Improvement in UWB Indoor Positioning by Using Multiple Tags to Filter Positioning Errors

Ming-Fong Tsai[1], Thanh-Nam Pham[2,3], Bo-Cai Hu[2], Fang-Rong Hsu[2]

[1] Department of Electronic Engineering, National United University, Taiwan, R.O.C.
[2] Department of Information Engineering and Computer Science, Feng Chia University, Taiwan, R.O.C.
[3] Department of Electronics and Communications Technology, Thai Nguyen University, Vietnam.
mingfongtsai@gmail.com, namfet@gmail.com, jhougo7928@gmail.com, and frhsu@fcu.edu.tw

Abstract

In this paper, we introduce an indoor-positioning system using ultra-wideband radio signals. To enhance the accuracy of indoor positioning, in our study, we propose to use multiple anchors installed at the same locations to filter positioning error to reduce the instability caused by receiving signals. In addition, we also propose a filtering algorithm referring to the absolute position and moving-direction information of the positioned object and a prediction method to predict the next position of the positioned object based on previous coordinates. When the error values are out of the acceptable range, it can adopt prediction results to conduct calibration using a control object. From the experimental results, our proposed method is effective in enhancing the accuracy of indoor positioning compared to other related works.

Keywords: Indoor positioning, Ultra-wideband, Filtering algorithm, Automated guided vehicle

1 Introduction

In recent years, indoor positioning technology is becoming a key technology of smart applications. Traditional satellite positioning technology fails to provide sufficient accuracy to position objects indoors. At the present time, there are many solutions for an indoor positioning system, such as Bluetooth, wireless, ultrasound, infrared, and video connection methods. These existing systems experience some problems such as short-range positioning and fluctuating received signals. The accuracy of these systems is limited to acceptable levels for their applications. For example, these systems often have a positioning error greater than 50 cm, especially systems using Bluetooth technologies with errors greater than 1 m. These systems are inefficient or do not meet the requirements for positioning systems in industrial or complex environments. These environments require high accuracy and real-time operation. In order to meet the stringent requirements of industrial environments, ultra-wideband (UWB) positioning technology has been introduced and developed. The systems based on Ultra-Wideband radio signals are among the most promising solutions and are becoming more and more popular [1-4]. The use of UWB radio signals in indoor positioning systems helps achieve positioning accuracy with ranging error of the order of centimeters and helps reduce the negative effects of multipath propagation. However, existing UWB-based systems are not yet effective in filtering out the large deviations of received signals. This leads to them still not being good enough to meet the strict requirements of industrial systems such as small positioning error, orientation error, and end-to-end delay. In this paper, we propose a method to solve this problem. Unlike other positioning systems using UWB, our system uses multi-anchors installed in the same positions to filter out errors and accurately predict the direction of movement of the object. We introduce a method to develop an information filtering algorithm and a prediction algorithm to avoid false positioning values and estimate the possible path via vector prediction.

Our proposed method uses absolute distance information to filter false positioning values. Its concept is based on the implementation of mutually overlapping indoor-positioning modules as two identical items of equipment that generate the same absolute coordinates and absolute distance and utilize this information to conduct analysis and filtering. This is combined with inertial navigation and a virtual map to diagnose whether the positioning coordinates and orientation of the next moment match a reasonable position and to conduct filtering. When they do not match a reasonable position, the information is excluded and the possible position is predicted according to the vector formed from the moving tracks. We tested the proposed method on an automated guided vehicle (AGV) in a test room, and the results

*Corresponding Author: Ming-Fong Tsai; E-mail: mingfongtsai@gmail.com
DOI: 10.3966/160792642019052003003

achieved showed higher accuracy than other related methods. Our proposed methods can achieve high accuracy (1-30 cm) very well suited to requirements for real-time indoor navigation and tracking of AGVs.

The remainder of this paper is organized as follows: Section 2 presents the related works. Section 3 provides a description of the proposed method and algorithms. Section 4 reports the simulation results of the system and presents a comparison of the proposed system with other related systems. The final section presents the conclusion of our paper.

2 Related Works

Currently, many home-positioning systems have been proposed. However, these systems have large tolerances that are unsuitable for high-accuracy applications such as industrial ones. Lin et al. [5] built an indoor positioning system using iBeacon based on Bluetooth Low Energy (BLE) technology. They designed a medical application service for handheld devices for hospitals to obtain position information of patients and medical equipment. This system is sufficiently accurate to satisfy the medical staff's need to track the locations of patients. However, the positioning error is greater than 1 m and the system does not work well in an unstable signal environment such as a garage in a basement or industrial environment. Rida et al. [6] proposed an indoor positioning system based on the RSSI (Received Signal Strength Indicator) of the Bluetooth Low Energy 4.0 (BLE 4.0) technology. In their system, they installed equally spaced nodes on ceilings which enter sleep mode when there are no objects approaching and utilized the three nodes with the best positioning signals to deal with approaching objects using a trilateral positioning algorithm. However, its positioning accuracy error is up to 1 m.

Chang et al. [7] proposed a method to solve the problem of instable RSSI signals in the systems using BLE technology. Their method used distributed overlapping beacons so that the position information of the objects to be measured can be predicted via the signal intensity of the received RSSI. The RSSI deviation can be filtered out, which can reduce the probability of unstable RSSIs. However, positioning methods based on RSSI signals are always subject to constant fluctuations in signal strength, and as a result the positioning accuracy error of this method ranges from 50 cm to 1 m. In order to improve the positioning accuracy, the above methods can all simply use filtering of analysed data. When there is only one set of data, filtering can be conducted by comparing the distance difference, time difference, and vectors. On the other hand, we can simply use an absolute standard to conduct matching when the data are from the same environment and are generated at the same time to obtain better filtering results.

Systems using ultra-wideband technology were introduced in previous studies [1-2]. These systems proposed to solve the problem of multi-path environments and to locate objects at a long distance in complex industrial environments. These methods achieve high accuracy in the centimetre range and are suitable for deploying applications in industrial environments. However, the evaluation of these methods is not based on the different moving trajectories of the object but rather is almost always based on the straight trajectory of the object. These methods also do not provide a way to filter out the deviations of the positioning results that exceed the allowable error so that the positioning object can be adjusted to achieve higher accuracy. The authors in the studies [8-9] also proposed methods using a Kalman filter to estimate the angle of the moving object and predict the direction of the motion. Based on these studies and in combination with the inertial navigation methods as in [10], we proposed an algorithm for motion vector filtering and motion vector prediction.

