WiFi Based Indoor Localization with Adaptive Motion Model using Smartphone Motion Sensors

Xiang He, Jia Li, Daniel Aloi
Electrical and Computer Engineering
Oakland University, OU
Rochester, MI 48309, U.S.A
xhe2@oakland.edu, li4@oakland.edu, aloi@oakland.edu

***Abstract*—We present an adaptive motion model for tracking the movement of smartphone user by using the motion sensors (accelerometer, gyroscope and magnetometer) embedded in the smartphone. A particle filter based estimator is used to seamlessly fuse the adaptive motion model with a WiFi based indoor localization system. The system applies Gaussian process regression to train the collected WiFi received signal strength (RSS) dataset, and particle filter for the estimation of the smartphone user's location and movement. Simulations were conducted in MATLAB to provide more insights of the proposed approach. The experiments carried out with an iOS device in typical library environment illustrate that our system is an accurate, real-time, highly integrated system.**

***Keywords*—WiFi RSS, indoor localization, Gaussian process regression, particle filter, smartphone, motion sensors, adaptive motion model**

I. INTRODUCTION

Indoor localization is a key component to many location based services, such as patient monitoring in hospitals, tour guiding in museums or spot finding in parking lots. As Global Positioning System (GPS) lacks the ability to function indoor, people are looking for other solutions to solve the indoor localization problem. In recent years, WiFi received signal strength (RSS) based location fingerprinting technique are attracting more and more interest for indoor localization, as they can provide good accuracy with no modification to the infrastructure. But WiFi RSS has the problem of signal fluctuation due to the fact of multipath fading in indoor environment. To meet the uncertainty of signal fluctuation, many probabilistic methods have been proposed. The state-of-the-art is the Bayesian filtering technique, implemented as particle filter.

Particle filter consists of two components: a measurement likelihood model and a motion model. In [1], Ferris et al propose an algorithm to construct the measurement likelihood model using Gaussian process regression during the offline survey phase. In the online localization phase, they apply particle filter to determine the target location. In order to not only locate the target in one spot, but also track the target movement, they develop a motion model to update the particles' coordinates on the map. Limited by using only WiFi signal, they adopt a naive conditional probabilistic motion model, which does not unleash the total power of the particle filter.

As the smartphones are equipped with more and more different sensors, the future of localization and tracking system will most likely evolve towards systems that are able to fuse the information provided by multiple sensors in a mobile device. However, most of the existing localization systems nowadays either lack the ability to process the multisensory integration problem or rely on data collected from separated MEMS sensors and WiFi signals sniffed from a mobile handset [2]. The localization is performed by post-processing the data from different sensors on a laptop instead of real-time processing on a smartphone.

In our previous work [3], we adapt the Gaussian process modeling of WiFi RSS and particle filter based localizer to smartphone platform and develop an iOS app called WiFi iLocate. In this paper, we investigate the motion sensors embedded in the smartphone. Currently, smartphones are equipped with various low cost motion sensors, such as accelerometer, gyroscope and magnetometer. The localization technique based on them is called pedestrian dead reckoning (PDR). However, due to the drifting nature of the motion sensors, PDR could only achieve limited accuracy and the localization error will accumulate in the long run. To overcome the problem, Li et al [4] applied particle filter to fuse PDR with indoor map information: In each particle propagation step, the algorithm checks whether the particles ended into obstacles or cross the walls. If they did, their weights are set to zero and get eliminated in the resampling step. This approach greatly improves the PDR based indoor localization performance but requires extra infrastructure information and detailed modeling of the indoor map.

Inspired by the previous study trying to combine the WiFi based indoor localization with PDR [5] [6]. We developed an adaptive motion model by fusing the data provided by motion sensors embedded in smartphone. This motion model is incorporated with a Gaussian process regression based WiFi RSS model into the particle filter framework.

Specifically, we make the following contributions: By carefully examine the information provided by various sensors, we are able to fuse the data effectively, and process the multisensory integration problem in real time. To the best of our knowledge, our upgraded iOS app WiFi iLocate is the first one to achieve accurate, highly integrated indoor localization by seamlessly leveraging information from WiFi and motion sensors on smartphone.

