A Real Independent Centimeter-Grade 3D Indoor Localization System On Smartphone

Fan Dang, Xuan Ding, and Pengfei Zhou
School of software
Tsinghua University
Beijing, China
Email: dangf13@mails.tsinghua.edu.cn, {dingxuan, zhoupf05}@tsinghua.edu.cn

***Abstract*—Fine grained indoor localization is attractive for its wide usage in indoor navigation system, infrastructure management, and blooming augmented reality applications. In this paper, we propose a smartphone based indoor localization system called Plotter, providing a centimeter-grade localization service without any prior knowledge or additional devices. Leveraging the simultaneous localization and mapping (SLAM) technology, Plotter not only learns its relative position among surroundings, but also simultaneously constructs and updates the map of unknown area. We take advantage of a modified Kalman Filter algorithm in the system in order to eliminate unacceptable errors produced by motion sensors on smartphones. Evaluation result shows that Plotter achieves centimeter-grade accuracy, which is competitive comparing with prior works assisted by additional devices.**

***Keywords*—indoor localization; Kalman filtering; SLAM**

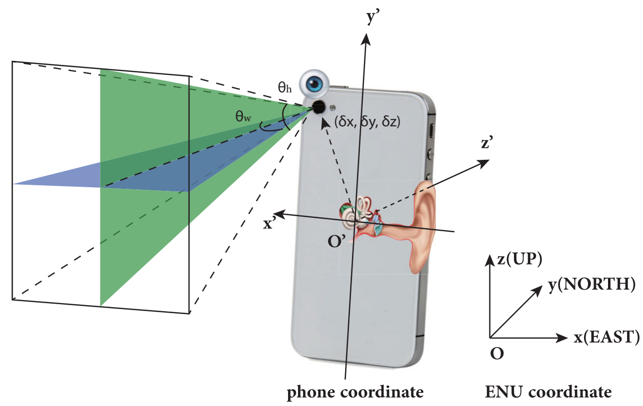


Fig. 1. Smartphone gets everything to localize itself like human being–camera (eye) and motion sensors (cochlea). Sensors use phone coordinate, rather than ENU coordinate.

I. Introduction

Accurate indoor localization is becoming increasingly attractive and important in pervasive computing technology nowadays. In applications like indoor navigation, motion sensing games, and human computer interaction, position information is one of the most essential concerns. In recent years, high accuracy positioning can be achieved by a bunch of fingerprinting based or model based localization approaches introduced by thousands of researchers. Innovative approaches have been constantly raising the bar. However, when trying to find a low cost and accurate localization system for real deployment, we find the choices are quite limited.

Generally speaking, fingerprinting based approaches require huge amount of manual work to collect fingerprints at every point of interest, and it is also required to store the fingerprints in the database. In recent years, approaches have been weakening these requirements. Recent works usually start with only a few known fingerprints. Then the systems constantly expand the scope of recognition, even though, in this process, they may sacrifice some flexibility when facing environmental changes and some user privacy. Nevertheless, manual work for initiation is still indispensable. In model based approaches, Angle of Arrival (AoA), Time of Arrival (ToA), and geometric constraint are widely used. Compared to the aforementioned fingerprinting based approaches, model based approaches have gained higher precision. One of the best [1] achieved a sub-centimeter-grade accuracy. Although fingerprinting based and model based approaches are mainstreams in research, additional devices are needed in both approaches, such as Wi-Fi routers, cellular base stations, or even Universal Software Radio Peripheral (USRP). Because of the expensiveness of those devices and harsh operating conditions, available area is usually limited. Most of the systems must be deployed limited in office buildings or laboratories. They will fail to work in mountainous areas without GPS or other RF signals. They will also fail when no auxiliary devices equipped.

In this paper, we propose an independent (of additional devices) and accurate localization system – Plotter, employing the idea of SLAM technology. The idea is quite simple. Take ourselves as an example, we, as mankind, were born to localize ourselves among surroundings. The reason is that we have our sensory system built up with optesthesia (visual sense), equilibrium (balance sense), etc. Optesthesia lets us know which direction the reference objects, like a building, a car, a door, and a corner of a wall, are in. Equilibrium tells us whether we are slant or accelerated. When we were babies, by crawling on the ground with turning heads around, we were able to recognize how wide those doors were, where those walls located, and how tall those buildings were constructed, combining both visual sense and balance sense. Though it is possible to determine the directions when our eyes are covered, we rely on our eyes more often, because optesthesia is much more precise than equilibrium. Now consider the elements in smartphones: after substituting the main characters of this scenario with cameras (for optesthesia) and motion sensors (for equilibrium), as shown in Fig. 1, we can reveal the full view of the Plotter system. Plotter makes use of its imprecise motion sensor for distance estimation roughly. Then the camera together with this moving distance are used together to localize some key points, such as corners, roughly again. For example, when we move left, noticing a key point moving fast from left to right in our visual field, we consequently know that it is getting closer to us, and vice versa. Camera and motion sensors compensate for each other and correct each other to form a group of accurate key points in the beginning of localization process. Later, it mainly uses visual based positioning approach for localization.