3 Proposed Method

3.1 Two-way Ranging Method

There are many positioning algorithms that can be used in UWB technology based on the estimated time of flight (TOF) [time of arrival (TOA) or time difference of arrival (TDOA)] over the angle of arrival or received signal strength localization for UWB. However, these methods encounter problems in synchronizing the time between anchors. In this paper, we propose to use the two-way ranging (TWR) method to estimate the TOA value.

The basic TWR procedure for communicating between tag and anchor is illustrated in Figure 1. To measure distance, two messages need to be exchanged. The tag initializes TWR by sending a poll message to the known addresses of all of the anchors in the test room in a time referred to the T_{SP} (time of sending poll). The anchor records the time of poll reception (T_{RP}) and replies with the response message at time T_{SR}, including the message ID, T_{RP}, and T_{SR}. After that, the tag will receive the response message and record the time T_{RR}. Based on T_{SP}, T_{RR}, $\mathrm{T_{RP}}$, and T_{SR}, the TOF will be estimated (TOF_{est}) and hence the distance to each individual anchor will be deduced. The calculation of the current position of the vehicle in the test room is derived from Eq. (3).

From this figure, we can estimate the distance and TOF_{est} as follows:

$$distance = TOF_{est} \times c \tag{1}$$

where c is the speed of light ($c \approx 3 \times 10^8\, m/s$) and

$$TOF_{est} = \frac{\left[\left(T_{RR} - T_{SP}\right) - \left(T_{SR} - T_{RP}\right)\right]}{2} \quad \textbf{(2)}$$

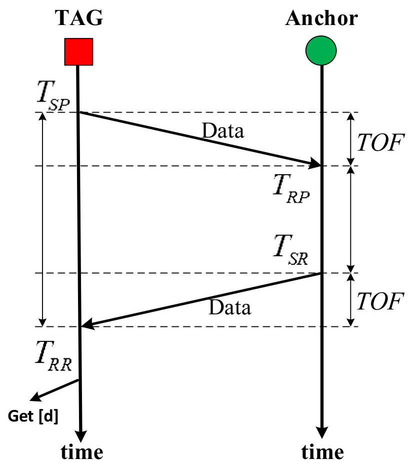


Figure 1. Estimation of the TOF between tag and anchor

Let *S(X, Y)* be the current position of the AGV in the room. Figure 2 describes the method for determining the object's coordinates in the test room.

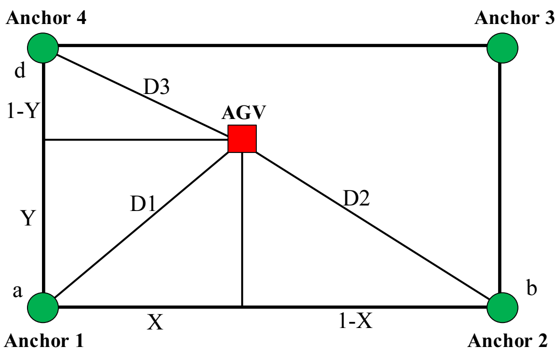


Figure 2. Location of AGV by using the TWR method

As shown in Figure 2, we can calculate the current location of the AGV *S(X, Y)* using the Pythagorean theorem in triangular.

$$S\left(X,Y\right) = \begin{cases} X = \dfrac{D_1^{\ 2} - D_2^{\ 2} + 1}{2} \\ Y = \dfrac{D_1^{\ 2} - D_3^{\ 2} + 1}{2} \end{cases} \quad \textbf{(3)}$$

where D_1, D_2, and D_3 are the relative distances between the vehicle tag and anchors. These values are estimated based on the TWR TOA, as mentioned above.

3.2 Indoor-positioning Algorithm Based on Absolute Distance

In this paper, we propose a positioning method based on absolute distance. We installed two tags on a positioned object (AGV) to obtain its absolute distance, as shown in Figure 8. When two tags are overlapping at the same location, we have an absolute distance value of zero, as shown in Figure 8(a). In contrast, an absolute distance greater than zero is shown in Figure 8(b). Let us assume that the positions of two tags T_1 and T_2 obtained are $A_1\left(x_1, y_1\right)$ and $A_2\left(x_2, y_2\right)$. By calculating the absolute distance $\overline{AB}$ at each positioning time, we can determine the location of the object (AGV) at these times. We have a limit parameter *P* called the margin of absolute distance error. The value of the absolute distance is compared with this margin to determine the location of the object. At each time of positioning, when the value of the absolute distance is within ±*P*, then the positional data are stored and the location of the object is obtained by taking the average of the coordinates of points A_1 and A_2. In cases where this value exceeds the range of the margin of error, we will filter and remove this positioning result. Our method is illustrated in Figure 3.