The paper continues as follows: In Section II, we first give a brief overview of our previous work. Then we describe the adaptive motion model and how it can improve the particle filter based location estimator. In Section III, simulation is conducted in MATLAB to provide more insights of the algorithm. The upgraded iOS application and field tests are presented in Section IV. Finally, we give a conclusion and future plan in Section V.

II. SYSTEM SETUP

The workflow of WiFi iLocate is presented in Figure 1. During the offline training phase, we first import the floor plan into the system. Then we can set the survey points on the screen and scan WiFi around the indoor environment. The scanned RSS values and corresponding BSSIDs are stored in a training dataset and preprocessed through Gaussian process regression modeling. The detailed procedure of Gaussian process regression can be found in [1] [3].

After the offline training phase, we are ready to perform the online localization and tracking by pressing the "Locate" button on the screen. When the localization task initiates, the smartphone starts scanning the WiFi RSS. Particle filter is initialized through KNN method, which searches for K closest matches of known locations in signal space from the offline-built dataset. The initial estimated location is acquired by averaging these K location candidates. It served as the starting point for all the particles. Next, the particles start to move randomly, as we apply a random walk motion model for particle propagation. When the particles are in motion, Gaussian process model is used to continually update the weight of each particle. After the particle weights get updated, we perform resampling to update the particle locations. The weight of each particle is treated as a probability where this particular particle is chosen to be the estimated location. In such a way, those particles with higher weights are picked more frequently than others, resulting in the elimination of wrongly moved particles and correctly tracking of the smartphone location. Figure 2 is the snapshots of our WiFi iLocate system [3]. The left-hand side shows the survey points during offline training phase, and the right-hand side shows the online localization result.

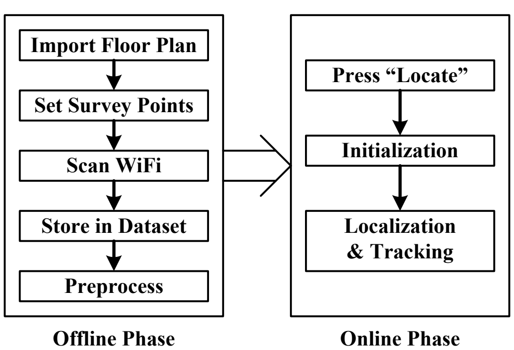

Figure 1: WiFi iLocate workflow [3]

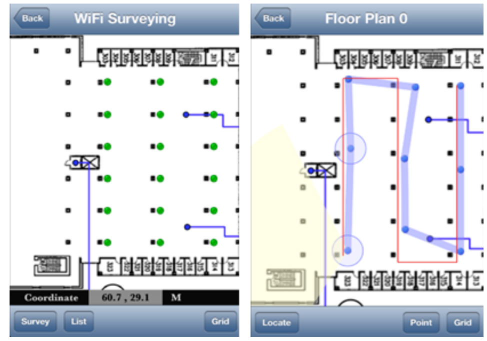

Figure 2: Snapshots of WiFi iLocate app [3]

Since we apply a simple random walk model to control the particle movement, we need a large amount of particle to reach sufficient accuracy, as the location estimation fully relies on the resampling step during the online localization [3]. We are hoping that some of the particles will randomly hit the right spot and others will be eliminated. If we have enough particles, it is working for most of the time. However, there're cases when particles can be fooled to a wrong place. We will demonstrate this in Section III.

In order to fully solve the problem, we develop an adaptive motion model to better control the particle movement.

A. Adaptive motion model

A motion model enables the prediction of the smartphone user's movement based on his current location. It is usually represented by the conditional probability $p(x_{t+1} \mid x_t)$, where x_t is current state of the particle and x_{t+1} is the next state. In order to track the user in real time, it is important to develop a motion model providing accurate estimation of his actual movement. We present three steps adaptive motion models with the help of motion sensors embedded in the smartphone.

(1) Stage transition detection

We define two stages for particles, standing and walking. Initially, all particles stay in the standing stage. We perform stage transition detection based on data collected from accelerometer.