Our main contributions are summarized as follows:

- Plotter is a localization system without relying on additional devices or prior knowledge.
- Besides, it achieves a centimeter-level precision, which is comparable to the best-performance prior works assisted by additional devices.
- It keeps track of its location, while simultaneously updates and records ambient key points, i.e. a map of surroundings.
- Plotter is a standalone application without any network communication, which affords good privacy protection.
- We developed our experimental system on COTS device, a smartphone with 1.5GHz CPU and 1G memory. In evaluation section, this device is proved to be capable of executing this method. There is no need for high computing performance or large memory capacity in Plotter system.

In the following sections, we briefly review related works mainly on indoor localization and SLAM technologies in Section II, and present a global view on our system and basic localization methodology in Section III. We introduce our algorithm specifically in Section IV. In Section V, we demonstrate our experiments for evaluation, and show the attractive result of it. Finally, there is a simple conclusion in Section VI.

II. Related Work

A. Indoor Localization

Localization information [2], [3], [4] for indoor environments has become increasingly important as with the growing amount of indoor guidance applications, motion sensing games, and mobile social networks, etc. There are two research directions in the mainstream of non-visual approaches: one is fingerprinting based localization, and the other is model based localization [5].

Based on the idea that the most possible position is where the RF fingerprint matches the best, a large amount of fingerprinting based approaches were brought up. Since Bahl [6] introduced this system RADAR, precision has been increased gradually up to $0.225m$ in Jiang's work [7] by using a dynamic-circle-expanding mechanism. One of the most significant weakness is that they all require considerable manual work to gather fingerprints in every room or every place of interest to build a fingerprint database.

Instead, model based mechanisms are the other group of more accurace approaches. In their theories, locations are calculated instead of searching in a known database. They leverage ToA [8], Time Difference of Arrival (TDoA) [9], or AoA [10] to locate a point based on geometric constraints. Model based approaches are much more precise than fingerprinting based approaches, providing centimeter-grade positioning accuracy. But indispensable multi-antenna array and expensiveness of devices become a highlighted drawback of those methods, no matter how precise they are.

B. Simultaneous Localization and Mapping

Introduced for over 50 years, the idea using one single camera for localization is not a new term. Simultaneous localization and mapping (SLAM) technologies are mainly based on a monocular camera and some sensing devices, aiming at constructing and updating the map of unknown as well as localizing the agent device. Though this seems to be a chicken-and-egg problem, the effort of Leonard et al. introducing Kalman filtering into this field [11] made SLAM work, and as a result, Kalman filtering based solutions have become the main research direction.

Many researchers have been working on it, and generated lots of excellent results, such as wheelchair robot based on RGB-D sensor [12], indoor navigation robot made by Wieser et al. [13], iSAM system using multi-session visual mapping [14], etc. To the best of our knowledge, current SLAM systems are all based on high accuracy sensors, like infrared distancer or ultrasonic detector, which of course raise the bar of hardware requirements.

III. Overview

Plotter system leverages the SLAM technique when localizing itself and recognizing surroundings. The main difference between Plotter and traditional SLAM systems is that Plotter makes use of low-end motion sensors in smartphones, including accelerometer and orientation sensor, instead of high accuracy sensors like laser rangefinders or ultrasonic rangefinders used by traditional SLAM system. It consequently suffers much higher errors compared to other SLAM systems when performing Kalman filtering algorithm. Besides, in general, SLAM is often used in the field of robot navigation, which outputs a floor plan when the robot is moving, while in plotter, we propose a 3D indoor localization service with higher accuracy and shall be utilized in AR/VR applications.

