Expert Systems with Applications 36 (2009) 7552–7561

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

A soft computing approach to localization in wireless sensor networks

Sukhyun Yun [a], Jaehun Lee [a], Wooyong Chung [a], Euntai Kim [a,*], Soohan Kim [b]

[a] School of Electrical and Electronic Engineering, Yonsei University, C613, Sinchon-dong, Seodaemun-gu, Seoul 120-749, Republic of Korea
[b] Samsung Electronics, Network Systems Division Internet Infra Team, Republic of Korea

ARTICLE INFO

Keywords:
Wireless sensor networks
Localization
Fuzzy logic system
Genetic algorithm
Neural network

ABSTRACT

In this paper, we propose two intelligent localization schemes for wireless sensor networks (WSNs). The two schemes introduced in this paper exhibit range-free localization, which utilize the received signal strength (RSS) from the anchor nodes. Soft computing plays a crucial role in both schemes. In the first scheme, we consider the edge weight of each anchor node separately and combine them to compute the location of sensor nodes. The edge weights are modeled by the fuzzy logic system (FLS) and optimized by the genetic algorithm (GA). In the second scheme, we consider the localization as a single problem and approximate the entire sensor location mapping from the anchor node signals by a neural network (NN). The simulation and experimental results demonstrate the effectiveness of the proposed schemes by comparing them with the previous methods.

1. Introduction

Wireless sensor networks (WSNs) are a framework for the future ubiquitous environment and have many applications for home, health, military, and industry (Akyildiz, Su, Sankarasubramaniam, & Cayirci, 2002). There are two types of nodes in WSNs: anchor nodes and sensor nodes. Anchor nodes possess sufficient energy and accurate information about their position, while sensor nodes do not. In WSN applications, one of the essential problems is the localization of the unknown sensor nodes for the location-based service.

Heretofore, there have been some researches about sensor network localization and they can be divided into two classes: range-based and range-free schemes. The range-based schemes need either node-to-node distances or angles for estimating locations (Bahl & Padmanabhan, 2000; Capkun, Hamdi, & Hubaux, 2001; Cong & Zhuang, 2002; Hightower, Boriello, & Want, 2000; Klukas & Fattouche, 1998; McGuire, Plataniotis, & Venetsanopoulos, 2003; Niculescu & Nath, 2003b; Priyantha, Chakraborty, & Balakrishnan, 2000; Rappaport, Reed, & Woerner, 1996). This information is estimated by using one or more of the following measurements: time of arrival (TOA), time difference of arrival (TDOA) and angle of arrival (AOA). The range-based schemes typically have higher location accuracy than the range-free schemes, but require additional hardware to obtain distances or angles and have weakness in the noisy environments.

In contrast, the range-free schemes do not need the distance or angle information to the sensor nodes from the anchor nodes for their localization (Bulusu, Heidemann, & Estrin, 2000, 2001; He, Huang, Blum, Stankovic, & Abdelzaher, 2003; Kim & Kwon, 2005; Niculescu & Nath, 2003a; Savarese, Rabaey, & Langendoen, 2002; Yun, Lee, Chung, & Kim, 2008). The range-free schemes provide more economic and simpler estimates than the range-based ones, but their results are not as precise as those of the range-based methods. These days, the range-free schemes get more popularity than the range-based methods.

In this paper, two range-free localization schemes based on RSS information are presented. They employ soft computing techniques to overcome the limitations of previous range-free localization methods. In the first scheme, the localization is decomposed into a collection of individual problems in which we compute the proximity of a sensor node to each anchor node. That is, we consider the *edge weight* of each anchor node separately and combine them to compute the location of the sensor nodes. The edge weights are modeled by the fuzzy logic system (FLS) and optimized by genetic algorithm (GA). Contrary to the first scheme, we consider the localization as a single problem in the second scheme and approximate the whole mapping from the anchor node signals to the locations of sensor nodes by a neural network (NN).

The rest of the paper is organized as follows: in Section 2, preliminary fundamentals along with previous localization methods and soft computing techniques are introduced. In Section 3, the proposed individual edge weight approach based on the FLS is presented. In Section 4, the whole approximation approach based on the NN is proposed. Results for the simulations and experiments are shown in Sections 5, and 6 concludes this paper.

* Corresponding author.
E-mail address: etkim@yonsei.ac.kr (E. Kim).

0957-4174/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2008.09.064

2. Preliminary fundamentals

2.1. Previous localization methods

Bulusu et al. proposed a range-free, proximity-based, and coarse-grained localization algorithm in Bulusu et al. (2000). In this method, the anchor nodes broadcast the their position (X_i, Y_i) and each sensor node computes its position as a centroid of the positions of all the connected anchor nodes to itself by

$$(X_{\text{est}}, Y_{\text{est}}) = \left(\frac{X_1 + \cdots + X_n}{N}, \frac{Y_1 + \cdots + Y_n}{N}\right) \tag{1}$$

where $(X_{\text{est}}, Y_{\text{est}})$ represents the estimated position of the sensor node and N is the number of the connected anchor nodes to the sensor node. This scheme is simple and economic but exhibits large amount of error.

Kim and Kwon proposed an improved version of Bulusu et al. (2000) in Kim and Kwon (2005). In this method, anchor nodes are weighted according to their proximity to the sensor nodes, and each sensor node computes its position by

$$(X_{\text{est}}, Y_{\text{est}}) = \left(\frac{w_1 \cdot X_1 + \cdots + w_n \cdot X_n}{\sum_{i=1}^{n} w_i}, \frac{w_1 \cdot Y_1 + \cdots + w_n \cdot Y_n}{\sum_{i=1}^{n} w_i}\right) \tag{2}$$

This method has the weakness that the choice of the weights (w_1, $w_2, \ldots, w_n$), is very heuristic, and the performance highly depends on the design of the weights.

2.2. Soft computing

Soft computing is an emerging problem solving technology and, in particular, it is appropriate for uncertain and nonlinear problems. Soft computing can be considered as collaboration between the FLS, NN, GA and other AI (artificial intelligence) related fields. In this paper, soft computing techniques play a crucial role in the proposed localization methods.

2.2.1. Fuzzy logic system

The fuzzy logic system (FLS) (Sugeno & Kang, 1986; Takagi & Sugeno, 1985) is an inference system which mimics the human thinking and its basic configuration consists of a fuzzifier, some fuzzy IF–THEN rules, a fuzzy inference engine and a defuzzifier, as shown in Fig. 1.

