A RSSI-based Algorithm for Indoor Localization Using ZigBee in Wireless Sensor Network

Yu-Tso Chen[1], Chi-Lu Yang[1,2], Yeim-Kuan Chang[1], Chih-Ping Chu[1]
[1]Department of Computer Science and Information Engineering, National Cheng Kung University
[2]Innovative DigiTech-Enabled Applications & Service Institute, Institute for Information Industry
[1]Tainan, Taiwan R.O.C. [2]Kaohsiung, Taiwan R.O.C.
{ p7696147, p7896114, ykchang, chucp}@mail.ncku.edu.tw

Abstract

For the various applications in home automation, the service system requires to precisely estimate user's locations by certain sensors. It is considered as a challenge to automatically serve a mobile user in the house. However, indoor localization cannot be carried out effectively by the well-know Global Positioning System (GPS). In recent years, Wireless Sensor Networks (WSNs) are thus popularly used to locate a mobile object in an indoor environment. Some physical features are widely discussed to solve indoor localization in WSN. In this paper, we inquired about the RSSI solutions on indoor localization, and proposed a Closer Tracking Algorithm (CTA) to locate a mobile user in the house. The proposed CTA was implemented by using ZigBee CC2431 modules. The experimental results show that the proposed CTA can accurately determine the position with error distance less than 1 meter. At the same time, the proposed CTA has at least 85% precision when the distance is less than one meter.

Keywords: indoor localization, home automation, ZigBee modules, wireless sensor networks

1. Introduction

For a large number of applications in home automation, the service system requires to precisely sensing user's locations by certain sensors. Moreover, the system sometimes requires recognizing the time and the weather for making decisions. On the other hand, the users always hope to be served correctly and suitably by the service system in the house. For satisfying the users' demands, one of the most key successful factors is to accurately estimate the user's location. It is considered as a challenge to automatically serve a mobile user in the house.

Indoor localization cannot be carried out effectively by the well-know Global Positioning System (GPS), which is subject to be blockaded in the urban and indoor environments [1-4]. Thus in recent years, Wireless Sensor Networks (WSNs) are popularly used to locate mobile object in the indoor environment. Some physical features are widely discussed to solve indoor localization in WSN. Received signal strength indication (RSSI) is the power strength of radio frequency in a wireless environment. The RSSI value can be regularly measured and monitored to calculate distance between objects. 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 could approximately be estimated. Time difference of arrival (TDOA) is computed based on the emitted signals from three or more synchronized senders. It also refers to a solution of locating a mobile object by measuring the TDOA.

In this paper, we inquired about the RSSI solutions on indoor localization, and proposed a new RSSI-based algorithm and implemented it using ZigBee CC2431 modules in wireless sensor network. The rest of this paper is organized as follows. In Section 2, we briefly introduce the related work on indoor localization in WSN. In Section 3, we first define relevant arguments to describe our algorithm. We then carefully explain the proposed algorithm. In Section 4, the experimental results are analyzed and discussed to validate our algorithm. We show our algorithm is more accurate by comparing with the others methods. The conclusion and future work of the study are summarized in Section 5.

2. Related Work

ZigBee solutions are widely applied in many areas, such as home automation, healthcare and smart energy *(ZigBee Alliance)*. ZigBee is a low-cost, low-power, low data rate and wireless mesh networking standard originally based on the IEEE 802.15.4-2003 standard for wireless personal area networks (WPANs). The original IEEE 802.15.4-2003 standard has been superseded by the publication of IEEE 802.15.4-2006 for extending its features [5, 14]. While many techniques related to ZigBee have also been applied to indoor localization, we choose to focus on 2-dimension localization issues for the following introduction.