We first propose two localization methodologies in this section, and then bring the architecture up based on these methods. Note that for better understanding, methodologies in this section are illustrated in ideal conditions without measuring error unless explicitly specified. Bold mathematical

TABLE I. MEANING OF RAW DATA PRODUCED BY SENSORS

Linear Accelerometer (excluding gravity)	a'_x	Acceleration force along the x'-axis.
	a'_y	Acceleration force along the y'-axis.
	a'_z	Acceleration force along the z'-axis.
Orientation Sensor	γ_1	Azimuth (angle around the z'-axis).
	γ_2	Pitch (angle around the x'-axis).
	γ_3	Roll (angle around the y'-axis).

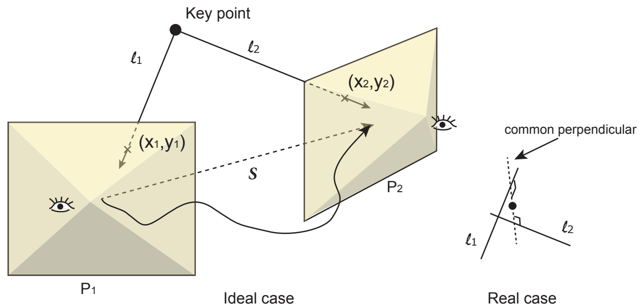


Fig. 2. Sight lines from two positions will intersect at key point, and vice versa, sight lines from two key points will intersect at lens.

symbol denotes a vector, and the symbol with a apostrophe means it is defined in phone coordinate, which is defined later.

A. Localization Methodologies

Prediction Model: Without any prior knowledge, Plotter provides only relative localization in a earth-fixed coordinate system or so called Earth North Up (ENU) coordinate system. The localization result is related to the position where this application starts. Against with ENU, smartphone utilizes its own coordinate as is shown in Fig. 1, which smartphone sensors mainly rely on. Briefly, when the device is held facing the screen, the x'-axis is horizontal and points to the right, the y'-axis is vertical and points up, and the z'-axis points toward the outside of the screen face. We make use of two sensors in our system, linear accelerometer and direction sensor, to develop our system on Android OS. Table I shows the meaning of values produced by each sensor. In Plotter, all computational works are based on ENU coordinate, so it is necessary to convert phone coordinate into ENU coordinate. Leveraging orientation sensor data, we can get the unit vector of $(x',\ y',\ z')^T$ under ENU coordinate system by Equation 1. We use $\boldsymbol{p_1},\ \boldsymbol{p_2},\ \boldsymbol{p_3}$ to denote projection vectors of each direction in phone coordinate:

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} \boldsymbol{p_1} \\ \boldsymbol{p_2} \\ \boldsymbol{p_3} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}. \tag{1}$$

Let $\boldsymbol{o} = (\gamma_1,\ \gamma_2,\ \gamma_3)^T$ be the orientation sensor values on three directions ,and now introduce an intermediate variable τ as,

$$\tau = \arccos\left(-\tan\gamma_2 \cdot \tan\gamma_3\right),$$

projection vectors can be expressed as the following three equations:

$$\boldsymbol{p_1} = \begin{pmatrix} -\cos\gamma_3 \cdot \sin\left(\gamma_1 + \tau\right) \\ -\cos\gamma_3 \cdot \cos\left(\gamma_1 + \tau\right) \\ -\sin\gamma_3 \end{pmatrix}^T$$

$$\boldsymbol{p_2} = \begin{pmatrix} -\cos\gamma_2 \cdot \sin\gamma_1 \\ -\cos\gamma_2 \cdot \cos\gamma_1 \\ \sin\gamma_2 \end{pmatrix}^T$$

$$\boldsymbol{p_3} = \boldsymbol{p_1} \times \boldsymbol{p_2}$$

Camera films at a constant speed, such as $15fps$. During the interval between two frames, sensors produce a series of data – accelerations $\boldsymbol{a_i'}$, orientations $\boldsymbol{o_i}$, and time duration between each data δ_{t_i}. When sensors sample data at their highest frequency, which is about $26Hz$, the time duration between two groups of sensored data is very short. Therefore, we assume the phone moves with a constant acceleration in each time slot. The cumulated speed $\boldsymbol{v}$ and displacement $\boldsymbol{S}$ are given by:

$$\boldsymbol{v_i} = \sum_{j=1\ldots i} \boldsymbol{a_j'} \cdot \begin{pmatrix} \boldsymbol{p_1} \\ \boldsymbol{p_2} \\ \boldsymbol{p_3} \end{pmatrix}^{-1} \cdot \delta_{t_j} + \boldsymbol{v_0} \tag{2}$$

$$\boldsymbol{S} = \sum_i \boldsymbol{v_i} \cdot \delta_{t_i} + \sum_i \boldsymbol{a_i'} \cdot \begin{pmatrix} \boldsymbol{p_1} \\ \boldsymbol{p_2} \\ \boldsymbol{p_3} \end{pmatrix}^{-1} \cdot \frac{\delta_{t_i}^2}{2} \tag{3}$$

By accumulating speed and displacement, Plotter is able to run the simplest localization. However, error in this model is not only huge, but also accumulates fast. As shown in Section V, it grows to $10m$ in only $40s$.