A fuzzy rule is written as the following statement:

$$R^i : \text{IF } x_1 \text{ is } A_1^i \text{ and } x_2 \text{ is } A_2^i \text{ and } \cdots x_n \text{ is} A_n^i \text{ THEN } y \text{ is} y^i \tag{3}$$

where R^i ($i = 1,2,\cdots,l$) denotes the ith implication; x_j ($j = 1,2,\cdots,n$) is input variables of the FLS; y_i is a singleton; A_j^i is the fuzzy membership functions, which can represent the uncertainty in the reasoning. When we use the product inference, center-average and singleton fuzzifier, the output of the fuzzy system for an input $X = (x_1, x_2, \cdots, x_n)$ can be expressed as

$$y = \frac{\sum_{i=1}^{N} \alpha_i y_i}{\sum_{i=1}^{N} \alpha_i} \tag{4}$$

where α_i implies the overall truth value of the premise of the ith implication, and are computed as

$$\alpha_i = \prod_{j=1}^{n} A_j^i(x_i). \tag{5}$$

2.2.2. Genetic algorithm

Genetic algorithms (GAs) are numerical optimization algorithms inspired from genetics and have been applied to a wide range of problems (Coley, 1999; Davis, 1991). GA typically maintains a population of individuals that represents a set of candidate solutions for the considered problem. The goodness of each candidate solution is evaluated based on its fitness to the objectives, and the population evolves by selection, crossover, and mutation. In the selection process, some individuals are selected to be copied into a tentative next population. The number of copies of each individual in the next generation is proportional to its relative fitness value. The promising individuals are therefore more likely to be selected for the next generation. The selected individuals are varied to search for a global optimal solution using mutation and crossover. GA is simple yet provides an adaptive and robust optimization methodology (Davis, 1991).

2.2.3. Neural network

Neural networks (NNs) imitate the human brain to perform intelligent tasks (Hagan, Demuth, & Beale, 1996). They can represent complicated relationships between input and output variables, and acquire knowledge about these relationships directly from the data. The type of NN used in this paper is the multilayer perceptron (MLP) and consists of an input layer, a nonlinear hidden layer and a linear output layer, as shown in Fig. 2.

When the numbers of neuron units of input, hidden and output layers are denoted by n, p, and m, respectively, the NN yields the output

$$P(\omega_k|X) \approx y_k = f\left\{\sum_{j=1}^{p} w_{jk} f\left(\sum_{i=1}^{n} v_{ij} x_i\right)\right\} \quad \text{for } k = 1, 2, \cdots m \tag{6}$$

where v_{ij}s are input-to-hidden layer weights; w_{jk}s are hidden-to-output weights; and f is an activation function.

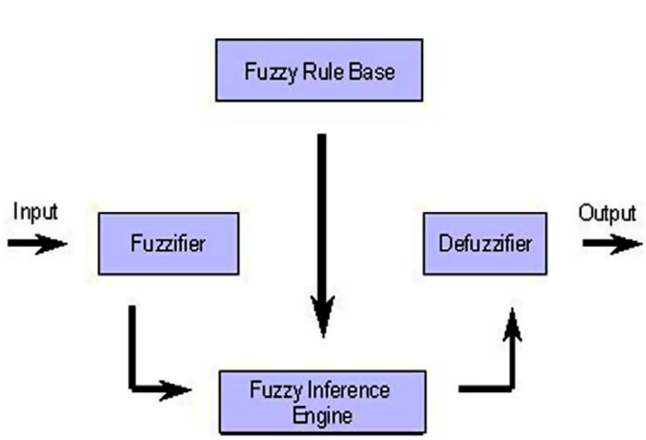

Fig. 1. An example of the fuzzy logic system (FLS).

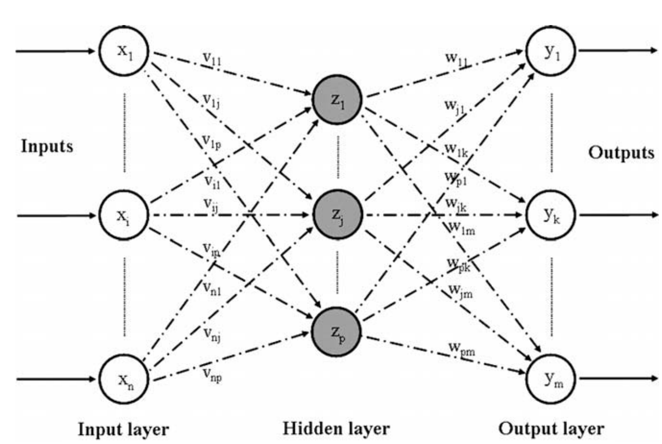

Fig. 2. An example of the MLP neural networks.

3. Individual localization by fuzzy logic and genetic algorithm

3.1. Environment setup and localization scenario

There are a set of anchor nodes and a set of sensor nodes in a WSN. A fixed number of anchor nodes are placed with the regions of coverage overlapped and serve as reference points, broadcasting periodic beacon signals. The sensor nodes are distributed randomly in the sensing field and receive messages from anchor nodes. The main responsibility of the anchor nodes is to send out beacon signals to help the sensor nodes to locate themselves. Each sensor node listens for a fixed time period and collects the RSS information of all beacon signals from adjacent anchor nodes. In this environment, it is assumed that

(1) The anchor nodes know their positions through GPS or by other means such as pre-configuration.
(2) The radio propagation is perfectly spherical and the transmission ranges for all radios are identical.

In this environment, each sensor node finds its position by the following procedures:

STEP1: Find the adjacent anchor nodes using connectivity.
STEP2: Collect the IDs and positions of anchor nodes and measure their RSSs.
STEP3: Calculate the edge weight of each anchor node. The edge weight should already be modeled by FLS. The FLS modeling is explained in the subsequent subsections.
STEP4: Use the edge weights of all anchor nodes to determine the position of sensor node by (2). More specifically, when the positions of the anchor nodes are $(X_1,Y_1),(X_2,Y_2),\ldots,(X_k,Y_k)$, the position of the sensor node is estimated as

$$(X_{\mathrm{est}}, Y_{\mathrm{est}}) = \left(\frac{w_1 \cdot X_1 + \cdots + w_k \cdot X_k}{\sum_{i=1}^{k} w_i}, \frac{w_1 \cdot Y_k + \cdots + w_k \cdot Y_k}{\sum_{i=1}^{k} w_i} \right) \tag{7}$$

where k is the number of adjacent anchor nodes.

3.2. Fuzzy model building for edge weight of anchor nodes

The RSS is simply the signal strength from the anchor node, not a distance measurement. But it can provide a cue for the distance from the anchor node to the sensor node, not precise though. To overcome the uncertainty of the RSS and the nonlinearity between the RSS and the distance, we use FLS and model the relationship between the weight of an anchor node and its RSS. The fuzzy model is the following form:

$$R^l : \text{IF } x \text{ is } A^l \text{ THEN } y \text{ is } B^l \tag{8}$$

The input variable x is the RSS from anchor node and takes a value in the interval $[0, \mathrm{RSS}_{\max}]$, where $\mathrm{RSS}_{\max}$ is the maximum RSS value. We decompose the input space of RSSI into five trapezoidal membership functions: VL, L, M, H, and VH, as shown in Fig. 3.

The output variable y is the edge weight of each anchor node for a given sensor node and takes a value in the interval $[0, \omega_{\max}]$, where $\omega_{\max}$ is the maximum weight. As in x, we decompose the output space of the weight into five membership functions: VL, L, M, H, and VH, as shown in Fig. 4.

