HiQuadLoc: A RSS Fingerprinting based Indoor Localization System for Quadrotors

Xiaohua Tian, *Member, IEEE,* Zhenyu Song, Binyao Jiang, Yang Zhang, Tuo Yu, Xinbing Wang, *Senior Member, IEEE*

Abstract—Indoor localization for quadrotors has attracted much attention recently. While efforts have been made to perform location estimation of quadrotors leveraging dedicated indoor infrastructures, the low-cost and commonly used RSS fingerprinting based approach utilizing existing Wi-Fi APs has yet to be applied. The challenge is that the high-speed flight reduces the RSS measuring opportunities for fingerprints comparison; moreover, the 3-D space fingerprints collection incurs more overhead than in the traditional 2-D case. In this paper, we present HiQuadLoc, a RSS fingerprinting based indoor localization system for quadrotors. We propose a series of mechanisms including path estimation, path fitting and location prediction to deal with the negative influence incurred by the high-speed flight; moreover, we develop a 4-D RSS interpolation algorithm to reduce the site survey overhead, where 3-D is for the indoor physical space and 1-D is for the RSS sample space. Experimental results demonstrate that HiQuadLoc reduces the average location error by more than 50% compared with simply applying the RSS fingerprinting based approach for 2-D localization, and the overhead of RSS training data collection is reduced by more than 80%.

Index Terms—Indoor Localization, Quadrotor, Interpolation, Path Correction

1 Introduction

Unmanned Aerial Vehicles (UAVs) such as quadrotors are playing important roles in many civil and military applications [1], [2], [3]. Recently, applications of quadrotors in indoor spaces are increasingly popular. For example, Amazon Prime Air delivery system tries to utilize drones for indoor warehouse management [4], DroneService uses quadrotors to perform surveillance tasks within buildings [5], and the GimBall [6] tries to save lives in damaged buildings, where the flying parts of the quadrotor are suspended inside a flexible carbon fiber cage to avoid hurting people. The crux of these applications is the localization technique, since the vitally important tasks of quadrotors such as flight and mission control depend on the current location of the vehicle. How to localize the fast-moving quadrotors in indoor spaces is especially challenging due to the unreliability of the global positioning system (GPS) service. Efforts have been made to perform localization based on vision and ultrasound detection [7], [8], [9], [10], [11]. While such solutions could yield remarkable localization results, additional dedicated infrastructures need to be deployed, which leads to extra cost and hinders the wide deployment of quadrotors.

Localizing mobile devices such as cell phones in indoor spaces has been a hot research topic in past decades, where the RSS fingerprinting based approach is widely used [12], [13], [14]. The basic idea of the approach is to first construct a database to associate a location with the locally observed Wi-Fi RSS fingerprints in the offline phase. Then the location of a new coming device can be derived by comparing the currently observed fingerprints with those in the database [15], [16], [17]. As Wi-Fi access points (APs) are widely deployed inside almost every building, such an approach needs no additional dedicated devices installed. However, the advantage of traditional Wi-Fi fingerprinting based approach could not be directly borrowed by the localization system for quadrotors for the following two reasons:

First, the high-speed flight of quadrotors can severely impact localization accuracy. With the fingerprinting based approach, a mobile device normally conducts RSS measurements multiple times at a location in order to reduce the location estimation error incurred by the randomness of radio signal propagation. In particular, the device needs to obtain the SSIDs/MAC addresses [12] of surrounding Wi-Fi APs before measuring corresponding RSSes. The information is carried by the beacon signals periodically broadcasted by APs every 100-$500ms$. Consequently, a typical RSS measurement takes about 1-$2s$. Such latency is acceptable for localizing smartphones carried by people; however, a high-speed quadrotor cannot stay at the same position during the flight and the number of RSS measurements can be performed at a location is very limited. This could lead to significant error of location estimation due to the unpredictable radio propagation. Moreover, since the fingerprints database is stored in the server side and requires updating periodically, the online phase should be performed in the localization server, otherwise the communication overhead for retrieving fingerprints from the database is untractable. The communication delay between the vehicle and the server makes it difficult to accurately estimate the current location of the quadrotor in real time [18]. Our experiments show that the average location error by the scanning and communication latency could be 3-$4m$, which can hardly meet the demand of the quadrotor localization.

Second, the RSS fingerprinting based approach requires to collect a large amount of fingerprints in the offline phase. The purpose of traditional systems is to localize people, which can be regarded as a 2-D localization scenario. Although also needs efforts, the crowdsourcing based schemes could save the cost of data collection by leveraging mobile users' unconscious participa-

• *Xiaohua Tian, Zhenyu Song, Yang Zhang, Tuo Yu and Xinbing Wang are with the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China, 200240. Xiaohua Tian is also with National Mobile Communications Research Laboratory, Southeast University, China. E-mail: {xtian, sunnyszy, emberspirit, zhangyang93, yutuo, xwang8}@sjtu.edu.cn.*

tory sensing in the 2-D case [19], [20]. In contrast, the scenario of localizing quadroters is a 3-D case, where the RSS fingerprints at different heights also need to be collected. This increases not only the amount of training data to be collected but also the degree of difficulty in the site survey process, as finish the task with crowdsourcing is unpractical. Techniques such as surface-based interpolation has been used to reduce the required sampling rate of RSS fingerprints for the 2-D case [45], [46], [47]; however, they basically record the average value of RSSes at each surveyed point, which loses the information included in the probability distribution of RSS caused by the complex channel environment. How to reduce the overhead of collecting 3-D RSS training data with taking the statistical properties of RSSes into account is still an open issue.

In this paper, we present *HiQuadLoc*, a high-speed quadrotor localization system, which sheds light on how to resolve the two issues mentioned above. Our design of the system is based on the following two observations: First, quadrotors tend to fly in the straight line when moving at high speed indoor, and the localization at turning points is critical; second, the probability value for a given RSS value to appear at a given position is continuous in the 3-D indoor physical space as well as in the 1-D RSS sample space.

With HiQuadLoc system, we divide the physical space into cubes in the offline data training phase, and only measure the RSS probability distributions at parts of the cubes. We then estimate the missing RSS distributions at the rest of the cubes with interpolation. In the online localization phase, a quadrotor periodically measures RSSes and monitors the motion state of its own. These data are uploaded to a localization server, and the server derives the flight path of the quadrotor with the historical localization results taken into account. Our contributions are as following:

First, we present a series of solutions to counteract the negative effect of the high-speed of quadrotors on localization. We propose a *path estimation* scheme, which enhances the Kalman filter [19] to track the quadrotor and obtain the preliminary localization results. We design a *path fitting* scheme that leverages the approximate-rectilinear motion feature of quadrotors in high-speed state, which further smooths the flight path between turning points so that the accuracy of location estimation can be improved. Further, we propose a *location prediction* scheme to predict the real-time location of the quadrotor, which mitigates the communication latency between the vehicle and the localization server.

Second, we propose a 4-D RSS interpolation scheme to reduce the site survey overhead in the offline phase, where 3-D is for the indoor physical space and 1-D is for the RSS sample space. With our scheme, there is no need to sample RSS values in the entire physical space. Our interpolation scheme estimates the probability for a given RSS reading to appear in the non-surveyed space, in contrast to the interpolation schemes in the literature, which only estimates the average values of RSS readings at non-calibration points in the 2-D localization case. Experiment results show that our scheme could reduce the number of necessary RSS samplings by 80% in the offline data training phase.

Third, we implement a prototype system of HiQuadLoc with a quadrotor platform and conduct comprehensive experiments in a building with the volume of $2700m^3$. The experiment results show that HiQuadLoc reduces the average location error by more than 50% compared with traditional RSS-based systems. We also compare HiQuadLoc with the channel state information (CSI) based localization scheme by building CSI retrievable APs with off-the-shelf hardware. HiQuadLoc still presents the performance no less than the CSI counterpart, and an outstanding advantage of HiQuadLoc is the convenient deployability.

The remainder of the paper is organized as follows. Section 2 presents the system architecture. Section 3 describes the working process of HiQuadLoc. Section 4 shows the preliminary localization algorithm, which also includes the scheme of the 4-D RSS interpolation. Section 5 describes the path correction algorithm. Experiment results and related work are given in Section 6 and Section 7, respectively. Conclusion remarks are given in Section 8.

2 HiQuadLoc System Architecture

HiQuadLoc system consists of two parts: *localization server* and *quadrotor client*, as shown in Fig. 1. The localization server contains the RSS fingerprints database and is connected to the Internet. A quadrotor needs to use our system must have installed the quadrotor client and can connect to the localization server through WLAN or cellular networks. Also, it has to be covered by at least one Wi-Fi AP. In a flight, the quadrotor client keeps measuring the RSS and other physical quantities periodically. The computation for location estimation is at the server side so that the communication overhead for reading fingerprints database can be avoided.

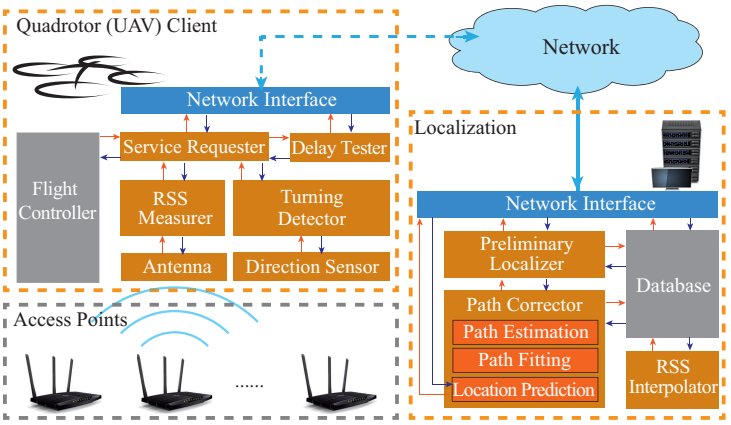


Fig. 1. Architecture of HiQuadLoc.

Our system can work in two modes: *continuous-tracking mode* and *single-locating mode*. In the continuous-tracking mode, the quadrotor monitors the latest location of its own continuously during the flight. The quadrotor client periodically uploads the latest metrical data to the localization server, and the server keeps estimating the up-to-date location of the quadrotor. The most recent localization results are replied to the client periodically. In the single-locating mode, the quadrotor communicates with localization service only when needed. The quadrotor client uploads part of the historical metrical data to the server when the quadrotor needs to localize itself. The server then estimates the current location of the quadrotor and returns the result. Since the principles of the two modes are similar, we mainly describe the continuous-tracking mode in this paper due to limitation of the space.

The quadrotor client mainly contains four components. During the flight, the *RSS Measurer* periodically measures the RSS for

each available AP. Meanwhile, the *Turning Detector* makes use of the direction sensor to detect the flight attitude of the quadrotor and judges whether it is turning. Since the actual flight attitude may not follow the control commands due to external disturbance, the attitude information cannot be obtained from the flight controller directly. The *Delay Tester* measures the communication delay between the quadrotor and the localization server, which denotes the time interval between when the client requests the localization service and when the result is received. All these components are controlled by the *Service Requester*, which uploads service requests and receives localization results. The service requester is also supposed to have access the quadrotor's flight controller if the flight controller exposes any developer interface.