Observation Model: The second localization methodology is based on computer vision and geometric constraints, which is much more accurate than acceleration accumulation in prediction model. When two lines in the space intersect at one point, this point is determined uniquely. Moreover, when given one point and one unit vector, a unique line is determined. As shown in Fig. 2, the phone moves from P_1 to P_2. Two sight lines l_1 and l_2, which is from the camera to the key point, together with displacement $\boldsymbol{S}$ make a triangle. Let $(x_j,\ y_j)$ be the key point's image location on the screen, where x_j is pixels from left bound and y_j is pixels from the top bound, and the camera is at position P_j $(j = 1, 2)$. Then the sight line vectors in phone coordinate is:

$$\boldsymbol{u_j'} = \left(\frac{p_w}{2} - x_j,\ y_j - \frac{p_h}{2},\ \frac{p_w}{2 \cdot \tan\theta_w/2}\right),$$

where p_w and p_h are max horizontal and vertical resolution of lens, θ_w is the horizontal lens angle as shown in Fig. 1. Additionally, define $\boldsymbol{\delta}_{lens}'$ in phone coordinate as the relative position from the lens to the center of the phone. Then line l_j passes through the point $\boldsymbol{c_j'} = \boldsymbol{P_j'} + \boldsymbol{\delta_{lens}'}$ with the direction $\boldsymbol{u_j'}$. Since it is a relative position, we simply define $\boldsymbol{P_1}$ as $(0, 0, 0)$, so that $\boldsymbol{P_2} = \boldsymbol{S}$. So far, we get two lines l_1 and l_2 intersecting at the key point. However, in real cases, due to measuring errors, l_1 and l_2 will not intersect all the time. Fortunately, they are quite close and seem to intersect at one point, despite they stagger a very small distance. To keep things simple, we define their intersection in real situation as the middle point of their common perpendicular, as shown in Fig. 2.

Vice versa, if we are tracking two key points at the same time, the camera or the lens is at the intersection of two sight lines connecting lens and key points. Although this initially appears to be a chicken-and-egg problem, there are several algorithms known to solve it. To be introduced in the following parts, Kalman filtering is one of the most popular approximate solution methods.

B. System Architecture

In this section, we present the overall view of Plotter, as is shown in Fig. 3. The working process of Plotter is a cycle containing data collection, Kalman filtering, output and storage.

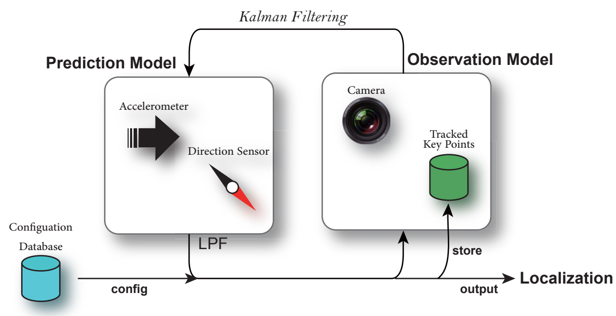


Fig. 3. Plotter architecture

The system starts with a configuration database, storing basic parameters of different smartphone models, including $\boldsymbol{\delta}'_{\boldsymbol{lens}}$ and θ_w we mentioned in last section. During localization process, these parameters are indispensable. The collected data are separated into two parts, corresponding to two models working in Kalman filter – prediction model and observation model. In prediction model, we use direction sensor and linear-accelerometer data to calculate position and posture. Meanwhile, observation model corrects localization error by using camera and tracked key points. Afterwards, the system provides an estimated position of phone for output, and positions of some new key points for storage which will be used in the following iterations.

