Proceeding of the IEEE
International Conference on Information and Automation
Hailar, China, July 2014

WiFi Signal Strength-based Robot Indoor Localization

Yuxiang Sun*, Ming Liu†, Max Q.-H, Meng*
* Department of Electronic Engineering, The Chinese University of Hong Kong
† Department of Electronic & Computer Engineering, The Hong Kong University of Science & Technology
{yxsun,qhmeng}@ee.cuhk.edu.hk, eelium@ece.ust.hk

***Abstract*— Due to the unavailable GPS signals in indoor environments, indoor localization has become an increasingly heated research topic in recent years. Researchers in robotics community have tried many approaches, but this is still an unsolved problem considering the balance between performance and cost. The widely deployed low-cost WiFi infrastructure provides a great opportunity for indoor localization. In this paper, we develop a system for WiFi signal strength-based indoor localization and implement two approaches. The first is improved KNN algorithm-based fingerprint matching method, and the other is the Gaussian Process Regression (GPR) with Bayes Filter approach. We conduct experiments to compare the improved KNN algorithm with the classical KNN algorithm and evaluate the localization performance of the GPR with Bayes Filter approach. The experiment results show that the improved KNN algorithm can bring enhancement for the fingerprint matching method compared with the classical KNN algorithm. In addition, the GPR with Bayes Filter approach can provide about 2*m* localization accuracy for our test environment.**

I. INTRODUCTION

The unavailability of GPS signals in indoor environment makes robots difficult to localize themselves. The problem of indoor localization has become an increasingly heated research topic in recent years. Many researches have tried many approaches in the past decade. Due to the development of powerful computer vision algorithms, image sensors are widely employed for indoor localization[1]. Unfortunately, the image matching often fails when there are major changes in the scene although many approaches to enhance the ability of scene recognition[2] [3] are available. The laser scanner-based SLAM system[4] can help robots position precisely. However, the extremely expensive laser equipment hampers the wide applications. The state-of-the-art Visible Light Communication (VLC) can also provide high accurate localization[5], however, this approach requires specific designing for LED lights. With the introducing of WiFi routers and the widely deployed low-cost WiFi infrastructure, WiFi signal strength provides a great opportunity for indoor localization. The advantages of ubiquitous infrastructure and no hardware modification requirement make this approach be extensively studied in the past years.

In this paper, we develop a system for WiFi signal strength-based indoor localization and implement the improved KNN-based fingerprint matching and Gaussian Process Regression (GPR) with Bayes Filter methods for this system. Both approaches can be broken up into two stages[6]. The first is the calibration stage, during which the robot records the signal strength at different positions and builds a fingerprint database. In the second stage, which we call the measurement stage, the robot determines its position by pattern recognition and probabilistic methods.

This paper is organized as follows. In section II, we explain the feasibility for WiFi signal strength-based localization and describe the system we developed for the WiFi localization. In section III, the details of Received Signal Strength Identification (RSSI) fingerprint[7] methods will be presented and the comparing experiments between improved KNN and classical KNN algorithms will be conducted. In section IV, we model the WiFi signal strength by Gaussian Process, and build the regression prediction for the whole RSSI field. During the online stage, we use Bayes Filter to determine the position. The localization performance of this method will be evaluated in this section. We conclude this paper and discuss our future work at the end.

II. WIFI FUNDAMENTALS AND SYSTEM DEVELOPMENT

In the first part of this section, we will introduce the WiFi fundamentals and explain the feasibility for WiFi signal strength-based localization. In the second part, we will describe our WiFi localization system.

A. WiFi Fundamentals

The word WiFi is a trademark name that is short for Wireless Fidelity. The WiFi Alliance defines WiFi as any Wireless Local Area Network (WLAN) products that are based on the IEEE802.11 standards[8]. The wireless radio channel serves as the medium for wireless communication. The software on the mobile station can calculate the received power by integrating over the WiFi beacon packages. The received signal strength[9] is measured in decibel in milliwatt as *dBm* with the following formula,

$$RSSI(\text{dBm}) = 10\log\frac{ReceivedPower(\text{mW})}{1\text{mW}}. \qquad (1)$$

The variation of the received signal power over distance is due to the path loss, shadowing and multi-path effects. This variation is essential to form unique RSSI vectors at different locations, which explains why signal strength fingerprint is a feasible method. However, the channel is susceptible to noises, interference, impediments, etc. These factors change over time randomly due to the user movement, environment change and other unpredictable reasons, so the received power and RSSI will become unstable accordingly[10]. This instability of signal strength will bring errors for WiFi localization.

978-1-4799-4100-1/14/$31.00 ©2014 IEEE

B. System Development

The WiFi localization system consists of a mobile robot and 5 WiFi routers(*TP-Link WR841N*) that are deployed evenly in our test room. The laptop running our software developed in C++ by using Qt5 GUI library. Fig.1 shows

Fig. 1. The Mobile Robot and the Laptop

the mobile robot *Turtlebot* in the test room. The robot is controlled by our software. The test room is 11.7m long and 6.64m wide and divided by 39×22 square grids. The room is laboratory in which there are people, furniture, and instruments. The software is shown in Fig.2, which is a snapshot for the calibration process.