The localization server mainly contains three components. The *Preliminary Localizer* operates a probability-based localization algorithm according to the metrical data uploaded. Each time it processes *only one set* of RSS measurement results and estimates where the quadrotor is at the time point *when the measurement is finished*. The output result is further processed by the *Path Corrector*, which executes the algorithms of path estimation, path fitting and location prediction to improve localization accuracy. The *RSS Interpolator* reduces the site survey overhead during the construction of HiQuadLoc radio map by using the 4-D RSS interpolation algorithm to be described in Section 4.2.

3 Working Process of HiQuadLoc

In this section, we introduce the working process of our system in detail. Like other typical indoor location systems based on RSS fingerprints, the process of localization in our system consists of *offline data training phase* and *online localization phase*.

3.1 Offline Data Training Phase

We first divide the space that requires the localization service into cubes with constant size. As a cube theoretically contains unlimited number of points, we just measure the RSS readings of the APs at the center of each cube, which requires great efforts in the case of 3-D localization. In our system, we propose the RSS interpolation algorithm based on 4-D space, which reduces the demand for high fingerprint sampling rate. In practice, we only measure RSS at 1 of each 8 cubes only. All the fingerprints information collected will be uploaded to the localization server and the server then processes the data according to the scheme described in Section 4.2. A detailed description of how to collect fingerprints could be found in Section 6.1.

3.2 Online Localization Phase

In the online phase, we divide the time axis into slots, each of which has a duration of T. During the whole online phase, the quadrotor keeps measuring the RSS from APs in each time slot. The specific process of the continuous-tracking mode consists of the following stages:

Stage 1: When the quadrotor has access to network and intends to localize itself, it first establishes links to the localization server via WLAN or cellular networks, and then sends a message to the server including the length of time slot T set by the quadrotor client. Note that T must be longer than the minimal RSS measurement time according to the quadrotor's hardware performance.

Stage 2: In each time slot, the quadrotor measures the RSS from surrounding APs. The result is included in an RSS-result message, and the client sends the message to the server immediately. Also, the message includes the value of communication delay T_d between the client and the server measured in the last localization process. Moreover, if the Turning Detector finds that the quadrotor is making a turning, the information will also be contained in the message of that time slot (Section 3.3). Note that *Stage* 2 is executed periodically, and the client is allowed to send the RSS-result message even if the localization result for the last time slot has not been returned by the localization server.

Stage 3: Each time when the localization server receives a RSS-result message, it estimates the position of the quadrotor according to the data uploaded as well as the historical localization results. The algorithm structure will be described in Section 3.4. The localization result is sent back to the client and its copy is stored at the server for future localization.

Stage 4: Once the client receives the result, the Delay Tester calculates the time interval between the sending out of the RSS-result message and the return of the result, and sets the interval as the new T_d.

For the single-locating mode, the quadrotor still periodically measures RSS. The difference is that the results are temporarily stored in the client without being uploaded. When the Turning Detector finds that the quadrotor has passed a corner, the data measured before the corner are deleted to save memory space. When the quadrotor needs to locate itself, the Delay Tester first measures the communication delay T_d between the client and the server. Then all the stored data, including T_d, will be uploaded to the server.

3.3 Turning Detection

To improve the performance of path estimation at the corners, the quadrotor client needs to detect the turning motion of the quadrotor. Our system utilizes the direction sensor installed on the quadrotor to detect turning.

Different from regular UAVs, it is harder for the client to detect the flight direction of quadrotors because a quadrotor can make lateral movements without changing its heading direction. However, we notice that when a quadrotor is moving in a specific direction, its normal vector will have a drift angle for the same direction, as shown in Fig. 2(a). This is caused by the propulsion principle of the quadrotors. Thus we measure the normal vector instead of the heading direction of the quadrotor.

When fixing on a quadrotor, a typical direction sensor returns three values d_x, d_y and d_z, as shown in Fig. 2(a). Among them we use d_y and d_z only to calculate the drift angle of the normal vector. According to the figure, we get (omit the cases where $d_y = 0$ or $d_z = 0$ for concision):

$$\alpha = \arctan(\frac{\tan d_z}{\tan d_y}) - 90^\circ \mathrm{sgn}(d_z)[\mathrm{sgn}(d_y) - 1], \quad (1)$$

$$\beta = |\arctan(\frac{\tan d_y}{\cos \alpha})|, \quad (2)$$

where α is the included angle between the projection of the normal vector and the front of the quadrotor. In Equation (1) we project the range of α to $(-180^\circ, 180^\circ]$, as shown in Fig. 2(b). The angle

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2016.2636823, IEEE Transactions on Mobile Computing

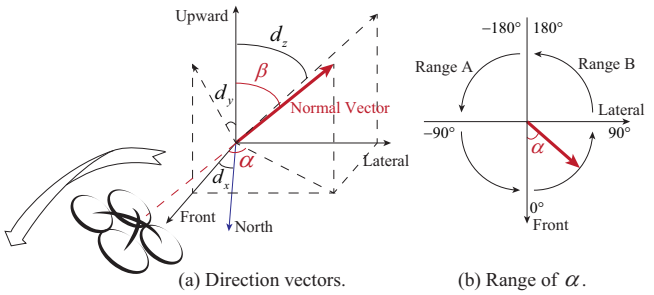


Fig. 2. During a turning the direction of the normal vector (determined by α and β) changes.

β is the drift angle of the normal vector. Note that we cannot use the heading direction determined by d_x alone to judge whether the quadrotor is turning, because during a turning based on lateral movement, d_x may not change (Fig. 3).

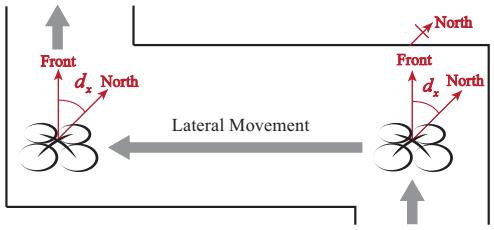


Fig. 3. During the turning based on lateral movement, d_x does not change.

During a flight, the Turning Detector periodically measures d_y, d_z, processes them by low-pass filter, and calculates β for each time interval T_s, where $T_s \ll T$. With constant thresholds α_T and β_T, two different points are simultaneously followed:

- If $\beta < \beta_T$ for continuous duration nT_s, the quadrotor is hovering, during which it may change its direction. A *turning-start signal* will be included in the next RSS-result message. Once $\beta \geq \beta_T$ for another nT_s, a *turning-end signal* will be included in the next RSS-result message. Note that even if the quadrotor does not change its heading direction during the hovering, it is still necessary to notice the Path Corrector adjust the parameters of Kalman filter, otherwise it cannot perform well for the case of hovering.
- When $\beta \geq \beta_T$ for more than nT_s, the Turning Detector starts to calculate α and continuously updates the mean value $\bar{\alpha}$ for its recent N_α values. To avoid the jump between $-180°$ and $180°$, if the recent N_α values simultaneously contain the angles in Range A and B (Fig. 2(b)), we add $360°$ to the angles in Range A, and add $360°$ to the newest α if it is also in Range A. Once $|\alpha - \bar{\alpha}| \geq \alpha_T$ for more than mT_s, which means that the quadrotor is turning, the Turning Detector includes the turning-start signal in the next RSS-result message. After that, when $\beta \geq \beta_T$ and $|\alpha - \bar{\alpha}| < \alpha_T$ for more than mT_s, which indicates the end of the turning, the turning-end signal is included in the next RSS-result message. Note that if $\beta < \beta_T$ happens during this time interval, the previous point is still followed. Moreover, the turning based on lateral movement can also be detected by this method.

With the turning-start and turning-end signals, the localization algorithm can take the impact of turning into account, which is to be described in Section 5.2 in detail.

3.4 Structure of Localization Algorithm

The localization algorithm in our system consists of two subalgorithms: *preliminary localization algorithm* (Section 4.3) and *path correction algorithm* (Section 5). The preliminary localization algorithm is based on the statistical properties of RSS fingerprints, and the 4-D RSS interpolation (Section 4.2) is applied to reduce efforts for radio map construction. The path correction algorithm contains the methods of path estimation, path fitting and location prediction, which further improves the accuracy of localization based on the result obtained from the preliminary localization algorithm.

In the continuous-tracking mode, when the localization server starts to process a new set of RSS data measured at time slot k, the following steps will be executed:

Step 1: The preliminary localization algorithm is executed first. The estimated position derived by this algorithm is still preliminary and has only low accuracy. We denote it by $\check{s}_k$.

Step 2: Along with the historical localization results, $\check{s}_k$ is revised by the path estimation module. The new estimated position is denoted by $\bar{s}_k$.

Step 3: Based on the method of path fitting, the motion path is smoothed and $\bar{s}_k$ is replaced by $\hat{s}_k$.

Step 4: According to the communication delay T_d, $\hat{s}_k$ is further processed by the location prediction method, and the final output is $\tilde{s}_k$.

In the single-locating mode, the server continuously executes *Step* 1 and 2 for all the RSS data uploaded, and then execute *Step* 3 and 4 for the last $\bar{s}_k$ only. Figure 4 shows an example for the relationship among $\check{s}_k$, $\bar{s}_k$, $\hat{s}_k$ and $\tilde{s}_k$. The algorithms mentioned will be introduced in detail in the following sections.

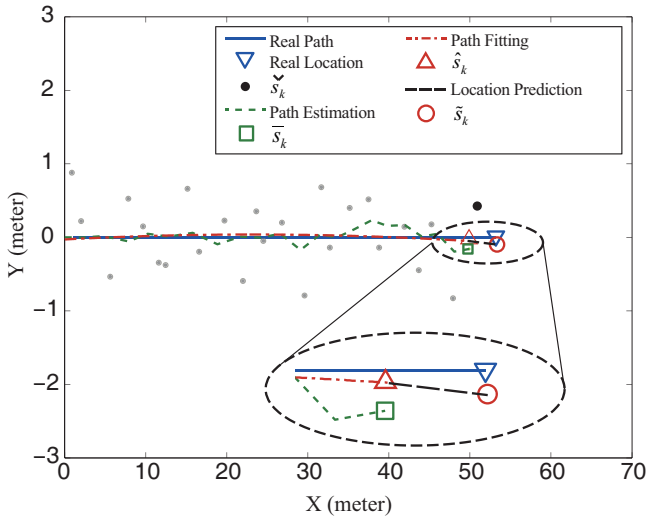


Fig. 4. The relationship among $\check{s}_k$, $\bar{s}_k$, $\hat{s}_k$ and $\tilde{s}_k$.

4 Preliminary Localization Algorithm

In this section, we present the algorithm for determining the quadrotor location in an 802.11 WLAN environment. The tradition algorithm for data processing in offline data collection is to be modified to accommodate the 4-D RSS interpolation algorithm.

4.1 Theoretical Basis

Our algorithm uses probability distributions to enhance accuracy and tackle the noisy nature of wireless channels. Generally, we use l to denote a location in the indoor localization region, o to denote a training datum of RSS from an AP, and $\bar{o}$ to denote an observation variable of RSS.

