SmartLoc: Push the Limit of the Inertial Sensor Based Metropolitan Localization Using Smartphone

Cheng Bo, Xiang-Yang Li, Taeho Jung
Department of Computer Science, Illinois Institute of Technology, Chicago IL
Email: cbo@hawk.iit.edu, xli@cs.iit.edu, tjung@hawk.iit.edu

ABSTRACT

GPS is the most widely used solution for localization and navigation. However, it often works poorly in area with tall buildings in metropolises. We present *SmartLoc*, a system to estimate the location and the traveling distance by leveraging the lower-power inertial sensors embedded in smartphones. To minimize the negative impact of sensor noises, *SmartLoc* exploits the intermittent strong GPS signals and uses the linear regression to build a prediction model which is based on the trace estimated from inertial sensors only and the one computed from the GPS. Furthermore, we extensively rely on automatically detected landmarks (*e.g.*, bridge, traffic lights) and special driving patterns (*e.g.*, turning, uphill, and downhill) from inertial sensor data to improve the localization accuracy when the GPS signal is weak. Our evaluations of *SmartLoc* in a large city demonstrates its technique viability and significant localization accuracy improvement compared with GPS and other approaches: the error is approximately 20m for 90% of time while the mean error of GPS is 42.22m. *SmartLoc* system can also reduce the energy consumption by at least 10% for localization in other scenarios by carefully turning on GPS periodically without sacrificing localization accuracy.

Keywords

SmartLoc, Inertial Sensor, Localization, Smartphone.

1. INTRODUCTION

Localization have attracted significant attentions in the past few decades, and numerous techniques have been proposed to achieve high accuracy localization. In outdoor scenarios, GPS (Global Positioning System) or its variants are the most common technologies to provide accurate position [9] for various applications, such as trace and tracking [40, 48] in the wild, and environmental monitoring [27]. However, problems regarding low accuracy of GPS in critical regions such as metropolises have proposed the idea of war-driving and created the state of art large scale WiFi/GSM fingerprint database for positioning, like Skyhook [1]. These methods often sample and establish fingerprint databases, which are computationally intensive.

To study the severity of the GPS localization errors in metropolises, we conducted comprehensive experiments to evaluate the location accuracy through integrated GPS in smartphones. We collected the GPS information (location, and the location accuracy) in all road segments in downtown Chicago. Our measurement demonstrates that, due to complicated road structures(*e.g.*, tunnels, highrises, and underground), the GPS signals are very weak and unstable, or even totally blocked. In addition, the largest location error is over 100m on the ground, and nearly 400m in the underground segments (see Fig. 1 for more details). Thus, improving the location accuracy is imperative when the GPS signal is weak in metropolises.

A number of novel techniques have been proposed in the literature to remedy the inaccuracy of GPS localization in metropolises [1, 9, 15, 32, 42], to reduce the energy consumption of GPS process [23, 25] or time complexity [17]. In this work, we leverage existing sensors (*e.g.*, GPS, accelerometer, gyroscope, and magnetometer sensors) that are already integrated in most smartphones to provide more accurate localization. Exploiting the data collected from these inertial sensors has been used in the literature to address a number of challenging and interesting tasks, *e.g.*, indoor localization [10, 24, 43, 45], road condition monitoring [12, 28], property tracking [15], activity recognition [4, 20, 39], measuring walking speed [10], gaming [26, 49], and outdoor localization [15, 18, 33]. In contrast to these novel solutions for various purposes, the main challenges in providing an accurate location (or trajectory) in metropolises for vehicles using inertial sensors include:

1. The inherent noise, inaccuracy, and imprecision in the sensor data [15, 43] cause significant drift (from cumulative errors) in estimated location or trajectory when simple dead-reckoning methods are used.
2. The coordinating systems on the smartphone (called body frame coordinate (BFC)) and earth (called earth frame coordinate (EFC)) are different. The conversion and extraction of linear acceleration from BFC to EFC introduces additional noise, which will incur large localization errors if not carefully addressed.
3. Even we mount the smartphone on the windshield during the navigation, its dynamic rotation (caused by driving vibration) still introduces additional errors in lo-

cation estimation.[1] It is thus extremely challenging to calculate the linear acceleration in EFC while minimizing the introduced errors.

Note that accelerometer has been used to measure the walking speed and distance of pedestrian in outdoor environment [7, 10, 37, 43]. Such applications exploit walking steps to estimate the distance using step counts and stride-length, and then use compass to estimate the direction. However, providing realtime localization of driving cars in metropolises is far more challenging as such activity does not have a cyclic pattern in sensor data, in contrast to people walking. To address these challenges, during the dead reckoning process for calculating the current position of a car using previously estimated location and the data from sensors, we propose a novel strategy to reduce the impact of inherent noise and inaccuracies of sensor data, and improve the estimation of driving speed and moving distance consequently.

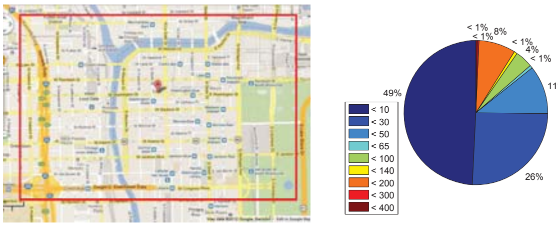
In order to further improve the localization accuracy, in *SmartLoc* we design a calibration strategy based on road infrastructures (*e.g.*, bridge, traffic lights, uphill, and downhill) and driving status (*e.g.*, turns, stops), which are inferred from the sensor data, such as accelerometer, gyroscope, and magnetometer. Our extensive evaluations show that using inertial sensors can accurately identify special road infrastructures using either fingerprint based approaches or pattern-matching technique. Note that unlike the traditional fingerprint based localization, *SmartLoc* exploits the current coarse-grained estimation of location from dead-reckoning to confine the search space which often has only one candidate matching left. Based on the matched road infrastructures or driving status, SmartLoc can provide a much accurate localization. Our evaluations show that turning (left or right), uphill and downhill provide localization accuracy within a few meters while traffic lights and stop signs provide less accurate localization: the curve of 'SmartLoc' is closer to the ideal diagonal line than the one of 'Inertial Sensor & Traffic Light' in Figure 9(e).

In this work, we implement a prototype, **SmartLoc** on Android, and evaluate the localization performance in both downtown Chicago and highway. Our extensive test results in the majority of blocks in Chicago indicate that SmartLoc improves the location accuracy significantly: 1) the mean localization error in each time slot is 11.65m; 2) the proportion of "good" road segments, where the localization error is less than 20m, is increased from $\leq 50\%$ (by purely using GPS) to $\geq 90\%$ using SmartLoc in downtown area. When testing SmartLoc on highway, the localization error is at most 12m for 95% of the time. In comparison the state-of-the-art localization scheme for driving vehicles, AutoWitness [15], only produces the error of distance estimation less than 10% for most of the cases, which could be large when the estimated distance is long (*e.g.*, 10% of the 2 miles driving is 320m). Our results also imply that SmartLoc can save the energy consumption by turning off the GPS periodically.

The rest of paper is organized as follows. We demonstrate our measurement results and observations on the GPS accuracy in Section 2. We then present the overview of SmartLoc in Section 3 and the backgrounds for our system presentation in Section 4. In Section 5, Section 6 and Section 7, novel calibration techniques of SmartLoc are presented one by one. We report our detailed real world experiment results in Section 8, review the state of the art localization techniques in Section 9, and conclude the work in Section 10.

2. GPS ACCURACY MEASUREMENT

To study how bad the GPS location accuracy could be, we first conduct a comprehensive measurements of GPS accuracies within the red rectangular area in Downtown Chicago, as shown in Figure 1(a). We find that in the test area, the largest location error reaches 400m, and the distance of the longest road segment between two GPS location with reasonable accuracies ($\leq 30m$) is about 1km.



(a) Downtown Chicago (b) Proportion of Error

Figure 1: GPS localization accuracy in Chicago.