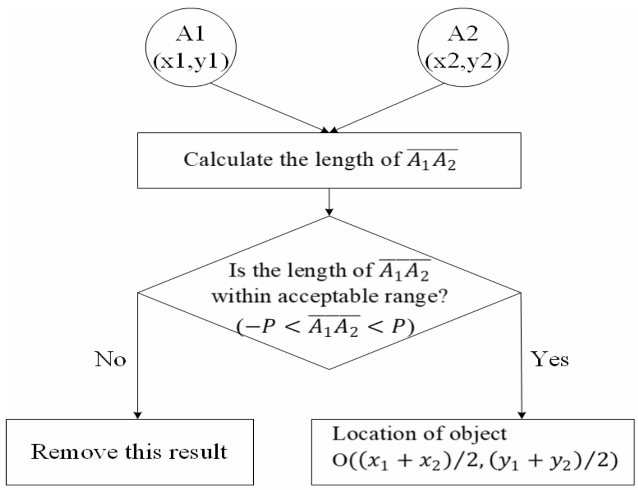


Figure 3. Proposed algorithm of positioning method based on absolute distance

3.3 Filtering Methods

In our paper, we adopt three filtering methods to process the received positioning data of two tags. The data that need to be considered are the timestamp, absolute distance, and motion angle of the positioned object; these data will be processed to decide whether to serve or remove. By using these three parameters, we will obtain three margins of errors corresponding to them. These margins will be used to eliminate irrelevant values. We rely on Algorithms 1 and 2 to process these data as described below. Our algorithms will carry out filtering according to the preset deviation of the timestamp, absolute distance, and motion angle. To obtain the positioning information value generated at the same time, if the timestamp difference between T_1 and T_2 is out of the allowed range, it must be filtered. Then, to obtain the positioning information value matching the absolute distance, if the distance between two points A_1 and A_2 is out of the allowed range, it must be filtered. Filtering data on the timestamp and absolute distance are described in the Algorithm 1. If the data of the absolute distance are removed, we need to adopt the range of change in angle of the moving path to filter the indoor-positioning position of the motion vector that is using the previous two moving angles to predict the next possible position of the object as shown in Algorithm 2. The filtered indoor positioning value will predict the next coordinates according to the previous coordinates via motion vector prediction as shown in Algorithm 3. Figure 4 illustrates the prediction of the moving angle when there is only one set of anchor hardware. When the system has obtained the position coordinates of O_1 and O_2 according to the history information, it can estimate angle θ_1 information. When the O_3 position coordinate has been obtained, it can refer to the O_2 position coordinate, estimate angle θ_2 information, and determine whether the difference in angle between θ_1 and θ_2 is within the allowed range. If the O_3 position coordinate is excluded by the algorithm, the next position coordinate O_4 will be taken to estimate the angle with the O_2 position coordinate. Our system will receive more than 10 position coordinates every second. It can effectively exclude unreasonable indoor position coordinates and ensure the accuracy of object positioning.

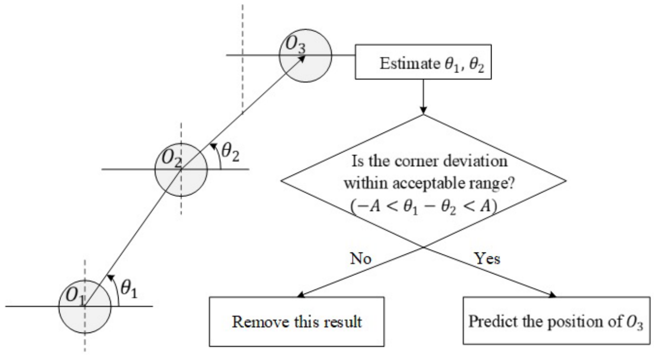


Figure 4. Method of prediction of moving angle

Algorithm 1. Timestamp and Absolute Distance Filtering Method

Input: Tag1 Data T1(longitude, latitude, time), Tag2 Data T2(longitude, latitude, time)

Output: filter true or false

```
List [] point_temp, point_temp2
If filter_time (T1[time], T2[time])
    if filter_distance (T1[longitude, latitude], T2[longitude, latitude])
        switch (point_temp2.length)
            case 0:
                point_temp2[0] = (middle (T1, T2))
                return true
            case 1:
                if Check_possibility (point_temp2[1], middle (T1, T2))
                    point_temp2[1] = middle (T1, T2)
                    return true
                else
                    return false
                end if
            case 2:
            if Algorithm1(point_temp2[0], point_temp2[1], middle (T1, T2)) and Check_possibility
            (point_temp2[2], middle (T1, T2))
                    point_temp.add (T1, T2)
                    return true
                else
                    return false
                end if
    else if point_temp.length > 3
        i = point_tem.length
        return Algorithm3(point_temp, middle (T1, T2))
    else
        add_error_distance_report (T1, T2)
        return false
    end if
Else
    add_error_time_report (T1)
    return false
End if
```

Algorithm 2. Motion Vector Filtering

```
Input: point1 (longitude, latitude), point2(longitude, latitude), point3(longitude, latitude)
Output: true or false
If check_same_point (point1, point2, point3)
     return false
Else
     Sita1 = find_angle (point1, point2)
     Sita2 = find_angle (point2, point3)
     return filter_angle (sita1, sita2)
End if
```

Algorithm 3. Motion Vector Prediction

```
Input: point_temp[], point (longitude, latitude)
Output: true or false
If check_same_point (point_temp[i-1], point_temp[i-2], point_temp[i-3], point_temp[i-4], point5)
     return false
Else
     x = find_mean_deviation_longitude (point_temp[i-1], point_temp[i-2], point_temp[i-3], point_temp[i-4])
     y = find_mean_deviation_latitude (point_temp[i-1], point_temp[i-2], point_temp[i-3], point_temp[i-4])
     write (point5.x + x, point5.y + y)
     return true
End if
```

4 Simulation Results

4.1 Scenario Setup

To evaluate our proposed method, we implemented algorithms on an AGV in a test room 750 cm in length and 500 cm in width. The testing room simulated industrial environments. An industrial AGV was adopted and was run on the track at a constant speed. As shown in Figure 5, the black line represents the actual fixed path of movement. Then we installed two overlapping tags on the AGV to overlap position and used two sets of anchors on the four walls for data analysis to verify the efficiency of the proposed algorithm. To accurately evaluate the results of our tests, we ran multiple experiments and obtained the average statistical result of these runs. For performance evaluation of the system, we also deployed other related methods in the same test run scenario. The requirements for real-time indoor navigation and tracking of AGV of our study are described in Table 1.

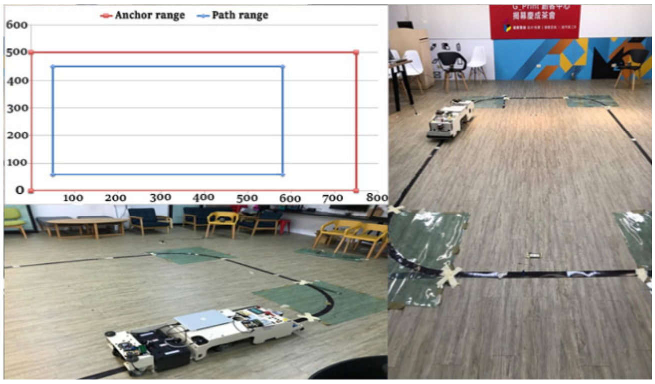


Figure 5. Controlling AGV in test room

Table 1. System requirements

Criteria	Requirements
Position error	1-30 cm
Oriention error	1-90 degress (dynamic case)
Update rate	10 Hz-1 KHz
End-to-End delay	<100 ms

4.2 Comparison of Indoor Positioning Methods

As mentioned above, we carried out an efficiency comparison between our test results and the results of related work done by Lin et al. [5], Rida et al. [6], and Chang et al. [7]. The results are shown in Figure 6. The error of the indoor-positioning position obtained by the proposed method is within ±10 cm, which occupies 53.85% of the overall, within ±20 cm, which occupies 86.10% of the overall, and within ±30 cm, which occupies up to 94.85% of the overall. Compared to related works done in recent years, our method has better filtering efficiency. The method of Lin et al. achieves the worst performance because the positioning error of this method is large (greater than 1 m), the beacon density used in this method is low, and the accuracy of positioning depends on only one beacon. The method of Rida et al. has greater accuracy, because they use three beacons to estimate the user's position. However, the error of this method is still large (up to 1 m). The method of Chang et al. achieves higher accuracy than those of Lin et al. and Rida et al. because their method filters out the deviation of the RSSI signal when using the beacon technology. However, the use of beacons has the disadvantage of a short working distance (in the range of 20 to 50 m). The frequency of beacon messages depends on the device. The beacon method does not work well in unstable signal environments and industrial environments. Therefore, our proposed method always achieves the best performance.

