2013 27th International Conference on Advanced Information Networking and Applications Workshops

A Comprehensive Study of Bluetooth Fingerprinting-based Algorithms For Localization

Li Zhang
Software College
Northeastern University
Shenyang, China
zhangl@swc.neu.edu.cn

Xiao Liu
Software College
Northeastern University
Shenyang, China
mythliu@gmail.com

Jie Song
Software College
Northeastern University
Shenyang, China
songj@swc.neu.edu.cn

Cathal Gurrin
School of Computing
Dublin City University
Dublin, Ireland
cgurrin@computing.dcu.iee

Zhiliang Zhu
Software College
Northeastern University
Shenyang, China.
zhuzl@swc.neu.edu.cn

***Abstract*—There is an increasing demand for indoor navigation and localization systems along with the increasing popularity of location based services in recent years. According to past researches, Bluetooth is a promising technology for indoor wireless positioning due to its cost-effectiveness and easy-to-deploy feature. This paper studied three typical fingerprinting-based positioning algorithms - kNN, Neural Networks and SVM. According to our analysis and experimental results, the kNN regression method is proven to be a good candidate for localization in real-life application. Comprehensive performance comparisons including accuracy, precision and training time are presented.**

Keywords—Bluetooth indoor positioning, Fingerprinting, kNN, Neural Networks, SVM

I. INTRODUCTION

In recent years, there is a keen demand in indoor navigation and localization systems along with people's increasing need of context awareness [1,2]. There are many real-world applications depending on such systems. For example, many modern buildings, such as huge international airports, national libraries, exhibition halls, malls and etc., are becoming significantly comprehensive, complex and confusing to travel in without a navigation system. In addition, it's crucial for some extreme environments, such as underground working spots, fire scenes and etc., to be equipped with localization systems due to the fact that it would effectively increase the personnel's chance of survival. Moreover, for the commercial building owners, the locations of their customers are important information for making their business decisions.

One of the mainstream outdoor navigation technologies is Global Positioning System (GPS), which is really well established and very accurate for most outdoor positioning situations. However, due to the significant lost of the reception of satellite signals inside concrete modern buildings, it is notoriously difficult to acquire a reasonable positioning result using GPS in indoor environments, therefore alternative technologies are needed. Many different kinds of signals are used for indoor localization. To name a few, one can consider Wi-Fi, Bluetooth, infrared light, ultrasound and etc. Among all these technologies, Wi-Fi and Bluetooth are two most common choices in most indoor localization systems because of their widespread applications and relatively lower costs.

In many applications, localization using Wi-Fi has been shown to be fairly accurate. However, Wi-Fi devices are still not as cheap and widely installed in mobile devices in comparison to Bluetooth ones. In addition, only laptops and newer model smart phones have Wi-Fi component integrated, while almost every cell phone and other mobile devices have Bluetooth installed. On the other hand, according to many researches, Wi-Fi devices consume more energy than Bluetooth devices. Therefore, with good positioning algorithm, Bluetooth technology serves a better candidate for localization purpose.

In this paper, an elaborate discussion on fingerprinting-based Bluetooth localization algorithms is provided. According to previous works, even though there are several studies on each algorithm separately, no previous work has studied their effects on localization or provided comparison between them. In the remaining of this paper, and overview of existing wireless localization algorithms and fingerprinting is given in Section II. In Section III, three typical fingerprinting-based algorithms are described elaborately while Section IV depicts the system architecture. Experiments are presented in Section V whereas the paper is concluded in Section VI.

II. RELATED WORKS

In industry and academia, many efforts have been made both in theoretical researches and real-world implementations of such systems. Many localization theories have been established focusing on different types of signal parameters, localization techniques, positioning algorithms and wireless technologies.

