# Indoor User Tracking with Particle Filter

Incheol Kim
Department of Computer Science
Kyonggi University
Suwon, Korea
kic@kyonggi.ac.kr

Eunmi Choi, Huikyung Oh
Department of Computer Science
Kyonggi University
Suwon, Korea
{allychoi, ohkv770}@kyonggi.ac.kr

***Abstract*—Recently there have been developed a number of mobile personal assistants, which can provide their users with useful location-based services. In this paper, we propose a WiFi fingerprint-based localization algorithm for tracking the accurate position of a smartphone user in indoor environment. To meet high complexity of localization in a large continuous environment, our algorithm incorporates a graph-based space representation, a linear interpolation-based observation model, and three component motion models into the particle filter framework. In experimental evaluation, our WiFi localization algorithm showed high accuracy and robustness in indoor tracking.**

***Keywords-WiFi fingerprint; indoor tracking; particle filter; probabilistic model***

## I. INTRODUCTION

The location information of a user plays a key role in various mobile services. In outdoor environment, GPS is a common solution for obtaining location information, but it does not work well in indoor environments. WiFi fingerprint-based indoor positioning systems [1,2] are currently attracting interest, since they can reduce installation costs. WiFi network modules are readily embedded in a variety of mobile devices, and WiFi APs are commonly installed in modern buildings. However, indoor localization using WiFi signal strength has the problem of unpredictable signal propagation through indoor environments. To meet this uncertainty problem of WiFi fingerprint-based localization, many probabilistic/statistic approaches have been proposed [2,3]. The best known of them is the particle filter, in which the posterior probability distribution of the current position of a user is represented and propagated using the set of weight samples. For the particle filter localization to be successfully used in a large continuous indoor environment, however, decisions should be made on the following important factors: a space and/or state representation to reduce the size of the state space, an observation model to generate likelihoods at locations for which no calibration data is available, and a motion model to predict the accurate position of a pedestrian.

In this paper, we propose a WiFi fingerprint-based localization algorithm for tracking the position of a smartphone user in indoor environment. To meet high complexity of localization in a large continuous environment, our algorithm incorporates a graph-based space representation, an effective observation model, and three component motion models into the particle filter framework. In experimental evaluation, our WiFi localization algorithm showed high accuracy and robustness in indoor tracking.

The next section presents the representation of an indoor environment and the state of a pedestrian roaming within the environment. Section III describes the WiFi fingerprint map and the observation model built up from the calibration data set. Section IV details the motion model for tracking a pedestrian's motion in an effective manner. Section V describes precisely the particle filter algorithm for tracking the WiFi-enabled smartphone user's position in real-time settings. Section VI explains the experiments for evaluating the performance of our WiFi-based localization algorithm. Section VII concludes and discusses future work.

## II. REPRESENTATION OF ENVIRONMENT

We represent an indoor environment as a graph G=(V, E), where V is a set of vertices indicating pre-determined locations in the environment, and E is a set of edges connecting two adjacent vertices. Figure 1 shows an example, representing one floor in a research building.

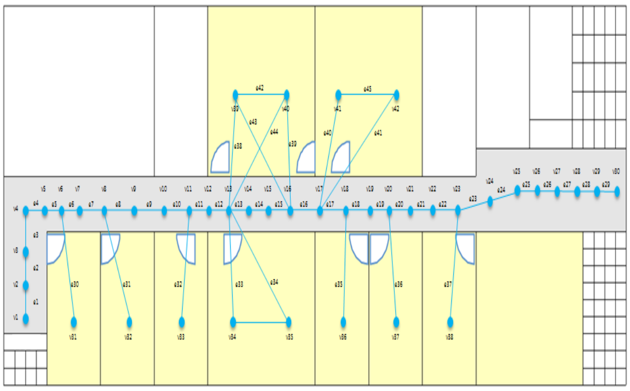


Figure 1. Graph representation of an indoor environment.

Considering important environmental factors such as the layout of corridors and rooms, the position of WiFi APs (WiFi Access Points), the typical motion pattern of residents, the number and the position of vertices are usually decided. We assume a WiFi-equipped smartphone user moves only along the edges of the graph. Hence, an instant position of the user can be viewed as a point on an edge $e_t$, and the state $x_t$ of the user is represented as a tuple $<e_t, s_t, d_t, m_t>$, as illustrated in Figure 2.