When the user is standing still, it is expected that their mobile device will register little acceleration. Therefore, the standard deviation in the magnitude of acceleration is selected to detect the stage transition. It tells whether the particle to stay or move. If $\sigma_{\|a\|} < 0.01$, it means that the user is standing still in a location, thus the particle's movement will be limited in a small circle with center in the previous location. If $\sigma_{\|a\|} \geq 0.01$, however, it is not sufficient to ascertain that the user is walking. For example, user hands' sudden movement could result in a larger acceleration. Thus, we exploit the repetitive nature of walking.

Figure 3 shows the acceleration data recorded by a smartphone carrying by a walking user. We can see that the acceleration data exhibits a highly repetitive pattern. This pattern arises due to the fact of rhythmic nature of walking. In order to determine whether the user actually enter the walking state, we calculate the auto-correlation of the acceleration signal $a(n)$ for lag τ at the m^{th} as follows [7]:

$$\aleph(m,\tau) = \frac{\sum_{k=0}^{k=\tau-1}\left[(a(m+k) - \propto(m,\tau)) * (a(m+k+\tau) - \propto(m+\tau,\tau))\right]}{\tau * \sigma(m,\tau) * \sigma(m+\tau,\tau)} \quad (1)$$

where $\propto(k,\tau)$ and $\sigma(k,\tau)$ are the mean and standard deviation of the sequence of samples from $a(k)$ to $a(k+\tau-1)$.

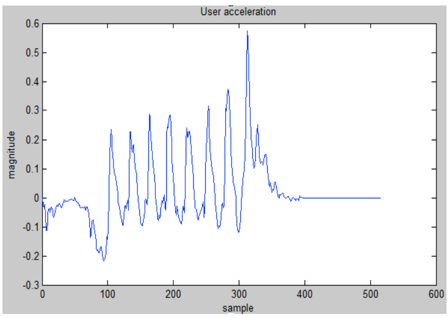

Figure 3: User acceleration during walking

If the user is walking then the auto-correlation will spike the periodicity of the walker. We define $\psi(m)$ as the maximum of the auto-correlation between $\tau_{\min} = 30$ and $\tau_{\max} = 60$. If both $\sigma_{\|a\|} \geq 0.01$ and $\psi(m) \geq 0.8$ are satisfied, then we set the state equals walking.

(2) Step counting and stride estimation

Once we have determined that state is walking, step counting and stride estimation are performed to calculate the walking distance of the user.

As shown in Figure 4, the step counting is realized by dividing the duration of sample when the maximum auto-correlation $\psi(m) \geq 0.8$ by τ_{opt}, and round up to an integer value. The τ_{opt} is determined by simply finding the most frequently occurred τ in the duration when $\psi(m) \geq 0.8$.

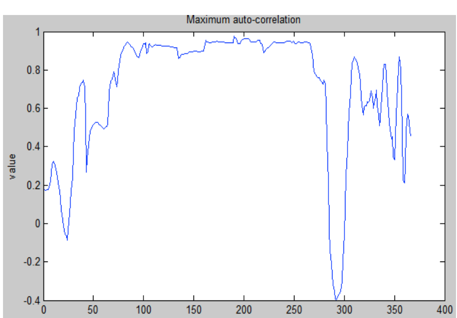

Figure 4: Maximum autocorrelation for step counting

Because human stride is not constant during walking, the stride size is determined by dynamically checking the acceleration sequence. We apply an empirical equation based on [8] to estimate the stride size.

$$Stride = 0.98 * \sqrt[3]{\frac{\sum_{k=1}^{N} |a_k|}{N}} \quad (2)$$

where a_k means the measured acceleration and N represents the number of sample in one period of walking.

(3) Heading detection

The mobile phone's magnetometer provides heading orientation of the phone relative to the magnetic north. There are many researches about how to induce the user walking direction from the magnetometer reading with phone placing on different parts of the human body [4] [5] [6] [7]. For example, the phone may be placed in pants, bounded on arms or hold in hands. In our case, we use the phone to track our location in an indoor environment, with our current location and walking path display on the screen. Thus, we can simply assume that the phone will only be hold in hand and in portrait direction. This is a valid assumption for navigation application on smartphone and it reduces the complicated orientation induction problem to simple heading detection.

