Y. Song, H. Adam, U. Buddemeier, A. Bissacco, F. Brucher, T. Chua, and H. Neven, "Tour the World: Building a Web-Scale Landmark Recognition Engine," in *Proc. ICCV*, 2009, pp. 1085-1092.
- [16] L.S. Kennedy and M. Naaman, "Generating Diverse and Representative Image Search Results for Landmarks," in *Proc. WWW*, 2008, pp. 297-306.
- [17] M. Larson, M. Soleymani, P. Serdyukov, S. Rudinac, C. Wartena, V. Murdock, G. Friedland, R. Ordelman, and G.J.F. Jones, "Automatic Tagging and Geotagging in Video Collections and Communities," in *Proc. ICMR*, 2011, p. 51.
- [18] T. Sattler, B. Leibe, and L. Kobbelt, "Fast Image-Based Localization Using Direct 2D-to-3D Matching," in *Proc. ICCV*, 2011, pp. 667-674.
- [19] J. Manweiler, P. Jain, and R.R. Choudhury, "Satellites in Our Pockets: An Object Positioning System Using Smartphones," in *Proc. MobiSys*, 2012, pp. 211-224.
- [20] O.V. Laere, S. Schockaert, and B. Dhoedt, "Finding Locations of Flickr Resources Using Language Models and Similarity Search," in *Proc. ICMR*, 2011, p. 48.
- [21] G. Friedland, J. Choid, H. Lei, and A. Janin, "Multimodal Location Estimation on Flickr Videos," in *Proc. MM*, 2011, pp. 23-28.
- [22] M. Azizyan, I. Constadache, and R.R. Choudhury, "SurroundSense: Mobile Phone Localization via Ambience Fingerprinting," in *Proc. MobiCom*, 2009, pp. 261-272.
- [23] C. Wu, Z. Yang, Y. Liu, and W. Xi, "WILL: Wireless Indoor Localization without Site Survey," in *Proc. INFOCOM*, 2012, pp. 64-72.
- [24] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R.R. Choudhury, "No Need to War-Drive: Unsupervised Indoor Localization," in *Proc. MobiSys*, 2012, pp. 197-210.
- [25] I. Constadache, X. Bao, M. Azizyan, and R.R. Choudhury, "Did You See Bob? Human Localization Using Mobile Phones," in *Proc. MobiCom*, 2010, pp. 149-160.
- [26] K. Lin, A. Kansal, D. Lymberopoulos, and F. Zhao, "Energy-Accuracy Aware Localization for Mobile Devices," in *Proc. MobiSys*, 2010, pp. 1-14.
- [27] J. Peak, K. Kim, J. Singh, and R. Govindan, "Energy-Efficient Positioning for Smartphones Using Cell-ID Sequence Matching," in *Proc. MobiSys*, 2011, pp. 293-306.
- [28] OpenCV. [Online]. Available: http://opencv.itseez.com/doc/tutorials/imgproc/imgtrans/sobel_derivatives/sobel_derivatives.html#code



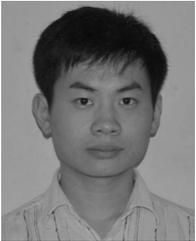
Longfei Shangguan received the BS degree from the School of Software, Xidian University, China, in 2011 and is currently working toward the PhD degree in the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong. His research interests include pervasive computing, wireless sensor networks and RFID system. He is a member of ACM. He is a Student Member of the IEEE.



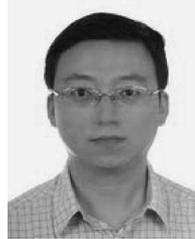
Zimu Zhou received the BE degree from the Department of Electronic Engineering, Tsinghua University, Beijing, China, in 2011 and is currently working toward the PhD degree in the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong. He is a Student Member of the IEEE and ACM.



Zhenjiang Li received the BE degree from the Department of Computer Science and Technology, Xi'an Jiaotong University, and the MPhil degree from the Department of Electronic and Computer Engineering, and the PhD degree from the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong. He is currently working in the Wireless and Distributed Sensing (WANDS) research group, School of Computer Engineering, Nanyang Technological University, Singapore. He is a member of the IEEE and ACM.



Zheng Yang received the BE degree from the Department of Computer Science, Tsinghua University, Beijing, China, and the PhD degree from the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong. He is currently an Assistant Professor at TNLIST, School of Software, Tsinghua University. He is a member of the IEEE and ACM. He has been awarded the 2011 State Natural Science Award (second class).



Xibin Zhao is now an assistant professor at School of Software, Tsinghua University. He received the BS, M.E and Ph.D degrees in School of Computer Science and Telecommunication Engineering from Jiangsu University in 1994, 2000, 2004 respectively. His research interests include reliability analysis of hybrid network systems and information system security.



Kebin Liu is currently an assistant researcher in school of software and TNLIST, Tsinghua University. He received his BS degree in Department of Computer Science from Tongji University, in 2004, and the MS and Ph.D degrees in Shanghai Jiaotong University, in 2007 and 2010. His research interests include sensor networks and distributed systems.



Yunhao Liu received the BS degree in automation from Tsinghua University, China, in 1995, and the MS and PhD degrees in computer science and engineering from Michigan State University, East Lansing, MI, in 2003 and 2004, respectively. He is now a Professor at TNLIST, School of Software, Tsinghua University. His research interests include wireless sensor network, peer-to-peer computing, and pervasive computing. He is a Senior Member of the IEEE, IEEE Computer Society, and an ACM Distinguished Speaker.

▷ For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.