There are three mainstream schemes of wireless location positioning techniques. The first one is Triangulation[1,2], which uses the geometric properties of triangles to calculate the object location. The second one is Fingerprinting[3] (also known as Scene Analysis), which refers to the type of algorithms that collect the signal features (fingerprints) of a scene and then estimate the target location by matching the real-time signal features got online with the prior collected data set and choose the closest one as result. The last one is Proximity[4]. Proximity algorithms provide symbolic relative location information. It relies upon a dense grid of antennas with fixed and well-known positions, and then the location of the target is determined by collocating with the antenna whose signal is the strongest. Each of these schemes has its advantages and disadvantages, and each of them has several derivations. Since the proximity method requires a big amount of beacons due to the fact that the mobile device needs to be close to at least one of them to acquire its location, it is expensive and difficult to deploy. Therefore the triangulation and fingerprinting methods are most widely used instead. Triangulation needs at least three beacons and

978-0-7695-4952-1/13 $26.00 © 2013 IEEE
DOI 10.1109/WAINA.2013.205

accurate estimations of distances between them to compute the location. However, in real-world application, different devices vary very much in their relations between distance and signal parameters; moreover such relations are impossible to determine in complicate environments with obstacles (e.g. consumption or reflection of walls or other objects), which are quite common in application. These defects render the triangulation a bad candidate for indoor localization. Therefore in this paper, fingerprinting is adopted as the primary positioning algorithm.

Fingerprinting, as stated above, refers to the type of algorithms that estimate the location by matching the features of a scene to prior records. In real-world application, RSS-based location fingerprinting is used most commonly. Usually, there are two stages for fingerprinting: offline training stage and run-time stage. During the first stage, a site inspection is performed in the target environment. Reasonable points are selected from the environment, and then their coordinates or labels and received signal strengths from nearby beacons are collected. After the collection of data, whether a model will be trained depends on the chosen algorithm. During the run-time stage, the location is figured out by comparing the currently observed signal strengths with previously collected information using selected algorithm. However, according to past literatures [4-6], a known challenge of location fingerprinting is that the received signal strengths are vulnerable to the influences of diffraction, reflection and obstruction of the objects in the environment or even human bodies. In this paper, solutions will be proposed to improve the existing technique in order to solve this drawback.

At least five fingerprinting-based algorithms have been explored in literature so far [7]: probabilistic methods, k-nearest-neighbor (kNN), neural networks, support vector machine (SVM), and smallest M-vertex polygon (SMP). Although each of them has been exploited, the horizontal comparison between them has not been proposed yet. In this paper, kNN, SVM and neural networks are chosen as preliminary candidates for comparison because of their relatively good outputs from former studies for regression and classification in fingerprinting.

III. Data Collection

A. Architecture

The data collection system consists of two main components. The first one is the Android mobile application for collecting data inside the building. It can collect both the RSSI parameters and the real locations of the points at the same time. The other one is a server application, at which the result is calculated with different candidate algorithms and stored in the database for further analysis. While calculating the result, the server also calculates the needed analysis data such as distance error at the same time. The procedure of a standard experiment procedure and topology using this system is showed in the diagram below (Figure 1).

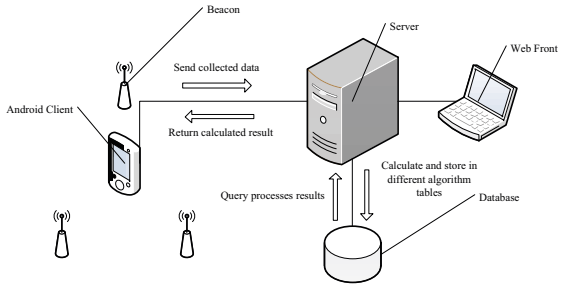

Figure 1 Experiment System Procedures and Structure

As showed in the figure 1, the mobile application collects the scene features and the coordinates of a specific location and transfers them to the server. Then the server calculates the result and returns it to the mobile client. During the calculation, the server calculates needed parameters and stores them in the database. A typical table for one algorithm in the database is as below (Table 1).