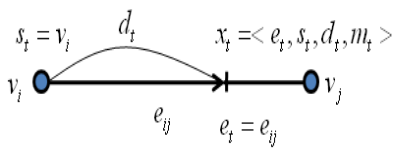



Figure 2. The state of a user roaming in an indoor environment at time t.

$e_t$ represents the edge the user currently walks along, and $s_t$ indicates the starting vertex the user entered the edge through. $d_t$ represents the distance from the position of the starting vertex to the current position of the user, and $m_t$ expresses the motion state indicating whether the user is *moving* or *stopped*.

## III. WIFI FINGERPRINT MAP AND OBSERVATION MODEL

To enable online tracking with the particle filter, we need an observation model to tell the likelihood of observing the specific WiFi signal strength at a certain location in the environment. In fingerprint-based localization systems, this observation model and so-called WiFi fingerprint map are obtained from the calibration data. To collect the calibration data over the environment, we scanned WiFi signal strength vectors at the locations indicated by the vertices $v \in V$ on the graph. Each entry of the calibration data consists of the WiFi signal strength vector and the location label, that, a pair of <WiFi RSS vector, location>. We assume the likelihood of observing the specific signal strength at a certain location is a Gaussian distribution. Assume there exist |A| number of APs (Access Points) discovered in the indoor environment. In practice, the WiFi fingerprint map includes |V|x|A| number of Gaussian distributions, each of them represented by two parameters: the mean $\mu_p$ and the variance $\sigma_p^2$ of WiFi signal strength of each AP as measured from each vertex $p$. Based on these two parameters $\mu_p$ and $\sigma_p^2$ of the WiFi distribution at the location $x_p$, we can compute the likelihood $p(z|x_p)$ of observing the specific signal strength $z$ at the location $x_p$, as formulated in the equations (1) and (2).

$$p(z \mid x_p) = \frac{1}{\sqrt{2\pi\sigma_p^2}} \exp\left(-\frac{(z-\mu_p)^2}{2\sigma_p^2}\right) \qquad (1)$$

$$p(\vec{z} \mid x_p) = \prod_{i=1}^{k} \frac{1}{\sqrt{2\pi\sigma_p^2}} \exp\left(-\frac{(z-\mu_p)^2}{2\sigma_p^2}\right) \qquad (2)$$

To obtain a full observation model covering over an entire large continuous state space, we should be able to compute likelihoods at locations for which no calibration data is available. As illustrated in Figure 3, we can estimate ($\mu_p$, $\sigma_p^2$), the likelihood distribution parameters at an arbitrary location $x_p$ on an edge by linearly interpolating ($\mu_i$, $\sigma_j^2$) and ($\mu_j$, $\sigma_j^2$), the WiFi distribution parameters of its two end vertices $x_i$ and $x_j$. This means that if we know the likelihood distributions of any two vertices, the distribution at every location on the edge connecting these vertices can be also estimated by using the linear interpolation.

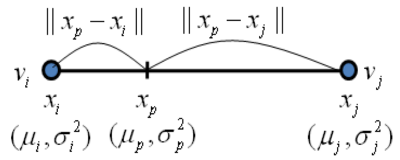



Figure 3. Linear interpolation to compute the likelihood distribution.

The linearly interpolated mean signal strength $\mu_p$ and the variance $\sigma_p^2$ can be computed by:

$$\mu_p = \frac{\| x_p - x_j \| \cdot \mu_i + \| x_p - x_i \| \cdot \mu_j}{\| x_i - x_j \|} \qquad (3)$$

$$\sigma_p^2 = \frac{\| x_p - x_j \| \cdot \sigma_i^2 + \| x_p - x_i \| \cdot \sigma_j^2}{\| x_i - x_j \|} \qquad (4)$$

Therefore, by building the WiFi fingerprint map in the form of the annotated graph G=(V, E), where each vertex is annotated with the mean and the variance of the WiFi signal strength distribution at the corresponding location, we can obtain a complete observation model covering over the entire space.