IV. Proposed Algorithm

We propose a centimeter-grade indoor localization algorithm in Plotter system based on motion sensors and camera on smartphone in this section. The main goal of this algorithm is to maximally avoid being affected by large error in sensor data. We apply Kalman filter in the first part, which is a very popular tool when solving SLAM problems. Later, we bring up a method to help Kalman filter eliminate noise interference, which is proved to be effective in the following evaluation section.

A. Traditional Kalman Filter

Kalman filtering has been playing an increasingly important role in computer vision, despite its 50-year history after R. E. Kalman proposed this theory. Kalman filtering is an algorithm that operates recursively on streams of noisy input data to produce a statistically optimal estimate of the underlying system state. In Plotter's scenario, we use Kalman filtering to get localization information from continuously received images, accelerometer data, and direction sensor values.

In general, we keep track of smartphone's position and speed, as well as k $(k \geq 2)$ positions of key points in 3D space together as the state $\boldsymbol{x}$. In prediction model, state at time t can be inferred by the state at the previous time $\boldsymbol{x_{t-1}}$ and current accelerations $\boldsymbol{a_t}$,

$$\hat{\boldsymbol{x}}_t^- = \begin{pmatrix} \boldsymbol{p_t} \\ \boldsymbol{v_t} \\ \boldsymbol{K_{1t}} \\ \vdots \\ \boldsymbol{K_{kt}} \end{pmatrix} = \begin{pmatrix} 1 & \delta_t & \boldsymbol{0} \\ 0 & 1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I_k} \end{pmatrix} \hat{\boldsymbol{x}}_{t-1} + \begin{pmatrix} \delta_t^2/2 \\ \delta_t \\ \boldsymbol{0} \end{pmatrix} \boldsymbol{a_t} \tag{4}$$

where $\boldsymbol{p_t}$, $\boldsymbol{v_t}$ are position and speed vector of phone at time t, and $\boldsymbol{K_{it}}$ is the ith key point's position at time t. Note that symbol with hat means it is an estimated value, not the real one. The bar on the top right corner means this estimated value is not calibrated by observation model yet. We denote the first matrix on the right side in Equation (4) as $\boldsymbol{F}$, and the second one as $\boldsymbol{B}$ habitually.

In prediction model, error covariance matrix $\boldsymbol{P_t}$ is slightly changing from time $t-1$ due to the effect of Equation (4). Prediction model itself also brings error to $\boldsymbol{P_t}$. Equation (5) describes this relation, where $\boldsymbol{Q}$ is the error covariance matrix caused by prediction model. Seen from Equation (1), we can get initial error covariance $\boldsymbol{P_0}$ by converting metadata from sensor, whose orientation and acceleration on each direction are treated as irrelevant and can be measured or found in hardware parameter handbooks. $\boldsymbol{Q}$ is simply defined as a diagonal matrix, with large variance on the 2 top left elements, like 1; and small variance on the k bottom right elements, like 0.01, because key points are fixed while smartphone is moving.

$$\boldsymbol{P_t^-} = \boldsymbol{F}\boldsymbol{P_{t-1}^-}\boldsymbol{F}^{\mathrm{T}} + \boldsymbol{Q} \tag{5}$$

In observation model, we take advantage of computer vision to get an observation of both phone position and key point positions. The observation state at time t,

$$\boldsymbol{z_t} = (1,\ 0,\ 1, \cdots, 1)^T \boldsymbol{x_t} + \boldsymbol{v}, \tag{6}$$

where $\boldsymbol{v}$ is observation error. We mark the observation matrix on the right side in Equation (6) as $\boldsymbol{H} = (1,\ 0,\ 1, \cdots,\ 1)$ and define $\boldsymbol{R}$ as observation error covariance matrix brought by observation model.

The next step is to get a best estimate value of state $\hat{\boldsymbol{x}}_t$, which is a linear combination of an a-priori estimate $\hat{\boldsymbol{x}}_t^-$ and an actual measurement $\boldsymbol{z_t}$ as shown in Equation (7).

$$\hat{\boldsymbol{x}}_t = \hat{\boldsymbol{x}}_t^- + \boldsymbol{K_t}(\boldsymbol{z_t} - \boldsymbol{H}\hat{\boldsymbol{x}}_t^-), \tag{7}$$

where $\boldsymbol{K_t}$ is Kalman gain that minimizes the a-posteriori error covariance $\boldsymbol{P_t}$:

$$\boldsymbol{K_t} = \boldsymbol{P_t^-}\boldsymbol{H}^{\mathbf{T}}(\boldsymbol{H}\boldsymbol{P_t^-}\boldsymbol{H}^{\mathbf{T}} + \boldsymbol{R})^{-1}. \tag{8}$$

At last, we update the a-posteriori error covariance estimate via Equation (9)

$$\boldsymbol{P_t} = (\boldsymbol{I} - \boldsymbol{K_t}\boldsymbol{H})\boldsymbol{P_t^-}. \tag{9}$$

After each pair of prediction and observation process, Equation (4)-(9) make up the main cycle of Kalman filtering. We try to simulate a series of smartphone movement in ideal sensing environment, Kalman filtering obtained a good result, but failed in practical tests due to heavy noise.