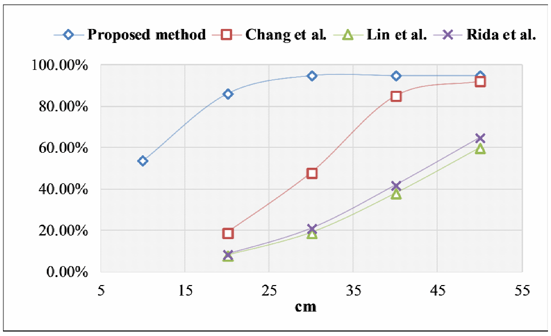


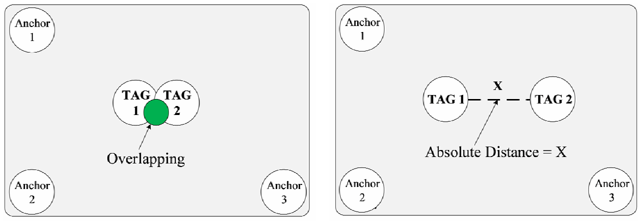
Figure 6. Efficiency comparison between proposed method and other related works

4.3 Data Analysis of Absolute Distance

In this section, we will evaluate the effect of the absolute distance between the positions of two tags on the performance of the system.

4.3.1 Absolute Distance Equal to Zero

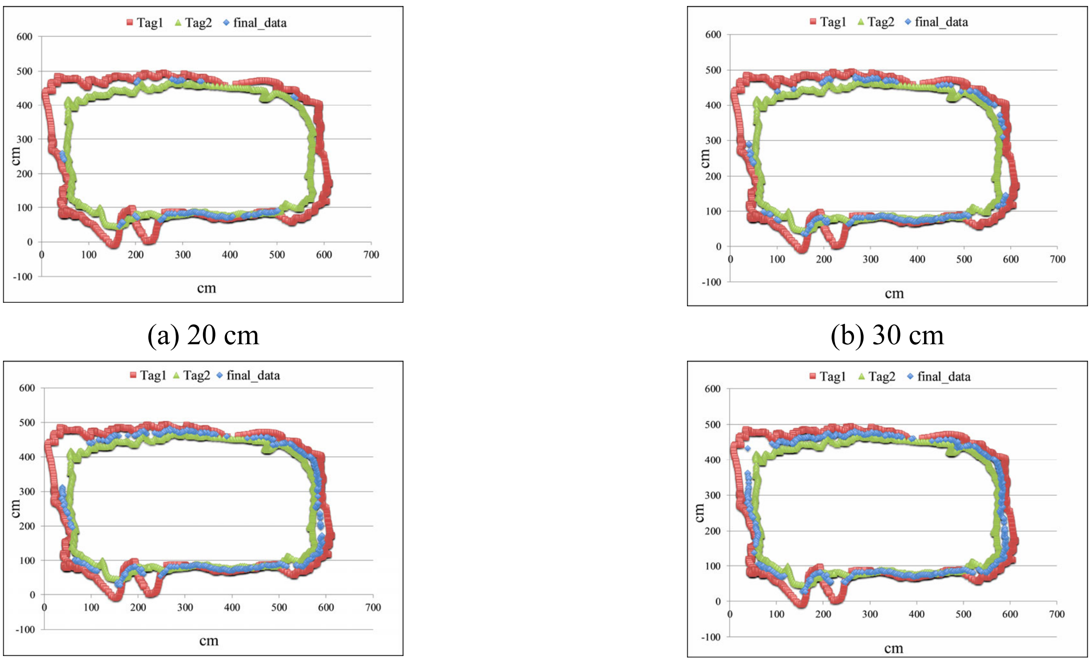
To filter the deviation of positional errors, we use absolute distance information. Figure 7 illustrates the method of determining the absolute distance between two tags. We install a set of anchors on four walls and two overlapping tags on the AVG to obtain a test scenario with absolute distance equal to zero, as shown in Figure 7(a). In this case, the margins of error of the parameters are set as follows: the allowed time error is set as 0.05 s, the allowed distance error is set as 40 cm, and the allowed angle error is set as ±50°.



(a) equal to zero (b) greater than zero

Figure 7. Absolute distance

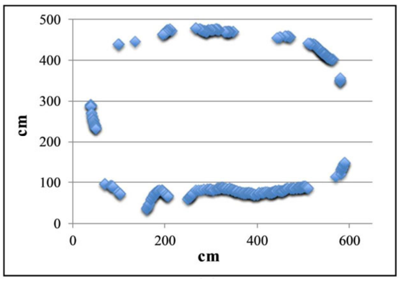
In Figure 8, the red line represents the data value obtained for tag 1, the green line indicates the data value of tag 2, and the blue line represents the data value obtained after filtering. From the experimental results shown in Figure 8, because the system filters too much indoor-positioning position information and fails to obtain the overall path, it is found that when the allowed distance error is smaller, more indoor-positioning positions will be left and the probability of obtaining false indoor-positioning position information will be higher. Then we changed the allowed time error to 0.001, 0.005, 0.01, and 0.05 s. As shown in Figure 9, the data indicated that the time factor has little effect on the overall performance because this system provides 10 indoor positioning data every second. In Figure 10, we changed the allowed angle error to 10°, 30°, 50°, and 70°. It is found that when the allowed angle error is smaller, more indoor positioning positions will be left and the probability of obtaining false indoor-positioning position information will be higher.