Now, we consider the rules for FLS. If an anchor node emits high powered signal, the anchor node is likely to be close to a given sensor node and it should have a high weight. Conversely, if an anchor node is connected to the sensor node but emits low powered signal, the anchor node is likely to be far from the given sensor node and should have a low weight. Consequently, we use the fuzzy rule bases and further tune the membership functions (see Table 1).

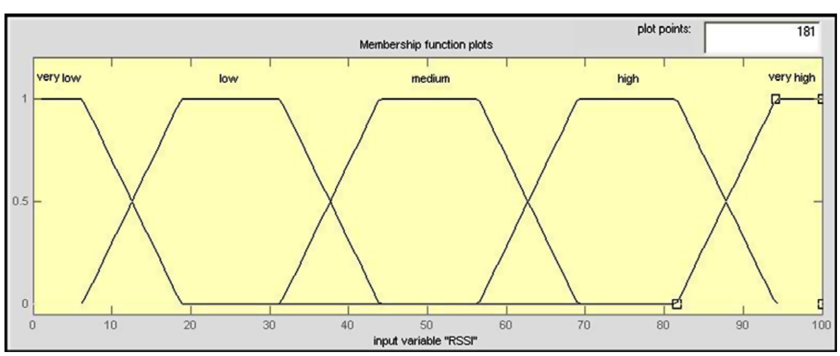

Fig. 3. Fuzzy membership function of RSS.

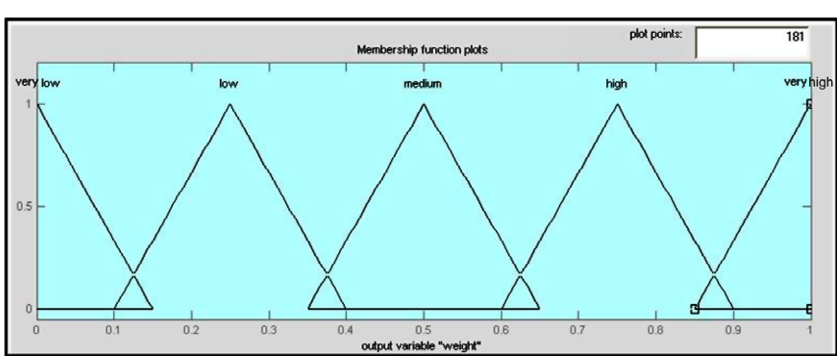

Fig. 4. Fuzzy membership function of weight.

Table 1
Rule base about edge weight

Rule	IF: RSS is ~	THEN: weight is ~
Rule 1	Very low	Very low
Rule 2	Low	Low
Rule 3	Medium	Medium
Rule 4	High	High
Rule 5	Very high	Very high

Table 2
Evolution parameters of GA

Parameter	Value
Generation number	200
Population size	100
Crossover rate	0.6
Mutation rate	0.05

Since FLS lacks the training method, we use GA to train and optimize FLS. For simplicity, we optimize only the membership functions of x, while leaving the membership functions of y and rule base unchanged. We use the standard genetic encoding and use the standard crossover and mutation. Table 2 shows the optimization parameters of GA.

We make 10 independent runs and Fig. 5 shows the optimization result of one specific run.

4. Overall localization using NN

4.1. Localization scenario and procedure

We use the same environment setup and localization scenario as in the previous section. In this environment, each sensor node finds its position using the following procedure:

STEP1: Find the adjacent anchor nodes using connectivity.
STEP2: Collect the IDs and positions of anchor nodes and measure their RSSs.
STEP3: Calculate the location of a sensor node using an NN based on the RSSs from the anchor nodes. More specifically, when the RSSs of the anchor nodes are represented by R, the position of the sensor node is estimated as

$$(X_{\text{est}}, Y_{\text{est}}) = NN_k(R) \tag{9}$$

where k is the number of adjacent anchor nodes. The NN should already be trained and the details of the NN would be explained in the subsequent subsections.

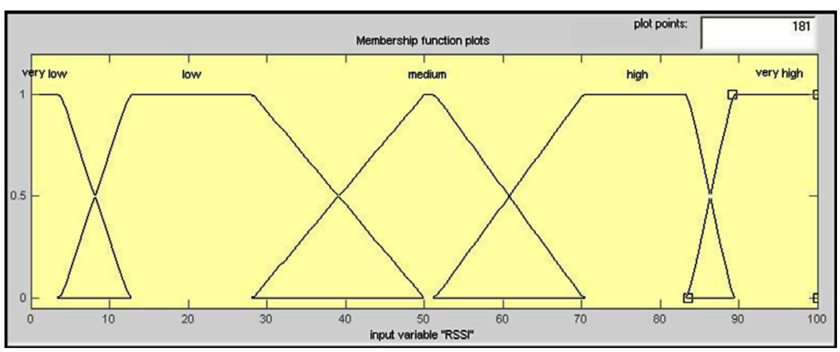

Fig. 5. Optimized fuzzy membership function of RSS using GA.

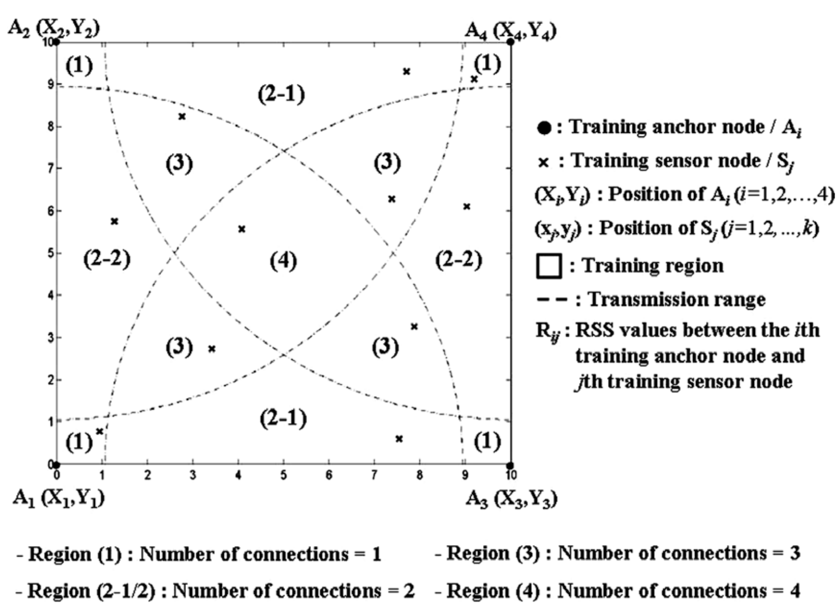

Fig. 6. The number of connections according to the position of sensor node.