For the purpose of localization, when an observation vector $\bar{O} = \{\bar{o}_1, ..., \bar{o}_K\}$ at a location is given, we need to find l such that $P(l|\bar{O})$ is maximized, that is, we want $argmax_l[P(l|\bar{O})]$. According to the Bayes rule, and since $P(\bar{O})$ is constant for all l, we have:

$$argmax_l[P(l|\bar{O})] = argmax_l[P(\bar{O}|l)P(l)], \tag{3}$$

where $P(l)$ is the prior probability of being at location l before knowing the value of observation variable. Since the distance between two contiguous localization results is limited by T, in the continuous-tracking mode $P(l)$ can be used to shrink the possible positions of the quadrotor. In our system, only the cubes which are less than $V_m T$ meters from the last localization result have $P(l) = 1$, otherwise $P(l) = 0$, where V_m is the maximum velocity of the quadrotor. Assuming the APs are independent, we have the following method to estimate $P(\bar{O}|l)$:

$$\begin{aligned} P(\bar{O}|l) &= \prod_{r=1}^{K} p(\bar{o}_r|l) \approx \prod_{r=1}^{K} [\frac{1}{N}\sum_{q=1}^{N} K(\bar{o}_r; o_l^q)] \\ &= \prod_{r=1}^{K} [\frac{1}{N}\sum_{q=1}^{N} \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{(\bar{o}_r - o_l^q)^2}{2\sigma^2})], \end{aligned} \tag{4}$$

where the function $p(\bar{o}_r|l)$ is the probability distribution of RSS value at the position l. $K(\bar{o}_r; o_l^q)$ is the *kernel function*, and we apply the widely-used *Gaussian kernel*. The parameter o_l^q is the q_{th} training observation at the cube l, N is the number of the training data at each cube, and σ is an adjustable parameter that determines the width of the kernel.

4.2 4-D RSS Interpolation Scheme in Offline Phase

In this section, we introduce the scheme of 4-D RSS interpolation applied in the offline phase. Assume that we have a room with $X \cdot Y \cdot Z$ cubes, where X, Y, Z denote the units in length, width, height direction respectively. We use G_{ijh} to denote a cube with coordinate (i, j, h), $i = 1, ..., X$, $j = 1, ..., Y$, $h = 1, ..., Z$.

In our localization algorithm, each AP is assigned a fingerprint map denoted by F_s, where $s = 1, ..., M$ and M is the number of APs in our system. For each fingerprint map, we use G_{ijh}^s to represent a specific cube in the map. As most of the previous work based on RSS fingerprint, we need to collect training data for each cube in each fingerprint map:

$$O_s = \{O_{ijh}^s | i = 1, 2, ..., X; j = 1, 2, ..., Y; h = 1, 2, ..., Z\}, \tag{5}$$

where

$$O_{ijh}^s = \{o_{ijh}^{sq} | q = 1, 2, ..., N\}. \tag{6}$$

Each O_{ijh}^s here includes N RSS signals measured in G_{ijh}^s and o_{ijh}^{sq} is the q_{th} training observation.

Normally, the cost of such training data collection is reasonable in 2-D localization, because it increases linearly with $X \cdot Y$, i.e., the area of the localization region. However, in the case of 3-D localization, the cost is proportional to $X \cdot Y \cdot Z$, i.e., the volume of the building. Thus the cost cannot be ignored. In our system, we propose the algorithm of 4-D RSS interpolation to accommodate the nature of 3-D localization.

We first let $C \triangleq \{(i, j, h) | i \in [1, X], j \in [1, Y], h \in [1, Z]\}$, $C_o \triangleq \{(i, j, h) | i = 1, 3, 5, ..., X; j = 1, 3, 5, ..., Y; h = 1, 3, 5, ..., Z\}$, and $C_e \triangleq C - C_o$, where we assume X, Y and Z are odd numbers without losing generality. Different from previous works, we collect training data for each 8 *volume units*:

$$O'_s = \{O_{ijh}^s | (i, j, h) \in C_o\}. \tag{7}$$

According to Equation (4), for each cube G_{ijh}^s the resulting density estimate for an observation $\bar{o}$ is a mixture of N equally weighted density functions

$$p(\bar{o}|G_{ijh}^s) = \frac{1}{N}\sum_{q=1}^{N} \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{(\bar{o} - o_{ijh}^{sq})^2}{2\sigma^2}), (i, j, h) \in C_o. \tag{8}$$

Specifically, to get a smoother estimation for RSS distribution, we use $\sigma = 10$. We iterate the observation $\bar{o}$ from its lower bound to upper bound to calculate the density distribution in the G_{ijh}^s. Specifically, we traverse $\bar{o}$ from $-150dBm$ to $0dBm$ and then normalize the density distribution to probability distribution for the G_{ijh}^s.

Let $p_s(\bar{o}, i, j, h) \equiv p(\bar{o}|G_{ijh}^s)$, then we could get the values of $p_s(\bar{o}, i, j, h)$ when $\bar{o} \in [-150, 0]$ and $(i, j, h) \in C_o$ according to the training data. To estimate the rest of the values, we could apply the method of interpolation. Note that $p_s(\bar{o}, i, j, h)$ is continuous in the 4-D space

$$\{(\bar{o}, i, j, h) | \bar{o} \in [-150, 0], (i, j, h) \in C\}. \tag{9}$$

Thus we can use the cubic spline interpolation in the 4-D space to estimate the $p_s(\bar{o}, i, j, h)$ when $(i, j, h) \in C_e$. Denote these interpolated values by $p_s(\bar{o}, i, j, h)' \equiv p(\bar{o}|G_{ijh}^s)'$, and we have the estimated probability for $\bar{o}$ to appear in G_{ijh}^s is

$$\hat{p}(\bar{o}|G_{ijh}^s) = \begin{cases} p(\bar{o}|G_{ijh}^s), & \text{if } (i, j, h) \in C_o \\ p(\bar{o}|G_{ijh}^s)', & \text{if } (i, j, h) \in C_e \end{cases}. \tag{10}$$

Finally, we train each G_{ijh}^s in each fingerprint map F_s with training data O'_s. Since the value of RSS measured is discrete, we can previously store $\hat{p}(\bar{o}|G_{ijh}^s)$ for each integral value of $\bar{o}$ in the database of the localization server. We are to show in Section 6.1 that the 4-D RSS interpolation algorithm can reduce the cost of fingerprint collection effectively, and the reduction in accuracy is acceptable.

4.3 Preliminary Position Estimation in Online Phase

We assume that the result of each RSS measurement operated by the quadrotor is denoted by

$$\bar{O} = \{\bar{o}_r^s | r = 1, ..., K, \ \ s = 1, ..., M\}, \tag{11}$$

where $\bar{o}_r^s$ is the r_{th} RSS value from the s_{th} AP. The symbol M is the total number of APs, and K is the number of times the quadrotor has measured for the same AP. Note that if the quadrotor is not covered by the s_{th} AP, $\bar{o}_r^s$ will be labeled by NaN and thus does not impact the following calculation.

For each $\bar{o}_r^s$, we can calculate $\hat{p}(\bar{o}_r^s|G_{ijh}^s)$ according to Equation (10). Since $\hat{p}(\bar{o}|G_{ijh}^s)$ has already been stored, we can get $\hat{p}(\bar{o}_r^s|G_{ijh}^s)$ directly by accessing the database in the localization server, which reduces our system's computation complexity.

In this paper, we assume that the signal strengths from the APs are independent. This assumption is justifiable for a well designed

802.11 network, where each AP runs on a non-overlapping channel; therefore, we could estimate the joint probability using the marginal probability. According to Equation (3) and (4), we get the probability for that the client observes $\bar{O}$ when it is in G_{ijh}:

$$P(G_{ijh}|\bar{O}) = P(G_{ijh}) \cdot \prod_{s=1}^{M} \prod_{r=1}^{K} \hat{p}(\bar{o}_r^s|G_{ijh}^s). \quad (12)$$

Note that $P(G_{ijh})$ is retained to shrink the possible location region. Finally, the joint probability matrix can be denoted by

$$P_{joint} = \{P(G_{ijh}|\bar{O})|(i,j,h) \in C\}. \quad (13)$$

Based on Equation (3), we then simply find the largest $P(G_{ijh}|\bar{O})$ in P_{joint} and its corresponding G_{ijh} denotes the estimated location for the vehicle. If this is the result for time slot k, we denote it by the preliminary localization result $\check{s}_k = [i,j,h]^T$, which will be further processed by the path correction algorithm. With the help of $P(G_{ijh})$, the algorithm complexity is $O(MKT)$, which are of the constant order.

5 Path Correction Scheme

In this section, we propose the path correction scheme based on a revised Kalman filter. The information of turning is taken into account to improve the performance of original Kalman filter at indoor corners. Also, we use the method of fitting to reach a higher accuracy, and also consider the negative effect of communication delay.

5.1 Path Estimation

The basic idea of path estimation is to further optimize the location estimation of the quadrotor, based on the location estimation result obtained from the RSS comparison as described in Section 4.3. We here utilize the concept of Kalman filter [30] for the optimization, where the preliminary location estimation result is regarded as the input of the "Predict" phase of the Kalman filter. The preliminary result is then passed to the "Correct" phase of the Kalman filter, which results in a more accurate estimation.

In the predict phase of Kalman filter, we assume that the quadrotor keeps moving in the 3-D physical space, and the RSS measurement is operated in every time slot. We denote the state of the quadrotor in the k_{th} time slot by

$$s_k = [x_k, y_k, z_k, v_{xk}, v_{yk}, v_{zk}]^T, \quad (14)$$

where (x_k, y_k, z_k) is the real position of the quadrotor at the k_{th} time slot, and (v_{xk}, v_{yk}, v_{zk}) is its velocity vector. Thus the motion could be modeled as

$$s_k = \begin{bmatrix} E_3 & TE_3 \\ \mathbf{0} & E_3 \end{bmatrix} s_{k-1} + w_{k-1} = Fs_{k-1} + w_{k-1}, \quad (15)$$

where E_3 is a three-rank identity matrix, T is the length of a time slot, and w_k is the process noise following Gaussian distribution $w_k \sim N(0, \sigma_w^2)$.

Recall that the preliminary localization result at the kth time slot is denoted by $\check{s}_k = [i,j,h]^T$ in Section 4.3. With the Gaussian noises for localization, the preliminary localization model can be written as

$$\check{s}_k = [E_3, \mathbf{0}]s_k + u_k = Hs_k + u_k, \quad (16)$$

where u_k is the preliminary localization noise that follows Gaussian distribution $u_k \sim N(0, \sigma_u^2)$.

Consequently, the details of the "predict" phase in our system are as following:

- Project the state ahead. This is as shown in Eq. 16.
- Project the error covariance ahead.

$$\bar{P}_k = FP_{k-1}F^T + Q_k, \quad (17)$$

where $Q_k = \mathbb{E}(w_k w_k^T)$ is the process noise covariance, and initially $P_0 = \mathbf{0}$.

The details of the "correct" phase in our system are as following:

- Compute the Kalman gain:

$$K_k = \bar{P}_k H^T (H\bar{P}_k H^T + R_k)^{-1}, \quad (18)$$

where $R_k = \mathbb{E}(u_k u_k^T)$ is the preliminary localization noise covariance. Note that different from normal Kalman filter, the value of u_k would be changed in each time slot, which is to be shown in Section 5.2.