2.1 Fingerprinting

The Fingerprinting (FPT) systems are built by analyzing the RSSI features. The RSSI features are pre-stored in a database and are approximately retrieved to locate a user's position [8-11]. The key step of FPT is that the blind node is put at pre-defined anchor positions in advance. By RSSI, the blind node continuously sends requests to its surrounding reference nodes and receives responses from these reference nodes. The FPT system can then continuously record these responses to analyze its features until the analyzed results are characteristically stable. In general, different anchors should be distinct from different RSSI features. In FPT, the mobile object is approximately located by comparing the current RSSI with the pre-stored RSSI features.

Denote a series offline training measurement of reference node k at location L_{ij} is $L=[l_{ij}^{k0},...,l_{ij}^{kM-1}]$ which enables to compute the histogram h of RSSI.

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

The reference nodes are indexed with k. The parameter δ represents the Kronecker delta function [8, 11].

2.2 Real-Time Tracking

The method, which can locate a mobile object by at least three reference nodes without pre-trained database, is named Real-Time Tracking (RTT) [1-4, 6-7]. The RTT System can convert the RSSI to a distance by specific formulas. Trilateration is a method to determine the position of an object based on simultaneous range measurements from at least three reference nodes at known location [1]. Trilateration requires the coordinates of at least three reference nodes (X_i, Y_i) and the distances d_p^i between the blind node and the pre-positioned reference nodes. The target's position $P(X_p, Y_p)$ can be obtained by MMSE [3]. The difference between actual and estimated distance is defined by formula *(2)* where i is a reference position and p is a mobile object.

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

Eq. (2) can be transformed into

$$(d_P^i)^2=(x_i-x_p)^2+(y_i-y_p)^2 \qquad (3)$$

Then Eq. (3) is able to be transformed into

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

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

$$b=A\begin{bmatrix}x_p\\y_p\end{bmatrix} \qquad (5) \qquad\qquad \begin{bmatrix}x_p\\y_p\end{bmatrix}=(A^TA)^{-1}*(A^Tb) \qquad (6)$$

Where

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

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

3. Proposed Algorithm

3.1 Definitions

A blind node refers to a mobile object. A reference node is a fixed node that responds its RSSI to assist locating the blind node. In this study, both the blind node and the reference node are ZigBee modules. In order to describe our proposed algorithm, the following terms are principally defined. These terms are categorized into primitive terms, original physical terms and derived terms. The primitive terms are defined as follows:

$N_{neighbor}$ = *the number of reference nodes which close to blind node within one hop currently*

BID = a pre-defined identification of a blind node, which is a mobile object.

RID = a pre-defined identification of a reference node (a fixed object), where $1 \leq RID \leq N_{neighbor}$

$R_{threshold}$ *[RID][d] = the RSSI of RID within the pre-defined threshold at distance d, where distance d is a set = {d(m) | 0.5, 1, 1.5, 2.0, 2.5, 3.0}*

M_{ACA} = *the mode of approximately closer approach for Tracking (the improved algorithm)*

M_{RTT} = *the mode of Real-Time Tracking*

The values of 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 ZigBee blind node, are defined as follows:

$R_{now}(x)$ = *the current value of the measured RSSI of x, where variable x refers to RID*

rid = an index of R_{now}, *where rid <* $N_{neighbor}$

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

CloserList[x] = a list RID of sorted by $R_{now}(x)$, *where* $R_{now}(x)$ *within* $R_{threshold}[x][d]$ *and* $R_{now}(x) \leq R_{now}(x-1)$, $1 \leq x \leq N_{neighbor}$

SortedList[x] = a list RID of sorted by $R_{now}(x)$, *where* R_{now}(SortedList[x]) $\leq R_{now}$(SortedList[x-1])

ClosestRID = a rid refers to RID, which is the closest node near the blind node (the mobile object; BID), and where R_{now} *(ClosestRID) is within* $R_{threshold}$