In the following part, we introduce low-pass filter to resist such heavy noise. Low-pass filter removes high frequency noise in raw sensor values and provides a better dynamic estimate of error covariance matrices.

B. Low-pass filtering

The process that a motion sensor producing sensor values is quite similar to the process that a recorder recording sound. Taking accelerometer as an example, sensor data can be treated as samples of mechanical wave, and the accelerometer can be treated as a recorder which samples acceleration values at a fixed frequency. The dash line in Figure 4(b) shows an example of acceleration on x'-axis when a user slightly wave this phone. After Fourier transform, acceleration spectrum is shown in Figure 4(a). This waving motion is a low frequency wave, with a peak appearing at around 0.5Hz. Other two spectrums are also shown in this figure, describing two scenarios when the phone is placed on the table or held in the hand. Seen from the spectrum, noise of "held in hand" is much larger than the stable one on the table in the low frequency part, while nearly the same in the high frequency part. Therefore, we assume that such noise appears in low frequency is mainly caused by slight shaking of hands.

When using Plotter system, user moves around, seeing through the screen. They won't make high-frequency vibration or high-speed movement during such process, because of the application scenario and human body physiology limitation. For a better understanding of how high the shaking frequency can be and how large the acceleration can reach, we tracked 5 students' moving parameters for over one minutes in our laboratory. Over 95.7% percent of accelerations are lower than $0.59m/s^2$, and over 57.2% of spectrum energy is distributed under 1Hz. So we implement a low-pass filter to process the sensor data, based on Traditional Kalman Filtering (TKF). Compared with raw data in Figure 4(b), filtered data is smooth with less noise.

V. Performance Evaluation

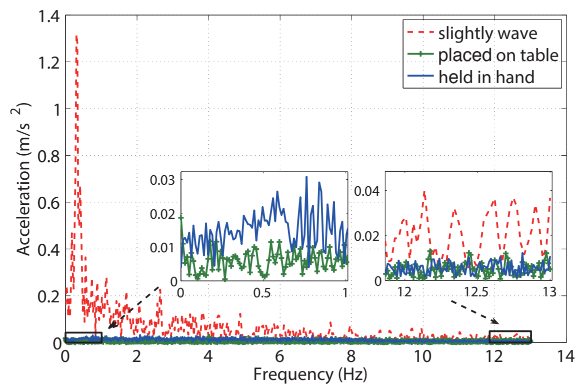
In this section, we conduct simulation tests for well controlled evaluations, and field tests on overall localization capability. We mainly focus on two algorithms, traditional Kalman filtering (TKF) and low-pass filtered Kalman filtering (LPKF).

A. Simulations

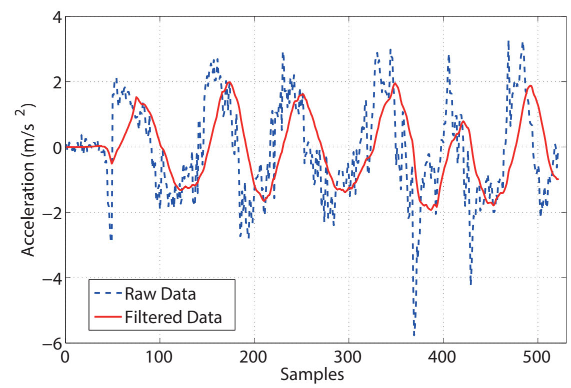
It is difficult to keep a constant acceleration or speed during field tests, and also hard for us to learn the ground truth. So we first conduct simulation evaluations with a better knowledge of environment parameters.

For better presentation, we simulate a rectangular motion with an acceleration and a deceleration process on each edge. When phone is at corners, its speed is 0. We employ sensor errors measured in field tests. The orientation sensor has an error variance of 0.235, and that of accelerometer is 0.011. Accelerations in simulation tests are under $0.01m/s^2$, which is a very small value even compared to the noise.