- Update the estimated state vector with $\check{s}_k$:

$$\bar{s}_k = F\bar{s}_{k-1} + K_k(\check{s}_k - HF\bar{s}_{k-1}), \quad (19)$$

where $\bar{s}_0 = [\check{s}_0^T, d_0^T]^T$ initially and d_0 is the initial unit direction vector provided by the direction sensor installed on the quadrotor. Output $\bar{s}_k = [\bar{x}_k, \bar{y}_k, \bar{z}_k, \bar{v}_{xk}, \bar{v}_{yk}, \bar{v}_{zk}]^T$.

- Update the error covariance:

$$P_k = (E_6 - K_k H)\bar{P}_k, \quad (20)$$

where E_6 is a six-rank identity matrix.

Since $(\bar{x}_k, \bar{y}_k, \bar{z}_k)$ is the estimated coordinate, we use $\bar{s}_k$ to denote the new estimated position directly in the following sections. Thus for each time slot the path estimation method optimizes the preliminary localization result $\check{s}_k$ and outputs $\bar{s}_k$ as the new estimated position. We are to show that a higher accuracy can be achieved by $\bar{s}_k$ in Section 6.2 with experiment results. As each execution of path estimation is to find the result of the "correct" phase once, the complexity is $O(1)$.

5.2 Parameter Readjustment During Turning

During the process of path estimation, one disadvantage of Kalman filter is that the localization accuracy at corners decreases obviously, as shown in Fig. 5. We can see that the estimated locations of the quadrotor deviate far from the real path of the quadrotor at the corner. This is caused by the unchanged weighting factor for the historical localization results. To overcome this we change the parameters of Kalman filter to adapt it to the turning motions of quadrotors.

As introduced in Section 3.3, once the Turning Detector detects a turning motion during a flight, it uploads turning-start and turning-end signals at the beginning and the end of the turning respectively. Then the following operations are executed:

Step 1 : At the beginning of the turning, change σ_u^2 to $\bar{\sigma}_u^2$, where $\bar{\sigma}_u^2$ is a smaller value than σ_u^2.

Step 2 : At the end of the turning, change $\bar{\sigma}_u^2$ back to σ_u^2.

The value of R_k should be recomputed after each step, since R_k actually controls the balance for the weighting factors of historical and newest localization results; reducing σ_u^2 leads to a higher weighting factor for the newest localization result, according to Equation (18) and (19). As shown in Fig. 5, at corners

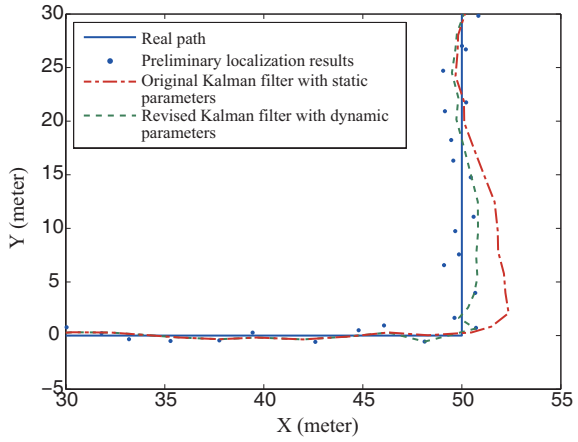


Fig. 5. The performance of Kalman filter at a corner (for 2-D case).

a high σ_u^2 would cause the overshoots of the path estimated by the Kalman filter. Thus relying more on the newest localization result at the corners can reduce these overshoots. We are to evaluate this revised scheme in Section 6.3. The change of σ_w^2 during turning has similar performance, thus we mainly focus on σ_u^2 in this paper.

5.3 Path Fitting

Kalman filter works well only when its parameters such as σ_w^2 match with the real situation, which can hardly be satisfied in practice since the indoor environment varies. Thus the jitter of the estimated path as shown in Fig. 4 cannot be totally avoided. We apply the method of curve fitting to mitigate this impact. This approach is based on the nature that a quadrotor tends to move in nearly straight lines in indoor environments, e.g., flying along a long corridor. Since the flight path of the quadrotor is actually a 3-D curve, the computation complexity of 3-D curve fitting is large and is severely impacted by the initial parameters of the curvilinear function. To reduce the response time of the localization server, we focus on the projected curve of the flight path on the 2-D ground and the changing curve of $\bar{z}_k$ with time respectively.

Still with the help of the messages about turning motions, the following operations are executed:

Step 1 : Once the quadrotor has finished a turning, the current time slot is labeled by T_c.

Step 2 : For $\{\bar{s}_k | k = T_c, T_c + 1\}$, output $\hat{s}_k = \bar{s}_k$ directly.

Step 3 : For $\{\bar{s}_k | k \geq T_c + 2\}$, we set $\bar{s}'_k = [\bar{x}_k, \bar{y}_k]^T$, and fit the data $\{\bar{s}'_l | T_c \leq l \leq k\}$ with the method of quadratic polynomial fitting. To avoid the case that the flight path is perpendicular to the x axis or the y axis, which leads to the bad performance of fitting, we exchange the two axes and choose the fitted curve with the maximum correlation coefficient. The result is denoted by curve L_k. Approximatively find the nearest point to $\bar{s}'_k$ on L_k, denote it by $[\hat{x}_k, \hat{y}_k]$. Also, we fit the data $\{[\bar{z}_l, l] | T_c \leq l \leq k\}$ with the quadratic polynomial fitting. The value of the result function at time slot k is denoted by $\hat{z}_k$. Output $\hat{s}_k = [\hat{x}_k, \hat{y}_k, \hat{z}_k, \bar{v}_{xk}, \bar{v}_{yk}, \bar{v}_{zk}]^T$. Go back to *Step* 1 once a new turning begins.

The length of a straight line segment between two corners is limited by the dimension of the building. If we use L_{max} to denote the maximum number of predicted locations along the longest straight line segment in the path predicting phase, the complexity for path fitting is $O(L_{max})$. The computational complexity of the polynomial fitting is limited and does not impact the responsiveness of the localization server.

Note that the reason why we do not fit $\{\check{s}_l | T_c \leq l \leq k\}$ directly is that, the error of the preliminary results $\check{s}_k$ is much larger than that of $\bar{s}_k$, which leads to severe changes between each L_k. Thus the path estimation method is still necessary.

5.4 Location Prediction

Due to the communication delay between the client and the server, there still exists non-negligible localization error. Assume that the client sends the RSS-result message at time point t, due to the cost of computation time and transmission delay, when the client receives the localization result returned, the time is $t+T_d$. Suppose the average velocity of the quadrotor during T_d is v_d, then the additional localization error will be at most $v_d \cdot T_d$. Thus for the high-speed moving quadrotors, it is necessary for our system to take the impact of communication delay into consideration.

In our system, we develop a location prediction scheme to predict the real location of a quadrotor according to the T_d it has uploaded. Moreover, the output of the path fitting scheme, $\hat{s}_k = [\hat{x}_k, \hat{y}_k, \hat{z}_k, \bar{v}_{xk}, \bar{v}_{yk}, \bar{v}_{zk}]^T$ contains the estimated velocity vector of the quadrotor; therefore, we can extend the motion curve of the quadrotor by

$$\tilde{s}_k = [\hat{x}_k + T_d \bar{v}_{xk}, \hat{y}_k + T_d \bar{v}_{yk}, \hat{z}_k + T_d \bar{v}_{zk}, \bar{v}_{xk}, \bar{v}_{yk}, \bar{v}_{zk}]^T. \quad (21)$$

Finally, the server responds the client with $\tilde{s}_k$, which is the final result of the localization process at time slot k. The relationship among $\check{s}_k$, $\bar{s}_k$, $\hat{s}_k$ and $\tilde{s}_k$ has been shown in Fig. 4, and we are evaluate the performance of the path correction scheme in Section 6.2.

It is straightforward that the complexity of the location prediction is $O(1)$. We can see that the computational complexity in each phase of localization is tractable. As the communication delay is dependent on the hardware and non-changeable, the low computational complexity could help improve the responsiveness of the server, which is important for high-speed quadrotors.

6 Experiment Results

In this section, we evaluate the performance of HiQuadLoc by conducting experiments with a quadrotor localization testbed. The experiments consist of the following parts: First, we evaluate the performance of the 4-D RSS interpolation algorithm in offline phase by operating an independent test in a lobby. Second, we depoly our system in a corridor and several lobbies, and evaluate the performance of path estimation, path fitting and localization prediction scheme. The benefit of turning detection is to be confirmed, and the impact of velocity on accuracy is to be shown. Third, we build CSI retrievable APs to implement CSI based localization for the quadrotor, and compare the performance of HiQuadLoc scheme with the CSI based scheme. Fourth, we verify that the power consumption of the HiQuadLoc is negligible compared with the basic power consumed by the quadrotor with experiment.

6.1 Evaluation of 4-D RSS Interpolation Algorithm

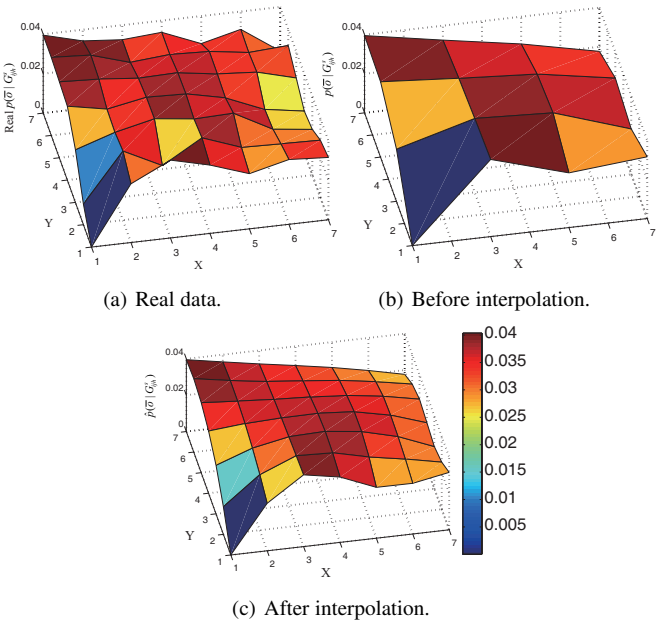
As the offline phase could significantly influence the accuracy of localization, we verify that the 4-D RSS interpolation algorithm can reduce the cost of RSS fingerprint collection without causing

severe drop in localization accuracy. We conduct the experiment in a lobby with an interior volume of more than $(9 \times 9 \times 5)m^3$. In the test, we deploy four Wi-Fi APs (HUAWEI E586Bs-2) at the four corners of the lobby.