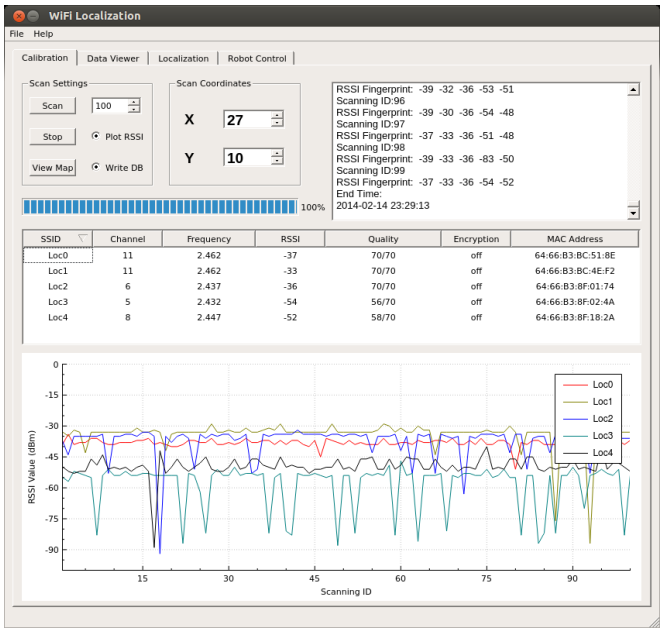


Fig. 2. The Calibration Process for WiFi Localization

III. Fingerprint Matching Approach

The RSSI vectors from different locations can be constructed in a database, with which the mobile robot can identify its position by using pattern recognition algorithms, such as the KNN algorithm. This kind of method is called the RSSI fingerprint matching method. Fig.3 shows the schematic overview of this approach. There are two stages for this method[11]. The first is the calibration stage, during which the robot records the signal strength in different positions and builds a database. In the measurement stage, the robot determines its position by employing the improved KNN algorithm. In this paper, the robot records the RSSI fingerprints 100 times for each position. And the improved KNN algorithm that is similar to the approach proposed by Q. Tran *et al*[12] is implemented.

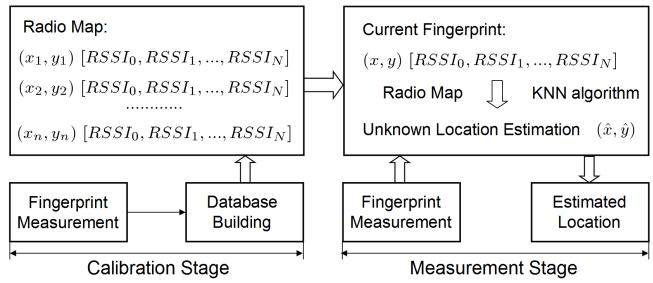


Fig. 3. The Schematic Overview of Fingerprint Matching Approach

A. Improved KNN Algorithm

The general idea of classical KNN[13] algorithm is like this. Assume we have an unknown data and a training dataset with known labels. What we want is to determine the label for this unknown data from the labels of the training dataset. Firstly, we need to choose an appropriate distance metric[14] and calculate the distances from the unlabeled data to each of the data in the training dataset, and then sort all the data in the training dataset in ascending order by the calculated distances. We just consider the top K data in the training dataset. Among these K data, we count the votes for the available labels. The label that has the maximum vote will be the label for the unknown data. In this paper, we use the raw measurement to construct the fingerprint database, which means for every calibrated points there will be 100 fingerprints. The 100 fingerprints are annotated with position as the label. The unlabeled fingerprint is acquired during the online measurement stage. What we want to do is to discriminate the type of this unlabeled fingerprint. Based on the assumption that different location has different fingerprint, there will be only one estimated location for the fingerprint.

However, the classical KNN algorithm does not take into account the variation of the training data. The RSSI will be subject to fluctuation that is caused by unpredictable reasons, such as walking people, electromagnetic interference, air moisture, etc. If the fingerprint with large variation serves as the training data, the localization accuracy will be degraded. The ideal training data is the fingerprint of which each value is closed to the mode value of the raw RSSI. In order to eliminate the large varying training data, a threshold is needed to handle the variation. We assume that if the total variation is greater than the threshold, this training data will be eliminated. Let T denote the training fingerprint vectors, the corresponding mode vector is M. The total variations

can be found by calculating the error from the $L-1$ norm between M and T,

$$TotalVariations = \|M - T\|_1 \tag{2}$$

The table below summarizes the Improved KNN algorithm. The first step is to calculate the Euclidean distance from the current fingerprint to each training data, and then sort the training dataset by the Euclidean distance in ascending order. The third step is to find each location vote in the top K training data, and at the same time check the training data whether its total variation is larger than the empirical threshold. If yes, eliminate this training data from the top K training dataset. The last step is to return the position with the maximum vote, which is the estimated location.

Algorithm 1: Improved KNN

Input: Training Dataset, Current Fingerprint
1.Calculate the Euclidean distance for each training data
2.Sort the training dataset in ascending order
3.Find the vote for each postion in top K training data
for $i \leftarrow 0$ ***to*** L **do**
  for $j \leftarrow 0$ ***to*** K **do**
    if $trainingData[i].totalVariations < threshold$ **then**
      $location[i].vote++$
    else
      delete $trainingData[i]$
4.**return** *location with Max vote*

B. Experiment Results

In this experiment, the performance of the improved KNN algorithm will be evaluated. We implement the algorithm in C++. The software runs on a PC with an Intel Atom CPU and 1GB memory. We conduct the same localization process at two distinct points $(20,10)$ and $(27,10)$. At each point, we employ the KNN and improved KNN algorithm to perform localization 500 times respectively.