The iOS navigation API provides device orientation information based on magnetometer reading. However, this information is easy to be disturbed indoor. Therefore, we only use it as the initial heading. Further heading direction is calculated from device yaw attitude, which acquired from gyroscope. Figure 5 shows the device yaw attitude changing when user is walking.

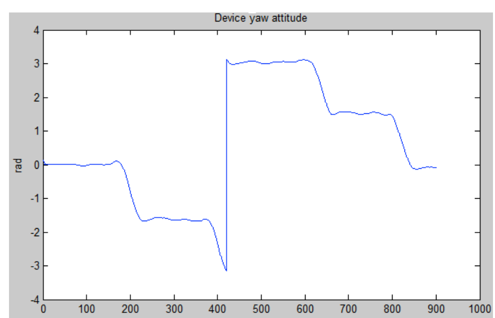

Figure 5: Device yaw attitude changing during walking

B. *The hidden Markov model*

The incorporation of adaptive motion model with Gaussian process modeling of WiFi RSS is implemented through particle filter to provide recursive location estimation over time. Particle filter is one kind of Bayesian filtering technique, which follows the Bayesian decision rule. The location estimation is based on posterior probability

$$p(x_t \mid z) \propto p(z \mid x_t) p(x_t \mid x_{t-1}) \quad (3)$$

Here $p(z \mid x_t)$ and $p(x_t \mid x_{t-1})$ represent a measurement likelihood model and a motion model, respectively.

The measurement likelihood model can be calculated using the posterior distribution of the signal strength at each location determined by the Gaussian process [3].

The motion model is based on a hidden Markov model (HMM). The state represents the location under estimation, it is not directly visible, but the state is depended on the observation output, which is the WiFi RSS in our case. Figure 6 illustrates the general structure of an HMM. We can clearly see that the conditional probability of the hidden state $X(t)$ at time t depends only on $X(t-1)$ at previous epoch, and the observation $O(t)$ only depends on $X(t)$ of the same epoch [9].

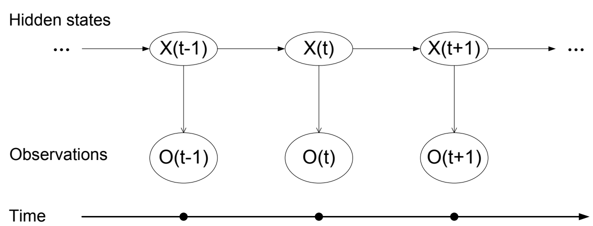

Figure 6: HMM general structure

The initial state $X(0)$ is determined through KNN method, as discussed before. Next, particle filter is applied to solve the hidden state problem. Each particle's coordinates represent a possible state. The particles propagate to a new location at the next epoch ($t=1$). From the observation $O(1)$, we are able to remove those wrongly moved particles by resampling. The state $X(1)$ is calculated as the mean of all the resampled particles' location. This state transition process is iterated for $t = 2, 3, 4 \ldots$ until the localization stop.

During the state transition process, we observe that the particle propagation plays an important role, which represents the state transition probability. The more accurate the particles propagate towards the right location, the better the localization performance will be. In our previous work, we apply a simple random walk model, which cannot guarantees the particles propagate towards the right location. In this paper, the motion sensors embedded in the smartphone are used to provide motion information about the user.

Based on the motion information, an adaptive motion model is developed to control the particle propagation. If the particles stay in standing stage, they follow a random walk model, which assigns a circle with center at the particle's current location and radius of 1 meter for the particle to move randomly within it. This may result in a disturbance when the user is standing still but the location display on the screen is moving slightly. But in general, the disturbance is small if the localization is correct. On the other hand, this freedom of movement gives particle filter the ability to correct the error during localization. Thus it greatly increases the system robustness. Once the stage is detected as walking, the system calculates the approximated heading direction and the walking distance within one particle filter epoch. Then we use the information to guide the particle's movement. All particles update their coordinates according to [4]

$$x(t+1) = x(t) + ((l(t) + \delta l(t)) * \cos(\theta(t) + \delta\theta(t)) \quad (4)$$

$$y(t+1) = y(t) + ((l(t) + \delta l(t)) * \sin(\theta(t) + \delta\theta(t)) \quad (5)$$

where $l(t)$ and $\theta(t)$ are the estimated step length and heading direction, while $\delta l(t)$ and $\delta\theta(t)$ are the zero mean Gaussian noise on the length and direction, respectively.