TABLE 1 THE PATTERN OF A STANDARD ALGORITHM TABLE

Field	Type	Length	Feature
Id	INT	11	Primary Key
Mac	VARCHAR	255	Not null
Name	VARCHAR	255	Allow null
Time	DATETIME	0	Not null
direction	VARCHAR	255	Not null
X	INT	10	Not null
Y	INT	10	Not null
Xe	DOUBLE	255,20	Allow null
Ye	DOUBLE	255,20	Allow null
Xyd	DOUBLE	255,20	Allow null
RSSI	VARCHAR	255	Allow null

In the table, the fields mac and name store the basic information of a testing device such as MAC address of the Bluetooth adapter and the recognizable name of the device. The field time stores the system time when the calculation is done. The field direction stores the integer number gained from the Android accelerator that represents the direction of the device. The fields x and y store the real location of the device while xe and ye store the error generated by comparing the calculation result with x, y. Whereas xyd represents the distance error of result point and real point. At last, obviously the field RSSI stores the string that made up by the RSSI values and comma between them.

B. Testbed

Although the system is means to be used for the whole building, due to personnel (only myself) and hardware (only 5 beacons) reasons, the experiments of the algorithms are only conducted in one room of the building. However, it is a room big enough to validate the experiments. It is the post graduation laboratory on the second floor of school of computing, DCU. The area is showed below in Figure 2.

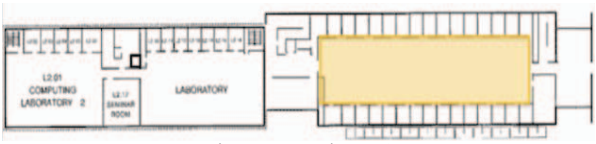

Figure 2 Experiment Area

The lab is about 41 meters long and 10 meters wide. However, the room is not an empty place like showed above. A cubicle wall surrounds the center space of the room, around which there is a pathway about 1 meter wide between the wall and the room border. Inside the wall, there are many cubicles for post-graduation students. Therefore, besides the wall there are many obstacles in this room for Bluetooth signals, which makes it difficult to achieve perfect result but provides more situations for analysis and research. In the room at least 30 locations are used for experiments, they are showed in Figure 3.

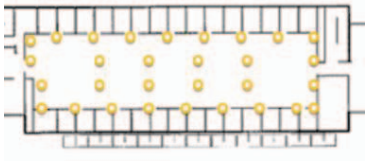

Figure 3 Experiment points

In the experiments, a HTC Desire mobile phone is used for collecting the signals and five other devices are used as beacons. It would be ideal if all the beacons were of same type, however, the experiments are still implementable with different type of devices. The devices used as beacons are as following: a Bluetooth adapter on a desktop PC, an iPad, an iPhone, a Vodafone 845 Android phone and my Macbook Air computer. They are deployed on each corner and the center of the room respectively (Figure 4).

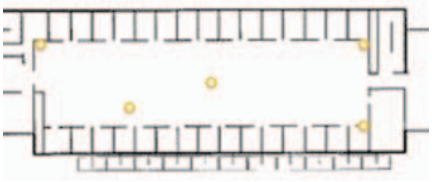

Figure 4 Locations of experiment beacons

One of the beacons is not set exactly at the corner because there is no PC for the Bluetooth adapter to deploy there. It could be seen from the experiments that the southwest corner of the room is kind of becoming confusion. However since all the algorithms are suffered from the same problem, it doesn't affect the comparison between them.

IV. FINGERPRINTING-BASED ALGORITHMS

Fingerprinting localization approach is based on the matching of the online data to the existing database. In this case, the online data with the RSSI values are gathered from each beacon in the building, which can represent the features of a specific location. In general, the localization with fingerprinting can be interpreted as a simple nonlinear equation, in which the value of each parameters are entered and the outputs are the coordinates of the location. The mathematical concept is showed in Figure 5 below.