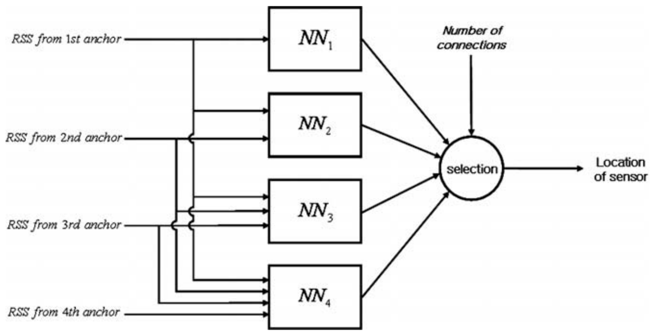

Fig. 7. Structure of the NN ensemble for the overall localization.

4.2. Neural network for overall localization

In this overall localization scheme, each sensor node uses all the RSSs from the connected anchor nodes as an input to an NN. We place the anchor nodes as shown in Fig. 6 such that the maximum number of connected anchor nodes is four and we design NNs with inputs up to four.

As shown in Fig. 6, the number of connected anchor nodes depends on the location of the sensor nodes and based on the number of the connected anchor nodes, the sensor node should use the most suitable NN. Fig. 7 shows the structure of the NN ensemble for the overall localization.

The NN ensemble receives the RSSs from the connected anchor nodes as an input and outputs the estimated location of the sensor node. When we train the NN ensemble, we use the popular back-propagation algorithm (Hagan et al., 1996). Depending on the regions shown in Fig. 6, we use the different training sets. More specifically, the following training sets are used.

4.2.1. Region (1): number of connections = 1

When the number of connections is one, we output the location of the anchor node as the estimated location for the sensor node.

$$\text{Input data} = [R_{11}, R_{12}, \cdots, R_{1k}]$$
$$\text{Output data} = \begin{bmatrix} x_1, x_2, \ldots, x_k \\ y_1, y_2, \ldots, y_k \end{bmatrix} = \begin{bmatrix} X_1 \\ Y_1 \end{bmatrix} \tag{10}$$

4.2.2. Region (2): number of connections = 2

First, for region (2-1) (# = 2, $Y_1 - Y_2 = 0$), we train the NN_2 using

$$\text{Input data} = \begin{bmatrix} R_{11}, R_{12}, \ldots, R_{1k} \\ R_{21}, R_{22}, \ldots, R_{2k} \end{bmatrix}$$
$$\text{Output data} = \begin{bmatrix} x_1, x_2, \ldots, x_k \\ y_1, y_2, \ldots, y_k \end{bmatrix} \tag{11}$$

where

$$X_1 < x_j < X_2$$
$$Y_1 = y_j = Y_2, \text{ for } j = 1, 2, \cdots, k$$

For the sensor nodes in the region (2-2) (# = 2, $X_1 - X_2 = 0$), we use the same NN_2, but we have to swap X and Y information.

4.2.3. Region (3): number of connections = 3

We train the NN_3 using

$$\text{Input data} = \begin{bmatrix} R_{11}, R_{12}, 0, R_{14}, \ldots, R_{1k} \\ 0, R_{22}, R_{23}, R_{24}, \ldots, 0 \\ R_{31}, R_{32}, R_{33}, 0, \ldots, R_{3k} \\ R_{41}, 0, R_{43}, R_{44}, \ldots, R_{4k} \end{bmatrix}$$

$$\text{Output data} = \begin{bmatrix} x_1, x_2, \cdots, x_k \\ y_1, y_2, \cdots, y_k \end{bmatrix} \tag{12}$$

where

$$(X_1 = X_2) < x_j < (X_3 = X_4)$$
$$(Y_1 = Y_3) < y_j < (Y_2 = Y_4), \text{ for } j = 1, 2, \cdots, k$$

In (12), "0" input data corresponds to no connection. The location of "0" is determined from the relation of other three training anchor nodes.

```
procedure Overall Localization ()
for i = 1 : Num
    if Count = 1
        (x_est, y_est) = NN_1(R);
        (X_est, Y_est) = (x_est, y_est);
    elseif Count = 2
        (x_est, y_est) = NN_2(R);
        (X_est, Y_est) = (x_est, y_est) + (X_1, Y_1);
    elseif Count = 3
        (x_est, y_est) = NN_3(R);
        (X_est, Y_est) = (x_est, y_est) + (X_1, Y_1);
    else
        (x_est, y_est) = NN_4(R);
        (X_est, Y_est) = (x_est, y_est) + (X_1, Y_1);
    end
End
```

Fig. 8. The pseudo code of location estimation using trained NN. NN_i: a kind of trained NN as described in Section 4.2, $i = 1, 2, \cdots, 4$; Count: number of the connected anchor nodes; Num: Number of sensor nodes; R: RSS information matrix; (X_{est}, Y_{est}): estimated position of the sensor node in the whole region; (x_{est}, y_{est}): estimated position of the sensor node in the training region; (X_i, Y_i): position of the training anchor node, for $i = 1, 2, \cdots, N$, where N is the number of training anchor nodes.

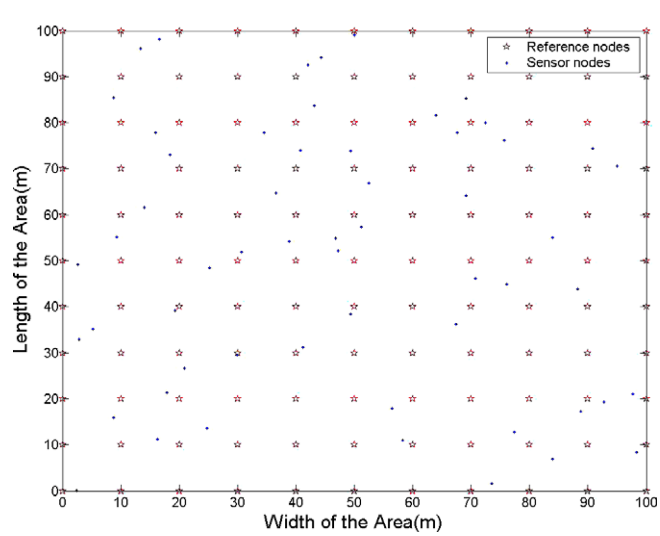

Fig. 9. Distribution of all nodes for the simulation.

4.2.4. Region (4): number of connections = 4

We train the NN_4 using

$$\text{Input data} = \begin{bmatrix} R_{11}, R_{12}, \ldots, R_{1k} \\ R_{21}, R_{22}, \ldots, R_{2k} \\ R_{31}, R_{32}, \ldots, R_{3k} \\ R_{41}, R_{42}, \ldots, R_{4k} \end{bmatrix} \qquad (13)$$
$$\text{Output data} = \begin{bmatrix} x_1, x_2, \ldots, x_k \\ y_1, y_2, \ldots, y_k \end{bmatrix}$$

where

$$(X_1 = X_2) < x_j < (X_3 = X_4)$$
$$(Y_1 = Y_3) < y_j < (Y_2 = Y_4), \text{ for } j = 1, 2, \cdots, k$$

All the NNs except NN_1 consist of three layers, as shown in Fig. 2. We use 500 hidden nodes and train them by utilizing the popular backpropagation algorithm (Hagan et al., 1996). The transfer function for the first layer is log-sigmoid and the transfer function for the second layer is linear.