2.1 Measurement in Chicago Downtown

We drive through every road segment in the area, and record the location information including Longitude, Latitude, Accuracy, and speed in every 50ms from smartphone. Meanwhile, we also collect all the sensory data with same frequency, which will be used to identify the driving condition. To remove time dependent GPS location errors, we conducted three independent measurements at three different times. The results reported here are the average of these three measurements.

2.2 Original Location Results

The location accuracy is not so high as expected based on our preliminary measurement. The smallest error we collect from GPS is 5m, while the largest error reaches 400m, which is nearly the length of three blocks in downtown, and the mean error in the Chicago downtown area is 42.22m. Such high errors will lead to wrong instruction for turning or stopping in navigation.

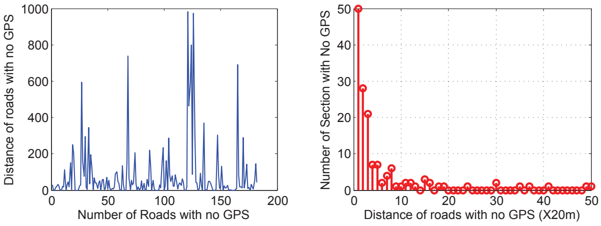
We also plot the localization accuracy information of downtown Chicago based on the measurement in Figure 1(b). According to the statistic data, only about half of the sampling

[1] The accelerometer values are the acceleration speeds in three axes of the BFC (coordinate is uniquely determined by the orientation of the smartphone). On the other hand, for localization, we need compute the integration of acceleration in EFC. It has been well-documented in the literature (*e.g.*, [15]) that the rotation of smartphones will introduce large errors in speed and location estimation.

points endure the error of less than 20m, over one quarter of the location have an error of about 50 m, and the rest quarter have an error larger than 50m.

As a fundamental step for navigation and location based services, we assume that the largest location error we could accept in the metropolis should be less than 30m, which is about a quarter of one block. For simplicity, we denote the position with GPS location error less than 30m as the location with **good GPS signal**, and the rest as **bad GPS signal**.

Typically, the navigation system works poorly only under the circumstance that a segment of roads are with bad GPS signal, and such high error will lead to wrong instruction for turning or stopping in navigation. Therefore, we found 182 bad segments of roads, see Figure 2(a), from our statistical information, with the longest length being almost one kilometer. Although the average length of these bad road segments is approximately 200m, as shown in Figure 2(b), the bad segment roads with length over 400m are concentrated in the center of downtown.



(a) Bad road segments. (b) Number of bad segments.

Figure 2: Road segments with poor GPS

Numerous techniques have been proposed in the literature to improve the GPS localization accuracy, such as A-GPS, D-GPS (Differential GPS) [34], and WAAS (Wide Area Augmentation System) [11]. WiFi signal [1, 9] and cellular signal [6, 42] have also been used to find the location. Unfortunately, these solutions are either prohibitive or inapplicable for navigation in metropolises. The localization based on cellular signal has the worst location accuracy (median error at 100m for downtown environment [6]), the localization based on WiFi signals also has poor location accuracy and it relies on nearby WiFi APs' locations. Unfortunately, for many critical road infrastructures, such as under ground roads and multilayered roads, the GPS signal is often lost, the GSM signal is weak, and there are no WiFi access points. Thus, traditional GSM or WiFi based approaches [9, 14] are all impractical for metropolises.

3. SYSTEM OVERVIEW

The purpose of **SmartLoc** is to use inertial sensors in the smartphone to estimate the movement of the vehicle, and lively provide locations based on the traveling distance and orientation with high accuracy but low energy consumption. Remarkably, we not only address the inaccuracy caused by the complex infrastructures in downtown area, but also exploit them to improve the localization accuracy in turn.

3.1 Main Idea

Our main idea is to propose a newly developed localization model to compensate the inaccurate localization in downtown area. SmartLoc collects both GPS information and sensor data, and conducts coarse noise reduction by computing the moving average with a certain sliding window. Since the large inherent error from sensors will lead to poor distance estimation, our system uses Extended Kalman Filter (EKF) to estimate the orientation before extracting the linear acceleration of the vehicle. In the following stage, we propose a self-learning predictive regression model to estimate the moving distance based on the extracted acceleration, in which the accumulated errors are minimized in the following way: SmartLoc switches to the training process when GPS signal is good, and train the predictive model. WHen GPS signals are unreliable, it uses the trained model to predict the travelling. Due to the complex road conditions and unpredictable driving behaviors, the training process should be updated periodically in our model. In addition, SmartLoc also detects the *landmarks* by finding special patterns from sensor data when the car drives through bridges, tunnels, traffic lights or turning points, and it calibrates the estimation accordingly.

3.2 Challenges

Many technical issues should be considered here. The first issue is how to deal with the huge noise in the magnetometer, which hinders correct estimation on the smartphone's orientation. The second one is how to minimize the errors caused by the noises coming from the accelerometer since naive method using Newton's Law will accumulate the noises (double integral on acceleration results in the displacement, but the noises are doubly accumulated also). The last challenge is how to detect and recognize the landmarks which will be used to improve the localization accuracy in our system.

4. BACKGROUNDS

4.1 Sensor Characteristics and Data Extraction

The behavior of the integrated accelerometer acts like a damped mass on a spring. Essentially, the mass will endure the force coming from both the gravity and the external force, and the readings reflect the resultant force. Therefore, we employ a low-pass filter to remove the gravity recursively and obtain the external force and thus its linear acceleration simultaneously. The gyroscope is used in smartphones to measure the angular velocity of the devices based on the principles of angular momentum. The magnetometer monitors the changes of the magnetic field surrounding the smartphone, and provides raw field strength data (in μT) for each of the three axes in BFC.

4.2 Mechanical Noise of Sensors

Although the sensors provide acceleration, angular velocity and the magnet strength, the intrinsic mechanical noise of the sensors make the naive distance estimation based on Newton's Law impossible due to the accumulated noises.

We put a smartphone on the ground in three different postures to analyze the noises in three sensors. We also calculate both the mean value and the standard deviation for all three axes for all the sensor (Figure 3). The readings from the accelerometer in three axes fluctuate frequently over the time domain, and the errors also change in different orientations. The fluctuation of the magnetometer is much larger than the one of the accelerometer even if the smartphone is deployed in the same posture. Meanwhile, we observe from our tests in inside and outside scenarios (of a car) that the magnetometer is sensitive to the environment. Both the mean and the standard deviation inside the car are approximately over 100 times larger than that of the outside case because of the engine. We have also found that the readings fluctuate more irregularly when the car is being driven, thus the traditional method which converts the linear acceleration from BFC to EFC by a rotation matrix brings huge errors consequently.

Surprisingly, the gyroscope is much stabler than the other two sensors with less standard deviation (Figure 3(e), Figure 3(f)). The readings from all three scenarios (outdoor, engine off, engine on) are similar, which provide us a good opportunity to improve the orientation estimation.

4.3 Obtaining Orientation

Although smartphones are usually fixed in cars when they are used as navigators, the orientation of the smartphone changes irregularly because of both the driving direction and the vehicle vibration. The orientation calculation is the basic component of any localization scheme, and it has more significant role in SmartLoc because our system detects the movement conditions (e.g., turning, downhill, uphill) and the landmarks (e.g., tunnel, bridge) based on the orientations. Therefore, orientation estimation is a key component in our system.

As a rigid body, smartphones' orientation can be considered as the space status of the phone in respect to the earth. Then, the orientation of any smartphone can be characterized by the readings extracted from the sensors, which can be further converted from the Body Frame Coordinate(BFC) to the Earth Frame Coordinate(EFC). Generally, the orientation of a smartphone is the posture relative to the EFC, or more specifically, the magnetic north. Theoretically, the orientation can be determined by the magnetometer combined with the accelerometer, represented by the Euler Angle *pitch* (θ), *roll* (ϕ), and *yaw* (φ) respectively, where

$$\theta = \arctan\frac{G_u x}{G_u z}, \phi = \arcsin G_u y, \varphi = \arctan\frac{E_u y}{N_u y} \quad (1)$$

In these equations, the unit vectors of the EFC project to the BFC for three axes are G_{u_d}, E_{u_d}, and N_{u_d}, where *d* refers to the three axes *x*, *y*, and *z*. *G* denotes the gravity extracted from the accelerometer, and *B* represents the readings from the magnetometer. *E*, in addition, illustrates the cross product as $E_u = G_u \times B_u$ while $N_u = E_u \times G_u = (G_u \times B_u) \times G_u$.

