

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

Fan Dang, Xuan Ding and Pengfei Zhou

Institute of Trustworthy Networks and Systems

School of software, Tsinghua University, Beijing, 100084

Email: dangf13@mails.tsinghua.edu.cn, dingx04@gmail.com, 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 (AR) applications. In this paper, we propose a smartphone based indoor localization system called Plotter, providing 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 constructing and updating map of the 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 smartphone. Evaluation result shows that Plotter achieves centimeter grade accuracy, which is competitive comparing with prior works assisted by additional devices.

## I. INTRODUCTION

Accurate indoor localization is becoming increasingly attractive and important in today's pervasive computing technology. In most applications of indoor navigation, motion sensing game and devices interaction, position information is one of the most essential contexts. 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 are constantly raising the bar, while when trying to find a low cost and accurate localization system for real deployment, we find its choices are quite limited.

Generally speaking, fingerprinting based approaches need manual work to gather fingerprints at every point of interest, and store them in the database. In recent years, these approaches weaken this requirements and start with a few known fingerprints, and constantly expand the scope of recognition, while it sacrifice flexibility to changes and part of user privacy conversely. Nevertheless, manual work for initiation is still indispensable. In the literature of model based approaches, Angle of Arrival (AoA), Time of Arrival (ToA) and geometric constraint are widely used. Compared with the former one, model based approaches have higher precision. One of the best of them[1] achieves a sub-centimeter-grade accuracy. Although fingerprint based and model based approaches are mainstreams in research, additional devices are needed in both of them, 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 limited. Most of them could be deployed only in office building or laboratory. They will fail in mountainous area without GPS or RF signals, or simply in buildings without electricity.

In this work, we propose an independent and accurate localization system—Plotter, employing the idea of SLAM technology. Considering the fact that, as mankind, we were born to localize ourselves among surroundings. The reason

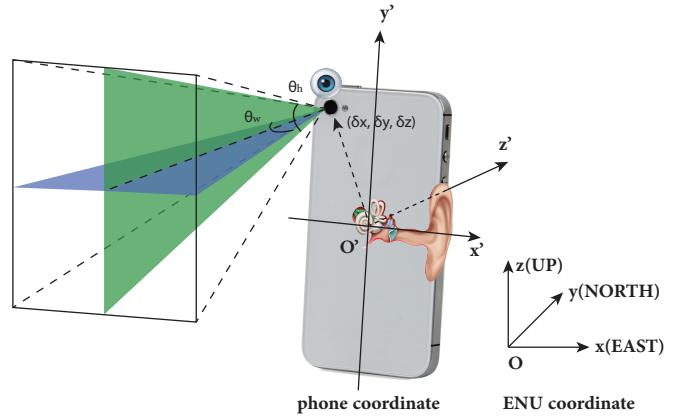


Fig. 1. Smartphone got everything to localize itself like human being—camera (eye) and motion sensors (cochlea). Sensors use phone coordinate, not ENU coordinate.

is, we have our sensory system—optesthesia (visual sense), equilibrium (balance sense), etc. Equilibrium tells us whether we are slant or accelerated. Optesthesia lets us know which direction those reference objects are on, like a building, a door, a corner of wall. When we were babies, by waving head, crawling around, combining both visual sense and balance sense, we can recognize how wide those doors are, where those walls located, and how tall those buildings constructed. Then we rely more on our eyes, because it's more precise than using equilibrium only. After substitute smartphone for the main character of this story, as is shown in Fig. 1, we can reveal the full view of Plotter system. Plotter make use of its imprecise motion sensor for distance estimation roughly. Then the camera together with this moving distance localize some key point, such as corners, also roughly. For example, when we moving left, and a key point moving fast from left to right in our visual field, we know that it is close to us, and vice versa. Camera and motion sensors compensate for each other and correct each other to get a group of accurate key points in the beginning of localization process. Latter, it use only visual based positioning approach for localization.

Our main contributions are summarized as follows:

- Plotter is a real sense of localization system without relying on additional devices or prior knowledge.
- Besides, it achieves a centimeter level precision, which is comparable with 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, affording good privacy protection.
- We develop 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 this work. There's 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. There's a simple conclusion in Section VI.