Based on these simulation tests, we work out smartphone and key point localization precision implemented by these algorithms. Fig. 5(a) shows a result in simulations. The phone starts from $(0,0,0)$, and moves along a $0.22m \times 0.22m$ square trail on $x-y$ plane. Fig. 5(b) describes key points localization process. At the beginning, because of the slow speed, both prediction and observation model are not reliable; coordinates of key point change severely. With the speed increasing, observation model provides more information when localizing phone itself and key points. Soon, key points' localizations converge to a constant value. Seeing from the result, we get the most precise coordinates on x-axis and y-axis which the phone moves on with only few millimeters error, and a poor



(a) Spectrum of acceleration when smartphone in 3 motion status



(b) Passing through a 1Hz low-pass filter, the sensing data filters out most of the noise

Fig. 4. Taking sensing data as a kind of wave, we use FFT to filter the raw data. The filtered data shows good features.

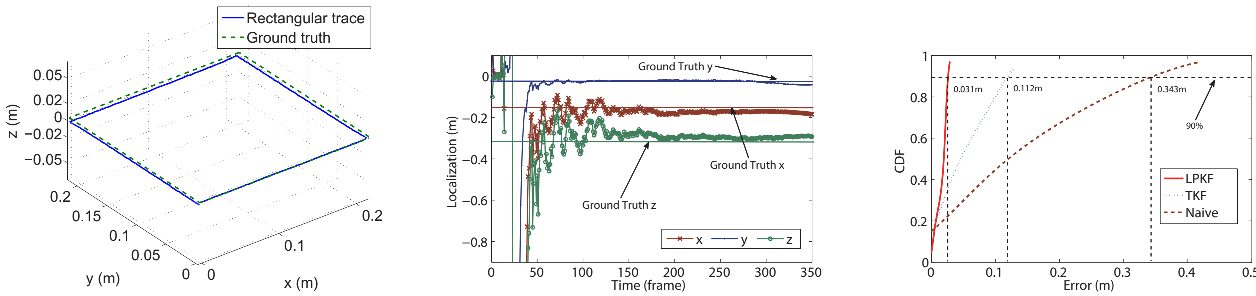
performance on z-axis which is parallel to sight line, even though error on z-axis is no more than $5cm$. From the start to the time when coordinates are stable, it takes about 50 frames, i.e. less than 4 seconds if camera films at $15fps$.

We conduct 100 groups of 10-second simulation, and Fig. 5(c) shows the cumulative distribution of errors produced by TKF, LPKF, and naive acceleration accumulation. Compared with naive method, both approaches in this work achieve much better results. LPKF performs the best, 90% of its error is under $2.6cm$. We also conduct simulations based on linear trail and circular trail, and achieve desired results.

B. Field Test

We implement Plotter on Sony Xperia 28i smartphone, with Android 4.0 operating system, and a linear accelerometer and an orientation sensor embedded. Accelerometer and orientation sensor provide sampling rate as high as 26Hz and error variance as is shown in section V-A. On the software side, we employ OpenCV 2.4.9 on Android SDK for CV analysis, and packaged Matlab program for data processing. The key point is defined as the most prominent corners in each frame and is tracked by using observing optical flow.

We recruit 5 students in our laboratory and let them move this phone near one fixed position, and return to that position



(a) A simulation trail with rectangular profile

(b) Key points' positions converge to the ground truth gradually

(c) Localization errors by TKF, LPKF and naive acceleration accumulation

Fig. 5. Simulation results

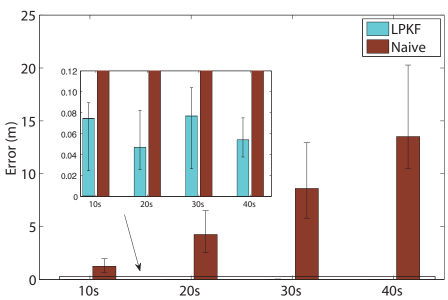


Fig. 6. Errors by LPKF and naive acceleration accumulation in field tests

at last. For better key point tracking effect, we choose a clean wall as the background and draw some black dots on it randomly. Four groups of experiment are conducted with different moving time. Naive acceleration accumulation method is also implemented as control group. With the growing of moving time, accumulated error in naive method is increasing at a high speed, about 3 meters per 10 seconds. Compared with naive method, LPKF shows an attractive error result around $6cm$ on average and $11cm$ maximally (Fig. 6). Seen from the result, there is no distinct increasing tendency with time grows.