5. ORIENTATION CALIBRATION

Although the differences of the orientations in different timeslots are computable from the above equations in Section 4.3, the actual orientation of the smartphone in regard to the earth is hard to achieve because of the noise of the magnetometer. Therefore, we propose an improved method by getting an initial orientation estimation based on accelerometer and magnetometer, and then building an estimation model through gyroscope and calibrate through Extended Kalman Filter (EKF).

In order to eliminate the device-dependent error and the fluctuation on the sensory data, we take a 10 seconds' initial orientation measurement by sampling average sensor readings, and then subtract the average readings from the consequent readings to calibrate them.

In the following sensor fusion process, the orientation at the time *k* is related to the posteriori estimation of the orientation at the time *k*-1. We present the non-linear stochastic difference equation as:

$$q_k = \widetilde{q}_k + A(Q_{am} - h_k \widetilde{q}_k); \qquad \widetilde{q}_k = R_k + q_{k-1} \quad (2)$$

where q_k indicates the estimated orientation at time *k*, Q_{am} denotes the orientation based on current accelerometer and magnetometer, and $\widetilde{q}_k$ presents the priori estimate at the time *k*, which is calculated from the last orientation and the rotation increment. Here A, h_k, and R_k are Kalman gain, measurement matrix, and rotation increment respectively. The rotation increment $\overline{R_k}$ is the angle increment at the time *k*, which is given as:

$$\overline{R}_k = \Omega \cdot [\textit{pitch roll yaw}]^T \cdot \Delta t \quad (3)$$

where Δt is the time interval of our sampling, and Ω is

$$\Omega = \begin{bmatrix} 1 & \sin\phi \cdot \tan\theta & \cos\phi \cdot \tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi / \tan\theta & \cos\phi / \tan\theta \end{bmatrix} \quad (4)$$

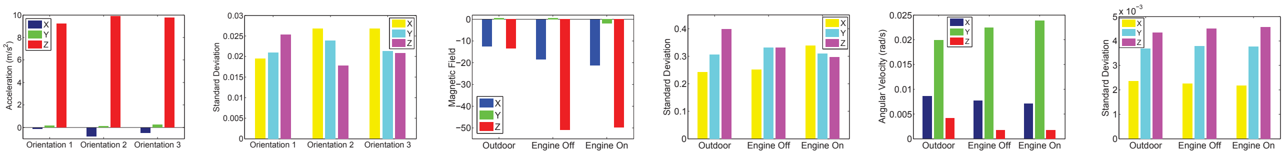
The EKF brings a continuously changing orientation of the smartphone, and the improved orientation in the consequent time slot can be achieved from:

$$\widetilde{q}_k = q_{k-1} + \overline{R}_k \quad (5)$$

6. DISTANCE CALIBRATION

6.1 Self-learning Predictive Model

According to the Newton's Law, we can achieve the distance after applying a double integration on the acceleration. However, the noises from the accelerometer are accumulated during the integral, and the error gets enormously huge in just several minutes. However, we observe from our preliminary experiment (Section 2) that the majority of the road



(a) Mean Acc. (b) SD Acceleration (c) Mean Magnetic (d) SD Magnetic (e) Mean Gyroscope (f) SD Gyroscope

Figure 3: Mean value and Standard Deviation for Accelerometer, Magnetometer, and Gyroscope respectively

segments with bad GPS signals (error $\geq$ 30m) are shorter than 400m, which takes only 20-30 seconds to drive through. This distance is long enough to navigate drivers to wrong places, but is also short enough to endure the errors to some extent. Therefore, we propose the following predictive linear regression model which adaptively calibrates itself using the GPS signals and the dead reckoning model.

Speed Estimator: The velocity V_i at the end of a timeslot i can be represented as

$$V_i = V_{i-1} + \beta \cdot a_i \cdot \Delta t + \mu \quad (6)$$

where β is the parameter to be learned and adjusted in real time, a_i is the mean of the measured acceleration during the timeslot i, and μ is the noise.

When GPS signals are strong, we achieve V_i and V_{i-1} from the GPS, and the mean acceleration a_i is computed from the accelerometer. Then we regress the model to find the best β.

When GPS signals are weak, we use the trained model above to predict the velocity V_i, where μ is sampled from a normal distribution with mean 0.

Distance Estimator: When GPS signals are strong, let $G(\Delta t_i)$ be the distance during a timeslot i read from GPS, which also contains errors (denoted as η). The distance provided by GPS can be denoted as

$$G(\Delta t_i) = \lambda_1 \cdot V_{i-1} \cdot \Delta t + \frac{1}{2} \cdot \widehat{a_i} \cdot \Delta t^2 + \eta$$

where $\widehat{a_i}$ is the actual acceleration in the time slot i. Here λ_1 is multiplied to reflect the error in the estimated speed V_{i-1} for the time slot $i-1$. We only know the measured acceleration a_i, which is defined as

$$a_i = (1+\varepsilon)\widehat{a_i} + \delta$$

Then, we use the following formula to estimate the distance $G(\Delta t_i)$:

$$G(\Delta t_i) = \lambda_1 \cdot V_{i-1} \cdot \Delta t + \lambda_2 \frac{1}{2} \cdot a_i \cdot \Delta t^2 + \lambda_3 \cdot \Delta t^2 + \lambda_4 \cdot \Delta t + \eta \quad (7)$$

where λ_i's are the parameters to be learned by our regression model. When GPS signals are strong (GPS error is $\leq$ 20m), based on the V_{i-1}, a_i computed from the sensor data and the distance from GPS, we train the model Eq. (7). This model is in turn used to predict the distance $G(\Delta t_i)$ in the time slot i when GPS signals are weak. From the predicted moving distance $G(\Delta t_i)$, we can predict the location at the time slot i.

6.2 Movement Detection

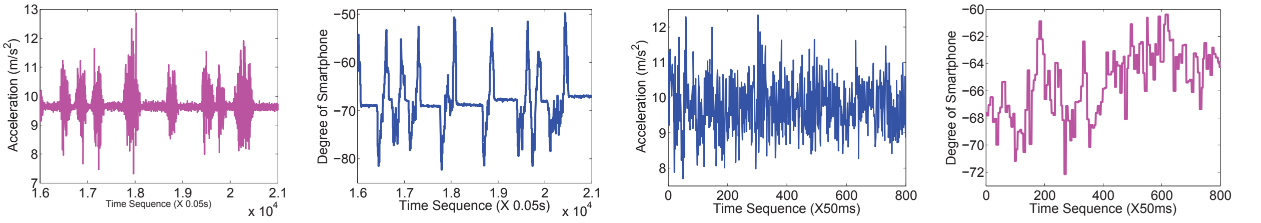
One important problem should be tackled before using our predictive regression model. Remember that the speed estimator estimates the speed based on the accelerometer, and the speed contains noises accumulated from the integral on the acceleration. Therefore, when the vehicle stops, the estimated speed is highly likely to be non-zero, which leads to a huge error in the final prediction. Therefore, determining whether the vehicle is moving or static further reduces the negative impact of the mechanical noises.

During our preliminary experiments, we discovered that the movement can be reflected precisely from the accelerometer and the gyroscope. We drove a few blocks, and plot the readings of both gyroscope and acceleration in Figure 4. The acceleration fluctuates frequently when the vehicle is being driven, and remains relatively stable when it stops (Figure 4(a)). We compute the variance for both cases. When the car stops, the variance is less than 0.01. the variance when the car is driven is much larger, which is over 0.6 most of time. Even in the cruise mode, which makes a car drive at a constant speed (in Figure 4(c)), due to the vibration of the car, the variance of the acceleration is still greater than 0.4. Although the smartphone is usually mounted to the windshield[2], due to the inertia while driving, especially speeding up and brake, the angle between BFC and EFC will change. In Figure 4(b) and 4(d), we found that the angle reflects the vehicle's movement as well. The variance for stopping, moving, and the cruise mode are 0.0228, 28.2620, and 5.638 respectively. SmartLoc continuously calculates the variance for both accelerometer and gyroscope sensors using a slide window, and calibrate the speed whenever the vehicle stops. In our experiment, we found that SmartLoc can distinguish between moving and stopping precisely.