## II. RELATED WORK

### A. Indoor Localization

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

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

While model based mechanisms are another group of more accuracy approaches. In their theories, locations are calculated instead of searched in a known database. They leverage ToA[8], Time Difference of Arrival (TDoA)[9] or AoA[10] to definitely locate a point based on geometric constraint. Model based approaches are much preciser 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 over 50 years ago, the idea using one single camera for localization is not a new term. Simultaneous localization and mapping technologies are mainly based on a monocular camera and some sensing devices, aimed 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, owe to the efforts of Leonard et al. introducing Kalman filtering into this field[11], it works out well and Kalman filtering based solutions become the main research direction.

Many researchers have been working on it, and there generate lots of excellent works, 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 system are all based on high accuracy sensors, like infrared

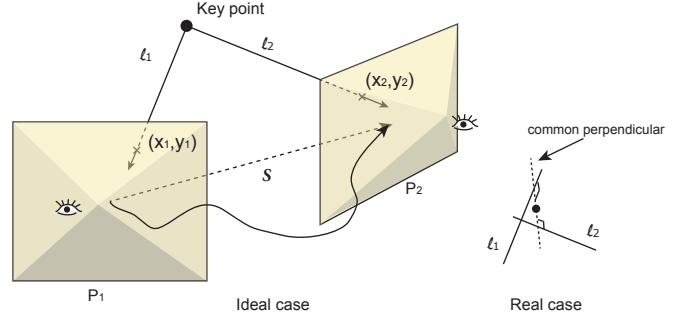


Fig. 2. Sight lines from two positions will intersect at key point, and vice versa. Sight lines from two key points will intersect at lens.

distancer or ultrasonic detector, which of course raise the bar of hardware requirements.

## III. OVERVIEW

Plotter system leverages SLAM technique when localizing itself and recognizing surrounding environments. There are two main differences between Plotter and traditional SLAM system. First, Plotter makes use of motion sensors in smartphone, like accelerometer and orientation sensor, instead of high accuracy sensors like laser rangefinders or ultrasonic rangefinders used by traditional SLAM system. It suffers much higher errors compared with others when processing Kalman Filter algorithm. Second, in general, SLAM is often used in the field of robot navigation, which outputs floor plan as the robot is moving. We propose a 3D indoor localization service with higher accuracy and shall be utilized in AR applications.

We propose two localization methodologies first in this section, and then the architecture based on these methods. Notice that, this section will be illustrated in ideal conditions without measuring error for better understanding unless explicitly specified. Bold mathematical symbol denotes vector, and symbols with apostrophe means it is defined in phone coordinate.

### A. Localization Methodology

**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 has its own coordinate as is shown in Fig. 1, which smartphone sensors mainly rely on. Briefly, 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

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	$\gamma'_x$	Azimuth (angle around the $z'$ -axis).
	$\gamma'_y$	Pitch (angle around the $x'$ -axis).
	$\gamma'_z$	Roll (angle around the $y'$ -axis).

sensor, and 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  $x'$ ,  $y'$ ,  $z'$  under ENU coordinate system by (1). We note  $\mathbf{P}_1$ ,  $\mathbf{P}_2$ ,  $\mathbf{P}_3$  as projection vectors of each direction in phone coordinate.

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \text{proj}(\mathbf{o}') = \begin{pmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \\ \mathbf{P}_3 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad (1)$$

where  $\mathbf{o}' = [\gamma'_x \ \gamma'_y \ \gamma'_z]^T$  are orientation sensor values on three directions. Furthermore, after we introduce an intermediate variable  $\tau$  as,

$$\tau = \arccos(-\tan \gamma_y \cdot \tan \gamma_z)$$

Projection vectors can be expressed as the following three equations.

$$\begin{aligned} \mathbf{p}_1 &= \begin{pmatrix} -\cos \gamma_z \cdot \sin(\gamma_x + \tau) \\ -\cos \gamma_z \cdot \cos(\gamma_x + \tau) \\ -\sin \gamma_z \end{pmatrix}^T \\ \mathbf{p}_2 &= \begin{pmatrix} -\cos \gamma_y \cdot \sin \gamma_x \\ -\cos \gamma_y \cdot \cos \gamma_x \\ \sin \gamma_y \end{pmatrix}^T \\ \mathbf{p}_3 &= \mathbf{p}_1 \times \mathbf{p}_2 \end{aligned}$$