4.3. Location recovery from NN

In order to recover the location of sensor nodes from the outputs of NNs that output only the relative location inside the training region, we add the output of NN to the location of the first anchor node (X_1, Y_1). More specifically, refer to Fig. 8.

5. Simulation and experimentation

5.1. Simulation

5.1.1. Setup and performance measure

First, we conduct the computer simulation to show the effectiveness of the proposed intelligent localization methods. The algorithms are coded in Matlab. In this simulation, we consider a $100 \times 100\ \text{m}^2$ region. One hundred and twenty one anchor nodes are placed regularly with each one 10 m apart, and 60 sensor nodes are placed randomly as shown in Fig. 9.

The transmission range of all anchor nodes is assumed 8.94 m. A sensor node can communicate with adjacent anchor nodes if its distance from the anchor node is smaller than the transmission range. For the simulation, we used the following RSS model which also takes into account noise

$$R_{ij} = (kd_{ij}^{-\alpha}) + (\text{AWGN} * \text{Var.}) \qquad (14)$$

where R_{ij} is the RSS value between the ith sensor node and the jth adjacent anchor node, k is a constant which takes into account car-

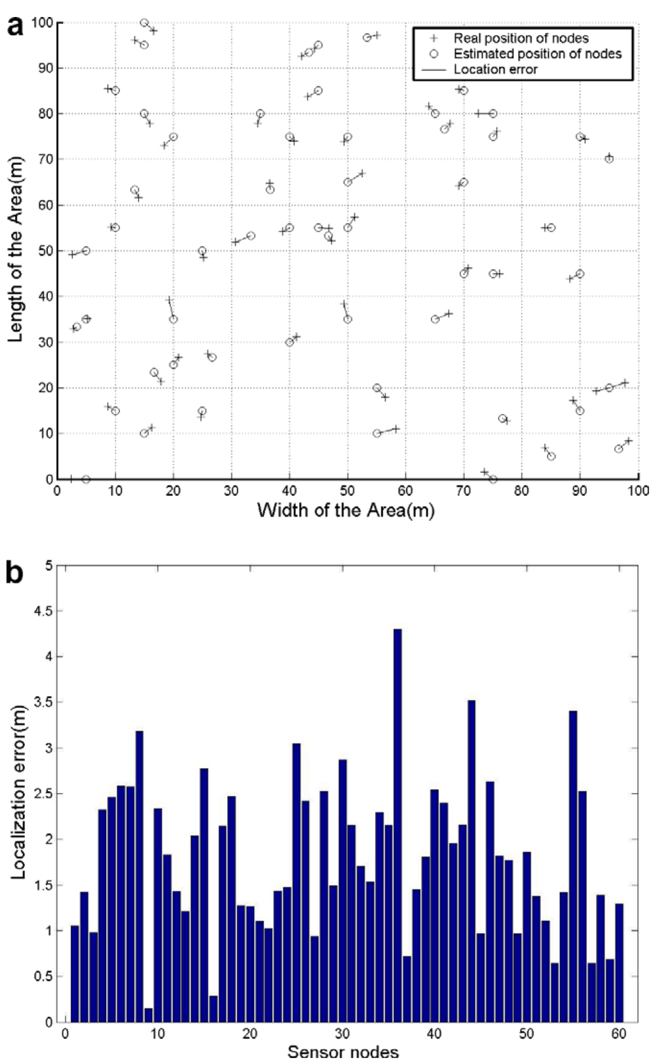

Fig. 10. Simulation result of the simple centroid method (Bulusu et al., 2000): (a) result of location estimation; (b) error result.

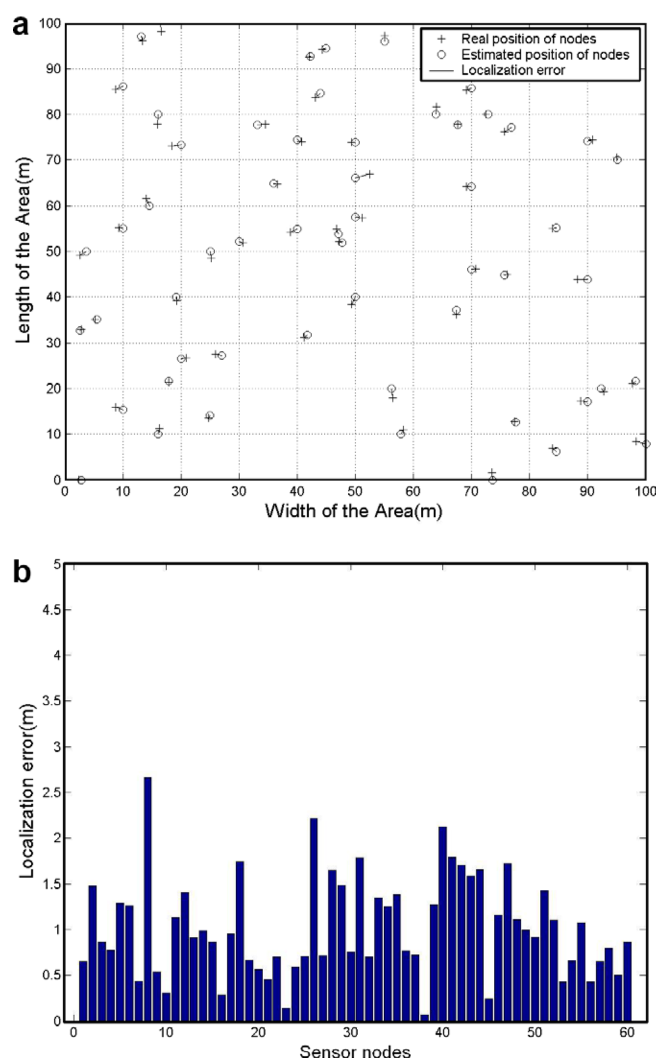

Fig. 11. Simulation result of the weighted centroid method (Kim & Kwon, 2005): (a) result of location estimation; (b) error result.

rier frequency and transmitted power, d_{ij} is the distance between the ith sensor node and the jth adjacent anchor node and α is the attenuation exponent. Here, we use $k = 50$ and $\alpha = -1$. Usually as in Bergamo and Mazzini (2002), *AWGN* (additive white gaussian noise) is not taken into account but it is contemplated here to mimic more realistic environment.

To evaluate the proposed schemes, we use the following two performance indices:

(1) *Location error:* The distance between the estimated position and the actual position of sensor node,

$$\text{Location error} = \sqrt{(X_{\text{est}} - X_a)^2 + (Y_{\text{est}} - Y_a)^2} \tag{15}$$

(2) *Average location error:* The average distance between the estimated position and the actual position of all sensor nodes,

$$\text{Average location error} = \frac{\sum \sqrt{(X_{\text{est}} - X_a)^2 + (Y_{\text{est}} - Y_a)^2}}{\text{number of sensor nodes}} \tag{16}$$

5.1.2. Results

We implement the four localization methods for comparison: the simple centroid method (Bulusu et al., 2000), the weighted centroid method (Kim & Kwon, 2005), and the two methods proposed in this paper. Figs. 10–13 show the results of Bulusu et al. (2000) and Kim and Kwon (2005), the individual and overall methods, respectively.