7. CALIBRATION BY LANDMARKS

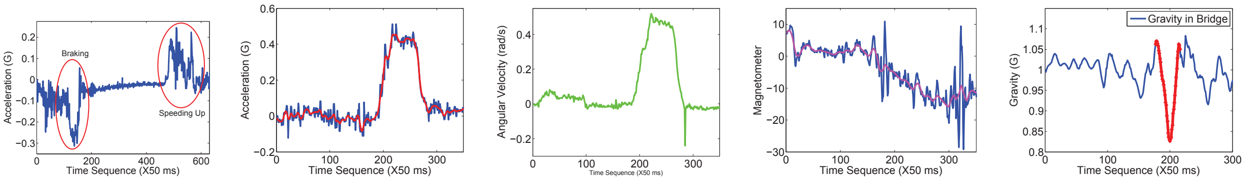
As we mentioned above, the road infrastructures, including tunnels, bridges, crossroads and traffic lights, cause large noises in the GPS data, which results in large drift in the distance estimation if it is not treated rigorously. In this work, we exploit the precise location of these infrastructures that are available in Google Map to calibrate the localization without any extra cost.

[2] In this case, the rotation of smartphone has only one degree of freedom. Thus we only report one relative angle between the BFC and EFC. The angle is computed using data from accelerometer, gyroscope, and magnetometer as in Section 4.3.



(a) Acceleration (b) BFC/EFC angle (c) Acceleration in Cruise (d) BFC/EFC angle in Cruise

Figure 4: The Acceleration and Angle while driving in city or Cruise model.



(a) Traffic lights (b) Centripetal force (c) Angular velocity (d) Magnetometer (e) Bridge

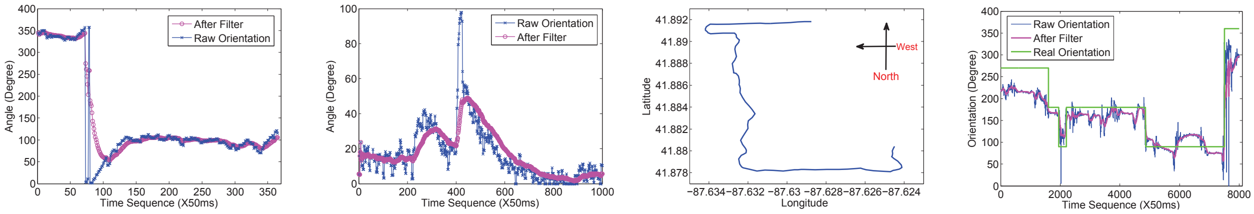
Figure 5: Pattern of the sensor data collected in different road infrastructures when driving: (a) car stopping and crossing a traffic light; (b), (c), and (d) car turning 90^o; and (e) car crossing a bridge.

Traffic Light: Driving in many large well organized cities, such as Chicago, one often encounters lots of traffic lights. When the vehicle stops at the traffic lights and drives through the crossroad, unique patterns appear in the readings from the sensors (Figure 5(a)). When vehicles encounters traffic lights, the whole process can be divided into braking and speeding up. The acceleration falls below zero when the car brakes, reaching the lowest point at the very moment when vehicle stops, and gets back to zero swiftly. As soon as the green light turns on, vehicle speeds up with the increasing acceleration. However, in rush hours with terrible traffic, the location where cars stop may not be near the crossroad, but with a certain distance from the crossroad. In this case, SmartLoc adjusts the moving distance based on the estimated stopping location from the empirical data, *i.e.*, subtracting the distance from the car to the crossroad. Empirically, the distance between the car and the crossroad is determined by the traffic condition. However, it is difficult to measure the exact distance from the car to the crossroad. The main approach adopted by SmartLoc is to subtract the $\frac{n \cdot L}{2}$, where L indicates the average length of a vehicle, and n represents the current possible number of vehicles waiting for the green light. We calculate the number of vehicle based on the observed data, and n is also related to different time periods.

Turning: Sometimes, vehicles may need to turn at the traffic light, and during the turning, all the sensors can detect the turning. Figure 5(b) indicates the centripetal force sensed by the accelerometer, and the scale of the acceleration depends on the speed at which the vehicle is turning. Simultaneously, the angular velocity sensed by the gyroscope also reaches up to 0.5 rad/s in our test case (Figure 5(c)), and the data from the magnetometer changes as well with a large fluctuation. Finally, the orientation of the smartphone also changes approximately 90 degrees when turning left or right. Such angle change is observed mainly along the Z axis on EFC, and the reading $0, 90, 180, 270$ represent north, east, south, and west respectively. Although the angle may not be accurate enough due to the large noise in the magnetometer (the maximum error we experienced was approximately 30^o), we are still able to correctly determine the road segment to which the car is turning. In Figure 6(a), vehicle turns from the north, the angle is from about 350^o to 100^o, which is east. We also compare the measured angle difference for turning and lane changing (Figure 6(b)) since lane changing can be wrongly detected as a turning. The angle difference when a car changes its lane is much smaller than the one when a car make a turn. In addition, we also calculated the standard deviation for the angle differences in lane changing, which is less than 10. Thus, distinguishing the turning and the lane changing is possible. We conducted more study on the driving orientation estimation. Figure 6(c) plots the raw trace of the vehicle achieved from the GPS with good signals, and Figure 6(d) illustrates the raw orientation generated only by the inertial sensors. We employ moving average to cancel some noises and calculate the driving orientation, which matches the ground truth.

Other possible road infrastructures that a vehicle may experiences are bridges, and tunnels. In our measurement, such patterns are more obvious and easier to be detected. After being converted from the BFC to the EFC, the acceleration along the gravity is less than $1G$ when the car is driving uphill and greater than $1G$ when driving downhill. A bridge or a tunnel often has both uphill and downhill patterns as shown in Figure 5(e).

In fact, certain driving patterns, such as turing left or right and stopping for traffic lights or stop signs, can be more accurately detected and thus classified. To classify other road infrastructures, we collect the sensor readings of those patterns to store as the fingerprints, and then match the real-



(a) Turning (b) Changing Lane (c) Driving Trace (d) Estimated Orientation

Figure 6: Turning or changing lanes, and Driving Trace

time sensor readings with the trained fingerprints. To improve the classification and the matching accuracy, we rely on the coarse-grained estimation of the location from dead-reckoning first, and then we further use our predictive regression model (Section 6) to confine the search space: only the road infrastructures (stored fingerprints) I within a certain distance δ from the estimated location x will considered as the matching candidate for the real-time pattern P achieved from the sensor data. We select the infrastructure that maximizes the *weighted matching score*:

$$\alpha M(I,P) + (1-\alpha)e^{-D(x,L(I))}$$

where $M(I,P)$ is the matching score between the fingerprint of an infrastructure I and the observed pattern P, $\alpha \in (0,1)$ is a constant, and $D(x, L(I))$ is the geodesic distance between the location x and the location $L(I)$ of infrastructure I. Then, the estimated location x is updated as the location $L(I^*)$ of the infrastructure I^* which maximizes the weighted matching score.

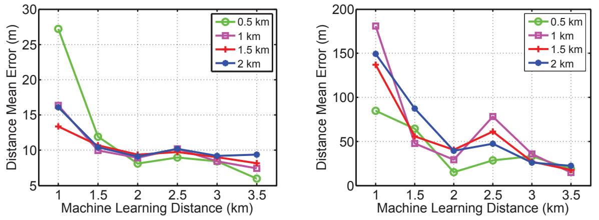
8. EXPERIMENTS AND EVALUATIONS

We conduct extensive evaluations of SmartLoc in different scenarios, both in downtown Chicago, and the highway. In our evaluation, Samsung Galaxy S3 with latest Android $4.1.2$ is mounted to the windshield, and we drive for over 100 different road segments in downtown Chicago ranging from 1km to 10km and over 30km in highway. Since the inertial sensors provide the driving orientations, we can compute the real-time location easily if we know the driving distance from the starting point and the driving orientation. Thus, the key problem becomes estimating the driving distance. We evaluate the travelling distance, road infrastructure recognition, accuracy, and energy consumption.