C_R = *a record for tracking the mobile object*

M_C = *Current localization mode =* $\{M_C \mid M_{ACA}, M_{RTT}\}$

3.2 Closer Tracking Algorithm

The locating style of the FPT has its own specific advantage and disadvantages, while the RTT style also has its own. The features of the two styles are characteristically complementary. Therefore, we proposed a compound algorithm to determine the usable mode at suitable time. Furthermore, we improved the FPT algorithm at the same time. This idea is also emerged from our observation on elder persons in the house. The elders usually stay on the same positions, such as sofa, table, water cooler or bed. They even frequently stay in front of the television or near the door for a long time. The time they are moving is much less than they are staying, while they are in their house. Since we look forward to provide automatic applications suitably for elders in their house, we can ideally design a position tracking algorithm based on above observation. The proposed algorithm for closer tracking (CTA) was specifically designed to improve the automatic applications. The CTA is carried out by the following four steps.

Step1 – [Build Neighbor List]

The blind node *BID* (the mobile object) periodically receives RSSI (R_{now}) from its neighbor nodes (*RIDs*) by broadcasting its requests. The neighbor nodes will be recorded by comparing their RSSIs with the pre-defined thresholds ($R_{threshold}$). In other words, if the RSSI of the *RID* is within the $R_{threshold}$ at *distance d*, the *RID* will be stored into the *CloserList.*

Step2 – [Determine Mode]

If there are records stored in *CloserList*, the improved FPT will be executed to locate the mobile object. In other words, if there is no record in the *CloserList*, the RTT will be executed for locating the mobile object.

Step3 – [Adapt Assistant Position]

It's likely that there is only one record in the *CloserList*. If the special situation occurs, we should need an extra data structure - *SortedList*. The *SortedList* is an array used to store the ordering *RIDs*, which are sorted by the received RSSIs. Nevertheless, the closest *RID* (*ClosestRID*) should not be stored into the *SortedList*. In next step, the *CloserList* and *SortedList* will be used to locate the mobile object more precisely under M_{ACA} mode.

Step4 – [Approximately Closer Approach]

The improved FPT, which is named approximately closer approach (ACA), is divided into two phases. In the first phase, *ClosestRID* is used to figure out a circular range, since the RSSI of the *ClosestRID* is within the pre-defined threshold at distance *d*. The plane of *ClosestRID* range can be conceptually divided into four sub-planes. In the second phase, the *RIDs* in the *CloserList* will be iteratively retrieved to select the sub-planes for narrowing down the outer range. For example, let's assume the *CloserList = {Ref4, Ref1, Ref5}* and *ClosestRID = Ref3*. In Fig. 1, a virtual circle surrounding the node *Ref3* will be first figured out, since the *ClosestRID* refers to *Ref3*. The plane of *Ref3* range can be conceptually divided into four sub-planes, such as *R1*, *R2*, *R3* and *R4*. In the second round, the sub-plane *R2* will be selected since the *Ref4* is the first *RID* in the *CloserList*. The other *RIDs* in the *CloserList* will be iteratively selected to narrow down the range. The iteration will be stopped until the *CloserList* is empty. The pseudo codes of the CTA, which contains the ACA, are showed in Table 1.