The crosses denote the actual position of sensor nodes and the circles denote the estimated position of sensor nodes in the first sub-figure of Figs. 10–13. In the second sub-figure of Figs. 10–13, the solid bars denote the location estimation error for each sensor node. The simulation results are summarized in Table 3.

It can be seen that the proposed two methods outperform the previous two methods Bulusu et al. (2000) and Kim and Kwon (2005). Between the proposed two methods, the overall approach based on NN is better than the individual approach based on the FLS.

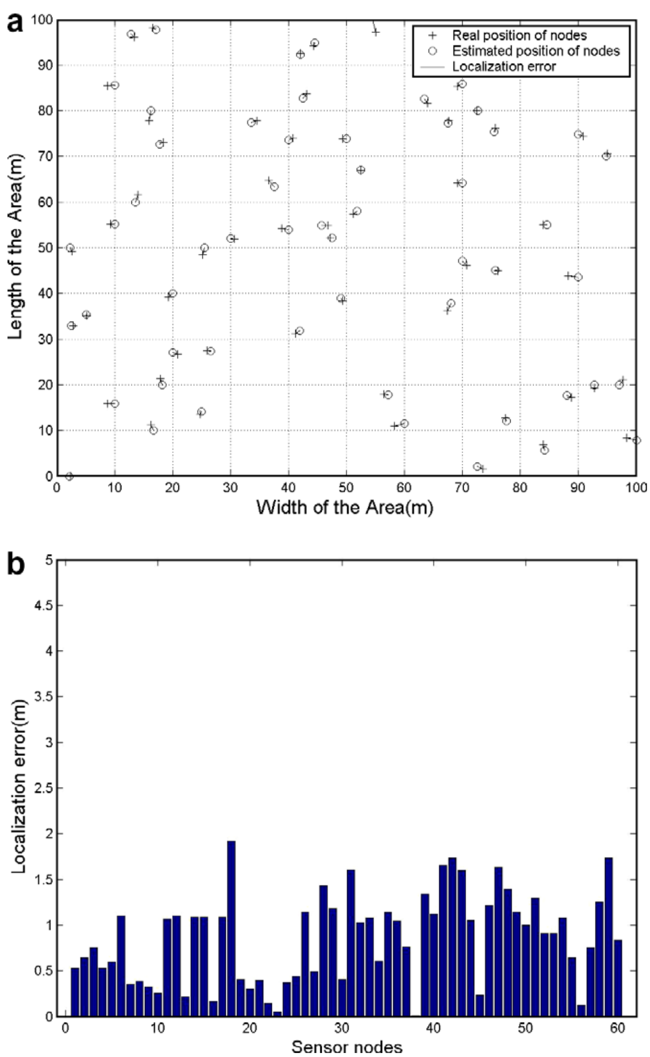

Fig. 12. Simulation result of the proposed individual method by FLS and GA: (a) result of location estimation; (b) error result.

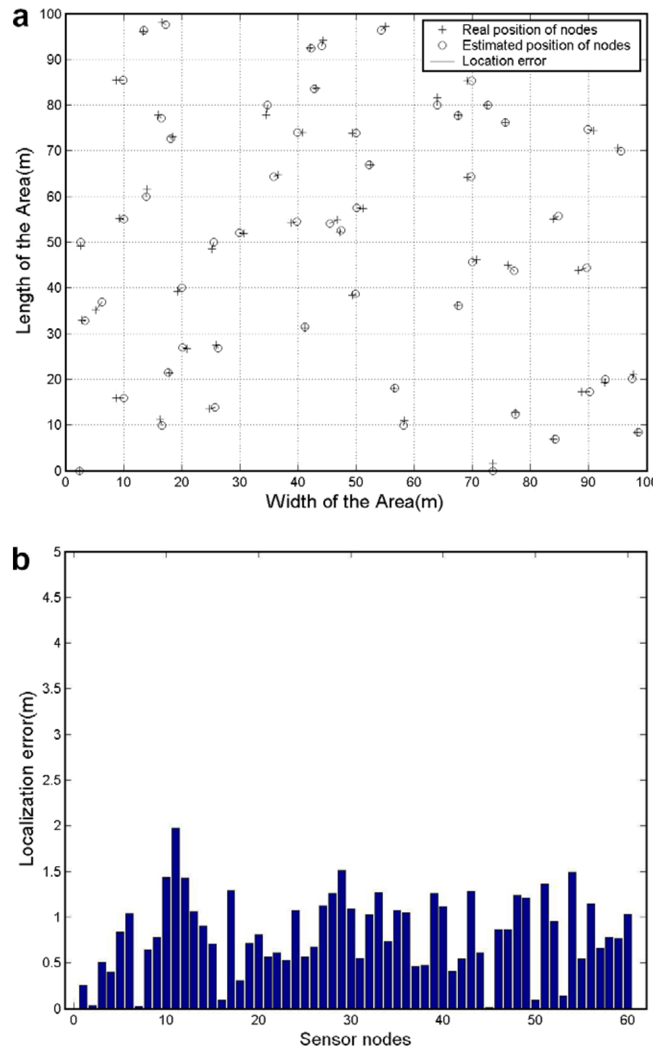

Fig. 13. Simulation result of the proposed overall method by NN: (a) result of location estimation; (b) error result.

Table 3
Comparison of simulation results of four different methods

Methods	Max. error (m)	Min. error (m)	Avg. error (m)
Simple centroid (Bulusu et al., 2000)	4.3485	0.1438	1.7519
Weighted centroid (Kim & Kwon, 2005)	2.6787	0.0826	1.0550
Fuzzy + GA	1.9201	0.0118	0.7802
NN	1.9718	0.0079	0.7119

Fig. 14. Photo of the MICAz with standard antenna.

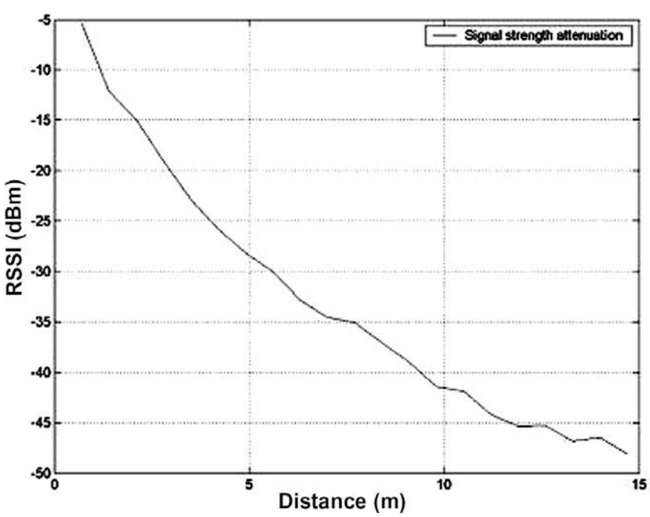

Fig. 15. Signal strength attenuation according to distance.