Figure 7 demonstrates the particle propagation from current epoch to next epoch and the particles are located within the shadow region. As shown in Figure 7, the left-hand side is related to the case when adaptive motion model is applied. The current state is at the center of the circle, the moving distance and heading direction are estimated by the adaptive motion model. The shadow region is considered as the state candidate of the next epoch, particles are guided to propagate to this region. In our previous work, we apply a random walk model for particle propagation. As we can see from the right-hand side of Figure 7, the shadow region covers the entire circle. Therefore, by introducing an adaptive motion model, we are able to accurately target the possible region for next state before doing particle resampling. It provides us two benefits: First, it greatly reduces the amount of particle needed for precise localization, thus decrease the computational complexity. Second, it prevents the cases when particles are fooled to a wrong place during resampling, making the system more robust against large errors.

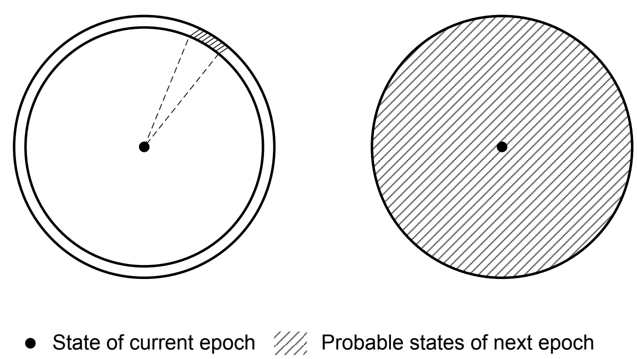

Figure 7: Particle propagation in state transition

III. SIMULATION AND RESULT ANALYSIS

In order to illustrate the insight of how the adaptive motion model can improve the localization performance and prevent large error occurs in particle resampling. We have setup a simulation environment in MATLAB.

The simulation setup is the same as in [3]. Wireless InSite, an EM solver by REMCOM, has been used to simulate a 40m by 40 m empty room with 4 APs, as shown in Figure 8.

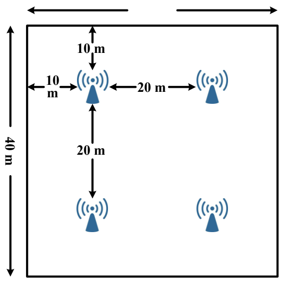

Figure 8: Simulated indoor environment [3]

The simulated WiFi RSS data are import into MATLAB, where we perform offline training and online localization and tracking. Figure 9 shows a localization result. The true path starts from coordinate (4.5, 4.5) and ends at coordinate (25.5, 10.5). It is shown on the left-hand side. The estimated path is shown on the right-hand side, where the blue dot and the red star represent the particle distribution and the estimated location [3].

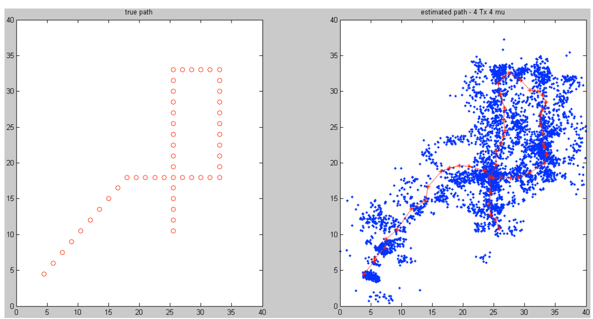

Figure 9: Particle filter based location estimation

As we can see from Figure 9, the particle filter is able to keep track of the true path. Here we use all 4 APs to help us locate the target. However, if we reduce the number of access point, the localization performance drops significantly, as shown in Table 1 [3].

