International Journal of Software Engineering
and Knowledge Engineering
Vol. 21, No. 1 (2011) 33–54

DOI: 10.1142/S0218194011005153

World Scientific
www.worldscientific.com

A SELF-ADAPTABLE INDOOR LOCALIZATION SCHEME FOR WIRELESS SENSOR NETWORKS

CHI-LU YANG[*,‡], YEIM-KUAN CHANG[*,§], YU-TSO CHEN[*,¶], CHIH-PING CHU[*,‖] and CHI-CHANG CHEN[†,**]

[*]*Department of Computer Science and Information Engineering*
National Cheng Kung University, Tainan, Taiwan

[†]*Department of Information Engineering, I-Shou University*
National Cheng Kung University, Tainan, Taiwan
[‡]*stwin@csie.ncku.edu.tw*
[§]*ykchang@mail.ncku.edu.tw*
[¶]*oilgo1124@gmail.com*
[‖]*chucp@csie.ncku.edu.tw*
[**]*ccchen@isu.edu.tw*

Service systems used for various applications in home automation and security require estimating the locations precisely using certain sensors. Serving a mobile user automatically by sensing his/her locations in an indoor environment is considered as a challenge. However, indoor localization cannot be carried out effectively using the Global Positioning System (GPS). In recent years, the use of Wireless Sensor Networks (WSNs) in locating a mobile object in an indoor environment has become popular. Some physical features have also been discussed to solve localization in WSNs. In this paper, we inquire into received signal strength indication (RSSI)-based solutions and propose a new localization scheme called the closer tracking algorithm (CTA) for indoor localization. Under the proposed CTA, a mechanism on mode-change is designed to switch automatically between the optimal approximately closer approach (ACA) and the real-time tracking (RTT) method according to pre-tuned thresholds. Furthermore, we design a mechanism to move reference nodes dynamically to reduce the uncovered area of the ACA for increasing the estimation accuracy. We evaluate the proposed CTA using ZigBee CC2431 modules. The experimental results show that the proposed CTA can determine the position accurately with an error distance less than 0.9 m. At the same time, the CTA scheme has at least 87% precision when the distance is less than 0.9 m. The proposed CTA can select an adaptive mode properly to improve the localization accuracy with high confidence. Moreover, the experimental results also show that the accuracy can be improved by the deployment and movement of reference nodes.

Keywords: Indoor localization; wireless sensor networks; RSSI; location-based services; home automation; e-health; ZigBee modules.

1. Introduction

The service systems for a large number of applications in home automation and security usually require accurate sensing of the locations of the user using certain

sensors. The sensed data in the systems will be meaningless if the locations of the sensors are unknown. The systems sometimes require the recognition of time and the weather to make decisions. Users always hope to be served correctly by service systems in an indoor environment. To satisfy the demands of users, one of the key successful factors is to estimate the location of the user effectively. However, serving a mobile user automatically by estimating his/her locations in an indoor environment is considered a challenge. Indoor localization cannot be carried out effectively using the Global Positioning System (GPS), which is blocked in urban and indoor environments [1–7, 15–17]. Hence, in recent years, the use of wireless sensor networks (WSNs) to locate mobile objects in the indoor environments has become popular. For example, personal activities are modeled and pre-stored in the database to provide automatic services accurately [18]. The activity patterns of the user can be sensed by the sensors in WSNs to trigger location-based services. Moreover, the application for monitoring remote cardiac patients can also be carried out effectively in WSNs [27]. The wearable device that transmits signals over the WSN can provide continuous electrocardiogram (ECG) monitoring by measuring electrical potentials between the various points of the body using a galvanometer. Through the sensors in WSNs, the network system can reduce the response time if emergency situations are detected. The specific features of the devices in WSNs are low data rate, short range, low power consumption, and low cost (*ZigBee Alliance*). Some physical signals have been discussed to solve the issues of indoor localization in WSNs. Received signal strength indication (RSSI) is the power strength of radio frequency in a wireless environment. RSSI values can be measured and monitored periodically to calculate distances among objects. RSSI solutions do not require special devices or sophisticated hardware. Time of arrival (TOA) means the travel time of a radio signal from one single sender to another remote receiver. By computing the signal transmission time between a sender and a receiver, the distance can be determined approximately. The time difference of arrival (TDOA) is computed based on the emitted signals from three or more synchronized senders. This also refers to the solution of locating a mobile object by calculating the TDOA. In general, a localization system is composed of a blind node, some reference nodes, and base stations with tracking programs. In this paper, we look into RSSI-based solutions for indoor localization and propose a self-adaptable indoor localization scheme for WSNs. The proposed scheme is evaluated using ZigBee CC2431 modules.

The rest of this paper is organized as follows. In Sec. 2, we briefly explain the related work on indoor localization in WSNs. The features of fingerprinting and real-time tracking are discussed. In Sec. 3, we define relevant arguments for describing our scheme and then illustrate the proposed scheme in detail. To validate the proposed scheme, the findings and experimental results are analyzed and discussed in Sec. 4. We show that our scheme has more accurate and precise than the other existing methods. Finally, the conclusions and future work are summarized in Sec. 5.

2. Background

The localization in WSNs has become one of the most popular topics in recent years. To locate and track a mobile object can be applied widely in an indoor environment. ZigBee solutions are for instance applied in a variety of areas, including home automation, environment security, remote healthcare, and smart energy management (*ZigBee Alliance*). ZigBee is a promising protocol for WSNs in industries. ZigBee is a low-cost, low data rate, low-power, easily deployed and wireless mesh networking standard [14, 24, 25, 27] that obeys the original IEEE 802.15.4-2003 standard belonging to wireless personal area networks. The features of the original standard have been extended by the publication of the IEEE 802.15.4-2006 [12–14]. TinyOS, as one of the most popular operating system for wireless sensor networks, is used to control the ZigBee modules. TinyOS was developed at Berkeley in 2000 [26]. A number of technologies for indoor localization in WSNs have also been studied. We intend to inquire into the RSSI-based solutions and solve two-dimensional localization issues. RSSI-based solutions only require common devices among other physical signals. In general, RSSI-based localization solutions in WSNs are categorized into two types: fingerprinting localization and real-time tracking localization.

2.1. *Fingerprinting localization*

The methods of fingerprinting localization (FPT) have been developed through analyzing and utilizing the RSSI characteristics of the environment. In general, a FPT method is partitioned into two phases [5, 8–11, 15]. (1) In the training phase, the analyzed RSSI characteristics are first tuned and then stored in a database. (2) In the estimating phase, these features are retrieved to locate the position of a mobile object. For example [9], the blind node (a mobile object) in the training phase is placed near all the positions of the pre-defined anchors, and the received RSSI values are tuned and stored in a database. Through the RSSI values, the blind node periodically sends requests to its surrounding reference nodes (anchors) and then receives response signals from these anchors. The FPT system can record these responses and analyze their characteristics until the analyzed results are stable. For example, a series offline training of a reference node k at location $L_{ij}, L = [l_{ij}^{k0}, \ldots, l_{ij}^{kM-1}]$, is used to compute a histogram h as an RSSI characteristic.

$$h_{ij}^{k}(\zeta) = \frac{1}{M} \sum_{m=0}^{M-1} \delta(l_{ij}^{km} - \zeta), \quad -255 \leq \zeta \leq 0. \tag{1}$$

The reference nodes are indexed with k. The parameter M is the total number of RSSI samples. The parameter δ represents the Kronecker delta function [9]. Different anchors should have generally different RSSI characteristics. In the FPT method, a mobile object is approximately localized by comparing the current RSSI

values with pre-stored RSSI features. The estimation accuracy of a mobile object is also associated with the deployment density and positions of the anchors. The advantage of the FPT is that the estimated results will be more accurate if the blind node is close enough to the pre-trained anchors. However, building an FPT system requires much effort, particularly when the RSSI features in an indoor environment have to be completed. The training effort and accuracy are traded off. Furthermore, the estimation accuracy also depends on the distance between the blind node and the reference nodes. In general, accuracy degrades as the distance increases.