## IV. THREE COMPONENT MOTION MODELS

In order to track effectively a WiFi-equipped smartphone user with the particle filter, we need a well-defined motion model of the user as well as a precise observation model. In this paper, we define the motion model $p(x_t|x_{t-1})$ by integrating three component models: motion state transition model $p(m_t|m_{t-1})$, edge transition model $p(e_t|e_{t-1})$, and motion distance model $p(d_t)$.

### A. Motion State Transition Model

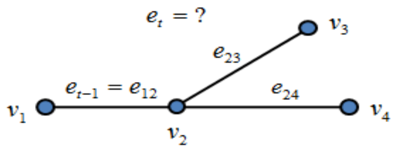
Motion state transitions $p(m_t|m_{t-1})$ represent the probability of motion state $m_t$ being *moving* or *stopped* given the previous state. Table 1 shows an example of the motion state transition model.

TABLE I. AN EXAMPLE OF THE MOTION STATE TRANSITION MODEL

| $p(m_t \mid m_{t-1})$ | | $m_t$ stopped | $m_t$ moving |
|---|---|---|---|
| $m_{t-1}$ | stopped | 0.55 | 0.45 |
| | moving | 0.25 | 0.75 |

### B. Edge Transition Model

Edge transitions $p(e_t|e_{t-1})$ represent the probability of choosing the next edge when reaching a vertex. They are stored at each vertex of the graph. An example of the edge transition model is illustrated in Figure 4.




$$p(e_t = e_{12} \mid e_{t-1} = e_{12}) = 0.2$$
$$p(e_t = e_{23} \mid e_{t-1} = e_{12}) = 0.3$$
$$p(e_t = e_{24} \mid e_{t-1} = e_{12}) = 0.5$$

Figure 4. An example of the edge transition model $p(e_t|e_{t-1})$.

### C. Motion Distance Model

Motion distance model $p(d_t)$ represents the probability of going away at distance $d$ from the previous position. The distance $d$ is sampled according to the Gaussian distribution with the mean $\mu_d$ and the variance $\sigma_d^2$: $p(d) \sim N(\mu_d\, \sigma_d^2)$. The mean $\mu_d$ and the variance $\sigma_d^2$ are manually set based on typical motion patterns of pedestrians, or learned from a specific user group's training data.

Sampling from the resulting motion model is done as follows. When $x_{t-1} = <e_{t-1}, s_{t-1}, d_{t-1}, m_{t-1}>$ is on an edge in the graph, we first sample the motion state $m_t$ with probability proportional to $p(m_t|m_{t-1})$. If $m_t = stopped$, then $x_t$ is set to be $x_{t-1}$. Otherwise, if $m_t = moving$, then we randomly draw a moving distance $d$ according to the Gaussian distribution $N(\mu_d\, \sigma_d^2)$. For this distance $d$, we determine whether the motion along the edge results in a transition over the end vertex of $e_{t-1}$. If not, then $d_t = d_{t-1} + d$ and $e_t = e_{t-1}$, $s_t = s_{t-1}$. Otherwise, we set $d_t = d_{t-1} + d + | e_{t-1} |$ and the next edge $e_t$ is sampled with probability $p(e_t|e_{t-1})$, and then $s_t$ is set to the starting vertex of the edge $e_t$.

As a summary, we define the motion model $p(x_t|x_{t-1})$ as follows.

$$\hat{p}(x_t \mid x_{t-1}) = p(m_t = moving \mid m_{t-1}) \cdot N(d_t; \mu_d, \sigma_d^2) + p(m_t = stopped \mid m_{t-1}) \cdot \delta(x_t, x_{t-1}),$$

$$\text{where } \delta(x_t, x_{t-1}) = \begin{cases} 1 & if \quad x_t = x_{t-1} \\ 0 & if \quad x_t \neq x_{t-1} \end{cases} \tag{5}$$

$$p(x_t \mid x_{t-1}) = \begin{cases} p(x_t \mid x_{t-1}) & if \quad e_t = e_{t-1} \\ p(e_t \mid e_{t-1}) \hat{p}(x_t \mid x_{t-1}) & if \quad e_t \neq e_{t-1} \end{cases} \tag{6}$$