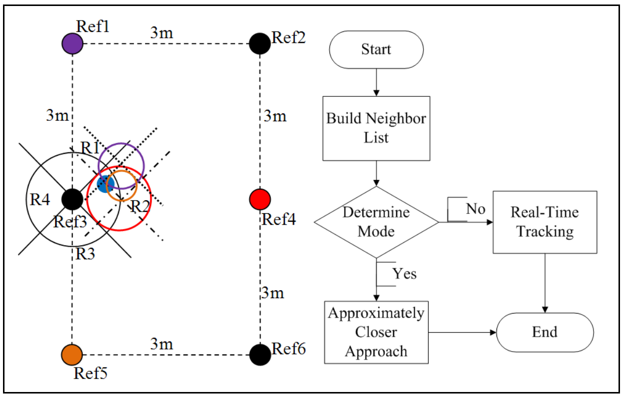

Fig. 1 Concept and flow of the Proposed Algorithm

Table 1 the pseudo codes of the CTA

```
Algorithm_Closer_Tracking(int *R_now)
{
//////Initial////////////////////////////////////////////////////////////////////////////////////
short CloseList [8]={-1};
int k=0;
const int row=3;
const int col=2;
//////Step1 – Build Neighbor List/////////////////////////////////////////////////////
for (dis = 0.5 ; dis <= 2.0 ; dis += 0.5){
    for (rid = 1 ; rid <= N_neighbor ; rid++){
        if (R_now[rid] within R_threshold[rid][dis]){
            CloseList[k] = rid;
            k++;
        }//end if
    }//end for
} //end for loop
//////Step2 – Determine Mode////////////////////////////////////////////////////////
if (k == 0) { // No record in the CloserList
    M_C = M_RTT
    break; //Change to Real-Time Tracking Mode
} //end if
//////Step3 – Adapt Assistant Position//////////////////////////////////////////
//////Only ClosestRID in the CloserList/////////////////////////////////////////
if (k == 1) {
    for (int x = 1; x < N_neighbor ; x++){
        CloseList[x] = SortedList[x-1];
    } //end for
    k = N_neighbor;
} // end if
//////Step4 –Approximately Closer Approach//////////////////////////////////
ClosestRID = CloseList[0];
for (int s = 0 ; s < k ; s++){    //FPT
    switch (CloseList[s+1] - ClosestRID){
        case 1:
            C_R[s] = R2;    break;
        case -1:
            C_R[s] = R4;    break;
        case col:
            C_R[s] = R3;    break;
        case -col:
            C_R[s] = R1;    break;
        default:    //other 4 direction
    } //end switch
} //end for
M_C = M_ACA
} //end Closer Tracking Algorithm
```

4. Implementation and Experiment

The ZigBee modules are used in this experiment. The CC2431 chip stands for the blind node and the CC2430 chips stand for the reference nodes. The specific features of these chips are listed in Table 2, and the figure of CC2431 is showed in Fig. 2. The RSSI values are long-term measured in the experiment, and all the values are stored in a database for further analysis. The proposed CTA is programmed by using the C#.NET language.

4.1 Findings

We measured 1-D RSSI in different environments, which electromagnetic waves are isolated, absorbed or normal. In Fig. 3, the x-axis represents the various distances between a blind node and a reference node, such as 0.5, 1, 1.5, 2.0, 2.5 and 3 meters. The y-axis represents the measured RSSI values. The RSSI values are measured until the statistic results are stable. In order to observe the data, all the measured values are added by one hundred. The statistic results and the standard deviation σ of the stable RSSI are shown in Fig. 3. The σ values are further utilized to define the thresholds.

The following formula provided by Texas Instruments (TI), which represented the relationship between RSSI and the estimated 1-D distance, is shown as follows:

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

While 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 1 meter distance. According to the formula (9), the 1-D distance d can be derived from the measured RSSI values of Fig. 3 and shown in Fig. 4.

Table 2 Features of CC2431

Features	Values
Radio Frequency Band	2.4GHz
Chip Rate(kchip/s)	2000
Modulation	Q-QPSK
Bit rate(kb/s)	250
Sensitivity	-92dBm
Data Memory	8KB
Program Memory	128KB internal RAM
Spread Spectrum	DSSS