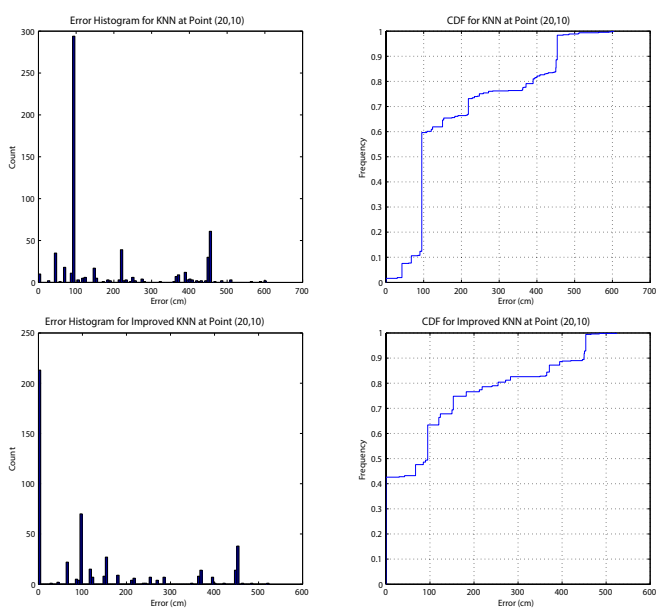


Fig. 4. The Experiment Results at Point$(20,10)$

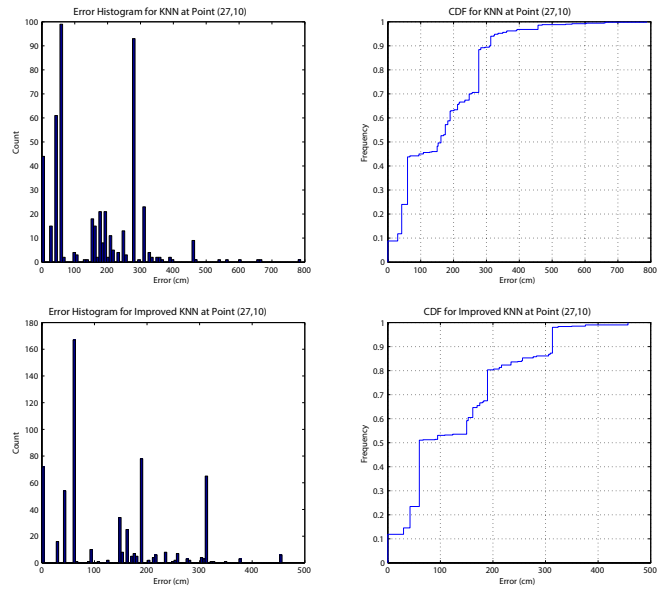


Fig. 5. The Experiment Results at Point$(27,10)$

The errors distributions and the plots of Cumulative Distribution Functions (CDF) are shown in Fig.4 and Fig.5. We can calculate from Fig.4 that the mean errors of KNN and improved KNN for point $(20,10)$ are $187.3cm$ and $126.4cm$, the accuracy is improved by 32.5%. In the same way, the mean errors for point $(27,10)$ are $159.9cm$ and $129.1cm$, the accuracy is improved by 19.3%. Therefore, we can see from the experiment results that the improved KNN algorithm can enhance the performance of the classical KNN-based matching method.

IV. GPR with Bayes Filter Approach

In this section, we implement the approach proposed by B. Ferris *et al* [15]. This approach models the RSSI field by Gaussian Process Regression (GPR), and employs the Bayes Filter to estimate the position. The schematic overview of this approach is shown in Fig.6.

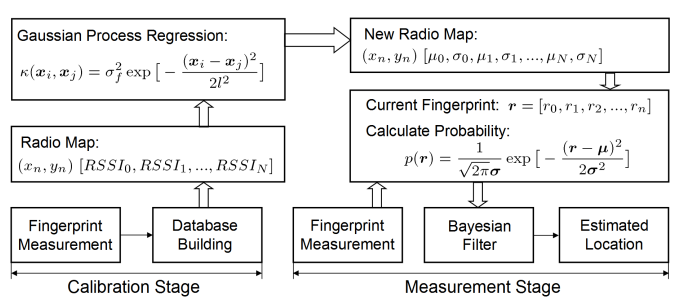


Fig. 6. The Schematic Overview of GPR with Bayes Filter Approach

Same as the fingerprint matching approach, this approach also requires a calibration process. However, the calibration process can only construct a sparse database, because we can't record the RSSI for all the appropriate positions due to the huge time consumption. The sparse database can't produce enough information for more precise location estimation.

However, we observed that the distribution of the WiFi signal strength normally obeys a Gaussian distribution. This intuitively let us approximate the RSSI filed as a Gaussian Process[16]. The Gaussian Process Regression (GPR) can predict the RSSI at the unrecorded positions. Therefore, by using GPR we can construct a RSSI database that can produce more information for the precise position estimation. During the measurement stage, we employ the Bayes Filter to find the posterior belief at each position. The iterative time and measurement update rules can produce a more reliable result, and the estimated location will converge to a fixed value.