## V. PARTICLE FILTER FOR INDOOR TRACKING

Particle filters provide a sample-based implementation of general Bayes filters [4,5]. The key idea of particle filters is to represent posterior over the state $x_t$ by sets $X_t$ of $M$ weighted samples: $X_t = \{<x_t^{[m]}, w_t^{[m]}> \mid m = 1, \ldots, M\}$. Here each $x_t^{[m]}$ is a sample state, represented by the tuple $<e_t^{[m]}, s_t^{[m]}, d_t^{[m]}, m_t^{[m]}>$ in our work, and $w_t^{[m]}$ is an importance weight of the state. Particle filters apply the recursive Bayes filter update to estimate posteriors over the state space. The basic form of the particle filter updates the posterior of the smartphone user's state according to the algorithm summarized in Figure 5. In this algorithm, the probability $p(x_t|x_{t-1})$ is the same one defined as motion model in the Section IV, while the probability $p(z_t|x_t)$ is the same one defined as observation model in the Section III. The input of this algorithm is the particle set $X_{t-1}$, along with the most recent measurement $z_t$. The algorithm then first constructs a temporary particle set $\hat{X}_t$ by systematically processing each particle $x_{t-1}^{[m]}$ in the input particle set $X_{t-1}$. Subsequently, it transforms these particles into the set $X_t$, which approximates the posterior distribution of the smartphone user's state $x_t$. Line 4 generates a hypothetical state $x_t^{[m]}$ for time $t$ based on the particle $x_{t-1}^{[m]}$, and this step involves sampling from the distribution $p(x_t|x_{t-1})$. Line 7 calculates for each particle $x_t^{[m]}$ the so-called importance factor, denoted by $w_t^{[m]}$. The importance is the probability of the measurement $z_t$ under the particle $x_t^{[m]}$. Line 11 through 15 implemented resampling by drawing with replacement $M$ particles from the temporary set $\hat{X}_t$.

## VI. EXPERIMENTAL EVALUATION

To evaluate the performance of our WiFi localization algorithm, we conduct experiments with a WiFi-equipped Android smartphone in the same indoor environment as shown in Figure 1. The size of the environment is about 52 $m$ x 18 $m$, and the average length of edges on the graph is about 2.5 $m$.

| | |
|---|---|
| 1 | **Algorithm Particle_filter(** $X_{t-1}$, $z_t$ ) |
| 2 | $\hat{X}_t = X_t = \{\}$ |
| 3 | for $m$=1 to $M$ do |
| 4 | // **Prediction Step** |
| 5 | Sample $x_t^{[m]}$ with probability $p(x_t \mid x_{t-1})$ |
| 6 | // **Update Step** |
| 7 | $w_t^{[m]} = p(z_t \mid x_t^{[m]})$ |
| 8 | $\hat{X}_t = \hat{X}_t + <x_t^{[m]}, w_t^{[m]}>$ |
| 9 | endfor |
| 10 | |
| 11 | for $m$=1 to $M$ do |
| 12 | // **Resample Step** |
| 13 | Draw i with probability $\propto w_t^{[m]}$ |
| 14 | Add $x_t^{[i]}$ to $X_t$ |
| 15 | endfor |
| 16 | return $X_t$ |

Figure 5. The particle filter algorithm.


ISBN: 978-1-61208-218-9

### A. Accuracy

In this test, we evaluate the accuracy of our localization algorithm by measuring the average error distances during 5 repeated traverses on along 3 different paths in the environment. Additionally, we investigate how the localization accuracy changes as the number of particles increases. Figure 6 shows the result of experiments. We find out that neither there are remarkable differences in error distance among different paths, nor among different particle numbers (200 ~ 400). The average error distance for individual paths is just about 0.94 *m* ~ 1.2 *m*. This result makes sure the high accuracy of our WiFi localization algorithm. In our experiments, when the number of particles is 250, we get the best performance.