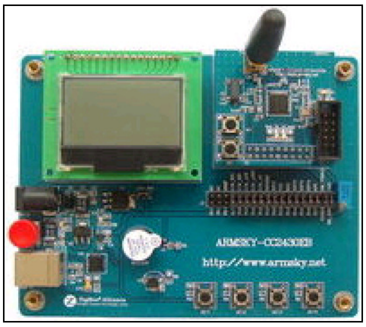

Fig. 2 CC2431 module

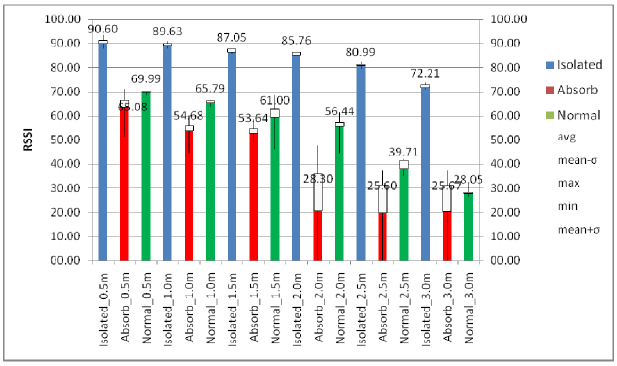

Fig. 3 RSSI thresholds

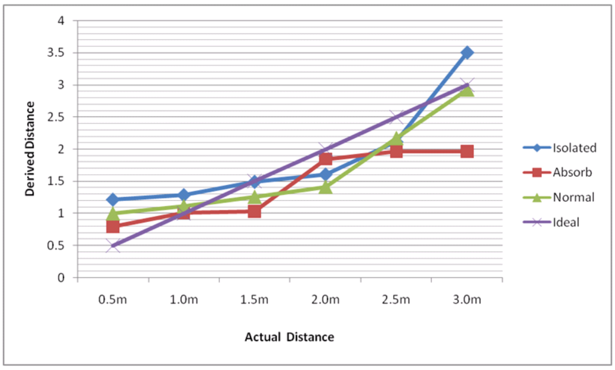

Fig. 4 Actual distance and derived distance (A, n) with Isolated (6, 4); Absorb (45, 10); Normal (30, 9)

4.2 Experimental Results

In this experiment, 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 and represent by an Error Distance formula as follows:

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

In order to validate accuracy of the proposed CTA, we implemented and compared the proposed CTA with the FPT [9] and RTT [12], which are experimented by using the CC2431 location engine. The experimental results are shown in Fig. 5 and Fig. 6. The x-axis represents the distance from the blind node to the closest reference node. The y-axis represents the difference between an actual position and the estimated position.

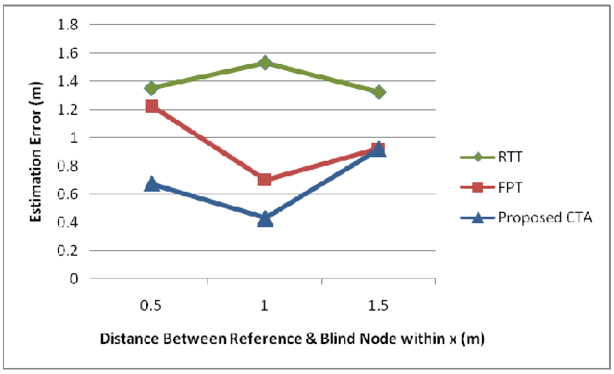

Fig. 5 Estimation errors at distance {0.5, 1.0, 1.5} meters (Accuracy)

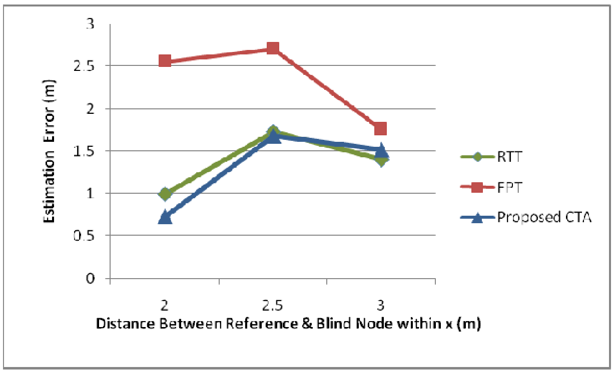

Fig. 6 Estimation errors at distance {2.0, 2.5, 3.0} meters (Accuracy)