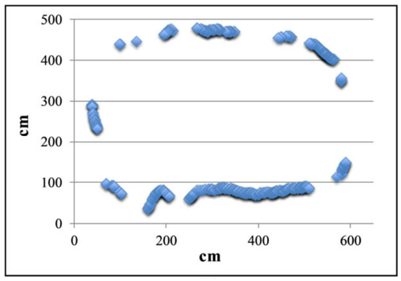
(a) 20 cm (b) 30 cm

(c) 40 cm (d) 50 cm

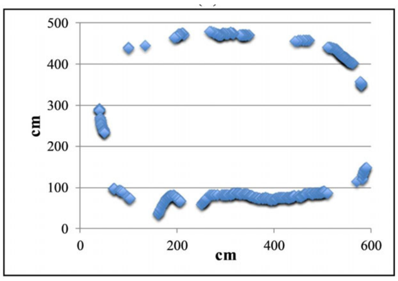
Figure 8. Allowed distance error for an absolute distance equal to zero



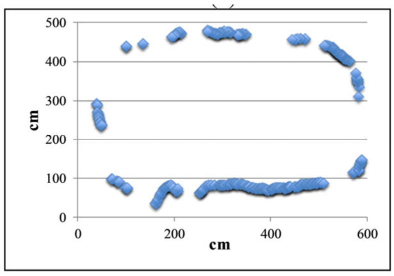
(a) 0.001 s



(b) 0.005 s

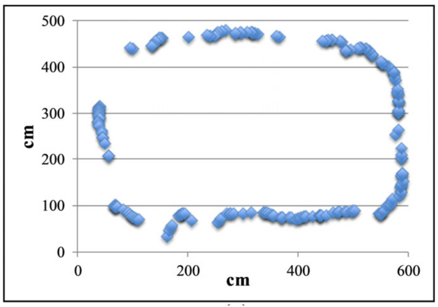


(c) 0.01 s

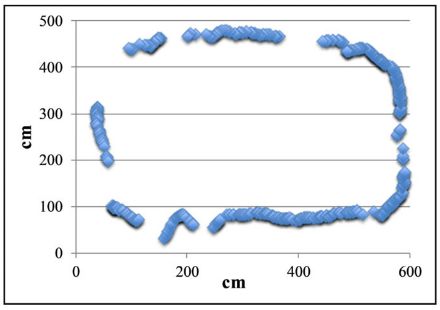


(d) 0.05 s

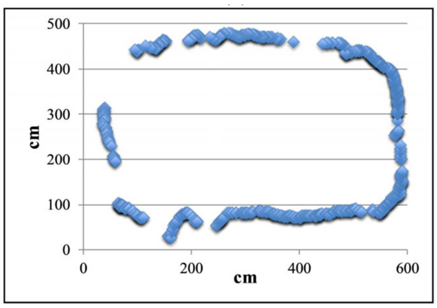
Figure 9. Allowed time error of (a)-(d) for an absolute distance equal to zero



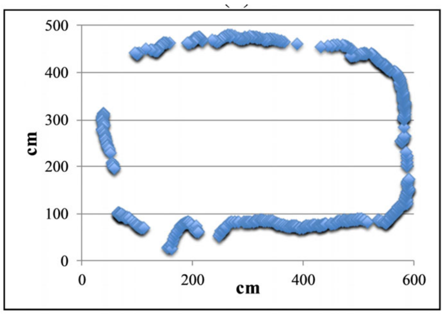
(a) 10°



(b) 30°



(c) 0.01 s



(d) 70°

Figure 10. Allowed angle error of (a)-(d) for an absolute distance equal to zero

Figure 11 shows the coverage rate under various changes in conditions. When the margin of the absolute distance error is set as 40 cm, the time error is set as 0.05 s, and the angle error is set as 50°, we can obtain the best filtering results. As shown in Figure 12, we obtain a visual difference for results after using a multi-tag sampling method and actual path of movement. From Figure 13, we obtained low accuracy when the margin of error is less than 30 cm; when the error margin was greater than 30 cm, we achieved almost 100% accuracy.

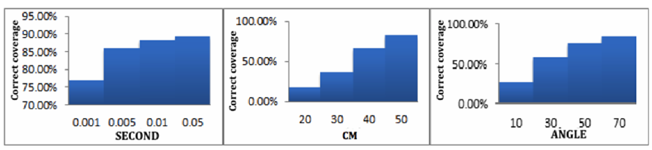


Figure 11. Correct coverage rate of various changing conditions of absolute distance equal to zero

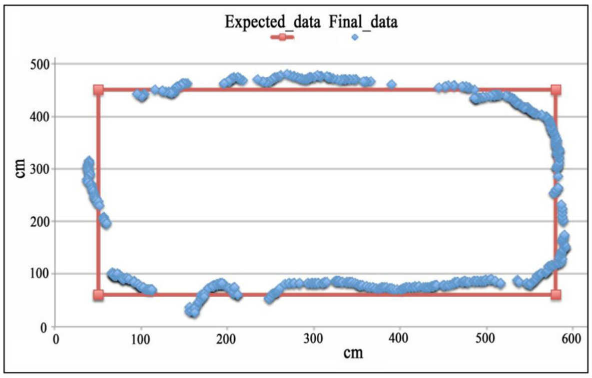


Figure 12. Comparison of proposed method after filtering and actual path in the case of an absolute distance equal to zero

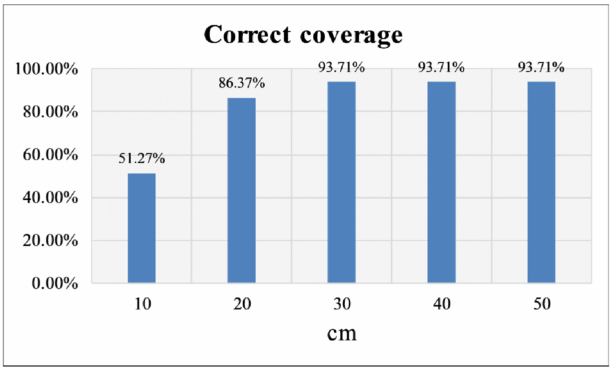
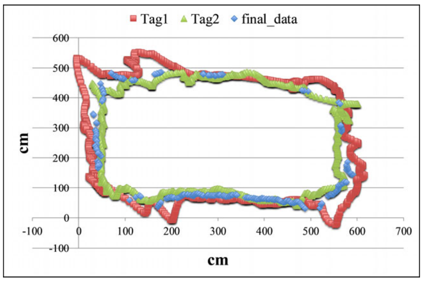