A. GPR Preliminaries

Gaussian Process (GP) can be viewed as a distribution over functions[17], it can be fully specified by its mean and covariance functions,

$$f(x) \sim \mathcal{GP}\big(m(\boldsymbol{x}), k(\boldsymbol{x},\boldsymbol{x}')\big), \tag{3}$$

$$m(\boldsymbol{x}) = \mathbb{E}[f(\boldsymbol{x})], \tag{4}$$

$$k(\boldsymbol{x},\boldsymbol{x}') = \mathbb{E}\big[\big(f(\boldsymbol{x}) - m(\boldsymbol{x})\big)\big(f(\boldsymbol{x}') - m(\boldsymbol{x}')\big)\big], \tag{5}$$

where $m(\boldsymbol{x})$ is the mean function, $k(\boldsymbol{x},\boldsymbol{x}')$ is the covariance function. Let

$$\boldsymbol{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} f(\boldsymbol{x}_1) \\ f(\boldsymbol{x}_2) \\ \vdots \\ f(\boldsymbol{x}_n) \end{bmatrix}, \tag{6}$$

$$\boldsymbol{\mu} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_n \end{bmatrix} = \begin{bmatrix} \mathbb{E}[f(\boldsymbol{x}_1)] \\ \mathbb{E}[f(\boldsymbol{x}_2)] \\ \vdots \\ \mathbb{E}[f(\boldsymbol{x}_n)] \end{bmatrix}, \tag{7}$$

$$\boldsymbol{\Sigma} = \begin{bmatrix} \kappa(\boldsymbol{x}_1,\boldsymbol{x}_1) & \kappa(\boldsymbol{x}_1,\boldsymbol{x}_2) & \cdots & \kappa(\boldsymbol{x}_1,\boldsymbol{x}_n) \\ \kappa(\boldsymbol{x}_2,\boldsymbol{x}_1) & \kappa(\boldsymbol{x}_2,\boldsymbol{x}_2) & \cdots & \kappa(\boldsymbol{x}_2,\boldsymbol{x}_n) \\ \vdots & \vdots & \ddots & \vdots \\ \kappa(\boldsymbol{x}_n,\boldsymbol{x}_1) & \kappa(\boldsymbol{x}_n,\boldsymbol{x}_2) & \cdots & \kappa(\boldsymbol{x}_n,\boldsymbol{x}_n) \end{bmatrix}. \tag{8}$$

The joint probability distribution function can be written as follows,

$$\begin{aligned} &f(\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n) \\ &= \frac{1}{(2\pi)^{\frac{n}{2}}(\det\Sigma)^{\frac{1}{2}}} \exp\big[-\frac{1}{2}(\boldsymbol{Y}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\boldsymbol{Y}-\boldsymbol{\mu})\big]. \end{aligned} \tag{9}$$

The key idea for the regression of GP is that the inputs are not independent, and they are correlated and can be indicated by the covariance. The covariance function can be modeled by kernel functions, which can measure the similarity between two inputs. There exist many types of kernel functions; however, in GPR we normally employ the Gaussian kernel that is also called the squared exponential kernel,

$$\kappa(\boldsymbol{x}_i,\boldsymbol{x}_j) = \sigma_f^2 \exp\big[-\frac{(\boldsymbol{x}_i-\boldsymbol{x}_j)^2}{2l^2}\big]. \tag{10}$$

Assume we have a training set $\mathcal{D}$,

$$\mathcal{D} = \{(\boldsymbol{x}_0,y_0),(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),...,(\boldsymbol{x}_n,y_n)\},$$

$\boldsymbol{x}$ denotes the input, y is the noisy measurement which can be expressed by $y = f(\boldsymbol{x}) + \varepsilon$, where ε is the i.i.d noise whose variance is σ_n^2, so the covariance function becomes,

$$\mathrm{cov}\big(f(x_i), f(x_j)\big) = \kappa(x_i,x_j) + \sigma_n^2\delta_{ij}, \tag{11}$$

where δ_{ij} is a Dirac function, the equation can be written in a compact form,

$$\mathrm{cov}(\boldsymbol{y}) = K(X,X) + \sigma_n^2 I, \tag{12}$$

where $X \in \mathbb{R}^{n\times n}$, $\boldsymbol{y} \in \mathbb{R}^{n\times 1}$, I is an identity matrix. The joint distribution of the training data is a multivariate Gaussian distribution,

$$\boldsymbol{y} \sim \mathcal{N}\big(0, K(X,X) + \sigma_n^2 I\big). \tag{13}$$

Assume n_* is the test point, $X_* \in \mathbb{R}^{n_*\times 1}$, we have the following joint distribution,

$$\begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{f}(X_*) \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(X,X)+\sigma_n^2 I & K(X,X_*) \\ K(X_*,X) & K(X_*,X_*) \end{bmatrix}\right). \tag{14}$$

We can derive the key predictive equations for GPR as follows,

$$\mathbb{E}[f(X_*|X,\boldsymbol{y},X_*)] = K(X,X_*)\big[K(X,X)+\sigma_n^2 I\big]^{-1}\boldsymbol{y}, \tag{15}$$