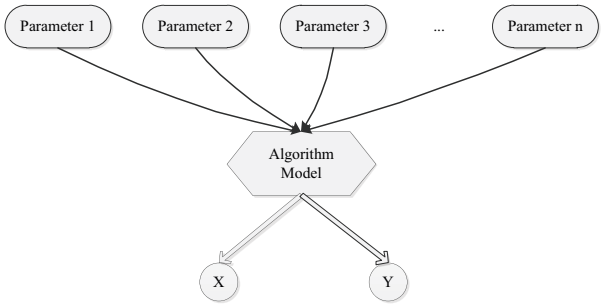

Figure 5 Concept of Localization Problem

In [6], direction is adopted as one of the input parameters. However, in the experiments with different algorithms, the using of direction as a parameter has been proven improper. In these algorithms, the slight change of any input variable has impact on the output value. In the experiments, even though the location remains the same, the output of the algorithms varies greatly along with the change of the direction, while the insensitiveness to direction is preferred. Nevertheless, even though the direction should not be adopted as a parameter, it still should be considered as a factor since the human body does have a significant effect on the RSSI values. Therefore in my experiments and implementation, the RSSI values are taken in four directions at each location in order to inform the system that these four varied data belong to one location.

In the remaining of this section, three fingerprinting-based algorithms are discussed. They are all based on the mathematical model above.

A. Neural Networks

According to the analysis of the problem, neural network algorithm can be suitable for the situation. A neural network, which is short for the term artificial neural network, is a simulation of biological neural networks, which are made up of biological neurons that are connected and functionally related in a nervous system. The real-life examples of neural networks are human and animal brains. An artificial neural network is composed of interconnecting artificial neurons. According to [9], each neuron is weighted and used to compute the output, which represents the strength of the connection. Usually, a learning algorithm automatically determines the weights. The basic concept of one of the most widely used neural network model, logistic function, is showed in Equation 1.

$$\text{Output}= \frac{1}{1+e^{-\text{sum}}} \quad (1)$$

Where sum is the aggregate of weighted inputs.

According to the empirical experiments on neural networks, feedforward neural network [9] is one of the simplest and most effective artificial neural networks. In this network, information travels only from the input nodes through the hidden nodes and then to the output nodes.

In order to improve the performance of a feedforward neural network, usually, a multi-layer perceptron is adopted. A multi-layer perceptron incorporates an input layer with input nodes and an output layer, which are connected with the outside world. In addition to these two layers, one or more hidden layers are used in a multi-layer perceptron. However, these hidden layers are not accessible directly; they are mainly used to represent the connections between the input and the output layers. The concept of a multi-layer feedforward network is showed in Figure 6.

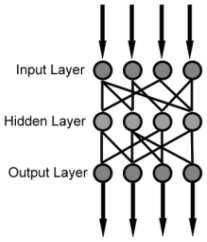

Figure 6 Structure of Multi-layer Feedforward Network

For the learning process of a multi-layer perceptron, back-propagation is one of the most popular learning algorithms. It is composed of two phases: the forward and the backward phase. In the forward phase, the error of the output is calculated. Then in the backward phase, adjustments are made in the neutrons in order to minimize the output error. After many times of this process, the network is trained and suitable for representing the relation between the inputs and outputs. In terms of these features of neural network, it is proper to be used to represent the nonlinear relationship between the RSSI values and the location. In the experiments, multi-layer feedforward networks are used to calculate the locations.

In the following experiments, the neural network method is implemented in two ways: one neural network model with two outputs for the coordinates and one neural network model for each of the coordinates. Their performance are tested and compared.

B. Support Vector Machines

A support vector machine (SVM) is able to analyze existing data and learn the relations between the input data and predicted outputs [10]. It could be used both for classification and regression. A standard SVM is a non-probabilistic binary linear classifier. A set of training data is used to build a model, during which a learning algorithm is needed. Then the model can recognize new inputs and assign them into each category.

Usually, in order to classify the examples, a SVM tries to widen the gap between different categories on a hyper plain if it is not visible currently. In a nonlinear classification (which is the case in this project), a nonlinear kernel function is used to maximize the margin between classes by transforming the space into a higher dimension, where the problem can be solved in a linear way. The basic concept of the function of a kernel is depicted in Figure 7.