Figure 13. Correct coverage rate with various margins of distance error for an absolute distance equal to zero

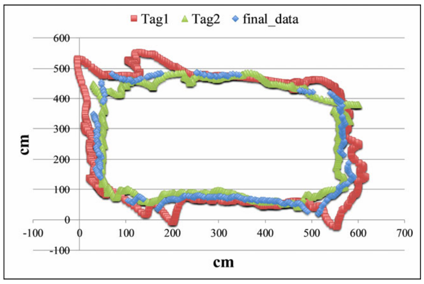
4.3.2 Absolute Distance Greater Than Zero

Another case that we consider is an absolute distance greater than zero. As shown in Figure 7(b), we install a set of anchors on four walls and two overlapping tags on the AGV to obtain a test scenario with an absolute distance greater than zero. The absolute distance is set as 85 cm, the allowed time error is set as 0.05 s, the allowed distance error is set as 20 cm, and the allowed angle error is set as ±30°.

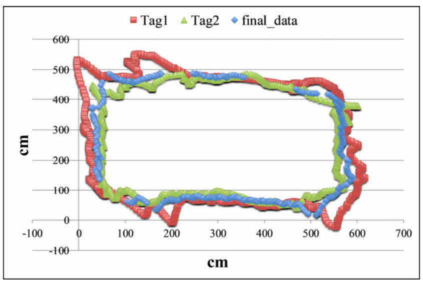
In the case with an absolute distance of 85 cm, we varied the allowed distance error between 10, 20, 30, and 40 cm. In Figure 14, the red line represents the data value obtained for tag 1, the green line indicates the data value for tag 2, and the blue line represents the data value obtained after filtering. From the experimental results, it is found that when the allowed distance error is smaller, more indoor-positioning positions will be left and the probability of obtaining false indoor-positioning position information will be higher. Then we varied the allowed time error between 0.001, 0.005, 0.01, and 0.05 s. As shown in Figure 15, the data indicated that the time factor has little effect on the overall performance because this system provides 10 indoor positioning data every second. Finally, we varied the allowed angle error between 10°, 30°, 50°, and 70°, as shown in Figure 16. It is found that when the allowed angle error is smaller, more indoor-positioning positions will be left and the probability of obtaining false indoor-positioning position information will be higher. As shown in Figure 17, for a correct coverage rate under various different conditions, when the absolute distance is set as 20 cm, the time error is set as 0.05 s, and the angle error is set as 30°, we can obtain the best filtering results. As shown in Figure 18, we obtain visual difference for results after using multi-tag sampling method and actual path of movement. From Figure 19, we obtained low accuracy when the margin of error is less than 30 cm; when the margin is greater than 30 cm, we achieve almost 100% accuracy.



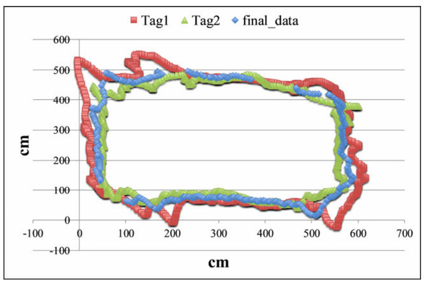
(a) 10 cm



(b) 20 cm

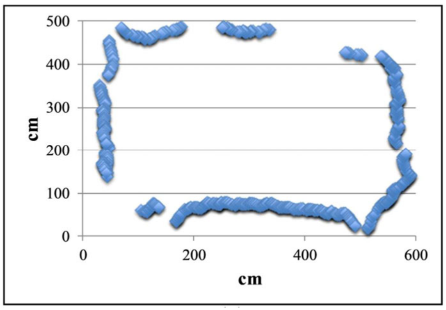


(c) 30 cm

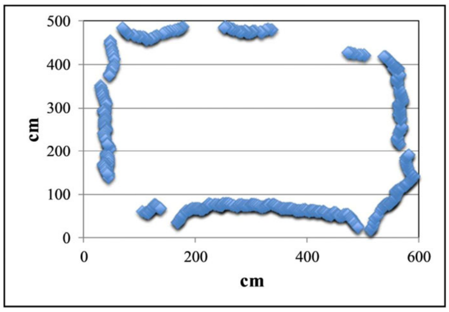


(d) 40 cm

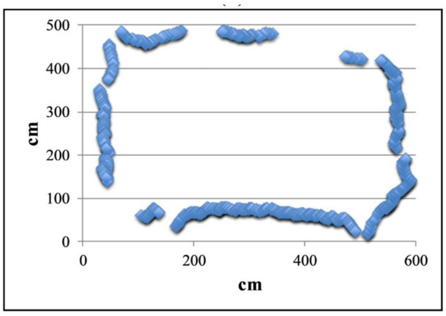
Figure 14. Allowed distance error for an absolute distance greater than zero



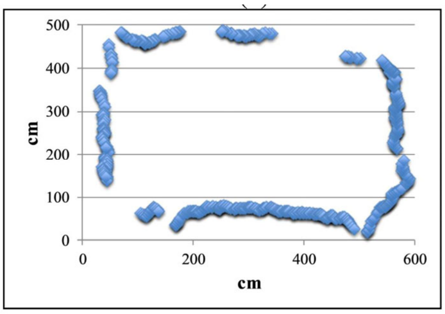
(a) 0.001 s



(b) 0.005 s

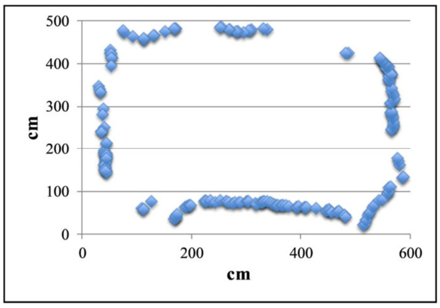


(c) 0.01 s

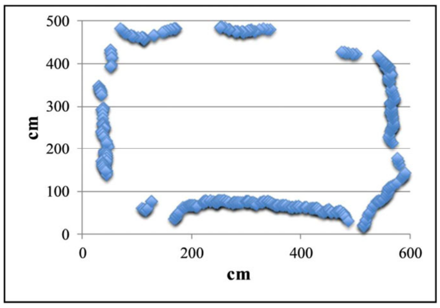


(d) 0.05 s

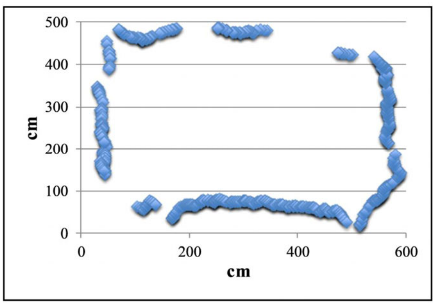
Figure 15. Allowed time error of (a)-(d) for an absolute distance greater than zero