2.2. *Real-time tracking localization*

Real-time tracking methods (RTT) can locate a mobile object by at least three reference nodes without a pre-trained database [1–4, 6, 7, 12, 15]. Trilateration graphs are ideal for locating a mobile object through at least three reference nodes with positions known in advance [1, 9, 12, 15, 19–23]. A triangle can be formed by three reference nodes. Three edges in the triangle can be used to determine the position of the blind node. Furthermore, each edge can be converted from the RSSI values using specific formulas. A trilateration graph for localization is determined recursively by adding an additional vertex until three edges exist to form a triangle. If the network contains a trilateration graph, one can search for the "seed" triangle in the graph exhaustively and greedily determine the trilateration extensions. Thus, an incremental algorithm can be adopted to search for other trilateration extensions of the sensor network. A mobile object can be progressively located from the incremental algorithms. Trilateration requires the coordinates of the reference nodes (X_i, Y_i) and the distances d_p^i between the blind node and the pre-positioned reference nodes. For example, the target's position $P(X_p, Y_p)$ can be calculated using the minimum mean square error (MMSE) [3]. The difference between actual and estimated distance is defined as Eq. (2), where i is a reference position and p is a mobile object.

$$d_P^i = \sqrt{(x_i - x_p)^2 + (y_i - y_p)^2}. \tag{2}$$

Equation (2) can be transformed into Eq. (3).

$$(d_P^i)^2 = (x_i - x_p)^2 + (y_i - y_p)^2. \tag{3}$$

Then Eq. (3) can be transformed into Eq. (4).

$$\begin{bmatrix} (d_P^1)^2 - (d_P^2)^2 + (x_2^2 + y_2^2 - x_1^2 - y_1^2) \\ (d_P^1)^2 - (d_P^3)^2 + (x_3^2 + y_3^2 - x_1^2 - y_1^2) \\ \cdots \\ (d_P^1)^2 - (d_P^N)^2 + (x_N^2 + y_N^2 - x_1^2 - y_1^2) \end{bmatrix} = \begin{bmatrix} 2(x_2 - x_1) & 2(y_2 - y_1) \\ 2(x_3 - x_1) & 2(y_3 - y_1) \\ \cdots & \cdots \\ 2(x_N - x_1) & 2(y_N - y_1) \end{bmatrix} \begin{bmatrix} x_p \\ y_p \end{bmatrix}. \tag{4}$$

Therefore, Eq. (4) is transformed into Eq. (5),

$$b = A \begin{bmatrix} x_p \\ y_p \end{bmatrix}, \tag{5}$$

which can be solved using the matrix solution given by Eq. (6). Position $P(X_p, Y_p)$ can be obtained by calculating Eq. (6)

$$\begin{bmatrix} x_p \\ y_p \end{bmatrix} = (A^T A)^{-1} * (A^T b) \tag{6}$$

where

$$b = \begin{bmatrix} (d_P^1)^2 - (d_P^2)^2 + (x_2^2 + y_2^2 - x_1^2 - y_1^2) \\ (d_P^1)^2 - (d_P^3)^2 + (x_3^2 + y_3^2 - x_1^2 - y_1^2) \\ \cdots \\ (d_P^1)^2 - (d_P^N)^2 + (x_N^2 + y_N^2 - x_1^2 - y_1^2) \end{bmatrix}, \tag{7}$$

$$\text{and} \quad A = \begin{bmatrix} 2(x_2 - x_1) & 2(y_2 - y_1) \\ 2(x_3 - x_1) & 2(y_3 - y_1) \\ \cdots & \cdots \\ 2(x_N - x_1) & 2(y_N - y_1) \end{bmatrix}. \tag{8}$$

The advantage of RTT is that we do not need to spend on the training effort in advance. The RTT method can be used directly to estimate the positions through real-time RSSI values. However, real-time RSSI easily fluctuates due to physical interference in the environment, causing the localization using the RTT method to become easily inaccurate with high variation. The estimation accuracy of the RTT is relative to environment conditions. Furthermore, the estimation accuracy slightly depends on the distance between the blind node and the reference nodes, too.

The FPT and RTT algorithms have their own specific strengths and weaknesses. Therefore, we utilize their strengths and design a self-adaptable scheme for indoor localization in wireless sensor networks.

3. The Proposed Scheme

The proposed self-adaptable scheme, including an improved FPT algorithm and an enhanced RTT method, provides a mode-change mechanism to choose a better executing mode according to pre-tuned thresholds. In subsections, we define relevant arguments to describe our scheme and then illustrate the proposed scheme in detail.

3.1. *Definitions*

A blind node is a mobile object that is movable and can be localized through RSSI values in an indoor environment. A reference node is a fixed node that responds to

RSSI to assist in locating the blind node. In this scheme, the blind node and the reference nodes are both ZigBee modules as ZigBee sensors are cheap, movable, easily available, and can be deployable conveniently. The terms defined in this subsection are used to represent our proposed scheme. These are categorized into primitive, original physical and derived terms. The primitive terms are defined as follows:

N_{neighbor} = *number of reference nodes that are neighbors of the blind node within one hop*;
RID = *pre − defined identification of a reference node* (*a fixed object*);
$R_{\text{threshold}}[RID][d]$ = *pre-trained RSSI thresholds of RID at distance d, where distance d is a set* = $\{d(m) \mid 0.5, 1, 1.5, 2.0, 2.5, 3.0\}$;
BID = *pre-defined identification of a blind node that is a mobile object*;
$RNPos[x]$ = *coordinates of the reference nodes, where variable x refers to RID*;
M_{ACA} = *mode of approximately closer approach* (*ACA*) *for localization* (*an improved algorithm*); *and*
M_{RTT} = *mode of RTT for localization* (*an improved algorithm*).

The values of the RSSI thresholds of *RID* within distance d are pre-trained and stored in the database. The terms of physical arguments, which are originally received from the ZigBee blind node, are defined as follows:

R_{now} (x) = *current value of the measured RSSI of x, where variable x refers to RID*;
rid = *an index of* R_{now}, *where* $1 \leq rid \leq N_{\text{neighbor}}$.