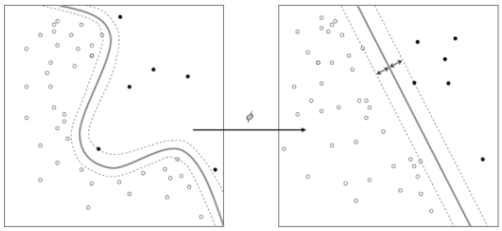

Figure 7 Mapping into Higher Dimension Using Kernel Function

There are three most popular kernel functions: polynomial, Gaussian radial basis function (RBF) and hyperbolic tangent. Since RBF is one of most popular and proven effective kernel function, it is adopted in this system. Its equation is showed below (Equation 2):

$$K(x, x_i) = \exp(-\gamma \|x - x_i\|^2) \quad (2)$$

Where γ represents the width of the function.

The training of SVMs is to minimize structural risk. Two factors need to be controlled in order to manage the generalization ability: empirical risk and fiducial range. Unlike neural networks, in the optimization of a SVM, training error is a constraint while gaining the least fiducial range is the training object. Normally two methods are used in the training process: k-cross validation and grid search. In the experiments, k-cross validation is used. In this method, training data are divided into k equal-quantity sub classes. One of them is used as testing data while the others are used as training data. The process is iterated for k times, during which the MSE is used to estimate the generalization error. At last, the optimized result is chosen.

SVM method could be used both for classification and regression. In this case, since the number of sample points in the building is too big for classification methods, and regression method could estimates the points that are not samples, the regression method is adopted.

C. K-nearest Neighbors

The k-nearest neighbor algorithm (kNN) is a very simple and effective classification method in pattern recognition. The basic procedure of kNN algorithm is as below. First, for an unclassified data example, the distance between it and every record in the database is estimated, which is represented by the sum of MSE of each input parameter. Second, the k instances with nearest distances are selected for calculating the final result.

A training phase is also required for kNN algorithm. Actually, the training phase is to collect abundant data from each situation needs to estimate. For example, in this project, the structure of a record consists of two parts: the RSSI inputs and the supposed outputs. For each location, several records are collected. Then during the run-time phase, the RSSI values of a specific point is compared to the RSSI values in the database, then the outputs of k nearest neighbors are returned for further calculation.

About the first results, there are several ways to put them in use. First, if the differences between each record are significant, there is no need to consider every neighbor, thus only the nearest one should be adopted as the final result. If

there are correlations between the first results, the final result should be generated by all of them. One of the most common methods is to calculate the mean value of k neighbors, which is unreasonable since not all the neighbors are of the same importance. Therefor the other method is emerged, which is to calculate the results with weights. The weights are the distances between results and example. Thus the strength of the connection between each result and the example is considered. In addition, this is a common way of kNN regression.

V. Experiments and Result Analysis

To test and compare the performances of the algorithms, we have done a comparative experiment between these algorithms, which compares the performance of the algorithms based on three different metrics. The remaining of this section discussed the environment, metrics and results of the experiments.

A. Performance Metrics

In the preliminary experiments the general comparison between the three algorithms are conducted in order to find out the advantages, disadvantages and effectiveness of each of them. The comparison is not only conducted between each algorithm but also between different methods in each algorithm. The types and compared parameters are presented in the following.

Usually, to compare a localization algorithm, two metrics are considered the most: accuracy and precision. In this case, accuracy is represented by the average distance error and precision is represented by how many points are within acceptable errors. In addition, in real-life implementation, these two parameters cannot be the only consideration. The deployment issues should also be considered. In the experiments, the training difficulty of each model is estimated. At last, the calculation time is also one of the considerations.

B. Algorithm Parameters and Data Collection

As stated before, multi-layer feedforward networks are used in the experiments. Many empirical experiments have proven that the structure of a network has significant effect on the training time and achieved accuracy of a network model. However, the number of training data also affects them. Normally, along with the increasing of the number of layers and neutrons, the training time tends to be reducing. However, too many neutrons might cause over learning and reduce the accuracy. In addition, within a reasonable amount, the increasing of training data is benefit to the accuracy. Nevertheless, the increasing of data definitely increases training time. Moreover, when it passes the reasonable amount, the training time and accuracy are both exacerbated greatly. The selection of structure and neutron number still remains unclear till now.