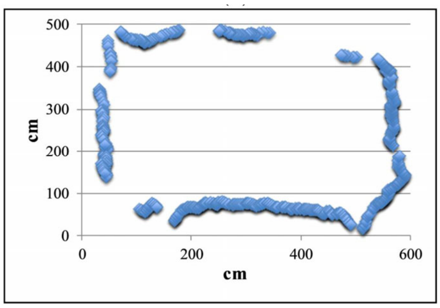
(a) 10°



(b) 30°



(c) 50°



(d) 70°

Figure 16. Allowed angle error of (a)-(d) for an absolute distance greater than zero

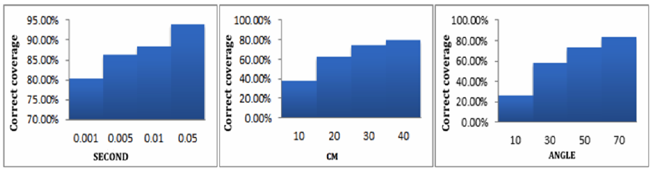


Figure 17. Correct coverage rate under various conditions for an absolute distance greater than zero

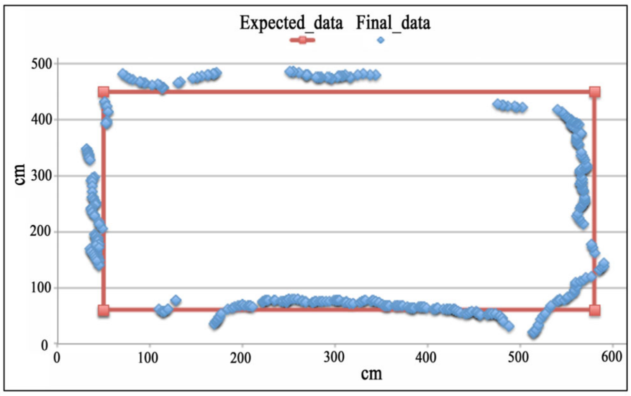


Figure 18. Comparison of the results of the proposed method after filtering and the actual path of movement in the case of an absolute distance greater than zero

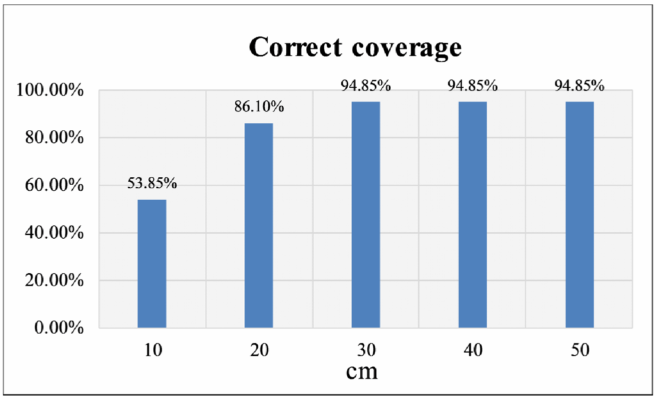
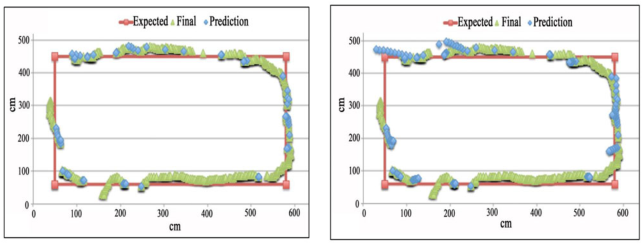


Figure 19. Correct coverage rate with various margins of distance error in the case of an absolute distance greater than zero

4.4 Motion Vector Prediction

For the case with an absolute distance equal to zero, we changed only the time length using motion vector prediction to 0.33 and 1 s. As shown in Figure 20, we obtain a visual difference in the results after using multi-tags with motion vector prediction and the actual path of movement. From Figure 21, we obtained the accuracy of various margins of error is at 0.33 and 1 s. From the experimental results, we know that the longer we use motion vector prediction, the more the correct coverage rate will decrease, but we can obtain more indoor-positioning position information.



(a) 0.33 s (b) 1 s

Figure 20. Comparison of the results of the proposed method and actual path of movement using motion vector prediction in the case of an absolute distance equal to zero when the time length is

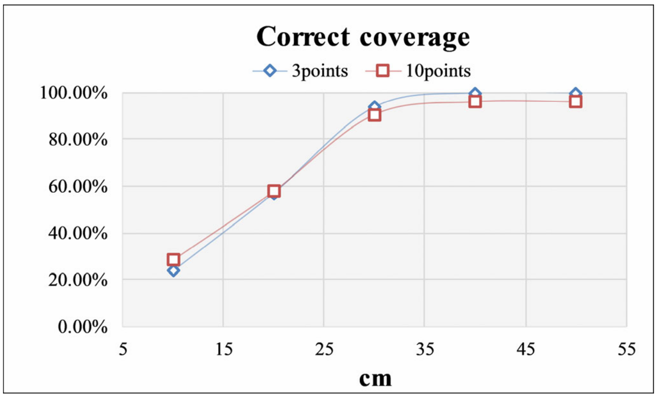
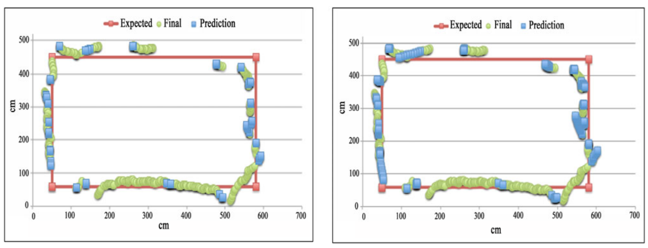


Figure 21. Comparison of correct coverage rate with motion vector prediction in the case of an absolute distance equal to zero when the error margin of absolute distance changes

For the case with an absolute distance of 85 cm, we also changed the time length using motion vector prediction to 0.33 and 1 s. As shown in Figure 22, we obtain a visual difference for results after using multi-tags with motion vector prediction and actual path of movement. From Figure 23, we obtained the accuracy of various margins of error is at 0.33 and 1 s. From the experimental results, we know that the longer we use motion vector prediction, the more the correct coverage rate will decrease, but we can obtain more indoor-positioning position information.