The derived terms, whose values are calculated from the physical terms and primitive terms, are defined as follows:

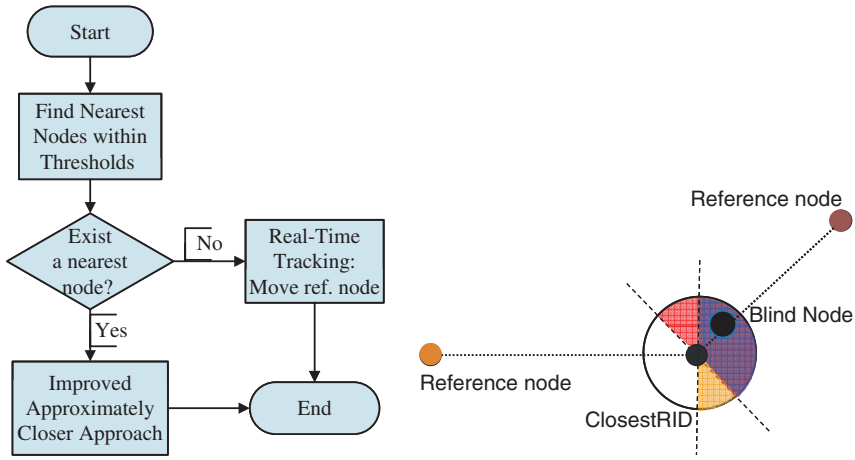
$CloseList[x]$ = *list of RIDs sorted by* $R_{\text{now}}(x)$, *where* $R_{\text{now}}(x)$ *is within* $R_{\text{threshold}}[x][d]$ *and* $R_{\text{now}}(x) \leq R_{\text{now}}(x-1), 1 \leq x \leq N_{\text{neighbor}}$;
$ClosestRID$ = *rid that is the closest node to the blind node* (*mobile object BID*) *where* R_{now} (*ClosestRID*) *is within* $R_{\text{threshold}}$;
M_C = *current localization mode* = $\{M_C | M_{ACA}, M_{RTT}\}$;
C_R = *record for tracking the mobile object*;
$NDis$ = *estimated distance between the blind node and the closest reference node*;
RN_{RSSI} $[x][rid]$ = *RSSI value between reference node x and the ClosestRID rid*;
$AngleList[x][y]$ = *angles among reference nodes where variable x and y are RIDs*;
$RA_{\text{est}}[x](s, e)$ = *estimated range* (*partial-circle*) *formed from the ClosestRID* (*the center of the circle*) *and two reference nodes* (*s, e*). *The two points* (*s, e*) *are on the circumference of the circle, where s is the start point, e is the end point, and x refers to RID*;
IRA_{est} (s, e) = *intersection range* (*partial-circle*) *of all other range, where s is the start point and e is the end point*;
X_{est} = *estimated value of X coordinate of the blind node*; *and*
Y_{est} = *estimated value of Y coordinate of the blind node.*

3.2. *The proposed closer tracking algorithm (CTA)*

The proposed CTA that contains an improved ACA method and a movement mechanism provides a mode-change mechanism to select a better executing mode automatically according pre-tuned thresholds. We tune and store the thresholds for the basic scheme in the initial step. The improved ACA is different from the FPT. The idea for ACA emerged from our observation on elderly persons in the house. The elderly usually stay in the same positions, such as the sofa, bed, in front of the TV, or near the door for a long time. The amount of time they stay in the same location inside the house is much longer than the amount of time they move. As we intend to provide automatic applications suitable for the elderly or persons in their houses, we can design a position tracking algorithm ideally based on observable features. Therefore, we specifically anchor the reference nodes near the positions where the elders are usually stay, while tuning the thresholds. The movement mechanism is an enhanced the RTT algorithm to reduce the uncovered area of ACA. The proposed CTA is designed specifically to enhance location-based services in home automation. The process of the proposed scheme that includes the improved ACA and the movement mechanism is shown in Figs. 1(a) and 2. The details are described in the steps as follows.

Step 0 — [Tuning Thresholds]

To decrease the error of converting the measured RSSI values to the estimated distances, we define the RSSI *thresholds* to estimate the distances between the blind



(a) Flow of the proposed CTA

(b) Conception of the improved Approximately closer Approach (ACA)

Fig. 1. Proposed CTA.

```
Algorithm_Improved_Closer_Tracking(int *R_now)
{
//////Initial////////////////////////////////////////////////////////////////////////
short CloseList [8]={-1};
short k=0;
float AngleList[8][8];    //angles of reference nodes each other
//////Step1 – Build Neighbor List/////////////////////////////////////////////////
01 call function Build_Neighbor_List( CloseList, &k, &NDis)
//////Step2 – Determine Mode//////////////////////////////////////////////////////
02 if (k == 0) { // It doesn't exist nearest reference node
03       M_C = M_RTT      //Change to Real-Time Tracking Mode
04       isRNMove = true;      //Active move a reference node
05 } // end if
06 else {
//////Step3 – Improved Approximately Closer Approach//////////////////
07       M_C = M_ACA
08       ClosestRID = CloseList[0];    //nearest
09       if(isRNMove)
10            call function Compute_Angle( AngleList);
11       call function Improved_ACA( ClosestRID, NDis,
         AngleList)
12 }//end of if-else
} //end Improved Closer Tracking Algorithm
```

Fig. 2. Process of the proposed CTA.

node and the reference nodes. A RSSI value will be filtered if it is out of *thresholds* at distance *d*. *Threshold* is defined as

Threshold = [*mean* − *n*σ*, *mean* + *n*σ*], *where the mean is an average of the measured RSSI values,* σ *is the standard derivation, and n is a constant value used to adjust the thresholds.*

For tuning *thresholds* in this initial step, the RSSI values of two *RIDs* were measured empirically in one-dimensional space at different distances *d*, such as 0.5, 1, 1.5, 2.0, 2.5, and 3.0 m. Second, these values were calculated through the mentioned definition of *thresholds*. They were stored permanently in the mentioned term: $R_{\text{threshold}}$. In other words, the values of *thresholds* can be determined using the *thresholds* definition, and the measured RSSI values can be computed in this step. Furthermore, *thresholds* can be defined as other statistic methods, such as *interval estimation*.

Step 1 — [Find the Nearest Node and Build the CloseList]

The blind node (*BID*; a mobile object) broadcasts its requests periodically and then receives RSSI values (R_{now}) from its neighbor nodes (*RIDs*). The neighbor nodes (*RIDs*) are sorted by comparing their RSSIs (R_{now}) with the pre-defined thresholds ($R_{\text{threshold}}$) and recorded in the *CloseList*. An *RID* is stored in the *CloseList* if the measured RSSIs of the *RID* are within *distance d* from $R_{\text{threshold}}$. The *RIDs* in the *CloseList* are also sorted by their R_{now} values. Therefore, the reference node closest to the *BID* should be recorded in the *CloseList* [0], which is called *ClosestRID*. The process of establishing the *CloseList* and finding the nearest node is shown in Fig. 3.

```
Function Build_Neighbor_List(short *CloseList, int *k, double
*NDis)
{
//////Initial////////////////////////////////////////////////////////////////////////////
short rid;     float dis = 0;
short N_neighbor = the number of reference nodes which close to
blind node within one hop currently;
int R_now[x] = the current value of the measured RSSI of x, where
variable x refers to RID;
//////Step1 – Build Neighbor List//////////////////////////////////////////////////
01 for (dis = 0.5 ; dis <= 2.0 ; dis += 0.5){
02      for (rid = 1 ; rid <= N_neighbor ; rid++){
03          if (R_now[rid] within R_threshold[rid][dis]){
04              C  loseList[*k] = rid;
05                *k++;
06                if (*k == 1) *NDis = dis;
07          }//end if
08      }//end for
09 } //end for loop
} //end Build Neighbor List Function
```

Fig. 3. Process of establishing a list of sorted neighbor nodes (Step 1).

Step 2 — [Determine the Mode]

If the nearest node is found in the *Step 1*, the improved ACA is selected to locate the mobile object further. In other words, if there is no record in the *CloseList*, the improved RTT (a movement mechanism) is executed to locate the mobile object. The process of determining the active modes is shown in Fig. 1(a).