8.1 Traveling Distance Estimation

First of all, the typical frequency of the GPS update in a smartphone (0.5Hz) is much lower than that of the sensors (1Hz-20Hz). We denote the driving distance in a time slot as a *traveling segment*. We focus on the evaluation of the traveling distance estimation in two aspects: (1) the accuracy in distance estimating in traveling segments, and (2) the accuracy in final distance estimation of long road segments. Then, we analyze the performance in details in the rest of the section.

8.1.1 Prediction in City Without Using Landmarks



(a) Mean error in each slot (b) Mean overall distance error

Figure 7: Accuracy vs. Learning Distance.

We first test SmartLoc in a city scenario. We drive in downtown Chicago for over 30 different roads, where some roads have reliable GPS signals and some do not. We separate these roads into road segments (more than 100 segments) whose sizes are determined by our evaluations presented in the rest of the section. Similarly, we analyze the performance of SmartLoc in two phases as aforementioned.

Before we describe the performance of SmartLoc in a city, we note that the GPS signals in downtown Chicago are relatively poor and also time dependent. It is difficult to get the ground truth of all locations using smartphones. To address this, we evaluate SmartLoc in the peripheral area of downtown Chicago, where the buildings are not so high as in the center of the downtown, and GPS signals are relatively good. It is much easier to get the ground truth through GPS data, and in our evaluation, we simulate a bad GPS area by removing the GPS values and apply SmartLoc to calculate the location in those area to compare our results with the ground truth.

Obviously, the driving habit and road conditions in a city are different from those in the highway, and slight deceleration makes the predicted result deviate from the ground truth. We first evaluate the reliability of SmartLoc when different driving distances are used to train the system, ranging from 0.5km to 3.5km. Generally speaking, the accuracy increases when the learning distance increases as illustrated in Figure 7(a). In this figure, the X axis indicates the driving distance used for training our predictive regression model in SmartLoc, and the Y axis represents the mean distance (between the actual location reported by the GPS and the location estimated by our SmartLoc) in every time slot when we sample GPS locations (*i.e.*, every 2 seconds, or about every 22m when driving at the speed 40km/h). This experiment measures the accuracy of the prediction when we drive for over four different road segments with length from 0.5km to

2km (24 different cases in total). Due to the unstable driving behavior, short road segments for training SmartLoc leads to a large estimation error in each time slot. When SmartLoc learns only using the trace of 1km, the mean error in every time slot in different scenarios is around 15m, and the largest one is nearly 30m. When SmartLoc trains our predictive regression model using a longer trace, the mean estimation error decreases in all the test cases. The smallest error is less than 6m, which is less than half of the error when the training trace is 1km. We also observe that the error grows with the increase of the length of the test road segment in most scenarios. For example, by training SmartLoc using a trace of 3.5km, the mean error of the estimation in a 2km road segment is nearly twice of that when testing a 0.5km road segment.

We then evaluate the error on estimating the overall driving distance (Figure 7(b)). We evaluate SmartLoc for all the road segments and measure the error between the predicted driving distance and the actual driving distance for each segment (of all segments with distance from 0.5km to 2km) under different training traces. If SmartLoc learns the model for only 1km, the parameters in Eq. (7) cannot be computed accurately enough. Thus, the estimation errors increase to 180m in all our tests. When SmartLoc learns enough samples, the parameters are much more reliable, and the average accumulated error is below 30m, which is significantly better than the GPS in Chicago downtown.

8.1.2 *Prediction in City Using Landmarks*

SmartLoc calibrates the location as soon as it detects specific patterns, especially traffic lights and turnings. We test the performance of SmartLoc in a real drive route with the calibration using landmarks, and the result is presented as Figure 8. which is a bird's-eye view of the driving trajectory.

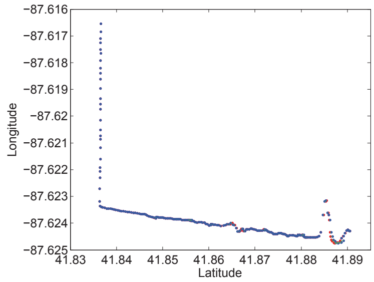
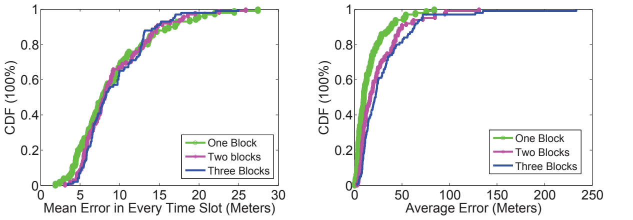


Figure 8: Localization In the Street.

The blue dots are the ground truth samples that we achieved from the GPS (where the GPS signals are good), and the red dots are the predicted locations from our SmartLoc with all calibration techniques. Both of them almost overlap with each other, and Figure 9(c) also shows the significant improvement brought by our landmark calibration.

We then compare the performance of three different methods: using inertial sensors only, using sensors and traffic lights calibration, and using SmartLoc with all calibration. In this experiment, we assume the first 3400m is with reliable GPS signals, and the precise locations are accessible. The estimation starts from 3400m, and the first three figures in Figure 9 indicate the driving distance from the starting point versus the elapsed time.



(a) Error of location estimation in every sampling timeslot

(b) Error in locating the final destination in different blocks

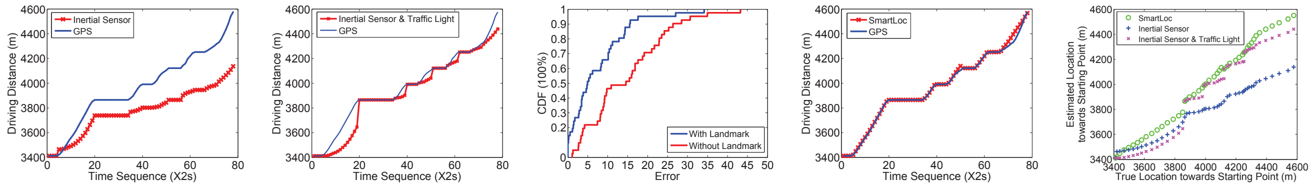
Figure 11: Navigation performance evaluation.

In Figure 9(a), we conducted the experiment based on sensors only, without any calibration or noise canceling. The double integration on acceleration leads to the final deviation of over 400m after driving about 1200m. When the road pattern detection is introduced, the location is calibrated when SmartLoc senses the road infrastructure pattern. During the same experiment, our vehicle crossed 5 traffic lights in total, and successfully detected all 5 traffic lights. The estimated locations are all then adjusted accordingly. The error in Figure 9(b) is still high, especially in the crossroads. Surprisingly, after combining our predictive regression model and the noise canceling technique, SmartLoc's result almost coincides with the ground truth, as shown in Figure 9(d). For the first 900m, the curve of SmartLoc nearly overlaps with the curve of the ground truth. For the first 450m, the vehicle passes three crossroads with all green lights, and the error is less than 20m in most of the time. After the final traffic lights, the vehicle has to drive at a relatively low speed because of the road construction. The predicted distance consequently deviates from the ground truth a little, but at the end of the road, the errors remain small. We plot all the estimated distances by three methods in Figure 9(e), with the X axis being the ground truth distance and Y axis being the predicted distance, *i.e.*, the perfect prediction will have a diagonal line. SmartLoc results are distributed almost along the diagonal line, and pure sensor approach deviates greatly.

The deviation of the results from the ground truth comes from the accumulated errors from all time slots. Based on the previous experiments, we plot the error in every time slot in Figure 10(a). SmartLoc with landmarks calibration has the smallest mean error of the estimated locations for all time slots: 90% of them are lower than 20m from the CDF in Figure 10(b). The other two approaches have larger errors, and the last figure describes the CDF of the total driving distance error.