Camera films at a constant speed, such as 15 frame/s. During the interval between two frames, sensors produce a series of data, accelerations  $\mathbf{a}'_i$ , orientations  $\mathbf{o}'_i$ , and time slots between each data  $\delta t_i$ . Sensors sample data at their highest frequency, about 26Hz. It is a very short time between two sensor data, so that we assume the phone moves with constant acceleration in each time slot. The cumulated speed  $\mathbf{v}$  and displacement  $\mathbf{S}$  is given by:

$$\mathbf{v}_i = \sum_{j=1 \dots i} \mathbf{a}_j \cdot \text{proj}(\mathbf{o}_j) \cdot \delta t_j + \mathbf{v}_0 \quad (2)$$

$$\mathbf{S} = \sum_i \mathbf{v}_i \cdot \delta t_i + \sum_i \mathbf{a}_i \cdot \text{proj}(\mathbf{o}_i) \cdot \delta t_i^2 / 2 \quad (3)$$

By accumulating speed and displacement, Plotter is able to operate the simplest localization. However, error in this model is not only huge, but also accumulated, as is shown in Section V, this error grows to 10m in only 40s.

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

$$\mathbf{u}'_j = \left[ \frac{p_w}{2} - x_j, y_j - \frac{p_h}{2}, \frac{p_w}{2 \cdot \tan \theta_w / 2} \right]$$

where  $p_w$  and  $p_h$  are max horizontal and vertical resolution of

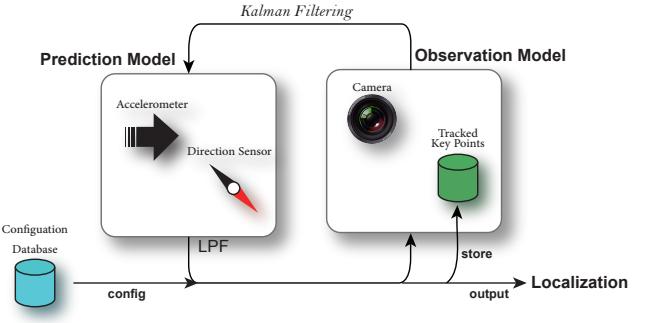


Fig. 3. Plotter architecture

lens.  $\theta_w$  is the horizontal lens angle as is shown in Fig. 1. Also,  $\delta'_{lens}$  is defined as the relative position from lens to the center of the phone. Line  $l_j$  passes through point  $\mathbf{c}'_j = \mathbf{P}'_j + \delta'_{lens}$  with direction  $\mathbf{u}'_j$ . Because it is a relative position, we simply define  $\mathbf{P}_1$  as  $(0, 0, 0)$ , so that  $\mathbf{P}_2 = \mathbf{S}$ . So far, we get two lines –  $l_1$  and  $l_2$  intersecting at the key point. In real case, due to measuring errors,  $l_1$  and  $l_2$  will not intersect at almost every moment. Fortunately, even though, they seems 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 is shown in Fig. 2.

On the other side, if we are tracking two key points at the same time, the camera or the lens is on 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 for solving 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.

The system starts with a configuration database, storing basic parameters of different smartphone types, like  $\delta'_{lens}$  and  $\theta_w$ . As is mentioned in the previous part, 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 are going to introduce Kalman filter in the first part, which is a very popular tool when solving SLAM problems. Later, we bring up two methods to help Kalman filter eliminate noise

interference, which is proved to be effective in the following evaluation section.

#### A. Traditional Kalman Filter

Kalman filtering is 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 receives images, accelerometer data and direction sensor values.

In general, we keep track of smartphone's speed and position, and  $k(k \geq 2)$  key point positions in 3D space as the state  $\mathbf{x}$ . In prediction model, state at time  $t$  can be inferred by the state at last second  $\hat{\mathbf{x}}_{t-1}$  and current accelerations  $\mathbf{u}_t$ ,

$$\hat{\mathbf{x}}_t^- = \begin{pmatrix} \mathbf{p}_t \\ \mathbf{v}_t \\ \mathbf{K}_{1t} \\ \vdots \\ \mathbf{K}_{kt} \end{pmatrix} = \begin{pmatrix} \mathbf{I}_3 & \delta t & 0 \\ 0 & \mathbf{I}_3 & 0 \\ 0 & 0 & \mathbf{I}_{3k} \end{pmatrix} \hat{\mathbf{x}}_{t-1} + \begin{pmatrix} \frac{\delta t^2}{2} \\ \delta t \\ 0 \end{pmatrix} \mathbf{u}_t \quad (4)$$

where  $\mathbf{p}_t$ ,  $\mathbf{v}_t$  are position and speed vector of phone at time  $t$ .  $\mathbf{K}_{it}$  is the  $i$ th key point's position. Note that symbol with hat means it is an estimate value, not real state. The bar on the top right corner means this estimate value is not corrected by observation model yet. Due to the limit of line width, the first  $\delta t$  stands for a third-order unit diagonal matrix  $\mathbf{I}$  times  $\delta t$ , and the second one is a column vector with three  $\delta t$ , so as the same with  $\frac{\delta t^2}{2}$ . We note the first matrix on the right side in (4) as  $\mathbf{F}$ , and the second one as  $\mathbf{B}$  habitually.