For the offline data training phase, the lobby is divided into 405 cubes and we collect the RSS fingerprints at the centers of them using 6 Android smartphones. All of the smartphones are Nexus 4 with Android Jelly Bean (4.2.2) as their operation systems. The size of each cube is $1m \times 1m \times 1m$, and we collect 30 sets of RSSes from each AP for each cube. We then generate two different sets of RSS fingerprints. For the *full fingerprints*, we use all the data collected from all the 405 cubes for localization (i.e., O_s in Equation (5)), which takes around 527 minutes. This is the traditional approach widely used by most of works in the literature. For the *interpolated fingerprints*, we use the RSS data of 1 cube for every 8 cubes, i.e., we could use the RSS data collected from only 50 cubes for the training phase. However, we use data collected from 75 cubes (i.e., O'_s in Equation (7)) to conduct the localization experiment due to the irregular shape of the lobby, which takes around 100 minutes. Then we use the 4-D RSS interpolation algorithm to approximate the data in the rest cubes. One example of the RSS probability distribution generated by an AP before and after the interpolation is shown in Fig. 6.

For the online localization phase, we use both the full fingerprints and the interpolated fingerprints to perform localization and compare the corresponding performance. To present a fair comparison, we only record preliminary localization results $\check{s}_k$ during each localization process. The localization tests are done by 405 times for each set of fingerprints, and for each test the smartphone will measure RSS 5 times for each AP. Thus we set $M = 4$, $K = 5$ in Equation (12).

The result is shown in Fig. 7. It can be seen that, even if the usage of interpolation reduces the accuracy of localization by $0.10m$ to $0.17m$ when the number of APs varies, the reduced accuracy is still trivial and it will be proved that this decrease is negligible compared with the accuracy gain achieved by the Path Corrector. Moreover, only 19% of the RSS training data are used by the interpolated fingerprints, which means that the workload of fingerprint collection can be reduced by 81%. Thus it is confirmed that our algorithm can significantly reduce the cost of data collection in the offline phase.



(a) Real data. (b) Before interpolation.

(c) After interpolation.

Fig. 6. The probability for RSS$= -75dBm$ to appear at the cubes which are $1m$ from the ground.

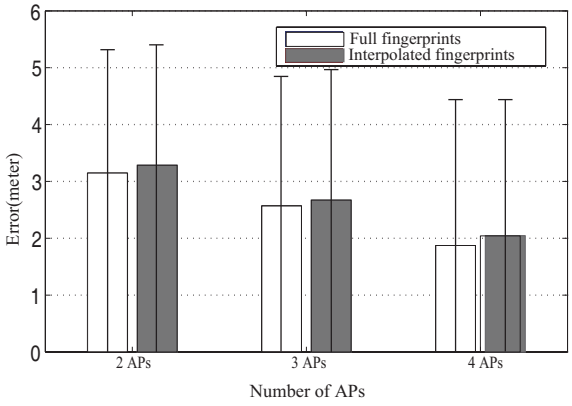
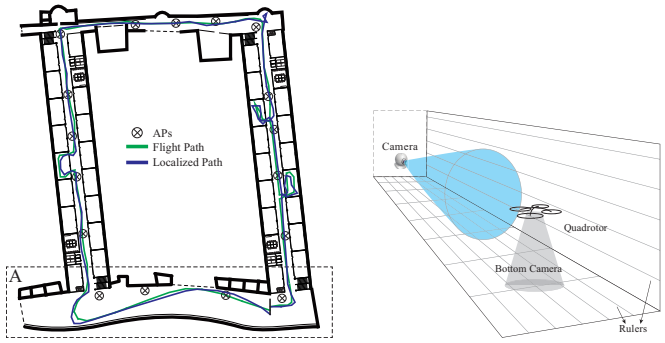


Fig. 7. Average localization errors using the two sets of fingerprints.

6.2 Evaluation of Localization Schemes

In this section, we evaluate the performance of the proposed schemes facilitating localization. The 2-D layout of the building where the experiments are conducted is illustrated in Fig. 8(a). The total area covered is more than $1100m^2$, and the total interior space volume is over $2700m^3$. We install 20 APs (HUAWEI E586Bs-2) along the corridor, and each reference point is covered by at least two APs. The quadrotor used in the experiment is shown in Fig. 9. The onboard computer is MSP430F5529, with 128KB Flash, 8KB RAM and 25MHz CPU frequency. The direction sensor is YAS530B-PZ. The WLAN card HLK-RM04 and MiFi HUAWEI E5776 are installed. The quadrotor can connect to our localization server via cellular networks. Since it is hard for a single AP to cover such a long corridor, M in Equation (12) is set equal to the real number of APs that the quadrotor has discovered. Since the quadrotor keeps moving during the test, $K = 1$ in Equation (12). Moreover, we set $T = 1.5s$, $T_s = 20ms$, $\alpha_T = 5°$, $\beta_T = 10°$, $m = 25$, $n = 25$, $N_\alpha = 20$, $V_m = 15m/s$,



(a) Experiment field. (b) Video recording.

Fig. 8. Experiments Settings. The flight path and localized path in Fig. 8(a) are the records of a single test (the height is not shown in the figure).

$\sigma_w^2 = 5$, $\sigma_u^2 = 20$ and $\bar{\sigma}_u^2 = 8$. The communication delay T_d between the quadrotor and the server is 0.335-$0.908s$ during the test. The computing time of each localization process is around $10ms$, which is negligible compared with T_d.

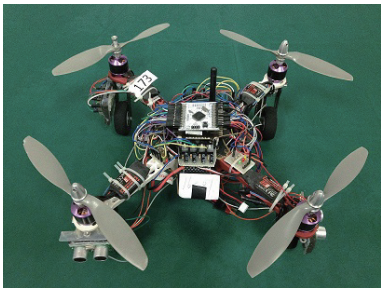


Fig. 9. The quadrotor experimental platform.

During the offline data training phase, we divide the experiment space into 2700 cubes with size $1m \times 1m \times 1m$, and collect the RSS fingerprints from 485 (18%) of them. For each cube we collect 30 sets of RSS from each AP. The data collected are uploaded to the sever and the 4-D RSS interpolation algorithm is executed.

During the online localization phase, we use a web camera to record the real location of the quadrotor. As shown in Fig. 8(b), the bottom of the camera on the quadrotor is used to record its 2D track during the flight. There are rulers drawn on the ground and we thus can get the real position of the quadrotor according to the test time and the video record. For the height of the quadrotor, we use an independent camera deployed at the ends of the corridors, which also serves as the reference point for rulers.

We initialize our client, and control the quadrotor to fly along the corridor. The test is repeated for 10 times, and 1268 localization results are recorded. For instance, according to the video record, the flight path during one test is shown in Fig. 8(a). We do not strictly constrain the speed of the quadrotor in this case, and the impact of the flight speed is to be evaluated in Section 6.4.

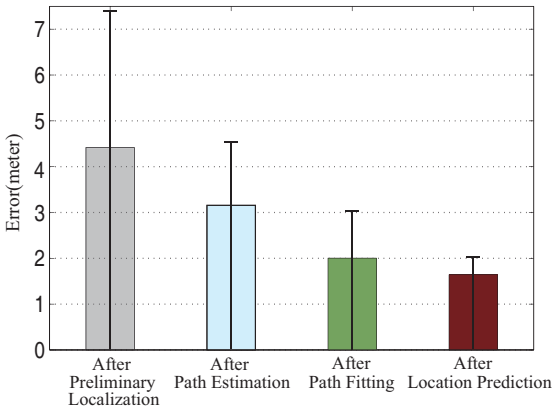


Fig. 10. Average location errors after the process of different algorithms.

During the flight, we use the continuous-tracking mode only, since the two modes have similar performance. The results of localization are recorded, and we analyze the results output by the methods of preliminary localization, path estimation, path fitting and location prediction respectively. The average location errors for these results are shown in Fig. 10.

For traditional localization systems based on RSS, their performance is the same with that of the preliminary localization algorithm. Thus if they are applied to the case of high-speed quadrotor directly, the average error will be $4.41m$, which can hardly meet with the need of quadrotor localization. If we apply the path estimation method, the average error can be reduced to $3.16m$. Based on the fact that quadrotors move in nearly straight lines for the indoor case most of the time, the method of path fitting further reduces the average error to $2.00m$. Finally, considering the delay of transmission, the location prediction method achieves an average location error of $1.64m$.

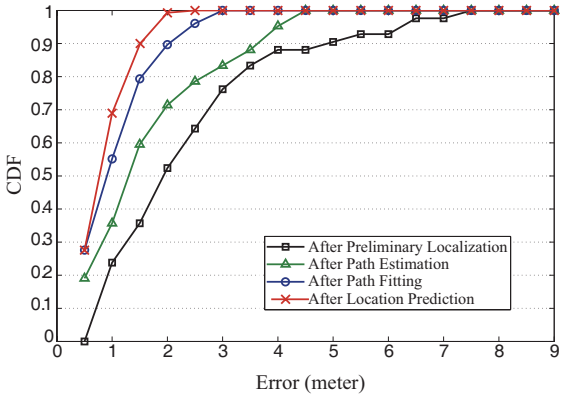


Fig. 11. CDF of location errors after the process of different algorithms.

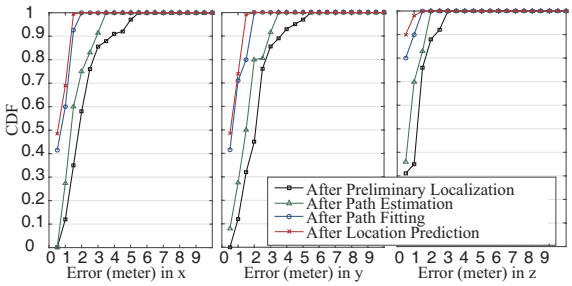


Fig. 12. CDF of location errors in three axis.

To better compare the performance of HiQuadLoc and normal RSS-based localization system, we also plot the cumulative distribution (CDF) of location errors in Fig. 11. As shown in the figure, the location error distribution for the final results of HiQuadLoc (red line) reaches its climax more quickly than the case of normal RSS-based localization system (black line). According to these results, in the field of high-speed quadrotor indoor localization, compared with traditional RSS-based systems, HiQuadLoc has reduced the location error by 62.8%, which is a remarkable improvement in accuracy. The localization errors incurred in each dimension (x, y and z asxis in the physical space) are illustrated in Fig. 12. Note that the location error is the square root of the quadratic sum of the error in each dimension.

6.3 Evaluation of Parameter Readjustment During Turning

We mentioned in Section 5.2 that we need to readjust the Kalman filter during the turning of the quadrotor. After the experiment in Section 6.2, we change the values of σ_u^2 and $\bar{\sigma}_u^2$ respectively

to confirm that the readjustment of Kalman filter can improve the accuracy of the path estimation. We analyze the output of the path estimation method $\bar{s}_k$ alone to rule out the additional gain of the path fitting and the location prediction method. Since the readjustment mainly happens during turning, we focus on the location error of the ± 5 localization results around each corner of the flight paths.

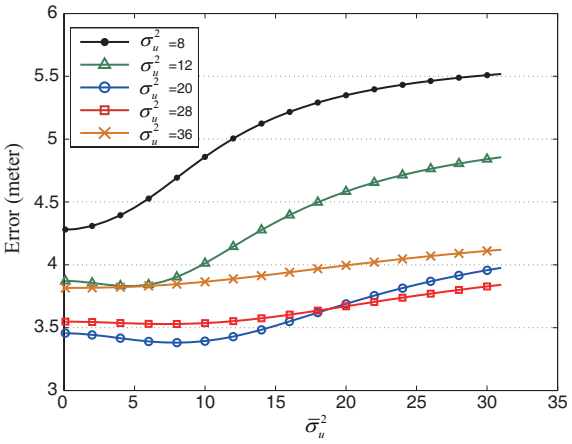


Fig. 13. Average location errors of $\bar{s}_k$ when σ_u^2 and $\bar{\sigma}_u^2$ vary.