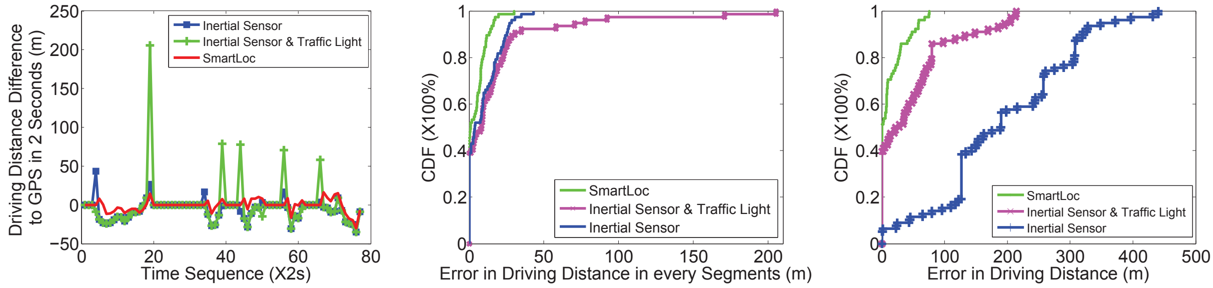
8.1.3 *Prediction in Highway*

In addition, we evaluate the SmartLoc on the highway to see whether our system works well when the velocity is relatively high. In the highway, GPS signals are almost always good, so the GPS data served as the ground truth only in this evaluation.



(a) Sensors Only (b) Sensors and Traffic Lights (c) Landmark vs No-Landmark (d) SmartLoc (e) Overall Comparison

Figure 9: Distance prediction comparison among three methods and ground truth.



(a) Error of driving distance in every time slot for three methods (b) Error of driving distance in every time slot for three methods (c) Error of long distance estimation for three methods

Figure 10: Comparison of three methods.

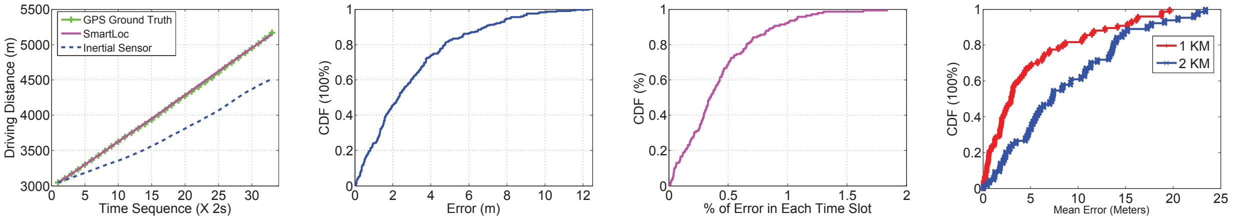
We drive over 10 different highway segments, and the total distance is over 60km (with driving speed 100km/h-120km/h approximately). The smartphone has access to the precise location information from the GPS, which is updated every 2 seconds and can be considered as the ground truth. Meanwhile, we collect the readings from the sensors and train our predictive regression model for 3km. Then, we predict the traveling distance for the next 2km and compare the distances from the ground truth, SmartLoc and the pure sensors.

Figure 12(a) illustrates the comparisons of driving distance estimation using SmartLoc (with sensors) and the GPS. The ground truth (GPS readings) is plotted by the green curve. It is obvious that the error between SmartLoc and the ground truth is becoming larger along the time. This is due to the accumulated errors without any calibration. By using our predictive regression model, SmartLoc calculate suitable parameters and apply them into the prediction. The estimation errors gets much smaller after then. Figure 12(b) indicates that the largest error is only 12m among the 10 different highway segments (each of length 2km), and in over 80% cases, the errors are less than 5m. Compared with the actual distance extracted from the ground truth (Figure 12(c)), at over 95% locations (among all locations where GPS location can be extracted), the errors are less than 1% of the actual driving distance, and the largest error is less than 2% of the actual driving distance. We also notice that the accuracy of the prediction decreases with the increase of the driving distance. We predict the driving distance for both 1km and 2km after taking the data of the first 3km to build a dead reckoning model. In our experiments, 80% of the errors in the first prediction (1km) are less than 10m, but they become 15m in the second prediction (2km). The largest error is around $19.8m$ and $23m$ respectively. We plot the results in CDF as Figure 12(d).

However, based on the evaluation, we discover that the estimation results cannot maintain high accuracies for a long distance. The main reason comes from the user dependent driving behaviors and the unpredictable road conditions. We also find that SmartLoc has a better estimation accuracy when the driving speeds remain stable, and when the driving speed fluctuates frequently, the SmartLoc's predicted results divert from the ground truth. Calibrating the location periodically is a feasible way to improve the location accuracy in real life applications.

8.1.4 *Evaluations Analysis*

Based on the evaluation results presented in this section, an obvious conclusion is that SmartLoc provides precise driving distance estimation in certain scenarios. In every time slot, the driving distance is estimated from the current sensor data as well as our predictive regression model. Suppose the error (denoted as D_i) in the estimation of each time slot i follows normal distribution: $D_i \sim \mathcal{N}(\mu, \sigma^2)$, with mean μ and variance σ^2. Then, the estimation of the total traveling distance S_t in t timeslots is the summation of the traveling distance in all time slots: $S_t = \sum_{i=1}^{t} D_i$. In this case, the error, from a long term perspective, will be accumulated. Obviously, $S_t \sim \mathcal{N}(t\mu, t\sigma^2)$. The variance of the variable S_t will be $t\sigma^2$. Thus, the mean error increases along the time, which leads to the conclusion that it is difficult to predict the traveling distance precisely in a long term, although sometimes the deviation in some continuous timeslots may be neutralized. For a given error bound δ, $Pr(S_t \geq \delta)$ is higher when



(a) Traveling Distance (b) Absolute error (c) Relative error (d) Long Distance Estimation

Figure 12: Traveling in a highway.

t is larger.

8.2 Localization in the City

We then present the localization results in Chicago downtown. As aforementioned, it is difficult to get the ground truth for the majority of the sampling locations.

We set the experiments of estimating the final location. Since, Section 2 has demonstrated that there are 9 bad road segments with lengths over 400m, which is less than 3 blocks in downtown Chicago. The goal of SmartLoc is then to obtain a relatively accurate distance estimation within three blocks. We randomly select 100 points as destinations in the experiment, and a destination could be one block, two blocks, or three blocks away from the starting point. We drive to these destination points to evaluate if the destination is precisely calculated by SmartLoc. We assume that the GPS signals are good before the starting point, and SmartLoc will train the dead-reckoning model during the driving. In this experiment, we test the accuracy of estimating the traveling distance in every time slot and of estimating the overall driving distance (*i.e.*, locating the final destination) as shown in Figure 11(a) and Figure 11(b) respectively. When SmartLoc only navigates to the destination within one block, with probability 70%, the error of estimating the location for each sampling slot is less than 10m, and with probability 85%, the mean error is less than 30m. When the destination is two blocks away, about 75% of the errors are less than 30m; when the destination is three blocks away, about 80% errors are less than 50m. From these figures, the error of destination locating within a few blocks is acceptable. We also plot the localization results for one road segment with length over 6400m in Figure 8. In this figure, the red spots denote the ground truth generated from GPS, and the blue spots represent the localization calculated by SmartLoc, where the green line between them is the localization error for every location.

8.3 Energy Consumption Analysis

SmartLoc provides the localization using both GPS and sensors. Empirically, the profiles of energy consumption for the GPS module and the sensors in smartphones are different. The average electric current of GPS modules in Android smartphones is approximately 135mA, which is larger than that of the sensors [47]. In addition, Android API provides four rate modes to sample the data, which are Fastest, Game, UI and Normal. The Fastest mode delivers the sensory data without any delay, thus the energy consumption is the largest (95mA [47]), while the Normal is the slowest, and also the most energy efficient one (15mA [47]). Then, SmartLoc is also able to reduce the energy consumption compared with the traditional approach using only GPS.

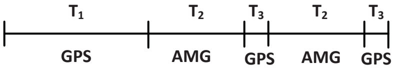


Figure 13: Working Pattern for SmartLoc.