In prediction model, error covariance matrix  $\mathbf{P}_t$  is slightly change from time  $t - 1$  due to the effect of (4). Prediction model itself also brings error to  $\mathbf{P}_t$ . Equation (5) describes this relation, where  $\mathbf{Q}$  is the error covariance matrix caused by prediction model. Seen from (1), we can get initial error covariance  $\mathbf{P}_0$  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.  $\mathbf{Q}$  is simply defined as a diagonal matrix, with large variance on the 6 top left elements, like 1; and small variance on the  $2k$  bottom right elements, like 0.01, because key points are fixed while smartphone is moving.

$$\mathbf{P}_t^- = \mathbf{F} \mathbf{P}_{t-1}^- \mathbf{F}^T + \mathbf{Q} \quad (5)$$

In observation model, we take advantages of computer vision to get an observation of both phone position and key point positions. We note observation state at time  $t$  as

$$\mathbf{z}_t = \begin{pmatrix} \mathbf{I}_3 & 0 & 0 \\ 0 & 0 & \mathbf{I}_{3k} \end{pmatrix} \mathbf{x}_t + \mathbf{v} \quad (6)$$

where  $\mathbf{v}$  is observation error. We mark the observation matrix on the right side in (6) as  $\mathbf{H}$  and define  $\mathbf{R}$  as observation error covariance matrix brought by observation model.

The next step is to get a best estimate value of state  $\hat{\mathbf{x}}_t$ , which is a linear combination of an a-priori estimate  $\hat{\mathbf{x}}_t^-$  and a weighted difference between an actual measurement  $\mathbf{z}_t$  and a measurement prediction  $\mathbf{H}\hat{\mathbf{x}}_t^-$  as shown in (7).

$$\hat{\mathbf{x}}_t = \hat{\mathbf{x}}_t^- + \mathbf{K}_t (\mathbf{z}_t - \mathbf{H}\hat{\mathbf{x}}_t^-) \quad (7)$$

where  $\mathbf{K}_t$  is Kalman gain that minimizes the a-posteriori error covariance  $\mathbf{P}_t$ .

$$\mathbf{K}_t = \mathbf{P}_t^- \mathbf{H}^T (\mathbf{H} \mathbf{P}_t^- \mathbf{H}^T + \mathbf{R})^{-1} \quad (8)$$

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

$$\mathbf{P}_t = (\mathbf{I} - \mathbf{K}_t \mathbf{H}) \mathbf{P}_t^- \quad (9)$$

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 parts, we introduce low-pass filter and innovation-based adaptive estimation to resist such heavy noise. Low-pass filter removes high frequency noise in raw sensor values, and then innovation-based adaptive estimation provides a better dynamic estimate of error covariance matrices.

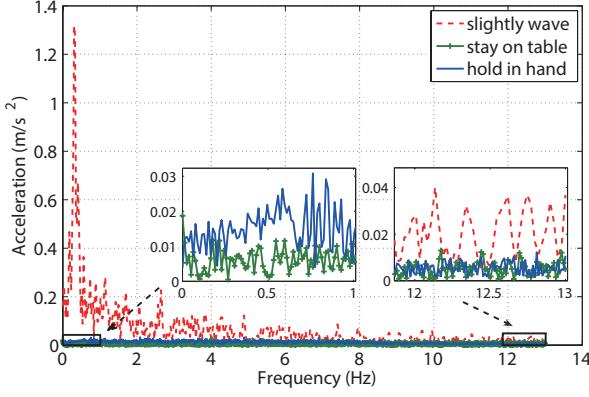
#### B. Low-pass filtering

Motion sensor producing sensor values is the same thing with recorder recording sound. Taking accelerometer as an example, sensor data could be treated as samples of mechanical wave, and mobile sensor is a recorder sampling 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 appears at around 0.5Hz. Another two spectrums are also shown in this figure, describing two scenarios when the phone is stay on table or held in hand. Seen from the spectrum, noise of hand hold mobile is much larger than stable one in the low frequency part, while nearly the same in the high frequency part. We assume that such noise appears in low frequency is mainly caused by slight shaking of hands.