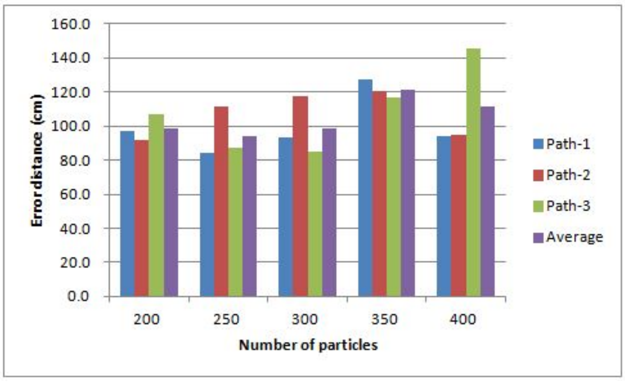



Figure 6. Number of particles vs. localization error.

### B. Robustness

In this test, we evaluate the robustness of our WiFi localization algorithm by investigating how the localization accuracy changes as the number of WiFi access points(APs) decreases. Figure 7 shows the result of experiments.

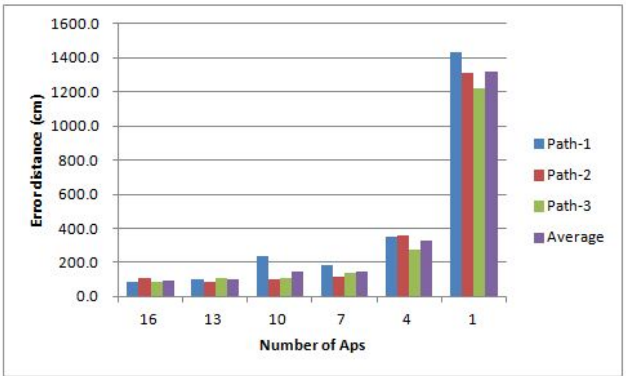



Figure 7. Number of APs vs. localization error.

We find out that until the number of APs decreases from 16 to 7, the average error distance for individual paths remains shorter than 1.5 *m*. That is, while the number of APs decreases to a certain degree, the localization accuracy does not decrease remarkably. This result makes sure the high robustness of our localization algorithm to meet possible changes of the WiFi environment.

## VII. CONCLUSIONS

We proposed a WiFi fingerprint-based localization algorithm for indoor user tracking. To track the position of a smartphone user in a large continuous environment, our algorithm incorporates a graph-based space representation, a linear interpolation-based observation model covering over the entire space, three component motion models into the particle filter framework. Through experiments, we proved the high accuracy and robustness of our localization algorithm. Our WiFi fingerprint-based localization algorithm has the limitation that it needs a large pre-built WiFi fingerprint map of the environment. A lot of effort and time is necessary to construct such a large WiFi fingerprint map. To overcome this limitation, we are now extending our algorithm to adopt some techniques [6,7] from SLAM research communities.

## ACKNOWLEDGMENT

This work was supported by the GRRC program of Gyeonggi province.

## REFERENCES

[1] J. Biswas and M. Veloso, "WiFi Localization and Navigation for Autonomous Indoor Mobile Robots", Proc. of IEEE International Conference on Robotics and Automation (ICRA), 2010. [retrieved: June, 2012]

[2] F. Duvallet and A. D. Tews, "WiFi Position Estimation in Industrial Environments Using Gaussian Processes", Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), 2008. [retrieved: June, 2012]

[3] B. Ferris, D. Hahnel, and D. Fox, "Gaussian Processes for Signal Strength-Based Location Estimation", Proc. of Robotics Science and Systems, 2006. [retrieved: June, 2012]

[4] F. Gustafsson, F. Gunnarsson, N. Bergman, and U. Forssell, "Particle Filters for Positioning, Navigation, and Tracking", IEEE Transactions on Signal Processing, Vol.50, No.2, pp. 425-437, 2002. [retrieved: June, 2012]

[5] S. Thrun, W. Burgard, and D. Fox, Probabilistic Robotics, MIT Press, 2005. [retrieved: June, 2012]

[6] B. Ferris, D. Fox, and N. Lawrence, “WiFi-SLAM Using Gaussian Process Latent Variable Models”, Proc. of Int. Joint Conf. on Artificial Intelligence (IJCAI), 2007. [retrieved: June, 2012]

[7] J. Huang, D. Millman, M. Quigley, D. Stavens, S. Thrun, and A. Aggarwal, “Efficient, Generalized Indoor WiFi GraphSLAM”, Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA), 2011. [retrieved: June, 2012]