In these experiments, after many tests (10 structures), a four-layer network with tangent neutron algorithm is chosen. It has 5 inputs in the first layer, 20 hidden neutrons in the second layer, 10 hidden neutrons in the third layer and output neutrons. Although it is only proven effective for many training data but not all, it is better than the other candidate structures. Since the output of a neural network could be multiple, yet the effect is not known, two kinds of experiments are conducted with 2 outputs and 1 output in order to compare the differences.

There are two parameters need to be set in SVM approach. However, it is generated by analyzing data; this is actually the training phase of SVM. And in this case, SVM regression is used.

Three kinds of kNN algorithms are tested in the experiments. The first is the regression one. The second is to adopt the nearest neighbor while the third one adds in the probability. During the test of the second one, quite a few records are found to have the same RSSI values but different coordinates. Therefor probability estimation is added in order to decide which point is more possible to be the result. In training data, some records might be oddities, and this approach could rule out some of them in theory.

Each of the algorithms uses the same set of training data. 496 records are collected at 31 points. At each point data are collected in four directions for four rounds in order to capture the features of each location. Then a set of 124 records is used for the estimation and comparison.

C. Results and Analysis

In the following, six algorithm models are implemented and estimated: the neural network with two outputs (NN2), the neural networks for x and y coordinates separately (NN1s), the SVM models (SVM), the kNN regression (kNN-r), the kNN nearest (kNN-n) and the kNN nearest with probability (kNNP-n). First, the comparison of the training phase time is taken place, and the result is showed in Table 2.

TABLE 2 THE TRAINING TIME OF DIFFERENT MODELS

NN2	NN1s	SVM	kNN-r	kNN-n	kNNP-n
8m48s	2m11s	2h45m31s	0s	0s	0s

As showed in the table above, the model training time of SVM is obviously the longest. And is a lot longer than the other models. This amount of time is proper only when its accuracy is as excellent, which is showed in the next experiment. The time NN2 took is about 4 times of the combination of two NN1 models. And when the convergence stopped, NN2 only achieves the error 0.076% while NN1 for x and NN1 for y each achieves 0.087% and 0.057% respectively. Although these errors cannot represent the effectiveness but they can represent the effects of trainings. All the kNN algorithms do not have any training time because their model is the training database. Their differences are in the calculation phase.

Secondly, the estimations of the accuracy and precisions are made. The accuracy results are showed in figure 8.

As showed in the figure 8, the NN1 has better accuracy than the one with 2 outputs while SVM is better than both of them. However, as stated before it takes too much time (about 3 hours) to train a model, which renders it a poor candidate. On the other hand, the kNN-r is most accurate compare to others. However, kNN-n and kNNP-n are among the least accurate ones.

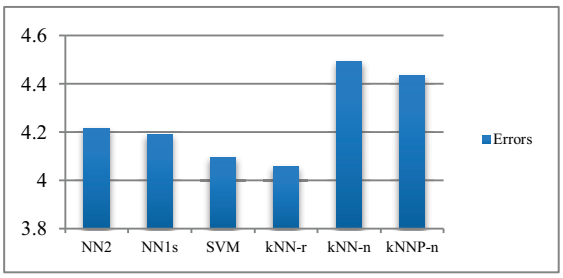

Figure 8 Accuracies of Various Algorithms

The other comparison is about the precision. The results of each point (in 4 directions) are used to calculate the average error of each point. And how many points of 31 points are within certain accuracy are estimated. There are three accuracy levels for this estimation: 1 meter, 2 meters and 3 meters. The result is showed below (Figure 9).

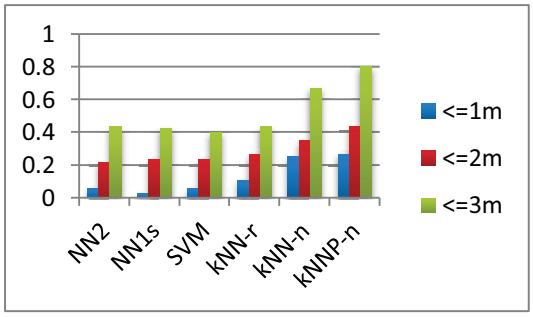

Figure 9 Precisions of Various Algorithms

As showed in the figure, the kNNP-n method has the best precision among all the candidates. Neither of the neural networks and SVM have results with errors below 1 meter. Generally, the kNN nearest methods have better precisions.