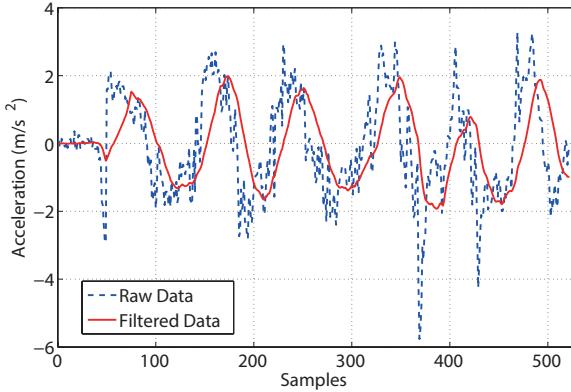
When using Plotter system, user moves around seeing through the screen. They won't make high-frequency vibration or high-speed movement during such process because of both the requirements of this application scenario and human body physiology limit. For a better understanding to how high the shaking frequency can be and how large the acceleration can reach, we tracked 5 students' moving parameters for over one minutes each in our laboratory. Over 95.7% percent of accelerations are lower than  $0.59m/s^2$  and 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 containing less noise.

#### C. Innovation-based adaptive estimation

In observation model, observation vector  $\mathbf{z}_t$  is extremely sensitive to position changing between key points and mobile device. Taking two closely spaced key points as an instance, seen from camera, these two points are very close, so sight lines to these two points almost coincide with each other. In ideal case, no matter how close they are, these two lines could be separated clearly. However, the pixel number of camera is limited, image may be blurry, and two key points positions can't be measured without error. These all make intersection (or the camera position) deviate in wide range. At this moment, observation model is much more unreliable than prediction model. On the other hand, when key points distribute equally,



(a) spectrum of acceleration when mobile at 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 implements FFT and filtering on raw data. The data shows good features.

observation model provides very precise localization based on camera only, like the basic technology in AR.

To deal with this problem, we employ the innovation-based adaptive estimation introduced by A.H.Mohamed[15], which is proved to improve localization precision by over 50% in INS/GPS scenario. Innovation in each time slot is defined as,

$$v_t = z_t - H\hat{x}_t^-$$

Based on the whiteness of the filter innovation sequence in the past  $N$  slots, the filter statistical information matrices are adapted as follows:

$$R_t = C_{vt} - H P_k^- H^T \quad (10)$$

$$Q_t = K_t C_{vt} K_t^T \quad (11)$$

where

$$C_{vt} = \frac{1}{N} \sum_{j=t-N+1}^t v_j v_j^T \quad (12)$$

Due to space constraints, the proof of (10) and (11) are omitted. Intuitively, when observation position is far from predict position,  $C_{vt}$  increases sharply, because  $C_{vt}$  is quadratic sum of distance between observation position and predict position.  $C_{vt}$  can be treated as errors including observation uncertainties

and prediction uncertainties, the smaller  $P_t$  is, the higher  $R_t$  will be. Under such circumstance,  $R_t$  and  $Q_t$  grow larger, which means both observation and prediction model are unreliable.

## 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 three algorithms, traditional Kalman filtering (TKF), low-pass filtered Kalman filtering (LPKF), and adaptive low-pass filtered Kalman filtering (ALPKF).

### A. Simulations

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

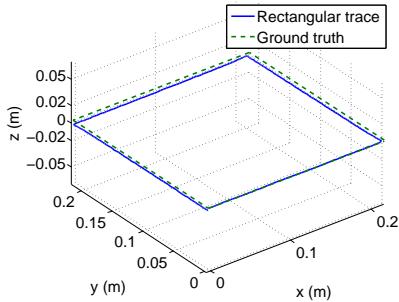
For better presentation, we simulate a rectangular motion with an acceleration and a deceleration process on each edge. When phone at corners, its speed is 0. We employ sensor errors measured in field tests. The orientation sensor has an error variance of 0.235, and that of accelerometer is 0.011. Accelerations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

Based on these simulation tests, we work out smartphone and key point localization precision implemented by three 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 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.