5.2. Experimentation

5.2.1. Experimentation setup

The results of the conducted experiments show the effectiveness of the proposed intelligent localization methods. In the exper-

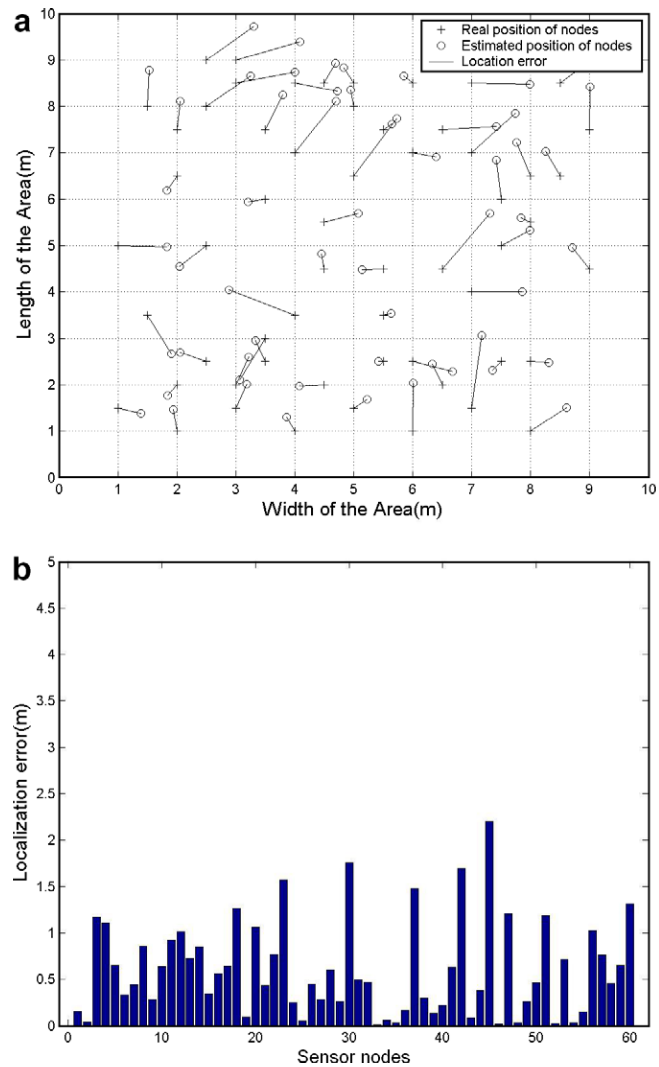

Fig. 17. Experimental result of the individual localization by FLS and GA: (a) result of location estimation; (b) error result.

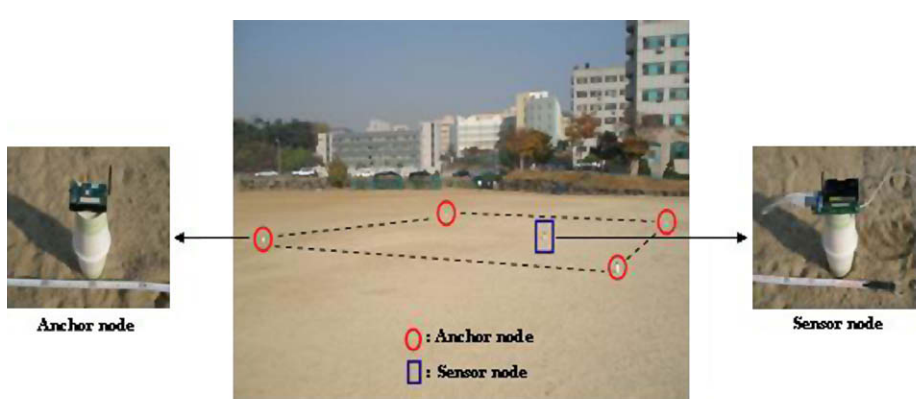

Fig. 16. Experimental environment.

imentation, the MICAz (MPR 2400) with standard antenna is used as both anchor and sensor nodes, as shown in Fig. 14. They are widely used in other researches (http://www.xbow.com/Products/productsdetails.aspx?sid=164).

MICAz operates using a TinyOS (http://today.cs.berkeley.edu/tos) which covers all the low level communication layers such as routing. Before setting up the experiment, we first measure the strength attenuation of the RSS in an ordinary environment. Shown in Fig. 15 shows the signal strength attenuation according to the distance between the nodes.

The communication range is about 15 m and RSS shows a similar result of (14). For simplicity, four MICAzs are used as anchor nodes at the four corners of a $10 \times 10\ m^2$ square in a playground, and one MICAz is used as a sensor node, as shown in Fig. 16. This square is divided into four hundreds of $0.5 \times 0.5\ m^2$ grids, and the RSSs are gathered at 441 grid intersections.

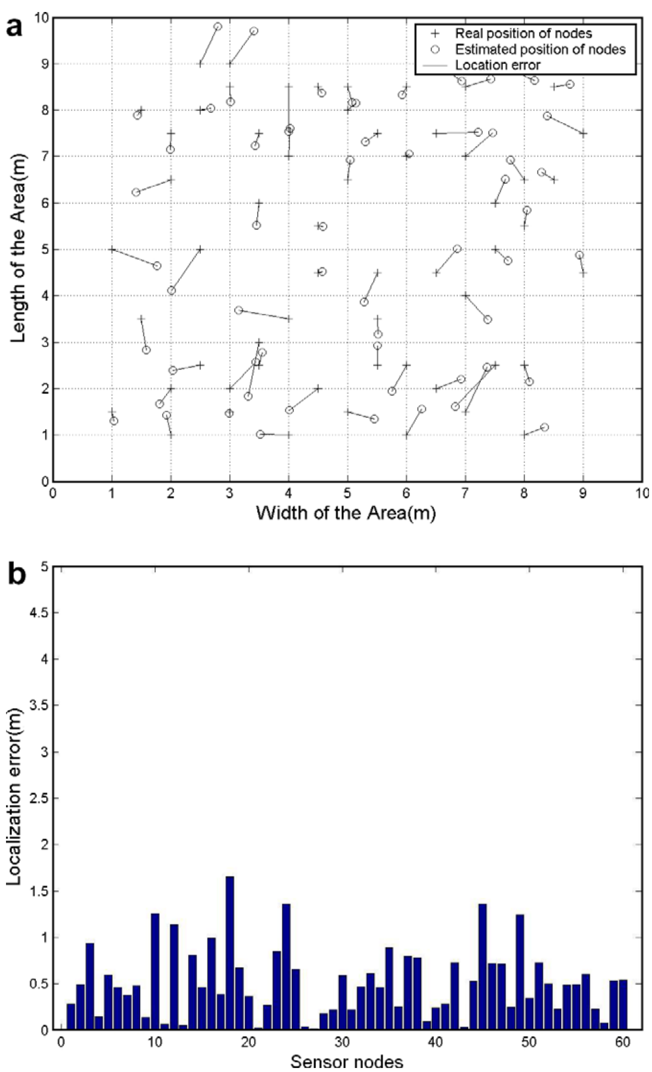

Fig. 18. Experimental result of the overall localization by NN: (a) result of location estimation; (b) error result.