(a) 0.33 s (b) 1 s

Figure 22. Comparison of the results of the proposed method and actual positioning error using motion vector prediction in the case of an absolute distance greater than zero when the time length is

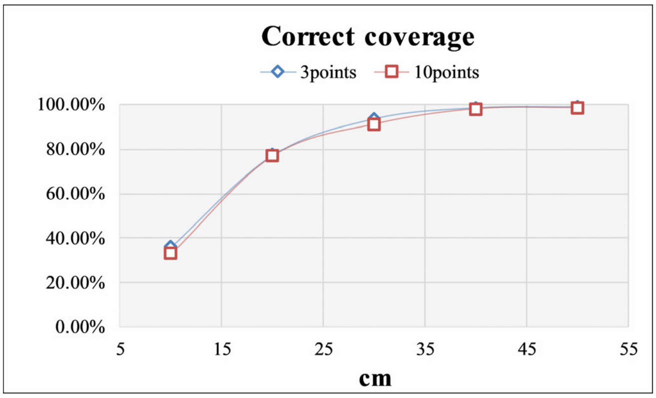


Figure 23. Comparison of correct coverage rate with motion vector prediction of absolute distance equal to zero when the error margin of absolute distance changes

5 Conclusion

In this study, we propose a method for improving the accuracy of indoor positioning using a UWB radio signal. In our system, we proposed to utilize multiple items of equipment installed in the same positions to filter the positioning error of absolute distance information to reduce instability caused by receiving signals. The proposed method is better than the use of a device installed in each position. Our method also utilizes vectors to predict the coordinate path of the moving object. From the experimental results, our method achieves higher accuracy than that obtained in other studies based on the proposed algorithms.

Acknowledgements

We thank the Ministry of Science and Technology of Taiwan for supports of this project under grant number MOST106-2221-E-239-036, MOST 106-2218-E-126-001 and MOST 106-2627-M-035-007. We thank co-authors and reviewers for their valuable opinions.

References

[1] B. Waldmann, R. Weigel, R. Ebelt, M. Vossiek, An Ultra-wideband Local Positioning System for Highly Complex Indoor Environments, *International Conference on Localization and GNSS (ICL-GNSS)*, Starnberg, Germany, 2012, pp. 1-5.

[2] E. García, P. Poudereux, Á. Hernández, J. Ureña, D. Gualda, A Robust UWB Indoor Positioning System for Highly Complex Environments, *IEEE International Conference on Industrial Technology (ICIT)*, Seville, Spain, 2015, pp. 3386-3391.

[3] A. K. M. M. Hossain, Y. Jin, W. -S. Soh, H. N. Van, SSD: A Robust RF Location Fingerprint Addressing Mobile Devices' Heterogeneity, *IEEE Transactions on Mobile Computing*, Vol. 12, No. 1, pp. 65-77, January, 2013.

[4] P. Bahl, V. N. Padmanabhan, RADAR: An In-building RF-based User Location and Tracking System, *Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies*, Tel Aviv, Israel, 2000, pp. 775-784.

[5] X. Y. Lin, T. W. Ho, C. C. Fang, Z. S. Yen, B. J. Yang, F. Lai, A Mobile Indoor Positioning System based on Ibeacon Technology, *37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Milan, Italy, 2015, pp. 4970-4973.

[6] M. E. Rida, F. Liu, Y. Jadi, A. A. A. Algawhari, A. Askourih, Indoor Location Position based on Bluetooth Signal Strength, *2nd International Conference on Information Science and Control Engineering (ICISCE)*, Shanghai, China, 2015, pp. 769-773.

[7] H. H. Chang, Y. C. Chu, Y. B. Huang, T. Y. Lin, M. F. Tsai, Enhancing the Accuracy of Indoor Positioning by using Multiple Bluetooth Low Energy Beacon Devices, *International Conference on Future Computer and Communication*, Nagoya, Japan, 2017, pp. 1-5.

[8] R. V. Garcia, H. K. Kuga, M. C. Zanardi, Unscented Kalman Filter Applied to the Spacecraft Attitude Estimation with Euler Angles, *Mathematical Problems in Engineering*, Vol. 2012, Article ID 985429, 2012.

[9] Y. S. Suh, Orientation Estimation Using a Quaternion-Based Indirect Kalman Filter with Adaptive Estimation of External Acceleration, *IEEE Transactions on Instrumentation and Measurement*, Vol. 59, No. 12, pp. 3296-3305, December, 2010.

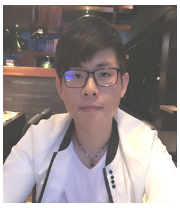
[10] Y. Jin, H. S. Toh, W. S. Soh, W. C. Wong, A Robust Dead-reckoning Pedestrian Tracking System with Low Cost Sensors, *IEEE International Conference on Pervasive Computing and Communications (PerCom)*, Seattle, WA, 2011, pp. 222-230.

Biographies

Ming-Fong Tsai received the Ph.D. degree from the Department of Electrical Engineering, Institute of Computer and Communication Engineering, National Cheng Kung University, Taiwan. He is currently an Assistant Professor with the Department of Electronic Engineering, National United University, Taiwan. His current research interests include Internet of Things, Vehicular Communications and Multimedia Communications.

Thanh-Nam Pham received the Ph.D. degree in Information Engineering and Computer Science from Feng Chia University, Taiwan, in 2018. He is currently with the Department of Electronics and Communications at

Thai Nguyen University of Information and Communications Technology, Vietnam. His research interests include Internet of Things Technologies and applications.



Bo-Kai Hu received the M.S. degree in Computer Science and Information Engineering from the Feng Chia University, Taichung, Taiwan, 2017. His current research interests include Indoor Positioning, industry 4.0 and cloud computing.

Fang-Rong Hsu received the Ph.D. degree in Computer Science from the National Chiao Tung University, Hsinchu, Taiwan, 1992. He is a professor and the chairperson in the Department of Information Engineering and Computer Science at Feng Chia University, Taiwan. His current research interests include information security, bioinformatics, cloud computing, data mining and graph algorithm.