Number of AP	1	2	3	4
Average error	14.8 m	7.4 m	6.5 m	3.8 m

Table 1: The localization error for different number of AP [3]

The root of this performance deterioration lies in the fact of WiFi RSS fluctuation. If we have sparse deployment of AP, it's highly possible that two faraway locations happen to share similar WiFi RSS. Figure 10 illustrates this situation in the simulation. As we can see from the figure, there are two places with much higher weight than other places, if particle reaches either places, they will survive the resampling process, as shown in Figure 11. Thus, large error occurs when we calculate the estimated location as the mean of particle distribution. Liu et al [10] also demonstrate this root cause of large error for WiFi based localization in real indoor environment.

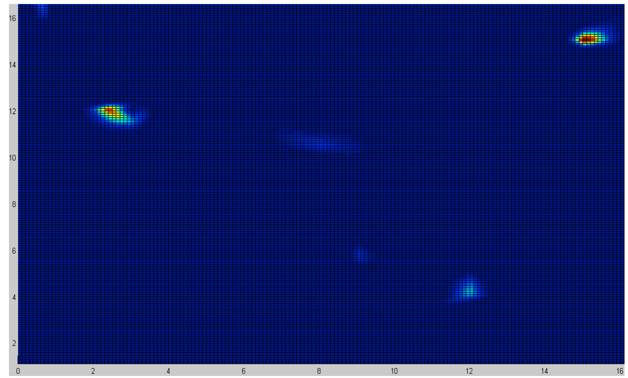

Figure 10: The situation when large error occurs

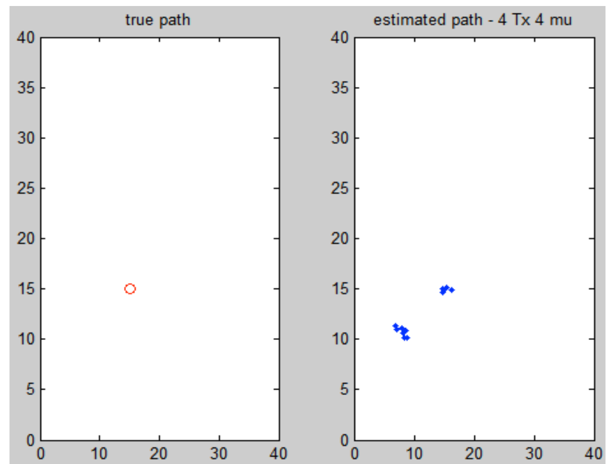

Figure 11: Particle distribution after resampling

By introducing an adaptive motion model, we are able to control the propagation of particles. Figure 12 shows the localization comparison in each time step between no motion model and with motion model in sparse AP situation. We can see that the motion information successfully helps solving the root cause of large error [3].

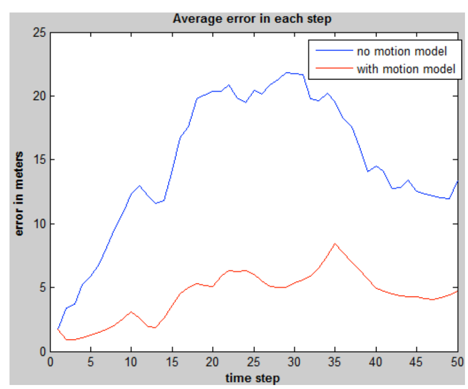

Figure 12: Localization error comparison, 1 AP situation [3]

IV. IMPLEMENTATION ON IOS DEVICE

We develop the adaptive motion model on our previous system WiFi iLocate. The updated system is tested and compared with the previous version in Oakland University library.

A. *System update*

The adaptive motion model is implemented within the online localization and tracking part in the system workflow shown in Figure 1. By introducing the iOS Motion API, we can record device motion data from accelerometer, gyroscope and magnetometer embedded in the smartphone in real time. These raw data related to the user acceleration, device attitude and heading direction are being logged simultaneously during the scanning of WiFi RSS. In each localization epoch, the motion information is processed by the adaptive motion model, and controlled the particle propagation.