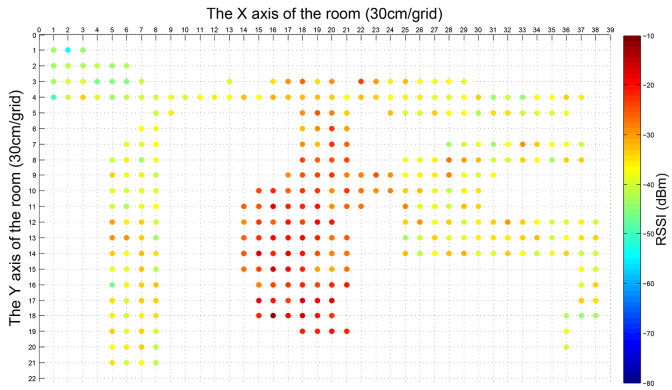
$$\begin{aligned} &\mathrm{cov}\big(f(X_*)\big) = \\ &K(X_*,X_*) - K(X_*,X)\big[K(X,X)+\sigma_n^2 I\big]^{-1}K(X,X_*). \end{aligned} \tag{16}$$

The key predictive equations not only give the mean, but also give the variance for each test input. The parameters σ_f, l, σ_n are called the *hyper parameters* which can be estimated by the gradient descent algorithm.

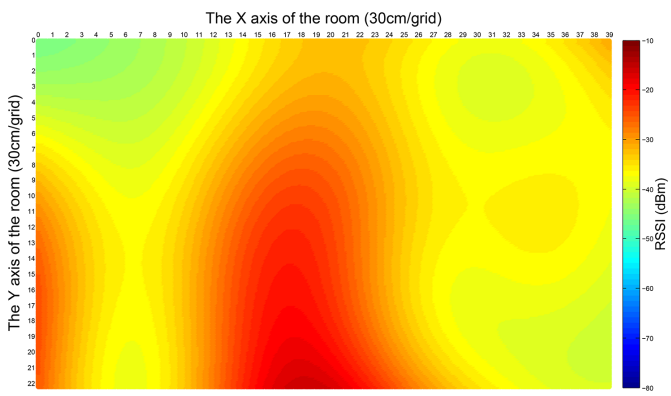
B. RSSI Prediction Results

In this paper, the training data is the mode vector $\boldsymbol{y} \in \mathbb{R}^{n\times 5}$ annotated with positions $\boldsymbol{x} \in \mathbb{R}^{n\times 2}$, here n is the number of training points, 5 represents there are 5 routers. For each router and each recorded point, there is only one RSSI vector for GPR prediction. The task is to predict the RSSI mean and Standard Deviation (S.D.) values for the unrecorded points. We implement the GPR by using the Matlab Toolbox[18]. Fig.7(a) describes the raw sparse measurement distribution for router *Loc0*, whose mean and S.D. of the RSSI value will be predicted by the GPR algorithm. The results are shown in Fig.7(b) and (c). The new radio map is 10 times denser than the original, so it seems that the mean and S.D. values are continuously distributed over the whole room. Router *Loc0* is placed at the center of the room where there is no larger impediment, metal object and moving people, so the mean RSSI distribution follows a path-loss model approximately. However, the decay rate of the signal strength is larger than

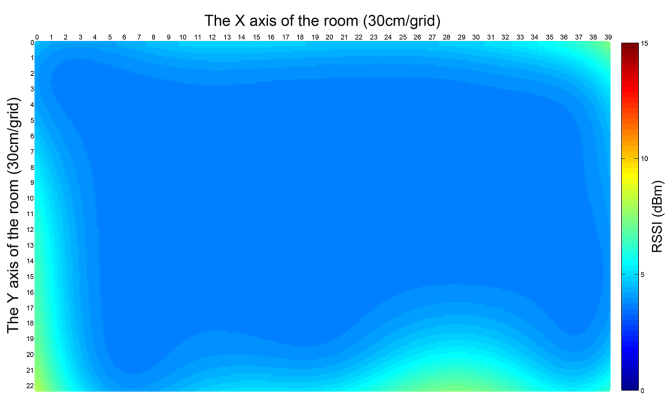
that of the free-space attenuation, whose average value is about $20dB$ per $10m$, so for this indoor environment we can infer the exponent of the approximate attenuation model is larger than that of the free-space.



(a) Mode Value of Raw RSSI(dBm) Measurement for Loc0



(b) Mean of GP Prediction for Loc0



(c) Standard Deviation of GP Prediction for Loc0

Fig. 7. The GPR Prediction Results for *Loc0*

C. Bayes Filter

During the measurement stage, the posterior belief at each point is iteratively computed by time and measurement update rules[19] of Bayes Filter. Let z_t denote the measurement data at time t, we assume the time is discrete and starts from t_0. Let u_t denote the control input, which is normally the odometry information. Let the state vector x_t represent the robot position. What we want to find is the probability $p(x_t|z_{1:t}, u_{1:t})$. What the GPR prediction can give us is the conditional probability $p(z_t|x_{0:t}, z_{1:t-1}, u_{1:t})$. The state x_t is irrelevant to the past states $x_{0:t-1}$, past measurements $z_{1:t-1}$ and control inputs $u_{1:t}$, so the probability can be written as $p(z_t|x_t)$. At time t, given the past measurement data $z_{1:t-1}$ and the control data $u_{1:t}$, $p(x_t|x_{0:t-1}, z_{1:t-1}, u_{1:t})$ is the prediction for the current state. Similar to x_t, x_{t-1} is also irrelevant to those variables, so we have $p(x_t|x_{0:t-1}, z_{1:t-1}, u_{1:t}) = p(x_t|x_{t-1}, u_t)$.