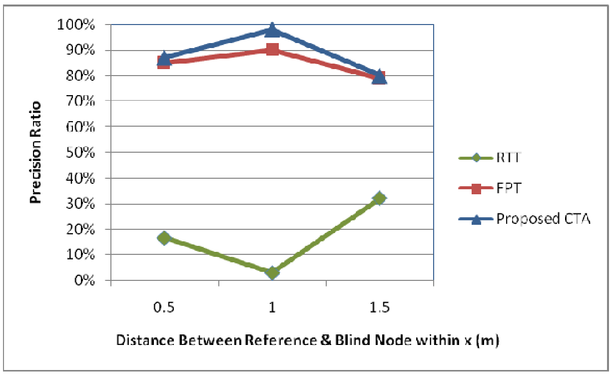

Fig. 7 Precision when error distance within 1.0 m

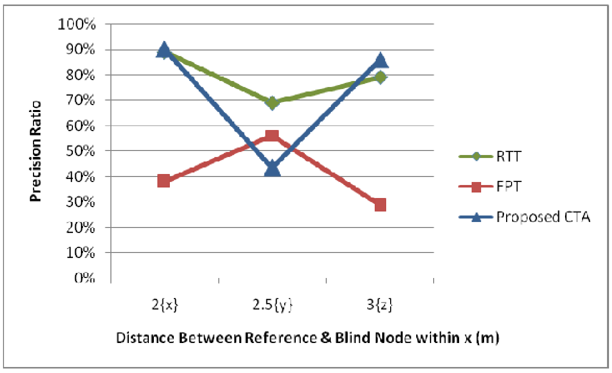

Fig. 8 Error distance {x} within 1.3m;
Error distance {y, z} within 1.7m

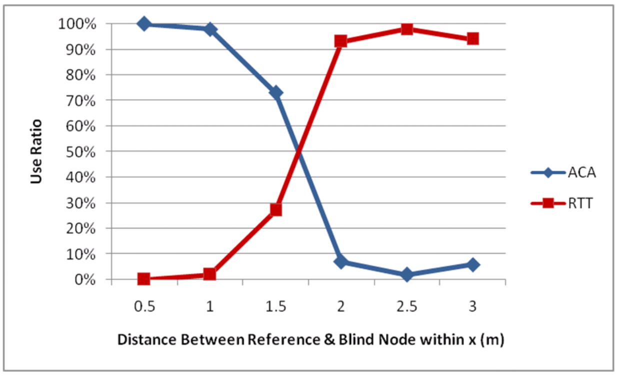

Fig. 9 Usage ratio of ACA & RTT modes

As we can see from the experimental results in Fig. 5, when a blind node approaches to any reference node, our algorithm can accurately determine the position with error distance less than 1 meter. The accuracy of the CTA is better than the other methods. At the same time, the FPT method is accurate enough when the blind node is moving close to the pre-trained positions. Furthermore, the estimation errors calculated by CC2431 are quite stable in Fig. 5, and the accuracy of RTT method is quite independent of the positions of the reference nodes.

In Fig. 6, the distances from the blind node to the closest reference node are increased. Therefore, the RSSI values are more interfered by background noise, and the variances are increased. In FPT method, the signal features are diminished, so that the estimation errors are obviously increased. In other words, the FPT method cannot determine the position accurately when the distance from the blind node to the closest reference node is more than two meter. Under this condition, our proposed CTA changed the operational mode from the ACA to the RTT mode. As a result, the accuracy of the proposed method is close to those of the RTT method. In the case of x = 2.0m, the proposed CTA is slightly more accurate than the RTT method. In the other case of x = 3.0m, the proposed CTA is slightly worse than the RTT method.