Suppose the energy cost of a GPS module for every time slot is E_{GPS}, and the energy cost of the sensors are E_{AMG}. Obviously, $E_{AMG} \leqslant E_{GPS}$. As shown in Figure 13, where *GPS* indicates the localization through the GPS, and *AMG* denotes the location estimation through the sensors. Initially, during T_1, the system will train our predictive regression model, and the trained model will be used to estimate the trajectory in T_2. Periodically, SmartLoc will calibrate the locations when GPS signals become good again in T_3, and our model is adaptively updated. Then, the ratio of saved energy is $\frac{(E_{GPS}-E_{AMG})\cdot T_2}{E_{GPS}\cdot(T_2+T_3)}$. The duration of both T_1 and T_2 affect the localization accuracy of SmartLoc. Our first experiment is taken under the circumstance in which all the sensors and the GPS is activated. The driving lasted for 30 minutes, and the battery of smartphone discharged from 84% to 70%. Similarly, GPS localization for half an hour led to the battery drop from 70% to 61%. In most of our cases, in order to get higher accuracy, the ratio between the training and the estimating should be between 6:3 - 8:3. In the first experiment, the training distance is approximately 3400m and SmartLoc estimates the following 1200m, which reduces the energy by approximately 10% without affecting the localization accuracy.

9. RELATED WORKS

Our work intersects with several techniques, and we mainly focus on the wireless localization, and dead-reckoning [22].

9.1 Wireless Localization

GPS [25], being the most popular outdoor localization method, provides extremely accurate location information to the user, and more improved techniques such as WAAS [11], and Differential GPS (DGPS) [34] have been adopted in general civilian devices. Another feasible method is GSM-based localization [42] such as Nericell [28], which is widely available but with low accuracy (up to hundreds of meters). It also needs to know the exact position of cellular towers in advance.

Other techniques for both outdoor and indoor localization mainly fall into two categories: Fingerprint-based and Modeling-based approaches. The main idea of the fingerprint is to first establish a database of signal fingerprints based on surrounding environment, and then match the measured pattern to the fingerprints in the database. The mapping returns the location with the most similar fingerprint. Modeling-based techniques rely on some theoretical or experimental signal propagation models, relating delay or strength to distance.

In indoor environment, Radar [3] collects WiFi fingerprints beforehand at known locations inside a building, and identifies the user's location using the matching fingerprint. It could achieve relatively high accuracy: with median error 2.94 meters. Similar systems using RF signals including Horus [46], LANDMARC [30]. Fingerprints generated through the combination of WiFi and FM are studied in [8], which increases the accuracy by 83% compared with WiFi fingerprint only. Unlike other systems mentioned above, SurroundSense [2] estimates the location in indoor environment through multiple ambience features, including the light, the color of the floor, the WiFi, and the sound. LiFS [45] does not have to know the APs' location in prior, and the high location accuracy is achieved through exploiting user's motion from crowdsourcing. PinLoc [38] uses physical layer information to locate users with the accuracy of 89%.

In outdoor environment, both PlaceLab [9], and ActiveCampus [14] make full use of both WiFi and GSM signal for location. The former creates a map by war-driving a region and maps both APs and cell tower's signal against the wireless map. The latter is quite similar except it assumes the APs' location is known prior. Taking advantages of both systems, CompAcc [10] uses dead-reckoning combined with AGPS to calibrate rather than preliminary war-driving. All these systems demand time-consuming calibration, and are not suitable for large scale area. Skyhook [1] could supply high accuracy location by hiring over 500 drivers to create the WiFi/GSM map in certain region.

Other common ways to calculate location are based on the geometric model. In time-based techniques, Time of Arrival (ToA) [21], Time Difference of Arrival (TDoA) [35], and Angle of Arrival (AoA) [31, 50] establish geometric relationship between two separate devices, and calculate the location. Although pleasing accuracy is achieved, the high cost can not be neglected. In acoustic ranging, the distance of transmitter and receiver is estimated more precisely. Existing work includes Active Bat [16], Cricket [35], ENSBox [13], and WALRUS [5]. Liu *et al.* [24] calibrate the location accuracy in the WiFi fingerprint, using a more accurate distance estimation from acoustic ranging. Centaur [29] combined acoustic ranging and RF measurement to locate a device in office environment with acceptable accuracy.

Several promising techniques such as crowdsourcing are introduced in localization recently, such as Zee [36], which also uses inertial sensors to track users' movement.

9.2 Dead-Reckoning

Recently, dead-reckoning using internal sensors to estimate motion activities attracts a lot of research interests. Strapdown Inertial Navigation System (SINS) [41] and Pedometer System [19] use MEMS to estimate the moving speed and trace. The key issue is to deal with the noise of internal sensors, and the accumulated error, which sometimes grow cubically [44]. Personal Dead-reckoning (PDR) system [32] uses "Zero Velocity Update" to calibrate the drift. The majority of the dead-reckoning results focus on walking estimation, such as UnLoc [43], and CompAcc [10]. They mainly use accelerometer to estimate the number of steps, and then measure the distance through estimated step length. AutoWitness [15] is the system with an embedded wireless tag integrated with vibration sensors, accelerometer, and gyroscope. The tag is attached to a vehicle, and accelerometer and gyroscope are used to track the moving trace.

10. CONCLUSION

This paper presented *SmartLoc*, a metropolis localization system by using the inertial sensors and the GPS module of smartphones. We showed that pure dead-reckoning cannot provide acceptable localization accuracy. We established a predictive regression model to estimate the trajectory using linear regression, and our SmartLoc detects the road infrastructures and driving patterns as landmarks to calibrate the localization results. Our extensive evaluations shows that SmartLoc improves the localization accuracy to less than $20m$ for more than 90% roads in Chicago downtown, compared with $\geq 50\%$ with raw GPS data.

This is only the first step stone for the localization in metropolises. Additional functionality and techniques could further improve the utility of SmartLoc. For instance, for long road segments without good GPS signals, additional data such as GSM signal and driving history can be used to improve the accuracy. A cloud based social network may also provide drivers more traffic information, which might further improve the accuracy on unpredictable roads.

11. REFERENCES

[1] Skyhook. http://www.skyhookwireless.com/.

[2] Azizyan, M., Constandache, I., and Roy Choudhury, R. Surroundsense: mobile phone localization via ambience fingerprinting. In *MobiCom* (2009), ACM, pp. 261–272.

[3] Bahl, P., and Padmanabhan, V. Radar: An in-building rf-based user location and tracking system. In *INFOCOM* (2000), vol. 2, IEEE, pp. 775–784.

[4] Bao, L., and Intille, S. Activity recognition from user-annotated acceleration data. *Pervasive Computing* (2004), 1–17.

[5] Borriello, G., Liu, A., Offer, T., Palistrant, C., and Sharp, R. Walrus: wireless acoustic location with room-level resolution using ultrasound. In *MobiSys* (2005), ACM, pp. 191–203.

[6] Chen, M., Sohn, T., Chmelev, D., Haehnel, D., Hightower, J., Hughes, J., LaMarca, A., Potter, F., Smith, I., and Varshavsky, A. Practical metropolitan-scale positioning for gsm phones. *UbiComp* (2006), 225–242.

[7] CHEN, W., FU, Z., CHEN, R., CHEN, Y., ANDREI, O., KROGER, T., AND WANG, J. An integrated gps and multi-sensor pedestrian positioning system for 3d urban navigation. In *Urban Remote Sensing Event, 2009 Joint* (2009), IEEE, pp. 1–6.

[8] CHEN, Y., LYMBEROPOULOS, D., LIU, J., AND PRIYANTHA, B. Fm-based indoor localization. In *MobiSys* (2012), ACM, pp. 169–182.

[9] CHENG, Y., CHAWATHE, Y., LAMARCA, A., AND KRUMM, J. Accuracy characterization for metropolitan-scale wi-fi localization. In *MobiSys* (2005), ACM, pp. 233–245.

[10] CONSTANDACHE, I., CHOUDHURY, R., AND RHEE, I. Towards mobile phone localization without war-driving. In *INFOCOM* (2010), IEEE, pp. 1–9.

[11] ENGE, P., AND VAN DIERENDONCK, A. Wide area augmentation system. *Progress in Astronautics and Aeronautics 164* (1996), 117–142.