Let $bel(x_t)$ denote the state probability $p(x_t|z_{1:t}, u_{1:t})$, $\overline{bel}(x_t)$ denote the prediction $p(x_t|z_{1:t-1}, u_{1:t})$. Then we have the two iterative update equations[20]:

- *Time Update*:

$$\overline{bel}(x_t) = p(x_t|u_t, x_{t-1}) bel(x_{t-1}). \tag{17}$$

- *Measurement Update*:

$$bel(x_t) = \eta p(z_t|x_t) \overline{bel}(x_t). \tag{18}$$

Algorithm 2: Bayes Filter

Input: Initial Guess for State x_0
1.$bel(x_0) = initialGuess$
2.Iterative Loop:
for $t \leftarrow 1$ ***to*** *Now* **do**
  for *all* x_t **do**
    $\overline{bel}(x_t) = p(x_t|u_t, x_{t-1}) bel(x_{t-1})$;
    $bel(x_t) = \eta p(z_t|x_t) \overline{bel}(x_t)$;
  return $bel(x_t)$

The above table shows the Bayes Filter algorithm. The η here is a normalization scalar which can make sure the posterior belief is a probability at each iteration. In our experiment, the robot remains static, so the probability $p(x_t|u_t, x_{t-1}) = 1$. Through the iterative computations, the distribution of the posterior belief $bel(x_t)$ will be finally convergent.

D. Software Implementation

Fig.8 shows the snapshot of our software implementation using GPR with Bayes Filter approach for WiFi indoor localization. In this figure, the red dots represent the WiFi routers. The shadow areas are desks, instruments, and other impediments. The green square with the coordinate $(18, 15)$ denotes the ground truth for the robot position (the robot remains static in this paper). The black dots in the picture encode the estimated location possibilities for different positions, and the larger radius corresponds to larger probability. As we assume the mobile robot can not get to the shadow areas, so we have not recorded RSSI at these positions. And there is no localization possibility for these areas accordingly. The blue dot represents the estimated location, which can be the point with maximum probability or calculated from the weighted sum of all the positions coordinates. The current estimated position in this figure is $(19.1, 13.1)$, and the error is 2.23 unit length. The radius of the transparent circle 4.2 encodes the distance to the 90% localization probability point.

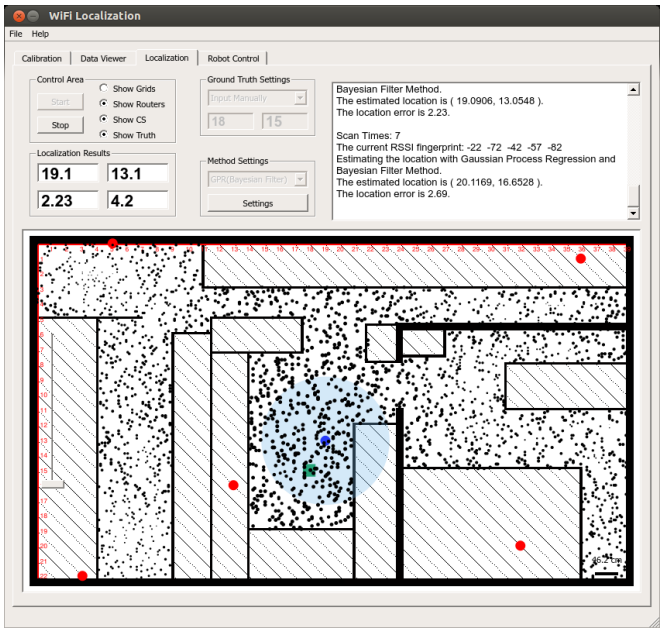


Fig. 8. The Localization Snapshot for GPR with Bayes Approach

The black dot cloud will change and shrink after each iteration. The dots in the cloud will concentrate on the positions with larger probabilities, and finally the cloud will converge to a single point. The localization error at the final status will be the metric for evaluating the performance of this approach.

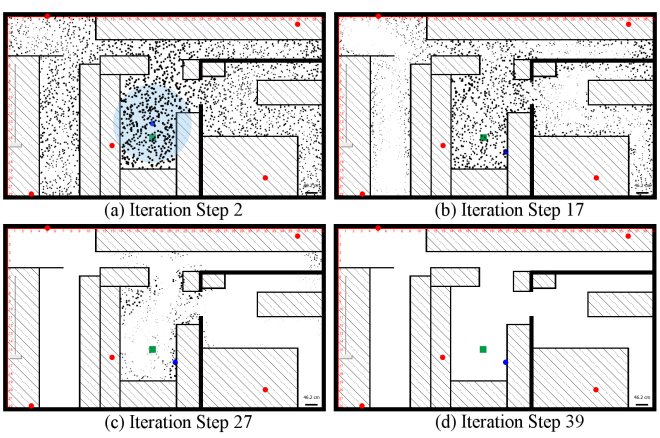


Fig. 9. The Convergence Process for GPR with Bayes Filter Approach

Fig.9 shows the convergence process at different iteration steps (2, 17, 27, 39 from the top left to the bottom right respectively). In Fig.9, we estimate the location by the weighted sum from all the possible points. The convergence process is normally very fast. We can see the estimated coordinates reach near $(20.8, 16.6)$ since the $17th$ step.