Step 3 — [The Improved ACA]

The improved ACA performs an initial step by drawing a complete circular area (*ClosestRID*-circle) as the measured R_{now} of the *ClosestRID* is within the pre-tuned *thresholds* of distance d in *Step 1*, where the center of the *ClosestRID*-circle is the *ClosestRID,* and the radius is the distance d. The *ClosestRID*-circle can be divided conceptually into several partial-circles. By comparing the R_{now} with the RN_{RSSI}, we can determine the closer node among the *ClosestRID* or the *blind node (BID)* to form a half-circle as shown in Fig. 1(b). For example, the closer node is the *BID* if the R_{now} is larger than the RN_{RSSI}. A half-circle of the *ClosestRID* can thus be determined. The half-circle is also close to the side of the reference *RID* of the *CloseList*. In other words, if the R_{now} is smaller than the RN_{RSSI}, the closer node is the *ClosestRID*. A half-circle of the *ClosestRID* can also be determined. However, the difference is that the half-circle is far from the reference *RID* of the *CloseList*. The intersection of the half-circle and the complete *ClosestRID*-circle can form a smaller partial-circle of the *ClosestRID*. We can iteratively intersect the half-circle and partial *ClosestRID*-circle to form another smaller partial-circle until all the *RIDs* in the *CloseList* are considered. The blind node is located in the final

```
Function Improved_ACA(short ClosestRID, double NDis, double
*AngleList)
{
//////Initial////////////////////////////////////////////////////////////////////////////
float X_est & Y_est= The estimated position of ClosestRID;
//////Step 3 – Improved Tracking Path to Blind Node////////////////////
    //Find out which range is covered by the formed ranges which
    // are produced by reference nodes.
01 for (short c = 1 ; c < k ; c++){
02      if(c == ClosestRID)     continue;
03      RA_est[c] (s) = AngleList[ClosestRID][c] – 90;
04      if (RA_est[c] (s) < 0) RA_est[c] (s) += 360;
05      if (R_now[c] < RN_RSSI[ClosestRID][c]) {
06      // the blind node is at the other side
07          RA_est[c] (s) += 180;
08          RA_est[c] (s) %= 360;
09      }//end if
10      RA_est[c] (e) = RA_est[c] (s) + 180;
11      Find out the intersection of all range from RA_est[c] (s, e)
12 }//end for
13 int Arg = (IRA_est (s ) + IRA_est (e ) )/ 2;
14 Arg %= 360;
15 X_est += NDis * Math.Sin(Math.PI / 180 * Arg);
16 Y_est += NDis * Math.Cos(Math.PI / 180 * Arg);
} //end Improved ACA Function
```

Fig. 4. Process of the improved ACA.

partial-circle of the *ClosestRID*. The other *RIDs* of the *CloseList* in this step are used to narrow down the *ClosestRID*-circle. The process of the improved ACA is shown step by step in Figs. 4 and 5.

3.3. *Movement mechanism*

In Step 2, the localization scheme is performed under the *RTT* mode if there is no record in the *CloseList*. However as indicated earlier, accuracy is not sufficient. An improved method would be one that chooses the Closest *RID* node from the neighbor nodes and then moves it towards the direction of the estimated position of the blind node. The position of the blind node can be estimated first under the RTT mode. Second, the mechanism selects the reference node that has the largest RSSI values and then moves the node toward the estimated position of the blind node. Thus, the direction and destination of moving the closest *RID* node can be determined using the estimated position of the blind node (*BID*) under the RTT mode. Finally, the localization scheme is switched to the ACA mode to estimate the position of the blind node again. The process of switching between the RTT mode and the improved ACA mode by moving a reference node is shown in Fig. 6. Through this process, we can ideally move a reference node close to the blind node and then switch to the ACA mode for further localization. Therefore, the proposed CTA can increase the localization accuracy.

```
Function Compute_Angle(float *AngleList){
//////Initial////////////////////////////////////////////////////////////////////////////////
double arctan(double x, double y){
    return Math.Atan(Math.Abs(x) / Math.Abs(y)) * (180 /
Math.PI);}
//////Compute Angles Between Reference Nodes//////////////////////////
01 for (short s = 1 ; s <= N_neighbor ; s++){
02      for(short z = 1 ; z <= N_neighbor ; z++){
03          Translation RNPos[s] to (0,0)
04          //RNPos[z] = (x,y);//after Translation
05          Check (x,y) in which quadrant
06          if(z == s) continue;
07          if(x == 0 && y == 0){     //origin
08          }else if(x == 0){
09              if(y > 0)     angle = 0;
10              else     angle = 180;
11          }else if(y == 0){
12              if(x > 0)     angle = 90;
13              else     angle = 270;
14          }else{
15              switch(quadrant){^
16                  case 1:
17                      angle = arctan(x,y);     break;
18                  case 2:
19                      angle = 360 – arctan(x,y);     break;
20                  case 3:
21                      angle = 180 + arctan(x,y);     break;
22                  case 4:
23                      angle = 180 – arctan(x,y);     break;
24              }//end switch
25          AngleList[s][z] = angle;
26 } //end for
} //end Compute Angle Function
```

Fig. 5. Computation of angles among the reference nodes.

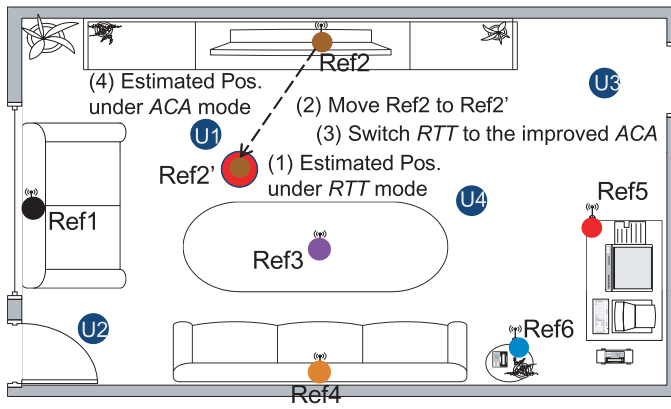


Fig. 6. Process of switching the RTT mode to the improved ACA mode by moving a reference node.

4. Performance Evaluation

The ZigBee modules, which include the CC2431 and CC2430 chips [12], are deployed to validate the proposed scheme in our experiments. The CC2431 chip represents the blind node, whereas the CC2430 chips represent the reference nodes. ZigBee has a radio frequency band of 2.4 GHz and a transmission bit-rate of approximately 250 kb/s. The other specific features of these chips are listed in Table 1; the snapshot of the CC2431 module is shown in Fig. 7. The RSSI values are measured continuously in the experiment, and all the values are stored in a database for further analysis. The proposed CTA is implemented using the C#.NET language.

4.1. *Findings*

To simplify the thresholds tuning, we first decrease the estimated three-dimensions RSSI to one-dimension and measure one-dimensional RSSI values in different environments, where the electromagnetic waves are isolated, absorbed, or normal. At the same time, the RSSI values are measured at different distances in the three

Table 1. Features of CC2431 [13].

Features	Values
Radio frequency band	2.4 GHz
Chip Rate (kchip/s)	2000
Modulation	Q-QPSK
Bit rate (kb/s)	250
Sensitivity	−92 dBm
Data memory	8 KB
Program memory	128 KB internal RAM
Spread spectrum	DSSS

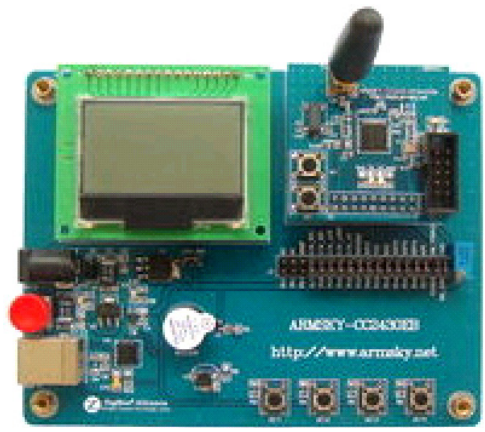


Fig. 7. CC2431 module.