[12] ERIKSSON, J., GIROD, L., HULL, B., NEWTON, R., MADDEN, S., AND BALAKRISHNAN, H. The pothole patrol: using a mobile sensor network for road surface monitoring. In *ACM MobiSys* (2008).

[13] GIROD, L., LUKAC, M., TRIFA, V., AND ESTRIN, D. The design and implementation of a self-calibrating distributed acoustic sensing platform. In *SenSys* (2006), ACM, pp. 71–84.

[14] GRISWOLD, W., SHANAHAN, P., BROWN, S., BOYER, R., RATTO, M., SHAPIRO, R., AND TRUONG, T. Activecampus: experiments in community-oriented ubiquitous computing. *Computer 37*, 10 (2004), 73–81.

[15] GUHA, S., PLARRE, K., LISSNER, D., MITRA, S., KRISHNA, B., DUTTA, P., AND KUMAR, S. Autowitness: locating and tracking stolen property while tolerating gps and radio outages. In *SenSys* (2010), ACM, pp. 29–42.

[16] HARTER, A., HOPPER, A., STEGGLES, P., WARD, A., AND WEBSTER, P. The anatomy of a context-aware application. *Wireless Networks 8*, 2 (2002), 187–197.

[17] HASSANIEH, H., ADIB, F., KATABI, D., AND INDYK, P. Faster gps via the sparse fourier transform. In *MobiCom* (2012), ACM, pp. 353–364.

[18] HWANG, S., AND YU, D. Gps localization improvement of smartphones using built-in sensors.

[19] JIRAWIMUT, R., PTASINSKI, P., GARAJ, V., CECELJA, F., AND BALACHANDRAN, W. A method for dead reckoning parameter correction in pedestrian navigation system. *Instrumentation and Measurement 52*, 1 (2003), 209–215.

[20] KWAPISZ, J., WEISS, G., AND MOORE, S. Activity recognition using cell phone accelerometers. *SIGKDD 12*, 2 (2011), 74–82.

[21] LANZISERA, S., LIN, D., AND PISTER, K. Rf time of flight ranging for wireless sensor network localization. In *WISES* (2006), IEEE, pp. 1–12.

[22] LEVI, R., AND JUDD, T. Dead reckoning navigational system using accelerometer to measure foot impacts, Dec. 10 1996. US Patent 5,583,776.

[23] LIN, K., KANSAL, A., LYMBEROPOULOS, D., AND ZHAO, F. Energy-accuracy trade-off for continuous mobile device location. In *MobiSys* (2010), ACM, pp. 285–298.

[24] LIU, H., GAN, Y., YANG, J., SIDHOM, S., WANG, Y., CHEN, Y., AND YE, F. Push the limit of wifi based localization for smartphones. In *MobiCom* (2012), ACM, pp. 305–316.

[25] LIU, J., PRIYANTHA, B., HART, T., RAMOS, H., LOUREIRO, A., AND WANG, Q. Energy efficient gps sensing with cloud offloading.

[26] MANWEILER, J., AGARWAL, S., ZHANG, M., ROY CHOUDHURY, R., AND BAHL, P. Switchboard: a matchmaking system for multiplayer mobile games. In *MobiSys* (2011), ACM, pp. 71–84.

[27] MO, L., HE, Y., LIU, Y., ZHAO, J., TANG, S., LI, X., AND DAI, G. Canopy closure estimates with greenorbs: sustainable sensing in the forest. In *SenSys* (2009), ACM, pp. 99–112.

[28] MOHAN, P., PADMANABHAN, V., AND RAMJEE, R. Nericell: rich monitoring of road and traffic conditions using mobile smartphones. In *SenSys* (2008), ACM, pp. 323–336.

[29] NANDAKUMAR, R., CHINTALAPUDI, K., AND PADMANABHAN, V. Centaur: locating devices in an office environment. In *MobiCom* (2012), ACM, pp. 281–292.

[30] NI, L., LIU, Y., LAU, Y., AND PATIL, A. Landmarc: indoor location sensing using active rfid. *Wireless networks 10*, 6 (2004), 701–710.

[31] NICULESCU, D., AND NATH, B. Ad hoc positioning system (aps) using aoa. In *INFOCOM* (2003), vol. 3, IEEE, pp. 1734–1743.

[32] OJEDA, L., AND BORENSTEIN, J. Non-gps navigation with the personal dead-reckoning system. In *Defense and Security Symposium* (2007).

[33] PAEK, J., KIM, J., AND GOVINDAN, R. Energy-efficient rate-adaptive gps-based positioning for smartphones. In *MobiSys* (2010), ACM, pp. 299–314.

[34] PARKINSON, B., AND ENGE, P. Differential gps. *Global Positioning System: Theory and applications. 2* (1996), 3–50.

[35] PRIYANTHA, N., CHAKRABORTY, A., AND BALAKRISHNAN, H. The cricket location-support system. In *MobiCom* (2000), ACM, pp. 32–43.

[36] RAI, A., CHINTALAPUDI, K., PADMANABHAN, V., AND SEN, R. Zee: Zero-effort crowdsourcing for indoor localization. In *MobiCom* (2012), ACM, pp. 293–304.

[37] RETSCHER, G. An intelligent multi-sensor system for pedestrian navigation. *Journal of Global Positioning Systems 5*, 1-2 (2006), 110–118.

[38] SEN, S., RADUNOVIC, B., CHOUDHURY, R., AND MINKA, T. You are facing the mona lisa: spot localization using phy layer information. In *MobiSys* (2012), ACM, pp. 183–196.

[39] TAPIA, E., INTILLE, S., AND LARSON, K. Activity recognition in the home using simple and ubiquitous sensors. *Pervasive Computing* (2004), 158–175.

[40] THORSTENSEN, B., SYVERSEN, T., BJØRNVOLD, T., AND WALSETH, T. Electronic shepherd-a low-cost, low-bandwidth, wireless network system. In *MobiSys* (2004), ACM, pp. 245–255.

[41] TITTERTON, D., WESTON, J., TITTERTON, D., AND WESTON, J. Strapdown inertial navigation technology. Institution of Electrical Engineers.

[42] VARSHAVSKY, A., CHEN, M., DE LARA, E., FROEHLICH, J., HAEHNEL, D., HIGHTOWER, J., LAMARCA, A., POTTER, F., SOHN, T., TANG, K., ET AL. Are gsm phones the solution for localization? In *WMCSA* (2006), IEEE, pp. 34–42.

[43] WANG, H., SEN, S., ELGOHARY, A., FARID, M., YOUSSEF, M., AND CHOUDHURY, R. No need to war-drive: Unsupervised indoor localization. In *MobiSys* (2012), ACM, pp. 197–210.

[44] WOODMAN, O., AND HARLE, R. Pedestrian localization for indoor environments. In *Ubicomp* (2008), ACM, pp. 114–123.

[45] YANG, Z., WU, C., AND LIU, Y. Locating in fingerprint space: wireless indoor localization with little human intervention. In *MobiCom* (2012), ACM, pp. 269–280.

[46] YOUSSEF, M., AND AGRAWALA, A. The horus location determination system. *Wireless Networks 14*, 3 (2008), 357–374.

[47] YOUSSEF, M., YOSEF, M., AND EL-DERINI, M. Gac: energy-efficient hybrid gps-accelerometer-compass gsm localization. In *GLOBECOM* (2010), IEEE, pp. 1–5.

[48] ZHANG, P., SADLER, C., LYON, S., AND MARTONOSI, M. Hardware design experiences in zebranet. In *SenSys* (2004), ACM, pp. 227–238.

[49] ZHANG, Z., CHU, D., CHEN, X., AND MOSCIBRODA, T. Swordfight: enabling a new class of phone-to-phone action games on commodity phones. In *MobiSys* (2012), ACM, pp. 1–14.

[50] ZHANG, Z., ZHOU, X., ZHANG, W., ZHANG, Y., WANG, G., ZHAO, B., AND ZHENG, H. I am the antenna: Accurate outdoor ap location using smartphones. In *MobiCom* (2011), ACM, pp. 109–120.