We can find from this figure that the convergent value may be not the optimal one. It is obvious that the error in the left top is smaller than the convergent value. From our observations, the error will oscillates within the first few iteration steps, and then the small probabilities will approach zero value quickly. It is quite difficult to change the weighted estimated positions when most points probabilities reach near zero values. This can be easily found especially when there is a big disturbance at the beginning. In that case, there may be a huge error for the original estimated position. Although the interference is removed afterwards, the possibility to pull the estimated position back to the expected value is little. Therefore, this approach is not robust for such case, but this problem can be solved when the robot is moving around the room because the odometry information can help to rectify the wrong possibilities. We can also find from Fig.9 that the distances from the estimated position to the 90% probability point become shorter as the Bayes Filter algorithm iterates. The points with larger probabilities get more and more closer.

E. Experiment Results

In this experiment, we are going to evaluate the localization performance of the GPR with Bayes Filter approach. The test environment is the same as before. The room is a laboratory that is full of moving people and electronic instruments. The people and instruments can be the interference sources for WiFi signal propagation. Thus, the RSSI is not quite stable in our test environment. We conduct the indoor localization at 335 positions through the whole room. The tested positions are the points shown in Fig.7(a). The mean and the S.D. value of the convergent errors for this experiment is $212.3cm$ and $201.2cm$ in this experiment.

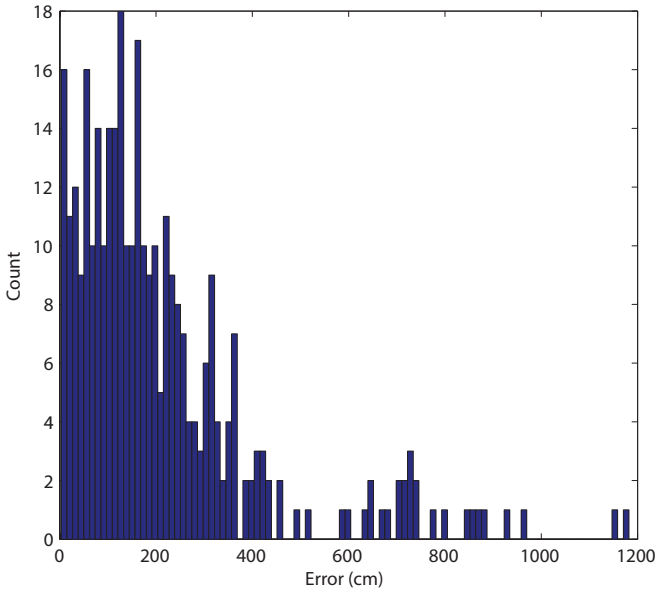


Fig. 10. The Error Distribution of GPR with Bayes Filter Approach

The distribution of the errors is shown in Fig.10. From this figure, we can see for some positions the errors are still so large that the localization can be considered as failed. This is normally due to the large RSSI disturbance at the beginning of localization. Another factor for large errors is the incorrect RSSI predictions, which are caused by the lack of enough calibration data around those positions.

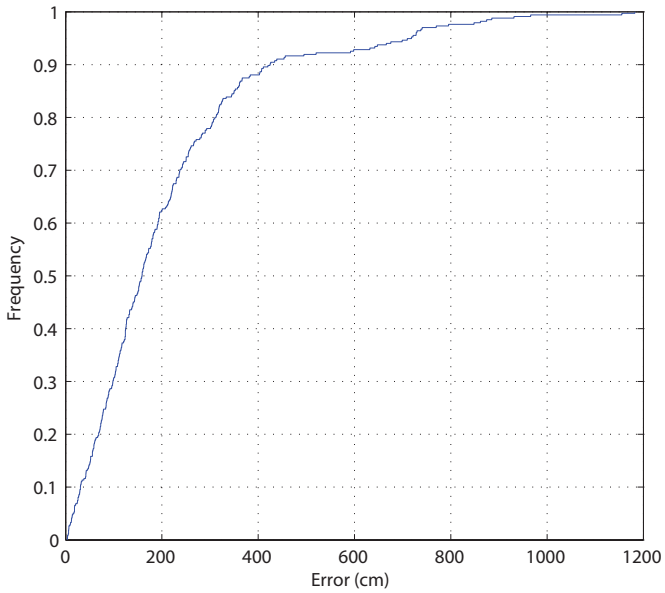


Fig. 11. The Error CDF Plot of GPR with Bayes Filter Approach

Fig.11 shows the CDF plot of the errors. We can see that this approach can provide $2m$ accuracy for nearly 60% positions and about 25% positions can get $1m$ accuracy. This is enough for applications that do not require high accurate positioning.