VI. Conclusion

In this paper, we propose a new indoor localization system on mobile devices taking advantage of the basic idea in SLAM technology. We employ low-pass filter for raw data processing in our localization algorithm. Simulation and field test result indicate that it can provide as precise as centimeter-grade accuracy, which is at the same level with other indoor localization approaches, without other devices or prior knowledge.

Acknowledgment

This work is supported in part by the Natural Science Foundation of China under Grant 61272429 and CERG-61361166009.

References

[1] L. Yang, Y. Chen, X.-Y. Li, C. Xiao, M. Li, and Y. Liu, "Tagoram: Real-time tracking of mobile rfid tags to high precision using cots devices," in *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, 2014, pp. 237–248.

[2] Y. Zheng, G. Shen, L. Li, C. Zhao, M. Li, and F. Zhao, "Travi-navi: Self-deployable indoor navigation system," in *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, 2014, pp. 471–482.

[3] G. Wang, C. Gu, T. Inoue, and C. Li, "Hybrid fmcw-interferometry radar system in the 5.8 ghz ism band for indoor precise position and motion detection," in *Microwave Symposium Digest (IMS), 2013 IEEE MTT-S International*, June 2013, pp. 1–4.

[4] C. P. Kumar, R. Poovaiah, A. Sen, and P. Ganadas, "Single access point based indoor localization technique for augmented reality gaming for children," in *Students' Technology Symposium (TechSym), 2014 IEEE*, Feb 2014, pp. 229–232.

[5] C. Wu, Z. Yang, Y. Liu, and W. Xi, "WILL: Wireless indoor localization without site survey," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 4, pp. 839–848, April 2013.

[6] P. Bahl and V. N. Padmanabhan, "Radar: an in-building rf-based user location and tracking system," in *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2, 2000, pp. 775–784 vol.2.

[7] J. A. Jiang, X. Y. Zheng, Y. F. Chen, C. H. Wang, P. T. Chen, C. L. Chuang, and C. P. Chen, "A distributed rss-based localization using a dynamic circle expanding mechanism," *IEEE Sensors Journal*, vol. 13, no. 10, pp. 3754–3766, Oct 2013.

[8] J. He, K. Pahlavan, S. Li, and Q. Wang, "A testbed for evaluation of the effects of multipath on performance of toa-based indoor geolocation," *IEEE Transactions on Instrumentation and Measurement*, vol. 62, no. 8, pp. 2237–2247, Aug 2013.

[9] M. Jung, G. Fischer, R. Weigel, and T. Ussmueller, "Low power 24 ghz ad hoc networking system based on tdoa for indoor localization," *ISPRS International Journal of Geo-Information*, vol. 2, no. 4, p. 1122, 2013.

[10] J. Xiong and K. Jamieson, "Arraytrack: A fine-grained indoor location system," in *Proceedings of the 10th USENIX Conference on Networked Systems Design and Implementation*. USENIX Association, 2013, pp. 71–84.

[11] J. J. Leonard and H. F. Durrant-Whyte, "Simultaneous map building and localization for an autonomous mobile robot," in *Intelligent Robots and Systems '91. 'Intelligence for Mechanical Systems, Proceedings IROS '91. IEEE/RSJ International Workshop on*, Nov 1991, pp. 1442–1447 vol.3.

[12] B. F. Wu, C. L. Jen, W. F. Li, T. Y. Tsou, P. Y. Tseng, and K. T. Hsiao, "Rgb-d sensor based slam and human tracking with bayesian framework for wheelchair robots," in *Advanced Robotics and Intelligent Systems (ARIS), 2013 International Conference on*, May 2013, pp. 110–115.

[13] I. Wieser, A. V. Ruiz, M. Frassl, M. Angermann, J. Mueller, and M. Lichtenstern, "Autonomous robotic slam-based indoor navigation for high resolution sampling with complete coverage," in *2014 IEEE/ION Position, Location and Navigation Symposium - PLANS 2014*, May 2014, pp. 945–951.

[14] J. McDonald, M. Kaess, C. Cadena, J. Neira, and J. Leonard, "Real-time 6-dof multi-session visual SLAM over large-scale environments," *Robotics and Autonomous Systems*, vol. 61, no. 10, pp. 1144 – 1158, 2013, selected Papers from the 5th European Conference on Mobile Robots (ECMR 2011).