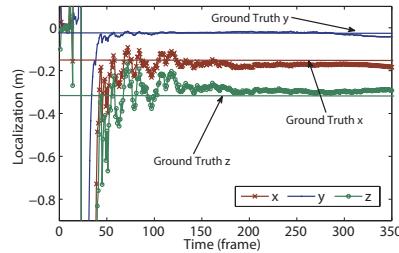
We conduct 100 groups of 10-second simulation, and Fig. 5(c) shows the cumulative distribution of errors produced by TKF, LPKF, ALPKF and naive acceleration accumulation. Compared with naive method, all three approaches in this work achieve much better results. LPKF performs the best, 90% of its error is under 2.6cm. ALPKF performs slightly poorer under the line of 90%, but its percentage of smaller-error result is much more than other approaches. We also conduct simulations based on linear trail and circular trail, and achieve desired results.

### B. Field Test

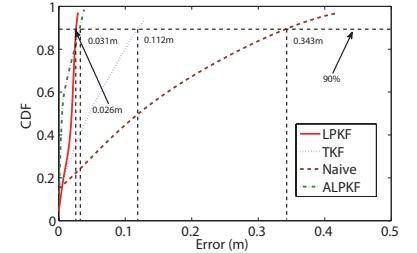
We implement Plotter on Sony Xperia 28i smartphone, with Android 4.0 operating system, embedded linear accelerometer and orientation sensor. Accelerometer and orientation sensor provide sampling rate as high as 26Hz and error variance as is shown in section V-A. On the software side, we employ OpenCV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



(a) A simulation trail with rectangular profile



(b) Key point's position converge to the ground truth gradually



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

Fig. 5. Simulation results

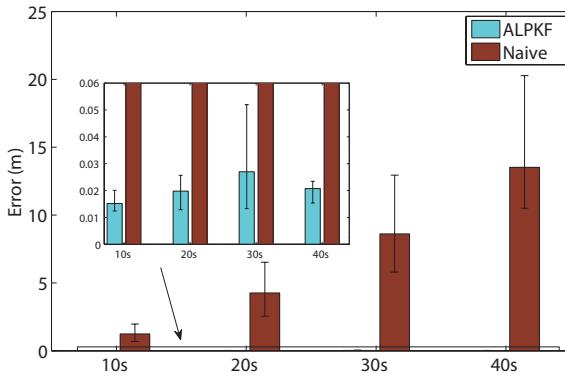


Fig. 6. Localization errors by ALPKF and naive acceleration accumulation in field tests

We recruit 5 students in our laboratory and let them move this phone near one fixed position, and return to that position at last. For better key point tracking effect, we choose a clean wall as the background and draw some black point on it randomly. Four groups of experiment are conducted with different moving time. Naive acceleration accumulation method is also implemented as control group. Filed tests result is shown in Fig. 6, where minimal and maximal errors are marked as a line segment on each bar. 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, ALPKF shows an attractive error result around 2cm on average and 6cm maximal. Seen from the result, there is no distinct increasing tendency with time grows in ALPKF.

## VI. CONCLUSION

In this paper, we propose a new indoor localization system on smart devices taking advantages of the basic idea in SLAM technology. We employ adaptive Kalman filtering for optimal position estimation and low-pass filter for raw data processing in our localization algorithm. Simulation and field test result shows it can provide as precise as centimeter grade accuracy, which is at the same level with other indoor localization approaches, without any additional devices or prior knowledge.

## 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*. ACM, 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*. ACM, 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*. IEEE, 2013, pp. 1–4.
- [4] C. P. Kumar, R. Poovalaah, 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*. IEEE, 2014, pp. 229–232.
- [5] C. Wu, Z. Yang, Y. Liu, and W. Xi, "Will: Wireless indoor localization without site survey," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 24, no. 4, pp. 839–848, 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. Ieee, 2000, pp. 775–784.
- [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," *Sensors Journal, IEEE*, vol. 13, no. 10, pp. 3754–3766, 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," *Instrumentation and Measurement, IEEE Transactions on*, vol. 62, no. 8, pp. 2237–2247, 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, pp. 1122–1135, 2013.
- [10] J. Xiong and K. Jamieson, "Arraytrack: A fine-grained indoor location system," in *NSDI*, 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*. Ieee, 1991, pp. 1442–1447.
- [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*. IEEE, 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 *Position, Location and Navigation Symposium-PLANS 2014, 2014 IEEE/ION*. IEEE, 2014, pp. 945–951.
- [14] J. McDonald, M. Kaess, C. Cadena, J. Neira, and J. 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.
- [15] A. Mohamed and K. Schwarz, "Adaptive kalman filtering for ins/gps," *Journal of geodesy*, vol. 73, no. 4, pp. 193–203, 1999.