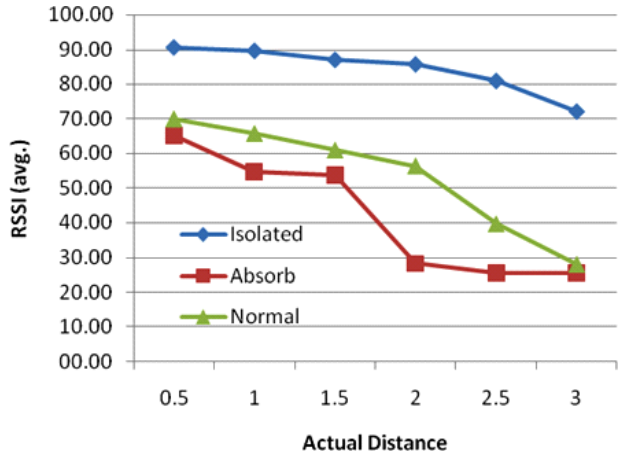


Fig. 8. Average RSSIs in different environments: Isolated, Absorb, and Normal. For example, thresholds can be defined as [mean $-1.0^*\sigma$, mean $+ 1.0^*\sigma$], where σ is the standard deviation.

environments. The average of the collected values is computed, as shown in Fig. 8. In the figure, the x-axis represents the various distances between a blind node and a reference node, namely, 0.5, 1, 1.5, 2.0, 2.5, and 3 m. In the y-axis, the averages of the measured RSSI values are represented. Furthermore, the measurements of the RSSI values are repeated until the statistical results are stable. To show the data points above the horizontal line, all the measured values are added to 100. The statistical results are shown in Fig. 8. The values of standard deviation σ are utilized to define the thresholds further.

The formula provided by *Texas Instruments* (*TI*) representing the relationship between the RSSI value and the estimated one-dimensional distance is shown as follows:

$$RSSI = -(10n \log_{10} d + A), \tag{9}$$

where n is a signal propagation constant or exponent, d is a distance from the blind node to the reference node, and A is the received signal strength at a 1 m distance. According to formula (9), one-dimensional distance d can be derived from the measured RSSI values of Fig. 8. The results of the derived distances in the findings are shown in Fig. 9. The minimal and accepted errors between the derived and actual distances in the three environments are at the distances of 1.0, 1.5, and 2.0 m with the given parameters (A, n). These errors are less than 1.0 m in the trial, providing us with opportunities to obtain the accurate localization using this characteristic. Therefore, we design our scheme to search for the probable nearest reference node as mentioned in step 1 of the proposed scheme. The defined thresholds in the initial step 0 can adapt the boundary sensibilities and enhance the searching hit-ratios by adjusting the standard deviation σ.

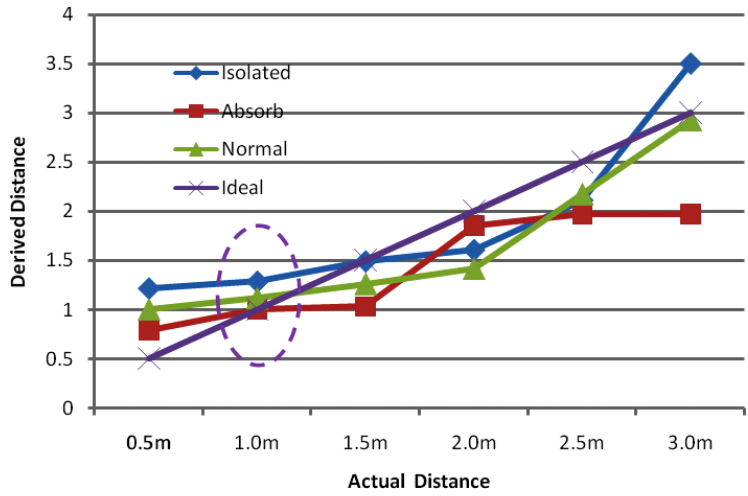


Fig. 9. Results of transferring the RSSIs to distances using formula (9), where (A, n) = Isolated (6, 4); Absorb (45, 10); Normal (30, 9).

4.2. *Results*

In this experiments, an actual position is represented by the coordinate (x, y), and an estimated position is represented by the coordinate (i, j). Therefore, we can simply define the accurate distance as an Error Distance formula as follows:

$$\text{Dist.}(L_{xy}, L_{ij}) = \sqrt{(x-i)^2 + (y-j)^2}. \tag{10}$$

To validate the accuracy of the proposed CTA, we implement the proposed CTA compared with the FPT [9], *original CTA* [15], and RTT [12] using the CC2431 location engine. The experimental results of deploying four reference nodes are shown in Fig. 10. The x-axis represents the distance from the blind node to the closest reference node. In the y-axis, the estimation error represents the difference between an actual and an estimated position.

Figure 10 shows that the accuracy of the proposed CTA is less than 1 m when the distance in x-axis is within 2 m; the proposed scheme can determine the position accurately with an error distance of less than 1 m when the blind node approaches any reference node. Furthermore, the accuracy in the experiment approaches that of the RTT methods if the distance in x-axis is greater than 2 m. The FPT is accurate enough when the blind node moves near the pre-trained positions. The estimation error is less than 1.5 m if the distance in the x-axis is within 1.5 m. However, the estimation error is more than 1.7 m if the distance in the x-axis is over 1.5 m, which is characteristic of the FPT method. The estimation errors of the RTT method are considerably stable at all distances. The accuracy of the RTT method is 1.5 m on average. Hence, the accuracy of RTT method is absolutely independent of the positions of the reference nodes.

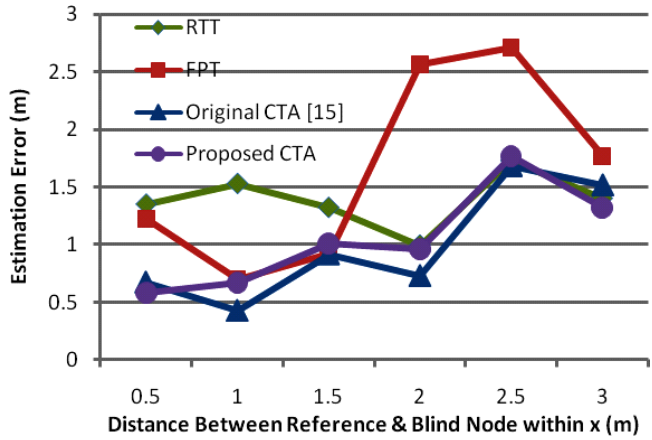


Fig. 10. Estimation error vs. the distance between the reference nodes and the blind node at distance $x(m) = \{0.5, 1.0, 1.5, 2.0, 2.5, 3.0\}$ m (accuracy: four reference nodes).

The distances between the blind node and the closest reference node increase along the x-axis. Therefore, the RSSI values are interfered increasingly by some background noise, causing the variances to increase as well. In the FPT method, the signal features diminish, and the estimation errors increase. Thus, the FPT method cannot determine the position accurately when the distance between the blind node and the closest reference node is more than 2 m. Under this condition, our proposed CTA changes the operational mode from the ACA to the RTT. As a result, the accuracy of the proposed method is close to the accuracy of the RTT if the distance of the x-axis is over 2 m. In the case of $x = 2.5$ m, the proposed CTA is slightly more accurate than the RTT method. In the case of $x = 3.0$ m, the proposed CTA is slightly worse than the RTT method.