B. *Experimental evaluation*

Figure 13 compares the estimated paths with the ground truth path. The dark red line represents the ground truth path, while the blue dot and the orange stroke illustrate the estimated path.

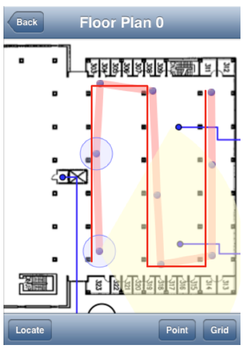

Figure 13: Comparison of the estimated path against the ground truth

Table 2 compares the localization performance between the pure PDR, previous WiFi iLocate system, and the updated one with Adaptive motion model (AMM).

Error	Mean	Median	Maximum
Pure PDR	9.5 m	8.8 m	>15 m
WiFi iLocate	3.6 m	2.9 m	5 m
WiFi iLocate with AMM	2.0 m	2.1 m	3.1 m

Table 2: Localization accuracy comparison

By seamlessly fusing the motion information into the system, we not only improve the localization performance, but also reduce the computational complexity, thus the system response time has decrease, resulting in a more sensitive, real time localization and tracking system.

V. CONCLUSION & FUTURE PLAN

In this paper, we have demonstrated an indoor localization and tracking system that is capable of integrating WiFi RSS and motion sensor information on smartphone. By introducing an adaptive motion model into the particle filter framework, we seamlessly fuse the motion information with the WiFi based localization technique. To the best of our knowledge, our updated WiFi iLocate is the first app delivering such accurate, highly integrated indoor localization system on smartphone platform.

As Google has announced in Project Tango [11], they have developed a prototype phone with powerful vision and 3D sensors. We believe the key technology for future localization lies in how to effectively fuse the information provided by various sensors. In the next step, we are looking to combine LiDAR and camera sensors into WiFi iLocate and fuse our system into vehicular information platform. So that smartphone and smart vehicle can better interact with each other.

REFERENCES

[1] B. Ferris, D. Hahnel, and D. Fox. "Gaussian processes for signal strength-based location estimation" in *Proceedings of Robotics: Science and Systems*, August 2006.
[2] H. Leppakoski, J. Collin, and J. Takala. "Pedestrian navigation based on inertial sensors, indoor map, and WLAN signals" in *Proceedings of Acoustics, Speech and Signal Processing*, March 2012.
[3] Xiang He, Shirin Badiei, Daniel Aloi, Jia Li. "WiFi iLocate: WiFi based indoor localization for smartphone" in *Proceedings of Wireless Telecommunication Symposium*, August 2014.
[4] Fan Li, Chunshui Zhao, Guangzhong Ding, Jian Gong, Chengxing Liu, Feng Zhao. "A reliable and accurate indoor localization mthod using phone inertial sensors", in *Proceedings of International Conference on Ubiquitous Computing*, September 2012.
[5] Frederic Evennou and Francois Marx. "Advanced integration of WiFi and inertial navigation systems for indoor mobile positioning", in *EURASIP Journal on Applied Signal Processing*, Volume 2006, Pages 1-11.
[6] M. Weyn, M. Klepal and Widyawan. "Adaptive motion model for a smartphone based opportunistic localization system" in *Proceedings of Mobile Entity Localization and Tracking in GPS-less environments*, 2009, Pages 50-65.
[7] A. Rai, K. Chintalapudi, V. Padmanabhan and R. Sen. "Zee: Zero-Effort Crowdsourcing for Indoor Localization" In *Proceedings of the 18th annual international conference on Mobile computing and networking*, August 22-26, 2012.
[8] Jeong Won Kim. "A step, stride and heading determination for the pedestrian navigation system" in *Journal of global positioning system*, volume 3, No. 1-2: 273-279.
[9] J. Liu, R. Chen, L. Pei, W. Chen, T. Tenhunen, H. Kuusniemi, T. Kroger and Y. Chen. "Accelerometer assisted robust wireless signal positioning based on a hidden Markov model" in *Proceedings of Position Location and Navigation Symposium*, May 2010.
[10] 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 *Proceedings of Mobile Computing and Networking*, August 2012.
[11] Google Project Tango, https://www.google.com/atap/projecttango/.