In Fig. 13, it is shown that when σ_u^2 is fixed, there exists a $\bar{\sigma}_u^2$ that makes the location error minimum. For the case of $\sigma_u^2 = 8$ and $\sigma_u^2 = 36$, the location error is minimum when $\bar{\sigma}_u^2 = 0$, which means that the path generated by Kalman filter should be interrupted during the motion of turning, and a new path is restarted at the first $\check{s}_k$ after the turning. On the other hand, when $\sigma_u^2 = 12$, $\sigma_u^2 = 20$ and $\sigma_u^2 = 28$, the error is minimum when $\bar{\sigma}_u^2 = 5$, $\bar{\sigma}_u^2 = 8$ and $\bar{\sigma}_u^2 = 7$ respectively. Thus we cannot simply set $\bar{\sigma}_u^2 = 0$, because sometimes the historical localization results are still useful during turning. Note that the change of the lines under different σ_u^2 shows that a too large or too small σ_u^2 also leads to the increase of error. Thus we cannot set $\sigma_u^2 = \bar{\sigma}_u^2$ directly.

The experiment results verify that changing the parameter of Kalman filter is necessary at the corners of the flight paths, and the benefit of the proposed revised Kalman filter is validated. In our system, we set $\sigma_u^2 = 20$ and $\bar{\sigma}_u^2 = 8$ according to these results.

6.4 Evaluation of HiQuadLoc for Different Flight Speeds

Since the quadrotor could flight in different speeds, we also evaluate the impact of speed on the accuracy of localization in our system. To ensure that the quadrotor can fly at approximately constant speed, we limit the flight path in a long lobby in the region A in Fig. 8(a). The length of the flight path is longer than $80m$. We control the quadrotor to fly in the lobby for 15 times at each speed: $3m/s$, $2m/s$ and $1m/s$. All the other settings are same to those in the previous experiments. The highest speed we set here is $3m/s$, which is a reasonable speed for the quadrotor. As a benchmark, a vision-based, autonomous quadrotor equipped with two cameras, an IMU, and an 1.6GHz Intel Atom processor can fly straight-line at the speed of 4m/s and non-straight-line at about 1-2m/s [48]. We note that some fully-loaded quadrotor could fly $20m/s$ [49], which however is equipped with "high-definition onboard cameras, LIDAR, sonar, inertial measurement units, and other sensors", and our means of localization are basically RSS fingerprints matching and sensors. The purpose of the experiment is to verify the feasibility of the low-cost fingerprinting based approach for localizing the quadrotor in the indoor space, and we do not claim that the high-speed quadrotor indoor localization issue has been completely resolved.

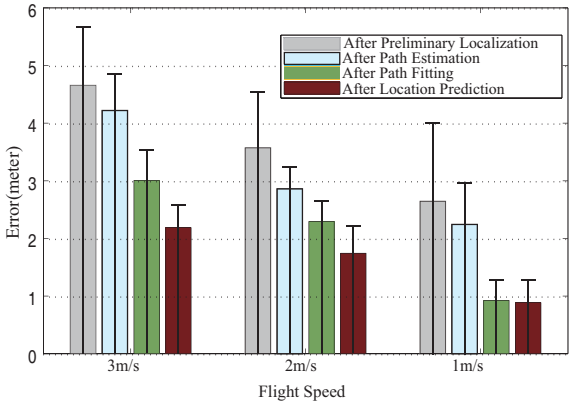


Fig. 14. Average location errors at different speeds.

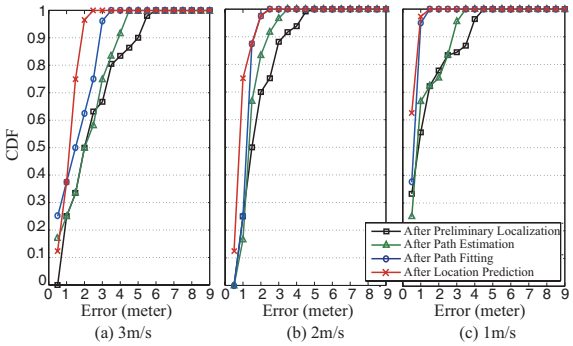


Fig. 15. CDF of location errors at different speeds.

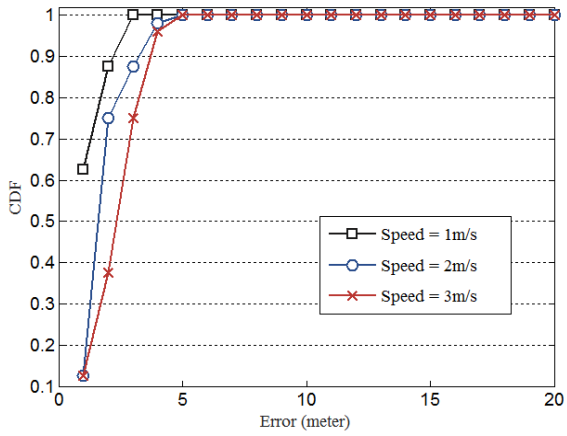


Fig. 16. CDF of location errors after location prediction at different speeds.

The results are shown in Fig. 14. It can be seen that the location error increases as the speed increases. This is caused by the fact that the number of times for RSS measurement is more limited when the quadrotor is moving at higher speed. However, HiQuadLoc still works well in this case. The average location

error is $2.19m$, which has been reduced by 53.0% compared with the original error $4.66m$ provided by regular RSS-based systems. For the case of a lower speed $2m/s$, that is $1.76m$, reduced by 51.0%. For $1m/s$, that is $0.89m$, reduced by 66.4%. Moreover, we can find that since the location error caused by communication delay is more serious in the higher-speed case, the contribution of the location prediction method in this case is much more obvious than that in the lower-speed case. Figure 15 illustrates the CDF of the location errors at different speeds. It can be seen that the path correction algorithm can effectively reduce the location error at all the speeds. Figure 16 presents how the localization accuracy decline with speed of the quadrotor after all the localization data processing.

6.5 Comparison with Channel State Information (CSI) based Scheme

We here compare the HiQuadLoc with channel state information (CSI) based mechanism. In particular, we choose to implement the angle-of-arrival (AoA) method [34], [32], and compare the localization performance with that of HiQuadLoc. The AoA method is chosen because it is the representative scheme utilizing CSI for localization, and it is also the basis of many localization systems such as ArrayTrack [34] and SpotFi [32]. We note that authors of ArraryTrack has built an AP-like hotspot system themselves; however, common deployment of such system is still unavailable to the best of our knowledge. The existing Wi-Fi APs in our experiment building have no interface for retrieving the CSI like most of the off-the-shelf APs.

We build CSI retrievable APs ourselves and the components of the AP are as shown in Fig. 17. The experiment setup is configured similar with that in SpotFi. In Fig. 17, the Intel 5300 network interface card (NIC) is for collecting the CSI data, Lenovo R400 is acting as the computing platform running the CSI data processing program [35]. Note that the NIC is equipped with 3 antennas, as each Intel 5300 NIC has only three slots for antennas. We built two such APs to cover an area of around $200m^2$ in the corridor as shown in Fig. 8(a), which has the same deployment density as in [32]. During the experiment, the target to be localized move in different speeds along a track in the shape of "L", and the APs record the corresponding CSI data, which are then used to derive the location information of the target.

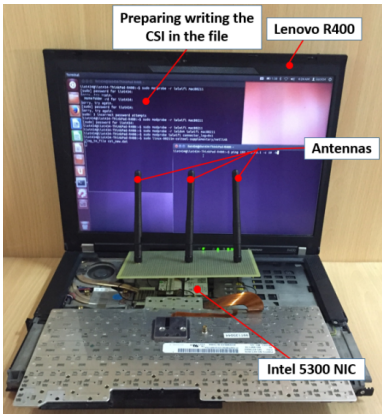


Fig. 17. CSI retrievable AP.

We compare the derived locations of the target and the real location of the target, and the CDF of location errors are illustrated in Fig. 18, where the performance of both HiQuadLoc and the CSI based scheme are shown. It can be seen that the performance of CSI based scheme is not as good as expected. This could be due to the fact that the moving quadrotor could incur fast changing multipath effects. The multipath effects have been considered in previous CSI based systems such as the ArrarTrack and SpotFi; however, those techniques are basically for the scenario of localizing mobile devices such as smartphones, where the holder of the smartphone has small scale movement. The quadrotor localization scenario is more challenging as the speed of the quadrotor is faster than the pedestrians; moreover, the flying part of the quadrotor could also incur multipath effect that changes very fast. This is corroborated by the fact that the low speed moving target could be more accurately localized as shown in Fig. 18.

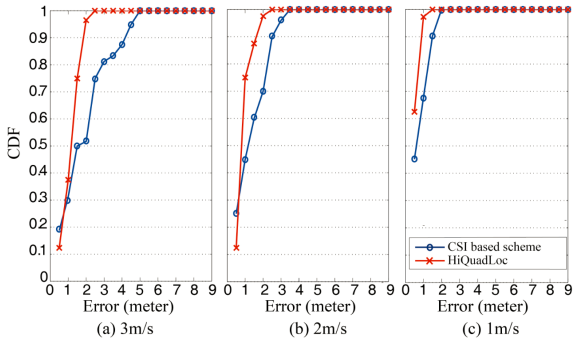


Fig. 18. Comparison with CSI based scheme: CDF of location errors at different speeds.

Another possible reason of the results is that the CSI based scheme is subtle but not that robust. We find that packet loss could be observed during the experiment, especially when the moving speed of the target is comparatively high and the distance between the target and the AP is far. As the underlying details of Intel 5300 NIC is not revealed, how the packet loss could impact the localization accuracy could not be evaluated. We note that the number of APs could also influence the performance of CSI based localization scheme as described in [32], which also hinders the wide deployment of CSI based localization systems.

In particular, most of APs do not provide the interface for retrieving CSI information. Intel 5300 NIC is a well-known hardware that could help retrieve CSI, which are adopted by a number of work in the literature. However, it is non-trivial to widely deploy APs created with Intel 5300 NIC and the PC in practice. For example, the integration of the NIC into the PC could incur software incompatibility issue. We collect 5 Lenovo R400 laptops that are officially announced supporting Intel 5300 NIC to create the AP [35], among which 3 of them have been found the incompatibility issue. It will be more challenging to widely deploy such systems in practice. In contrast to such systems, there is no need for the HiQuadLoc to retrieve the CSI information. The HiQuadLoc only requires reading RSS information that is naturally supported by all mobile devices, which is very convenient for wide deployment under the current infrastructure.