V. CONCLUSIONS

In this paper, we developed a system for WiFi signal strength-based indoor localization and implemented the two typical kinds of approaches. The experiments results show that the improved KNN algorithm can bring enhancement compared with the classical KNN algorithm for the fingerprint matching approach, and the probabilistic method can provide about $2m$ localization accuracy for our test environment.

In the future, we will incorporate the control input information, such as the odometry from the wheel for the time update rule of the Bayes Filter. In addition, visual information such as from omnidirectional cameras[21] and the current state-of-the-art visible light communication[22] technology can be employed to improve accuracy when the robot remains static. In addition, we can build a robust WiFi SLAM system based on the multi-sensor information.

REFERENCES

[1] M. Liu, C. Pradalier, F. Pomerleau, and R. Siegwart, "The role of homing in visual topological navigation," in Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on, 2012, pp. 567-572.

[2] M. Liu, D. Scaramuzza, C. Pradalier, R. Siegwart, and Q. Chen, "Scene recognition with omnidirectional vision for topological map using lightweight adaptive descriptors," in Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on, 2009, pp. 116-121.

[3] M. Liu and R. Siegwart, "Topological Mapping and Scene Recognition With Lightweight Color Descriptors for an Omnidirectional Camera," Robotics, IEEE Transactions on, vol. 30, pp. 310-324, 2014.

[4] H. Durrant-Whyte and T. Bailey, Simultaneous localization and mapping: part I, Robotics & Automation Magazine, IEEE, vol. 13, pp. 99-110, 2006.

[5] D. Ganti, W. Zhang, and M. Kavehrad, "VLC-based indoor positioning system with tracking capability using Kalman and particle filters," in Consumer Electronics (ICCE), 2014 IEEE International Conference on, 2014, pp. 476-477.

[6] F. Chen, W. S. A. Au, S. Valaee, and T. Zhenhui, "Compressive Sensing Based Positioning Using RSS of WLAN Access Points," in INFOCOM, 2010 Proceedings IEEE, 2010, pp. 1-9.

[7] H. Ren and M.-H. Meng, "Power adaptive localization algorithm for wireless sensor networks using particle filter," Vehicular Technology, IEEE Transactions on, vol. 58, pp. 2498-2508, 2009.

[8] P. Roshan and J. Leary, Wireless Local-Area Network Fundamentals: Cisco Press, 2003.

[9] Y. Chapre, P. Mohapatra, S. Jha, and A. Seneviratne, "Received signal strength indicator and its analysis in a typical WLAN system (short paper)," in Local Computer Networks (LCN), 2013 IEEE 38th Conference on, 2013, pp. 304-307.

[10] T. S. Rappaport, Wireless communications: principles and practice vol. 2: Prentice Hall PTR New Jersey, 1996.

[11] E. C. L. Chan, G. Baciu, and S. C. Mak, "Using Wi-Fi Signal Strength to Localize in Wireless Sensor Networks," in Communications and Mobile Computing, 2009. CMC '09. WRI International Conference on, 2009, pp. 538-542.

[12] Q. Tran, J. W. Tantra, C. H. Foh, A.-H. Tan, K. C. Yow, and D. Qiu, "Wireless indoor positioning system with enhanced nearest neighbors in signal space algorithm," in Vehicular Technology Conference, 2006. VTC-2006 Fall. 2006 IEEE 64th, 2006, pp. 1-5.

[13] S. A. Dudani, "The Distance-Weighted k-Nearest-Neighbor Rule," Systems, Man and Cybernetics, IEEE Transactions on, vol. SMC-6, pp. 325-327, 1976.

[14] K. Weinberger, J. Blitzer, and L. Saul, "Distance metric learning for large margin nearest neighbor classification," Advances in neural information processing systems, vol. 18, p. 1473, 2006.

[15] B. F. D. Hhnel and D. Fox, "Gaussian Processes for Signal Strength-Based Location Estimation," Robotics: Science and Systems II, p. 303, 2007.

[16] B. Ferris, D. Fox, and N. D. Lawrence, "WiFi-SLAM Using Gaussian Process Latent Variable Models," in IJCAI, 2007, pp. 2480-2485.

[17] C. E. Rasmussen, "Gaussian processes in machine learning", in Advanced Lectures on Machine Learning, ed: Springer, 2004, pp. 63-71.

[18] C. E. Rasmussen and H. Nickisch, "Gaussian processes for machine learning (GPML) toolbox," The Journal of Machine Learning Research, vol. 9999, pp. 3011-3015, 2010.

[19] D. Fox, J. Hightower, L. Lin, D. Schulz, and G. Borriello, "Bayesian filtering for location estimation," Pervasive Computing, IEEE, vol. 2, pp. 24-33, 2003.

[20] S. Thrun, W. Burgard, and D. Fox, Probabilistic robotics: MIT press, 2005.

[21] M. Liu, C. Pradalier, and R. Siegwart, "Visual Homing From Scale With an Uncalibrated Omnidirectional Camera," Robotics, IEEE Transactions on, vol. 29, pp. 1353-1365, 2013.

[22] M. Liu, L. Wu, K. J. Qiu, S. H. Li, F. Y. Che, and C. Patrick Yue, "Towards Indoor Localization using Visible Light Communication for Consumer Electronic Devices," in Intelligent Robots and Systems (IROS), 2014 IEEE/RSJ International Conference on, 2014.