From the experiment results above, the following conclusions could be drawn. First, SVM has good accuracy and precision. However, its total training time is too long to be reasonable. Neural network is only average in every aspect. Judging by appearance, kNN regression seems to be the best candidate. However, kNNP-n is more correlated to the points if the features of the point are unique and clear in the training data. This feature of kNN-n could be used to strengthen the representativeness of the training data and kNN model.

VI. Conclusion and Future Work

In this paper, firstly, three algorithms belonged to the fingerprinting positioning approach are compared horizontally. Several experiments were conducted in order to compare the training time, accuracy and precision of each algorithm.

First, in training time, the k nearest neighbor algorithm consumes; and it costs about 2-8 minutes for the neural network algorithm to train a model; whereas the training time of a SVM model is extremely long, which has reached up to about 3 hours. Second, about the accuracy, the SVM is better than the neural networks and kNN regression has the lowest error while the other kNN nearest methods are the worst in accuracy. Finally, according to the results, the kNN methods achieved the best precision among the candidates. Then based on these results, the kNN regression method has been proven generally the best candidate for the algorithm while the kNN nearest method is proven to be able to reflect the correlation between the training data and the current data sample. Therefore in the future a new method to collect data could be proposed, which is designed to increase the representativeness of the training data.

In this paper, only three fingerprinting algorithms have been compared, more comparison between other algorithms can be conducted in the future, such as the Bayesian algorithm, the SMP algorithm and so on. In addition, more fingerprinting algorithms can be discovered or proposed. Since Bluetooth indoor positioning is still a quite new area, more works focusing on improving the performance of localization could be conducted. Moreover, about the improved collecting method, it is only tested on the kNN regression algorithm in this paper, while actually it could be tested on all the classification and regression methods in order to evaluate its effectiveness. And more improvements can be made on the basis of this method.

Acknowledgment

This paper is supported by the National Natural Science Foundation of China under Grant No. 61173028, the Natural Science Foundation of Liaoning Province under Grant No.200102059;the Fundamental Research Funds for the Central Universities N110417002;).

References

[1] J. Hightower, G.Borriello,"Location sensing techniques," University of Washington, Computer Science and Engineering, technical Report,august 8, 2001.

[2] C. Savarese, J. Rabaey, and J. Beutel." Locationing in distributed ad-hoc wireless sensor networks,"In *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 2037-2040, Salt Lake City, UT, May 2001.

[3] K. Kaemarungsi and P. Krishnarmurthy, "Modeling of indoor positioning system based on location fingerprinting," in Proc. IEEE Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'04), Hong Kong, China, Mar. 2004, pp. 1012-1022.

[4] J. Hightower,G. Borriello, "Location systems for ubiquitous comput-ing" *Computer*, vol. 34, no. 8, Aug. 2001.

[5] K. Pahlavan, X. Li, and J. Makela, "Indoor geolocation science and tech- nology," *IEEE Commun. Mag.*, vol. 40, no. 2, pp. 112–118, Feb. 2002.

[6] Marco Altini, Davide Brunelli, Elisabetta Farella. Bluetooth Indoor Localization with Multiple Neural Networks [C] ISWPC [C]. 2010: 295-300.

[7] Hui Liu, H. Darabi, P. Banerjee. Survey of Wireless Indoor Positioning Techniques and Systems [J], Systems, Man and Cybernetics, 2007, 37(6): 1067-1080.

[8] A.K.M. Hossain, Wee-Seng Soh. A Comprehensive Study of Bluetooth Signal Parameters for Localization [C]. PIMRC [C]. 2007: 1-5.

[9] Simon Haykin. Feedforward Neural Networks: An Introduction [EB/OL].http://media.wiley.com/product_data/excerpt/19/04713491/0471349119.pdf.

[10] V. Kecman, *Learning and Soft Computing*. Cambridge, MA: MIT Press, 2001.