6.6 Evaluation of Energy Consumption

Since the battery capacity of quadrotors are always limited, it is necessary to confirm that our system does not reduce the maximum flight time of quadrotors obviously. The available battery capacity of our quadrotor is $2300mAh$. During a flight of $20min$,

	Basic Device	Direction Sensor	RSS Measurement
Energy Consumption	2089.16mAh	3.78mAh	91.98mAh
Percentage	95.62%	4.21%	0.17%

TABLE 1
Energy consumption during a flight.

we evaluate the energy consumed by the direction sensor and the RSS measurement. The results of the test are shown in Table 1. It can be found that compared with the basic energy consumption of the quadrotor during the flight, the energy consumed by the direction sensor and the function of RSS measurement is negligible. The results show that HiQuadLoc consumes only 4.38% of the battery capacity. Thus it is confirmed that our system will not impact the cruising ability of quadrotors obviously.

7 Related Work

Indoor Localization System for UAVs: Wireless techniques have been applied for localizing mobile devices in indoor spaces such as RFID technology [21], [22], UWB [23], [24], bluetooth [25], [26] and ultrasonic [27], [28]. Among the typical indoor localization technologies, WLAN positioning systems utilize RSS fingerprints, which do not need additional infrastructures except pre-existing WLAN APs. RADAR [15] is an in-building user localization system, adopt the method of kNN to achieve an accuracy of 2-$3m$. Yang *et al.* leverages user motions to construct the radio map of a floor [16], which only can be obtained by site survey previously. Laoudias *et al.* adopts Artificial Neural Networks (ANN) as a function approximation approach to match vectors of RSS fingerprints [17].

In recent years, research on UAV indoor localization has been increasingly popular, where the early systems are based on vision information [7], [8]. For example, Lee *et al.* propose a scale invariant feature transform algorithm based on image matching [10]. Kendoul *et al.* utilize an onboard camera and an Inertial Measurement Unit to develop a real-time 3D vision algorithm for estimating optic flow, aircraft self-motion and depth map [11]. Other solutions based on ultrasound [9] or infrared [29] are also proposed. However, additional infrastructures such as off-board sensors and cameras are needed in these works, which limits their application scenarios and hinders their wide deployment. Our system proposed in this paper is based on Wi-Fi RSS fingerprints, and Wi-Fi APs have been widely installed inside buildings.

Nonetheless, previous works based on RSS fingerprints cannot be applied to the case of quadrotor localization directly, because the high flight speed limits the chances of measuring RSS continuously at a position, which leads to large location errors. The adverse impact of communication delay is more obvious in the high-speed case. Our system applies the algorithm of path correction based on Kalman filter, turning detection, fitting and location prediction to resolve these problems.

Indoor Localization with Channel State Information (CSI): Vasisht *et al.* present a system termed as Chronos [31], which enables the quadrotor to follow the user with a controller around. Chronos utilizes a novel algorithm to compute the time-of-flight (ToF) of the signal to derive the distance between the quadrotor and the user's control device, where the key is to retrieve and utilize the CSI between the sender and the receiver. However, the high-speed motion of the quadrotor could incur the fast changing multipath effect, which could influence the accuracy of CSI. This may be the reason that the quadrotor in Chronos is moving very slowly. Whether the quadrotor under Chronos could fly around without the user's guidance as considered in this paper is not mentioned in [31].

Kotaru *et al.* propose SpotFi [32] system, which achieves decimeter level accuracy with Wi-Fi. With SpotFi, the localization server first analyzes CSI information including ToF and angle-of-arrival (AoA) collected from each AP, and then identifies the direct path profile between the target and each AP, which are then used for final location estimation. However, the APs used in SpotFi are factually computers equipped with Intel 5300 network interface card (NIC) [35] that is similar with the ones used in our experiment, which are not deployed in most if not all of the buildings.

Kumar *et al.* propose Ubicarse system [33] to utilize the principle of synthetic aperture radar (SAR) for localize mobile devices. With Ubicarse, users twist the held mobile device, and the device could derive the multi-path profile in the process, which is then utilized for location determination. Ubicarse can be hardly applied to high speed moving objects such as the quadrotor, as it is impossible to twist a quadrotor for localization; moreover, ubicarse requires the accurate location of each AP in the building, which may also require site survey in practice.

Xiong *et al.* develop an indoor location system ArrayTrack, which exploits multiple-input, multiple-output (MIMO) techniques to achieve an accuracy of tens of centimeters. The basic idea of ArrayTrack is to derive the AoA information of the user's frame with respect to multiple antennas. The AoA information is then aggregated and computed to estimate the user's location. While the components of ArraryTrack are off-the-shelf products, building the system requires great efforts. Although presenting outstanding performance, the ArrayTrack system is still not widely deployed as regular Wi-Fi APs.

Localization systems utilizing CSI as mentioned above normally present a higher level of accuracy compared with that utilizing fingerprinting methodology, as CSI is a finer-grained feature compared with RSS fingerprints. However, there are only limited types of hardware on the market supporting CSI retrieval [34], [35], and those those customized hardware are not widely deployed in existing buildings as regular Wi-Fi APs. The advantage of fingerprinting based localization is the convenient deployment, and our work in this paper shows that the fingerprinting method could also present reasonable accuracy that is feasible for quadrotor localization. Moreover, the high speed motion of the quadrotor could incur fast change of multipath effects profiles which makes the CSI information unreliable. We implement the CSI based scheme in Section 6. It turns out that the performance of CSI method is not as good as expected, which could be due to the limited number of antennas, customized APs and the high moving speed of the quadrotor.

Indoor Localization with Filter Techniques: Different filters have been applied to improve the performance of localization. For instance, both Kalman filter and Particle filter have been widely used for indoor tracking [30], [36], [37], [38], [39]. In our system, we apply Kalman filter to make use of historical localization results. For the case of UAV localization, Rullán-Lara *et al.* proposes an algorithm based on time difference of arrivals (TDOA) measurements of radio signal and an extended Kalman filter is leveraged to reduce the covariance error [40]. The systems in [41], [42] also apply an extended Kalman filter for enhancing the accuracy of the location estimation. In our system, we consider the motion pattern of quadrotors especially at turning corners, and

enhance the Kalman filter to adapt to the indoor localization for quadrotors. Eckert *et al.* propose a position prediction scheme to improve localization accuracy. In HiQuadLoc, quadrotors help to measure the communication delay and thus our path prediction algorithm can be more reliable.

RSS interpolation: To reduce the cost of the RSS fingerprints collection process, some systems apply the method of RSS interpolation, which completes the whole indoor fingerprint map according to the calibration grids with reduced density [43], [44], [45]. For instance, Dil *et al.* use linear Delaunay Triangulation interpolation in order to estimate the RSS over surface distribution of the calibration measurements [46]. Röhrig *et al.* leverage Delaunay Triangulation and interpolation to reduce the density of the calibration points in the radio map and thus minimize the manual effort to build the map [47].

However, most of the interpolation schemes in the literature are based on surface interpolation, which interpolate missing average RSS values. In HiQuadLoc, the preliminary localization algorithm is based on the probability distribution of RSS, thus surface interpolation cannot be applied in our system directly. To resolve this issue, we propose the algorithm of 4-D RSS interpolation. In constrast to the surface interpolation algorithm for 2-D indoor spaces, we consider the probability distribution of RSS in 4-D space: 3-D in indoor space and 1-D in RSS sample space. Moreover, what we estimate by interpolation is the probability for each RSS value to appear at each non-calibration cube, instead of the average values of RSS at non-calibration grids. To the best of our knowledge, this is the first work considering the RSS interpolation for probability-based localization algorithms.

8 Conclusion

In this paper, we have proposed HiQuadLoc, a RSS-based indoor localization system for quadrotors. While normal RSS-based localization systems cannot be applied directly to the quadrotors because the high speed flight reduces the amount of usable RSS data, our system leverages historical localization results as well as the motion states of the quadrotors to improve accuracy of localization. Mechanisms such as path estimation, path fitting and location prediction have been proposed to mitigate the negative impacts of the continuous motion on location estimation. The algorithm of 4-D RSS interpolation has been applied to relieve the overhead of collecting RSS training data. Experiment results have been demonstrated to show that HiQuadLoc can reduce the average location error by more than 50% compared with normal RSS-based systems, and the amount of RSS training data need to be collected can be reduced by more than 80%.

9 Acknowledgement

This work is supported by National Natural Science Foundation of China (No. 61572319, U1405251, 61532012, 61325012, 61271219, 61428205); National Mobile Communications Research Laboratory, Southeast University (No. 2014D07).

References

[1] Z. Liu, Y. Chen, B. Liu, C. Cao and X. Fu, "HAWK: An unmanned mini-helicopter-based aerial wireless kit for localization," *IEEE Trans. Mobile Comput.* , vol. 13, no. 2, pp.287–298, Feb. 2013.

[2] J. T. K. Ping, A. E. Ling, T. J. Quan and C. Y. Dat, "Generic unmanned aerial vehicle (UAV) for civilian application-A feasibility assessment and market survey on civilian application for aerial imaging," in *Proc. IEEE Sustainable Utilization and Development in Engineering and Technology (STUDENT),* pp. 289-294, Oct. 2012, October.

[3] R. Schneiderman, "Unmanned drones are flying high in the military/aerospace sector," *IEEE Signal Processing Magazine,* vol. 29, no. 1, pp. 8-11, 2012.

[4] https://www.amazon.com/b?node=8037720011

[5] http://www.asctec.de/en/droneservice-uas-indoor-surveillance/

[6] http://www.wired.com/2015/02/drone-bounces-around-disaster-sites-like-beach-ball/

[7] Y. M. Mustafah, A. W. Azman and F. Akbar, "Indoor UAV Positioning Using Stereo Vision Sensor," *Procedia Engineering,* vol. 41, pp. 575-579, 2012.

[8] F. Wang, T. Wang, B. M. Chen and T. H. Lee, "An indoor unmanned coaxial rotorcraft system with vision positioning," in *Proc. IEEE International Conference on Control and Automation (ICCA)*, pp. 291-296, Jun. 2010.

[9] J. Eckert, R. German and F. Dressler, "On autonomous indoor flights: High-quality real-time localization using low-cost sensors," in *IEEE ICC,* pp. 7093-7098, Jun. 2012.

[10] J. O. Lee, T. Kang, K. H. Lee, S. K. Im and J. Park, "Vision-based indoor localization for unmanned aerial vehicles," *Journal of Aerospace Engineering,* vol. 24, no. 3, pp. 373-377, 2010.

[11] F. Kendoul, I. Fantoni and K. Nonami, "Optic flow-based vision system for autonomous 3D localization and control of small aerial vehicles," *Robotics and Autonomous Systems,* vol. 57, no. 6, pp. 591-602, 2009.

[12] H. Liu, Y. Gan, J. Yang, S. Sidhom, Y. Wang, Y. Chen and F. Ye, "Push the Limit of WiFi based Localization for Smartphones," in *Proc. ACM MobiCom*, 2012, pp. 305–316.

[13] H. Liu, J. Yang, S. Sidhom, Y. Wang, Y. Chen and F. Ye, " Accurate WiFi Based Localization for Smartphones Using Peer Assistance," *IEEE Trans. Mobile Comput.* , vol. 13, no. 10, pp.2199–2214, Oct. 2013.