Table 4
Comparison of the experimental results

Methods	Max. error (m)	Min. error (m)	Avg. error (m)
Experiment (fuzzy + GA)	2.2016	0.0071	0.8025
Experiment (NN)	1.6565	0.0157	0.6938

Table 5
Comparison of simulation and experimental results

Methods	Max. error (m)	Min. error (m)	Avg. error (m)
Simulation (fuzzy + GA)	1.9201	0.0118	0.7802
Simulation (NN)	1.9718	0.0079	0.7119
Experiment (fuzzy + GA)	2.2016	0.0071	0.8025
Experiment (NN)	1.6565	0.0157	0.6938

Figs. 17 and 18 show the location estimation and error results for the individual and overall localization, respectively.

The experimentation results are summarized in Table 4.

It can be seen that the overall localization based on NN is slightly better than the individual approach based on FLS and GA. In Table 5, the experimentation results are compared with the simulation result.

For the overall localization based on NN, the experimentation shows almost the same result as the simulation. For the individual localization based on FLS and GA, there is a slight discrepancy between the experiment and simulation. The reason might be that for the individual localization by FLS and GA, we use only up to four anchor nodes in the experimentation but can use more than four in the simulation. For the overall localization by NN, however, the algorithm uses only up to four neighboring anchor nodes and makes no difference in the experiment and simulation.

6. Conclusions

In this paper, two intelligent range-free localization schemes for wireless sensor networks (WSNs) are presented. In our proposed methods, the sensor nodes do not need any complicated hardware to obtain the distance or TOA/AOA information. The sensor nodes can estimate their positions with only RSS information between itself and its neighbor anchor nodes. Soft computing techniques play the crucial role in our proposed schemes. In the first scheme, we consider the edge weight of each anchor node which is the neighbor of the sensor nodes separately and combine these edge weights to compute the location of sensor nodes. The edge weights are modeled by the fuzzy logic system (FLS) and optimized by the genetic algorithm (GA). In the second scheme, we consider the localization as a single problem and uniformly approximate the entire mapping from the anchor node signals to the locations of sensor nodes by a neural network (NN).

The proposed methods are applied to both computer simulation and outdoor experiments. In both situations, the proposed methods show the improved performance compared to the existing methods. The proposed localization methods are kinds of the distributed algorithms, i.e. each sensor node estimates its position independently. Therefore, the proposed methods can be applied to applications of the large scale network. The future work includes adapting the proposed localization methods to the noisy indoor environment and reduces the time requirement to optimize the FLS and to train the NN.

Acknowledgment

This work was supported by the Ministry of Commerce, Industry and Energy of Korea (HISP).

References

Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002). A survey on sensor networks. *IEEE Communications Magazine, 40*(August), 102–114.

Bahl, P., & Padmanabhan, V. N. (2000). RADAR: An in-building RF-based user location and tracking system. In *Proceedings of the IEEE INFOCOM 2000, Tel Aviv, Israel, March*.

Bergamo, P., & Mazzini, G. (2002). Localization in sensor network with fading and mobility. *IEEE Transaction on Communication*. June.

Bulusu, N., Heidemann, J., & Estrin, D. (2000). GPS-less low cost outdoor localization for very small devices. *IEEE Personal Communications Magazine, 7*(5), 28–34. October.

Bulusu, N., Heidemann, J., & Estrin, D. (2001). Adaptive beacon placement. In *Proceedings of IEEE international conference on distributed computing systems (ICDCS), Phoenix, AZ, April* (pp. 489–498).

Capkun, S., Hamdi, M., & Hubaux, J. (2001). GPS-free positioning in mobile ad-hoc networks. In *Proceedings of the 34th Hawaii international conference on system sciences*.

Coley, D. (1999). *An introduction to genetic algorithms for scientists and engineers*. World Scientific.

Cong, L., & Zhuang, W. (2002). Hybrid TDOA/AOA mobile user location for wideband CDMA cellular systems. *IEEE Transactions on Wireless Communications, 1*(July), 439–447.

Davis, L. (1991). *Handbook of genetic algorithms*. Van Nostrand Reinhold.

Hagan, M. T., Demuth, H. B., & Beale, M. (1996). *Neural network design*. PWS Publishing Company.

He, T., Huang, C., Blum, B., Stankovic, J., & Abdelzaher, T. (2003). Range-free localization schemes for large scale sensor networks. *MOBICOM*.

Hightower, J., Boriello, G., & Want, R. (2000). *SpotON: An indoor 3D location sensing technology based on RF signal strength*. Technical report UW CSE 00-02-02, University of Washington, February.

Kim, S. Y., & Kwon, O. H. (2005). Location estimation based on edge weights in wireless sensor networks. *Journal of Korea Information and Communication Society, 30*(10A) (Korean).

Klukas, R., & Fattouche, M. (1998). Line-of-sight angle of arrival estimation in the outdoor multipath environment. *IEEE Transactions on Vehicular Technology, 47*(Feb.), 342–351.

McGuire, M., Plataniotis, K. N., & Venetsanopoulos, A. N. (2003). Location of mobile terminals using time measurements and survey points. *IEEE Transactions on Vehicular Technology, 52*(July), 999–1011.

Niculescu, D., & Nath, B. (2003a). DV based positioning in ad hoc networks. *Journal of Telecommunication Systems*.

Niculescu, D., & Nath, B. (2003b). Ad hoc positioning system (APS) using AOA. *INFOCOM*.

Priyantha, N. B., Chakraborty, A., & Balakrishnan, H. (2000). The cricket location-support system. In *Proceedings of ACM international conference on mobile computing networking (MOBICOM), Boston, MA, August* (pp. 32–43).

Rappaport, T. S., Reed, J. H., & Woerner, B. D. (1996). Position location using wireless communications on highways of the future. *IEEE Communications Magazine, 34*(Oct.), 33–42.

Savarese, C., Rabaey, J., & Langendoen, K. (2002). Robust positioning algorithms for distributed ad-hoc wireless sensor networks. In *Proceedings of USENIX technical annual conference, Monterey, CA, June* (pp. 317–327).

Sugeno, M., & Kang, G. T. (1986). Fuzzy modeling and control of multilayer incinerator. *Fuzzy Sets and Systems, 18*, 329–346.

Takagi, T., & Sugeno, M. (1985). Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics, 15*(1), 116–132.

Yun, S., Lee, J., Chung, W., & Kim, E. (2008). Centroid localization method in wireless sensor networks using TSK fuzzy modeling. In *International symposium on advanced intelligent systems, Sokcho, Korea, September* (pp. 971–974).