In Figs. 11 and 12, we show the precision of the proposed CTA, original CTA, and the FPT and RTT methods. The precision representing the confidence of the localization is defined as follows:

$$\frac{Number\_of\_within\_Acceptable\_Error\_Distance}{Total\ Estimated\ Times}. \tag{11}$$

Similar to the experimental design of the precision in Fig. 11, the acceptable error distance is set to 1 m. Under this condition, if the estimated error is less than or equal to 1 m, the estimation record is selected for the precision calculation. The precision of the proposed CTA is at least 95%, 91%, and 80% at 0.5, 1.0, and 1.5 m, respectively. The RTT method presents relatively low precision in this condition. The CTA scheme has higher precision than the other methods. The scheme with high precision is especially useful in an indoor localization. In Fig. 12, the precision of the proposed CTA is over 83% at both 2 and 3 m. However, the precision is low (66%) in the case of $x = 2.5$ m. The reason is that most estimated errors remain

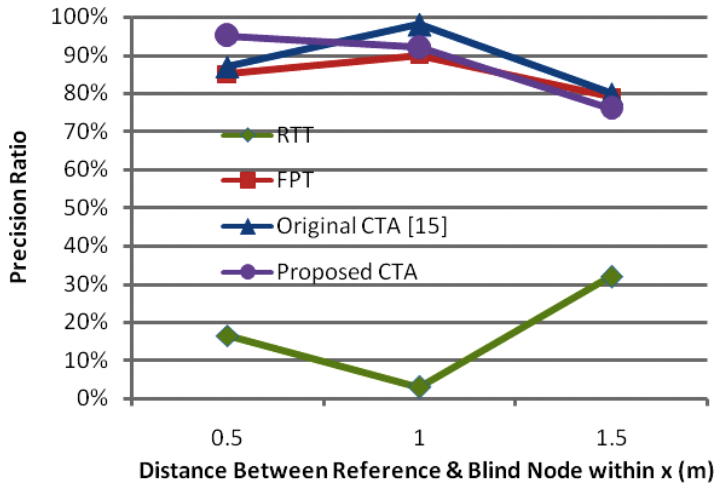


Fig. 11. Precision when acceptable estimation error is within 1.0 m at different distances (x meters) (Precision: four reference nodes).

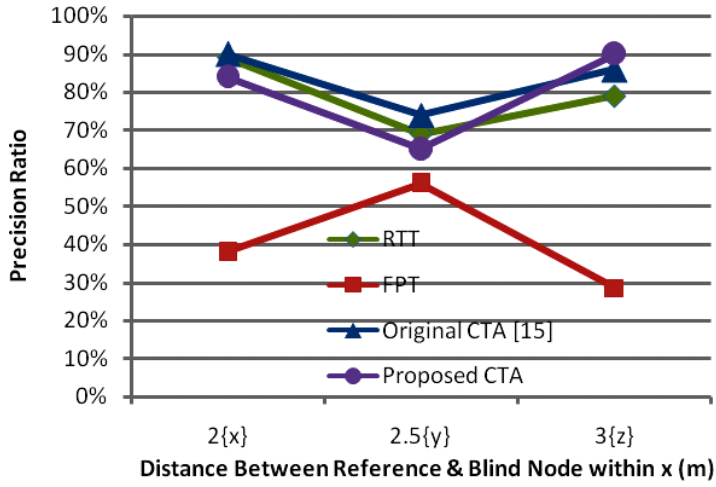


Fig. 12. Precision when acceptable estimation errors $\{x\}$ are within 1.3, 1.3, and 1.8 m at x, y, and z distances, respectively (Precision: four reference nodes).

in the ranges between 1.5 and 1.8 m, which is an interesting situation because the mode-changing functionality is just performed between these ranges.

The usability of the mode-changing operation of the proposed CTA corresponds to the usage ratio of the ACA and the RTT modes in the experiment. The usability of the mode-changing functionality is presented in Fig. 13. The usage ratios of the ACA and the RTT are shown at various distances in the y-axis. The ACA method is useful if the distance is less than 1.75 m. Furthermore, the ACA mode changes to

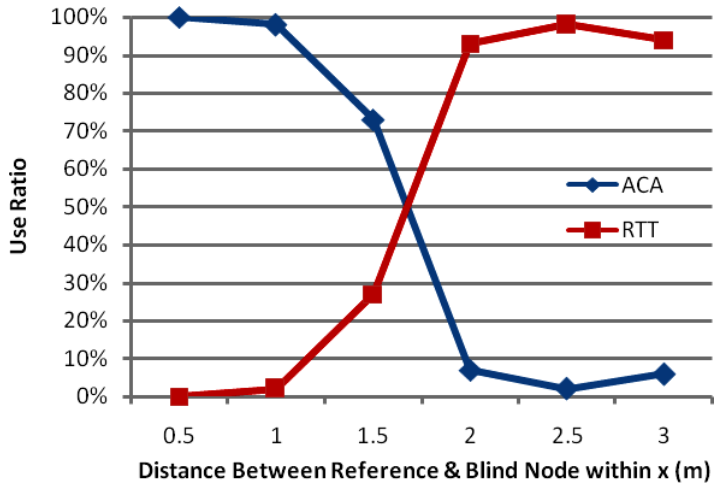


Fig. 13. Usage ratio of the ACA and RTT modes.

the RTT if the distance increases over 1.75 m. The mode-changing operation can be carried out practically according to the thresholds we set in advance. As a result, the proposed CTA can select an adaptive mode to acquire high precise localization.

The number of deployed references nodes can influence accuracy and precision. We validate the accuracy influence by deploying eight reference nodes and compare this with four reference nodes. In Fig. 14, the accuracy of estimation errors of the proposed CTA can be less than 0.9 m, which is better than the original CTA [15] and RTT.

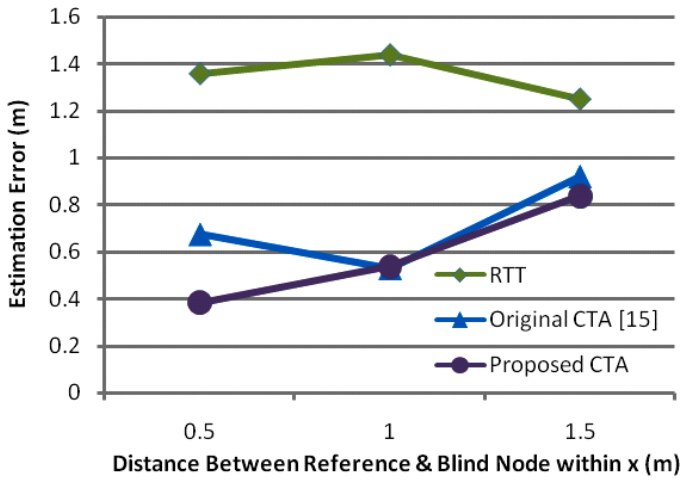


Fig. 14. Estimation error vs. distance between the reference nodes and the blind node at distance $x(m) = \{0.5, 1.0, 1.5\}$ m (accuracy: eight reference nodes).