[14] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan and R. Sen, "Zee: zero-effort crowdsourcing for indoor localization," in *Proc. ACM MobiCom*, 2012, pp. 293–304.
tracking system," Proceedings of IEEE International Conference on Computer Communications, vol.2, pp.775-784, Mar. 2000.

[15] P. Bahl and V.N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *Proc. IEEE INFOCOM*, vol.2, pp.775-784, Mar. 2000.
Localization with Little Human Intervention," Proceedings of the 18th annual international conference on Mobile computing and networking, pp.269-280, Aug.22-26, 2012.

[16] Z. Yang, C. Wu and Y. Liu, "Locating in Fingerprint Space: Wireless Indoor Localization with Little Human Intervention," in *Proc. ACM Mobicom*, pp.269-280, Aug.22-26, 2012.
"Indoor Localization Using Neural Networks with Location Fingerprints," Lecture Notes in Computer Science, vol.5769, pp.954-963, 2009.

[17] C. Laoudias, D.G. Eliades, P. Kemppi, C.G. Panayiotou and M.M. Polycarpou, "Indoor Localization Using Neural Networks with Location Fingerprints," *LNCS,* vol.5769, pp.954-963, 2009.

[18] D. Roberts, "Construction and Testing of a Quadcopter," 2013. http://digitalcommons.calpoly.edu/.

[19] K. Chintalapudi, A. Padmanabha Iyer and V. N. Padmanabhan, "Indoor localization without the pain," in *Proc. ACM MobiCom*, 2010, pp. 173–184.

[20] C. Wu, Z. Yang and Y. Liu, " Smartphones based Crowdsourcing for Indoor Localization," *IEEE Trans. on Mobile Comput.* , vol. 13, no. 10, pp.2199–2214, Oct. 2013.

[21] G. Jin, X. Lu and M.-S. Park, "An indoor localization mechanism using active RFID tag," in *Proc. IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing,* vol.1, pp.4, 5-7 Jun. 2006.

[22] C. Hekimian-Williams, B. Grant, X. Liu, Z. Zhang and P. Kumar, "Accurate localization of RFID tags using phase difference," *IEEE International Conference on RFID*, pp.14-16, Apr. 2010.

[23] H. Wymeersch, S. Marano, W.M. Gifford and M.Z. Win, "A Machine Learning Approach to Ranging Error Mitigation for UWB Localization," *IEEE Trans. Commun.* vol.60, no.6, pp.1719-1728, Jun. 2012.

[24] S. Krishnan, P. Sharma, G. Zhang and O. H. Woon, "A UWB based Localization System for Indoor Robot Navigation," in *Proc. IEEE International Conference on Ultra-Wideband*, pp.77-82, Sept. 24-26, 2007.

[25] M. Altini, D. Brunelli, E. Farella and L. Benini, "Bluetooth indoor localization with multiple neural networks," *IEEE International Symposium on Wireless Pervasive Computing (ISWPC)*, pp.295-300, May 5-7, 2010.

[26] S.S. Chawathe, "Low-latency indoor localization using bluetooth beacons," in *Proc. IEEE Conference on Inteliigent Transportation Systems (ITSC)*, pp.1-7, Oct. 4-7, 2009.

[27] E. Steen, M. Eichelberg, W. Nebel and A. Hein, "A Novel Indoor Localization Approach Using Dynamic Changes in Ultrasonic Echoes," *Part of the series Advanced Technologies and Societal Change*, pp.123-133, 2012.

[28] M. Scherhaufl, R. Pfeil, M. Pichler and A. Berger, "A novel ultrasonic indoor localization system with simultaneous estimation of position and velocity," in *Proc. IEEE Topical Conference on Wireless Sensors and Sensor Networks (WiSNet),* pp.21-24, Jan. 15-18, 2012.

[29] J. F. Roberts, T. Stirling, J. C. Zufferey and D. Floreano, "3-D relative positioning sensor for indoor flying robots," *Autonomous Robots,* vol.33, no. 1, pp.5-20, 2012.

[30] M. Nabaee, A. Pooyafard and A. Olfat, "Enhanced object tracking with received signal strength using Kalman filter in sensor networks,"in *Proc. IEEE International Symposium on Telecommunications (ISTEL),* pp. 318-323, Aug. 2008.

[31] D. Vasisht, S. Kumar, D. and Katabi, "Decimeter-level localization with a signle WiFi access point," in *Proc. USENIX NSDI,* 2016, pp. 165–178.

[32] M. Kotaru, K. Josi, D. Bharadia and S. Katti, "SpotFi: Decimeter Level Localization using WiFi," in *Proc. ACM SIGCOMM,* 2015, pp. 269–282

[33] S. Kumar, S. Gil, D. Katabi and D. Rus, "Accurate indoor localization with zero start-up cost," in *Proc. ACM MobiCom,* 2014, pp. 483–494.

[34] J. Xiong and K. Jamieson, "ArrayTrack: A fine-grained indoor location system," in *Proc. USENIX NSDI,* 2013, pp. 71–84

[35] D. Halperin, W. Hu, A. Sheth and D. Wetherall, "Tool release: Gathering 802.11n traces with channel state information," *ACM SIGCOMM CCR*, Jan. 2011.

[36] S. Outemzabet and C. Nerguizian, "Accuracy enhancement of an indoor ANN-based fingerprinting location system using Kalman filtering," in *Proc. IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC),* pp. 1-5, Sept. 2008.

[37] F. Evennou, F. Marx and E. Novakov, "Map-aided indoor mobile positioning system using particle filter," in *IEEE WCNC,* vol. 4, pp. 2490-2494, Mar. 2005.

[38] H. Wang, H. Lenz, A. Szabo, J. Bamberger and U. D. Hanebeck, "WLAN-based pedestrian tracking using particle filters and low-cost MEMS sensors," in *IEEE WPNC,* pp. 1-7, Mar. 2007.

[39] A. S. Paul and E. A. Wan, "RSSI-based indoor localization and tracking using sigma-point kalman smoothers," in *IEEE JSTSP,* vol. 3(5), pp. 860-873, 2009.

[40] J. L. Rullán-Lara, S. Salazar and R. Lozano, "Real-time localization of an UAV using Kalman filter and a Wireless Sensor Network," *Journal of Intelligent and Robotic Systems,* vol. 65(1-4), pp. 283-293, 2012.

[41] A. Benini, A. Mancini and S. Longhi, "An IMU/UWB/Vision-based Extended Kalman Filter for Mini-UAV Localization in Indoor Environment using 802.15. 4a Wireless Sensor Network," *Journal of Intelligent and Robotic Systems,* vol. 70(1-4), pp. 461-476, 2013.

[42] J. Eckert, F. Dressler and R. German, "Real-time indoor localization support for four-rotor flying robots using sensor nodes," in *Proc. IEEE Workshop on Robotic and Sensors Environments,* pp. 23-28, Nov. 2009.

[43] A. W. Tsui, Y. H. Chuang and H. H. Chu, "Unsupervised learning for solving RSS hardware variance problem in WiFi localization," *Mobile Networks and Applications,* vol. 14(5), pp. 677-691, 2009.

[44] P. Krishnan, A. S. Krishnakumar, W. H. Ju, C. Mallows and S. N. Gamt, "A system for LEASE: Location estimation assisted by stationary emitters for indoor RF wireless networks," in *Proc. IEEE INFCOM,* vol. 2, pp. 1001-1011, Mar. 2004.

[45] A. M. Hossain, H. N. Van, Y. Jin and W. S. Soh, "Indoor localization using multiple wireless technologies," in *IEEE MASS,* pp. 1-8, Oct. 2007.

[46] B. J. Dil and P.J.M. Havinga, "On the calibration and performance of rss-based localization methods," in *Proc. Internet of Things (IOT),* pp. 1-8, Nov. 2010.

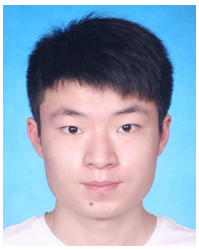
[47] C. Röhrig and F. Kunemund, "Estimation of position and orientation of mobile systems in a wireless LAN," in *Proc. IEEE Conference on Decison and Control,* Dec. 2007.

[48] Vijay Kumar Lab demonstration, http://www.kumarrobotics.org/videos/.

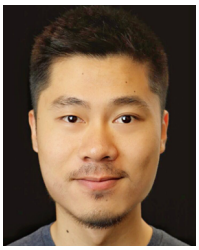
[49] NEW ATLAS, http://newatlas.com/darpa-drone-autonomous-45-mph/41810/.

Xiaohua Tian (S'07-M'11) his B.E. and M.E. degrees in communication engineering from Northwestern Polytechnical University, Xian, China, in 2003 and 2006, respectively. He received the Ph.D. degree in the Department of Electrical and Computer Engineering (ECE), Illinois Institute of Technology (IIT), Chicago, in Dec. 2010. Since Mar. 2011, he has been with the School of Electronic Information and Electrical Engineering in Shanghai Jiao Tong University, and now is an Associate Professor with the title of SMC-B scholar. He serves as an editorial board member on the computer science subsection of the journal SpringerPlus, and the guest editor of International Journal of Sensor Networks. He also serves as the TPC member for IEEE INFOCOM 2014-2017, best demo/poster award committee member of IEEE INFOCOM 2014, TPC co-chair for IEEE ICCC 2014-2016, TPC Co-chair for the 9th International Conference on Wireless Algorithms, Systems and Applications (WASA 2014), TPC member for IEEE GLOBECOM 2011-2016, TPC member for IEEE ICC 2013-2016, respectively.

Zhenyu Song is currently pursuing the B.E. degree in Computer Science from Shanghai Jiao Tong University, China and is expected to graduate in 2017. His research interests include wireless localization and big data.



Binyao Jiang is currently completing the B.E. degree in Computer Science from Shanghai Jiao Tong University, China and is expected to graduate in 2019.His recent research work is indoor localization.



Yang Zhang received his B.E. degree in Computer Science from Shanghai Jiao Tong University, China. He is currently pursuing his M.S. degree in Computational Data Science at Carnegie Mellon University, USA and is expected to graduate in 2016. His research interests include big data in cloud computing and human-centered data science.

Tuo Yu received the B.E. and M.E. degree in Department of Electronic Engineering from Shanghai Jiao Tong University, Shanghai, China in 2012 and 2015, respectively. He is currently pursuing his Ph.D. degree at Department of Computer Science in University of Illinois at Urbana-Champaign. His research interest focuses on wireless communications and mobile sensing.

Xinbing Wang received the B.S. degree (with hons.) from the Department of Automation, Shanghai Jiaotong University, Shanghai, China, in 1998, and the M.S. degree from the Department of Computer Science and Technology, Tsinghua University, Beijing, China, in 2001. He received the Ph.D. degree, major in the Department of electrical and Computer Engineering, minor in the Department of Mathematics, North Carolina State University, Raleigh, in 2006. Currently, he is a professor in the Department of Electronic Engineering, Shanghai Jiaotong University, Shanghai, China. Dr. Wang has been an associate editor for IEEE/ACM Transactions on Networking and IEEE Transactions on Mobile Computing, and the member of the Technical Program Committees of several conferences including ACM MobiCom 2012, ACM MobiHoc 2012-2014, IEEE INFOCOM 2009-2017.