In Fig. 7 and Fig. 8, we show the precision of the proposed CTA, the FPT, and the RTT. The precision is defined as follows:

$$\frac{Number\_of\_within\_Acceptable\_Error\_Distance}{Total\quad Estimated\quad Times} \qquad (11)$$

For the experimental design in Fig. 7, the acceptable error distance is set as 1 meter. Under this condition, the estimation errors, which values are less than or equal to 1 meter, are selected to calculate precision. As we can see, the proposed CTA has at least 85% precision when the distance is less than one meter. The CTA has higher precisions than the other methods. In Fig. 8, the precision is low yet in the case of x=2.5. This is because that most estimated errors stay in the range of 1.5 and 1.8. That's an interesting situation.

We showed the mode-changed functionality of the proposed CTA at various distances. The usage ratios of the ACA and the RTT are displayed in Fig. 9. As we can see, the ACA method is useful if the distance is less than 1.5 meters. Furthermore, the ACA mode will be changed to RTT if the distance increases over 1.5 meter. The mode-changed operation can be practically made according to the threshold we set. As a result, the proposed CTA can select an adaptive mode to obtain more precise location. The usage ratios of ACA and RTT are showed in Fig. 9.

5. Conclusion and Future Work

In this paper, we inquired about the RSSI solutions on indoor localization, and proposed a new RSSI-based algorithm using ZigBee CC2431 modules in wireless sensor network. Moreover, we improved the FPT algorithm at the same time. The mode-changed operation of the proposed CTA is even designed for combining the improved FPT and the RTT methods. The functionality can adapt the operational modes according to the thresholds, which we set and mentioned in the findings. As a result, the proposed CTA can suitably select an adaptive mode to obtain more precise locations. The experimental results show that the proposed CTA can accurately determine the position with error distance less than 1 meter. At the same time, the proposed CTA has at least 85% precision when the distance is less than one meter.

For the various applications in home automation, the proposed CTA can be applied to provide correct and suitable services by estimating user's locations precisely. In the future, the proposed CTA can even bring promising quality of services on caring elders in the house. At the same time, we will try to improve the real-time tracking algorithm of the CTA for increasing the accuracy of the uncovered ranges, which positions are beyond the reference nodes.

Acknowledgements

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

References

[1] Erin-Ee-Lin Lau, Boon-Giin Lee, Seung-Chul Lee, and Wan-Young 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, Vol. 1, No. 2, June 2008.

[2] Youngjune Gwon, Ravi Jain, and Toshiro Kawahara, "Robust Indoor Location Estimation of Stationary and Mobile Users," in Proc. of IEEE INFOCOM, March 2004.

[3] Masashi Sugano, Tomonori Kawazoe, Yoshikazu Ohta, and Masayuki Murata, "Indoor Localization System Using RSSI Measurement of Wireless Sensor Network Based on ZigBee Standard," in Proc. of Wireless Sensor Networks 2006 (WSN 2006), July 2006.

[4] Stefano Tennina, Marco Di Renzo, Fabio Graziosi and Fortunato 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.

[5] IEEE 802.15 WPAN™ Task Group 4, IEEE 802.15.4-2006, http://www.ieee802.org/15/pub/TG4.html

[6] Allen Ka, and Lun Miu, "Design and Implementation of an Indoor Mobile Navigation System," Master thesis of CS at MIT 2002.

[7] P. Bahl and V. Padmanabhan, "RADAR: An In-Building RF-based User Location and Tracking System," In Proceedings of the IEEE INFOCOM 2000, March 2000, pp. 775-784.

[8] Angela Song-Ie Noh, Woong Jae Lee, and Jin Young Ye, "Comparison of the mechanisms of the ZigBee's indoor localization algorithm," in Proc. SNPD, pp.13-18, 2008.

[9] Qingming Yao, Fei-Yue Wang, Hui Gao, Kunfeng Wang, and Hongxia Zhao, "Location Estimation in ZigBee Network Based on Fingerprinting," in Proc. IEEE International Conference on Vehicular Electronics and Safety, Dec 2007.

[10] Shashank 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, vol. 3, pp. 1360-1364, June 2004.

[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] ZigBee Alliance, ZigBee Specification Version r13, San Ramon, CA, USA, Dec. 2006.