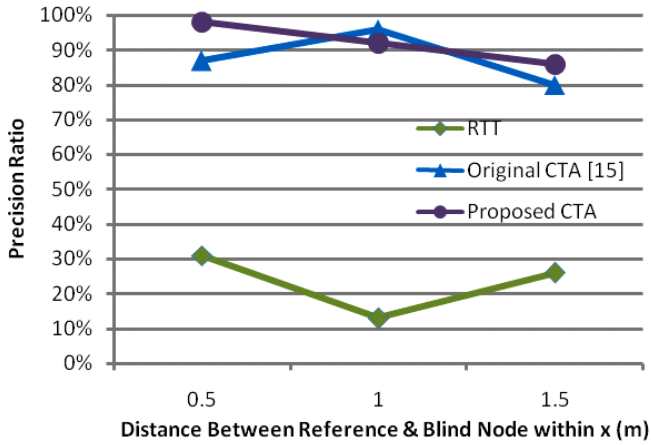
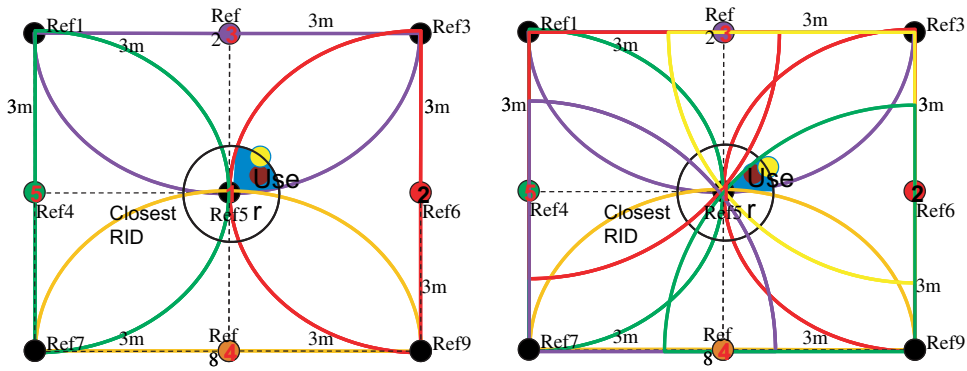


Fig. 15. Precision when error distance is within 1.5 m (precision: eight reference nodes).

Figure 15 shows the precision of the proposed CTA, which is more than 87% and is on average higher than the original CTA [15]. As a result of deploying more reference nodes, Step 3 of the improved ACA can intersect to form a smaller partial-circle range. The intersection of the eight reference nodes is compared with that of the four nodes, as shown in Fig. 16. The intersected area in Fig. 16(b) is smaller than that in Fig. 16(a).

In a living room, we deploy six reference nodes (*Ref1–Ref6*) with arbitrary positions in areas that the user usually stays close. We compare the proposed CTA with the RTT in this environment. The accuracy and precision are shown in Figs. 17 and 18, respectively. Figure 17 indicates that the accuracies are 0.5, 0.6, and 0.8, while the distances are at 0.5, 1.0, and 1.5 m, respectively. The estimated results are



(a) Four Reference Nodes (b) Eight Reference Nodes

Fig. 16. Four reference nodes vs. eight reference nodes (improved ACA).

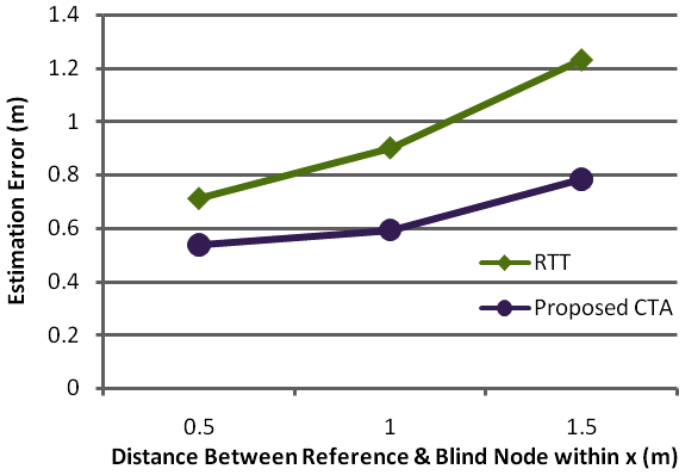


Fig. 17. Accuracy of user's activity area in the house.

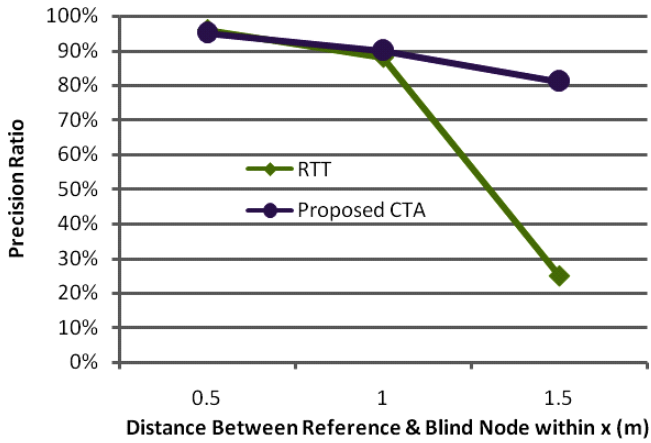


Fig. 18. Precision of user's activity area in the house.

accurate enough if we intend to provide location-based services in a real-indoor environment. In Fig. 18, the precision is more than 82% when the distance is less than or equal to 1.5 m. At the same time, the precision is at least 91% when the distance is less than or equal to 1 m. The precision is about 95% when the distance is less than or equal to 0.5 m. In other words, the estimated confidence is sufficient to provide location-based services in a real-indoor environment.

The dotted lines in Fig. 19 represent the movable tracks of the reference nodes (*Ref1*–*Ref5*). While the CTA scheme is under the RTT mode, the mechanism of switching to ACA mode is carried out by moving the reference node. *Ref1*, *Ref4* and *Ref5*, areas where the users are usually close to for various conditions in a living room, are automatically selected to be moved. The experimental results in Table 2

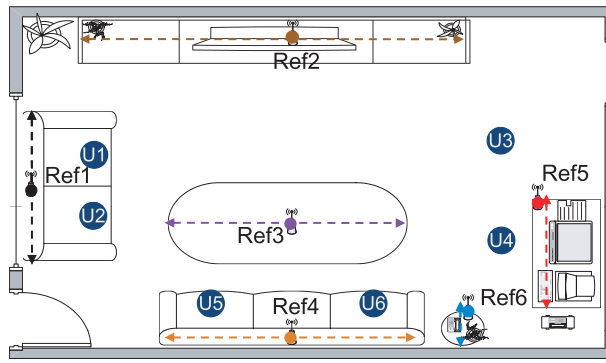


Fig. 19. Experimental environment of the movable reference nodes.

Table 2. Results of the estimated errors of switching the RTT mode to the ACA mode by moving the references nodes (*Ref1*, *Ref4*, and *Ref5*).

		Direction (Ratio)		Estimated error (m)	
User's position	Covered Ref.	Up/left (%)	Down/right (%)	Before movement	After movement
U1	Ref1	61	39	1.093426	0.535063
U2		0	100	0.472221	0.491547
U3	Ref5	87	13	1.535533	0.874122
U4		0	100	0.505131	0.534362
U5	Ref4	77	23	1.798067	0.684432
U6		0	100	1.490059	0.381617

show that the directions can be determined correctly while the users are at the six positions marked as $U1$–$U6$. The estimated errors can decrease effectively by 52% on average after movement. Therefore, the accuracy can be improved by the mechanism of moving the reference nodes.

5. Conclusions and Future Work

In this paper, we investigated the RSSI-based solutions for indoor localization and proposed a new self-adaptable indoor localization scheme called CTA for WSNs. The mode-changing operation of the proposed CTA was designed to switch between ACA and RTT methods. This functionality can adapt the optimal modes automatically according to the pre-defined thresholds, which we had tuned and mentioned in Sec. 4.1. We also designed a moving mechanism for the reference nodes to reduce the uncovered area of the ACA and increase the accuracy of the CTA scheme while the original positions of the estimated objects are beyond the reference nodes. The proposed CTA can select an adaptive mode properly and improve the localization accuracy significantly. Our experimental results show that the proposed CTA can accurately determine positions with an error distance of less than 0.9 m. At the

same time, the estimated precision of our proposed method is at least 87% when the distance is less than 0.9 m. Our results indicate that the accuracy can also be improved by the movement and deployment of the reference nodes.

The CTA scheme can be applied to trigger correct and suitable services for various applications in home automation and security according to the estimated locations of the user. In the future, CTA can provide promising quality of services geared towards taking care of the elderly and offer location-based services in an indoor environment.

Acknowledgments

This research was partially supported by the second phase of the Applied Information Services Development and Integration project of the Institute for Information Industry (III) sponsored by Ministry of Economic Affairs (MOEA), Taiwan R.O.C.

References

1. E.-E.-L. Lau, B.-G. Lee, S.-C. Lee and W.-Y. Chung, Enhanced RSSI-based high accuracy real-time user location tracking system for indoor and outdoor environments, *International Journal on Smart Sensing and Intelligent Systems* **1**(2) (2008) 534–548.
2. Y. Gwon, R. Jain and T. Kawahara, Robust indoor location estimation of stationary and mobile users, in *Proc. of the IEEE INFOCOM'04*, March 2004, pp. 1032–1043.
3. M. Sugano, T. Kawazoe, Y. Ohta and M. Murata, Indoor localization system using RSSI measurement of wireless sensor network based on ZigBee standard, in *Proc. of Wireless Sensor Networks 2006 (WSN 2006)*, Canada, July 2006, pp. 1–6.
4. S. Tennina, M. D. Renzo, F. Graziosi and F. Santucci, Locating ZigBee nodes using the TI's CC2431 Location Engine: A testbed platform and new solutions for positioning estimation of WSNs in dynamic indoor environments, in *Proc. of the First ACM International Workshop on Mobile Entity Localization and Tracking in GPS-less Environments (MELT 2008)*, Sep. 2008, pp. 37–42.
5. H.-S. Ahn and W. Yu, Environmental-adaptive RSSI-based indoor localization, *IEEE Trans. on Automation Science and Engineering* **6**(4) (2009) 626–633.
6. A. Ka and L. Miu, Design and implementation of an indoor mobile navigation system, Master thesis of CS at MIT, 2002.
7. P. Bahl and V. N. Padmanabhan, RADAR: An in-building RF-based user location and tracking system, in *Proc. of the IEEE INFOCOM2000*, March 2000, pp. 775–784.
8. A. S.-I. Noh, W. J. Lee and J. Y. Ye, Comparison of the mechanisms of the ZigBee's indoor localization algorithm, in *Proc. of the Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD 2008)*, Aug. 2008, pp. 13–18.
9. Q. Yao, F.-Y. Wang, H. Gao, K. Wang and H. Zhao, Location estimation in ZigBee network based on fingerprinting, in *Proc. IEEE International Conference on Vehicular Electronics and Safety*, Dec. 2007, pp. 1–6.
10. S. Tadakamadla, Indoor local positioning system for zigbee based on RSSI, M.Sc. Thesis report, Mid Sweden University, 2006.
11. C. Gentile and L. Klein-Berndt, Robust location using system dynamics and motion constraints, in *Proc. of the 2004 IEEE International Conference on Communications* **3** (2004) 1360–1364.

12. System-on-chip for 2.4 GHz ZigBee/IEEE 802.15.4 with location engine., Datasheet, Texas Instruments, http://focus.ti.com/lit/ds/symlink/cc2431.pdf, July 2007.
13. K. Aamodt, CC2431 location engine, *Application Note AN042*, Texas Instruments.
14. IEEE 802.15 WPAN^{TM} Task Group 4, *IEEE 802.15.4-2006, ZigBee Specification Version r13*, ZigBee Alliance, San Ramon, CA, USA, Dec. 2006.
15. Y.-T. Chen, C.-L. Yang, Y.-K. Chang and C.-P. Chu, A RSSI-based algorithm for indoor localization using ZigBee in wireless sensor network, in *Proc. of the 15th International Conference on Distributed Multimedia Systems (DMS 2009)*, Sep. 2009, pp. 70–75.
16. P. Barsocchi, S. Lenzi, S. Chessa and G. Giunta, Virtual calibration for RSSI-based indoor localization with IEEE 802.15.4, in *Proc. of International IEEE International Conference on Communications*, Dresden, Germany, June 2009, pp. 1–5.
17. C. Wang, J. Chen and Y. Sun, Sensor network localization using kernel spectral regression, *Wireless Communications and Mobile Computing*, published online, June 2009.
18. C.-L. Yang, Y.-K. Chang, C.-P. Chang and C.-P. Chu, A personalized service recommendation system in a home-care environment, in *Proc. of the 15th International Conference on Distributed Multimedia Systems (DMS 2009)*, Sep. 2009, pp. 76–81.
19. Y. Wang, S. Lederer and J. Gao, Connectivitys-based sensor network localization with incremental delaunay refinement method, in *Proc. of the IEEE INFOCOM'09*, April 2009, pp. 2401–2409.
20. T. Eren, D. K. Goldenberg, W. Whiteley, Y. R. Yang, A. S. Morse, B. D. O. Anderson and P. N. Belhumeur, Rigidity, computation, and randomization in network localization, in *Proc. of the IEEE INFOCOM'04* **4** (2004) 2673–2684.
21. D. K. Goldenberg, P. Bihler, M. Cao, J. Fang, B. D. O. Anderson, A. S. Morse and Y. R. Yang, Localization in sparse networks using sweeps, in *Proc. of the ACM/IEEE International Conference on Mobile Computing and Networking* (MobiCom 2006), pp. 110–121.
22. D. Goldenberg, A. Krishnamurthy, W. Maness, Y. R. Yang, A. Young, A. S. Morse, A. Savvides and B. Anderson, Network localization in partially localizable networks, in *Proc. of the IEEE INFOCOM'05* **1** (2005) 313–326.
23. D. Moore, J. Leonard, D. Rus and S. Teller, Robust distributed network localization with noisy range measurements, in *Proc. of the 2nd International Conference on Embedded Networked Sensor Systems* (SenSys'04), Nov. 2004, pp. 50–61.
24. W.-S. Jang and M. J. Skibniewski, A wireless network system for automated tracking of construction materials on project sites, *Journal of Civil Engineering and Management* **14**(1) (2008) 11–19.
25. T.-T. Do, D. Kim, T. S. Lopez, H. Kim, S. Hong, M.-L. Pham, K. Lee and S. Park, An evolvable operating system for wireless sensor networks, *International Journal of Software Engineering and Knowledge Engineering (IJSEKE)* **15** (2005) 265–270.
26. J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler and K. Pister, System architecture directions for networked sensors, in *Proc. of the International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS 2000)*, Nov. 2000, pp. 93–104.
27. F. Hu, M. Jiang and Y. Xiao, Low-cost wireless sensor networks for remote cardiac patients monitoring applications, *Wireless Communications and Mobile Computing* **8**(2) (2008) 513–529.