Using Machine Learning Approaches to Improve Ultra-Wideband Positioning

Che-Cheng Chang, Hong-Wen Wang, Yu-Xiang Zeng, Jin-Da Huang

Department of Information Engineering and Computer Science, Feng Chia University, Taiwan
checchang@fcu.edu.tw, hongwenwang7@gmail.com, leo32656884602@gmail.com, key9405618@gmail.com

Abstract

An ultra-wideband (UWB) positioning system consists of at least three anchors and a tag. Via the UWB transceiver mounted on each device in the system, we can use some techniques to obtain the distance between each anchor and the tag. Then we can further realize the tag localization by some classic algorithms. However, in the real environment, the uncertain measurement may bring incorrect distance as well as positioning information. Therefore, in this research, we intend to reconsider the positioning issue by incorporating some machine learning approaches with uncertain measurement in the real environment. Particularly, we utilize the concept of machine learning for overall consideration instead of using a model to evaluate the uncertainty. The experimental results show that our method can be applied to different cases, and some interesting properties in the practical experiments are presented.

Keywords: Positioning system, Uncertain measurement, Machine learning

1 Introduction

A positioning system is a mechanism for determining the location of an object in a specific space. There are a lot of positioning technologies ranging from interplanetary systems to workspace systems for different applications both indoors and outdoors. For outdoor tasks, the Global Positioning System (GPS) is the most significant development for safe and efficient navigation and surveillance. However, it may not be always precise and available due to the effects of signal attenuation and multi-path propagation for indoor tasks [1-6]. Thus indoor localization is challenging and important while GPS is not useful for positioning. There are many existing indoor positioning researches thus far, such as ultra-wideband (UWB) [7-8], radio frequency identification (RFID) [9], Bluetooth [10], Zig bee [11], dead reckoning (DR) [12-13], visible light communication (VLC) [14], WiFi systems [15], and mobile lidar [16]. Among these technologies, UWB is more attractive to us since it has been widely used for many indoor applications. It is a radio technology that utilizes a very low energy level for short-range, high-bandwidth communications over a large portion of the radio spectrum [17]. Thanks to the large spectrum, the signal, which is very broad in frequency, can be accurately measured in terms of the time it takes from a transmitter to a receiver. Also, with UWB, the signal is transmitted with low power, preventing interference with other systems using similar radio spectrum. A UWB positioning system consists of at least three anchors and a tag for a 2D plane application, and four anchors and a tag for a 3D one. Through the UWB transceiver mounted on each device in the system, we can utilize some techniques to obtain the distance between each anchor and the tag, e.g., time of flight (TOF), time of arrival (TOA), and so on. Then we can further realize the tag localization by some classic algorithms such as trilateration. However, in the real environment, the uncertain measurement may bring incorrect distance as well as positioning information. For example, multipath interference, which is a distortion of the signal, will take many different paths to the receiver with various phase shift and various polarisation shift [17]. Accordingly, it affects the transmission of UWB.

In this research, we reconsider the UWB positioning issue by incorporating machine learning approaches with uncertain measurement in real and distinct scenarios. Our method intends to utilize machine learning for overall consideration regarding the real environment instead of using a deterministic model to estimate the uncertainty. For example, in [18], the authors assume that the measurements of the range between the target and beacons are corrupted with white Gaussian noise with variance. Also, the Log-distance path loss model shown in [19] is composed of a normal random variable with a standard deviation. Next, so long as replacing the training set with the corresponding one, it is adapted to the different environment without modifying the procedure. Most important of all, we do not apply those complicated or combinatorial learning models to this research, we intend optimizing the classical trilateration algorithm

*Corresponding Author: Che-Cheng Chang; E-mail: checchang@fcu.edu.tw
DOI: 10.53106/160792642021092205008

by only considering some simple machine learning approaches for the real-time purpose. Thus our method is more suitable for the mobile edge computing applications that may not have powerful computing hardwares in practice.

Moreover, we test our method by many practical experiments, including distinct environments with various types of density of obstacles, different trajectories, and distinct number of reference points. These results from above experiments will show the ability of our design. In a word, this research has following main contributions:

1. Using machine learning to enhance the classical trilateration;
2. Proving that our method works well in various systemic and environmental conditions;
3. Proving that the measurement errors are not normal distributions.

This paper is structured as follows. In section 2, we review the necessary preliminary knowledge related to our work. In section 3, we start to introduce our idea, which is based on classic machine learning approaches to deal properly with uncertain measurement in real and distinct scenarios, in order to realize the tag localization based on a classical trilateration algorithm. The experimental results are provided in section 4, and some interesting properties of our method are presented in this section as well. Finally, section 5 concludes the paper and discusses some possible future works.

2 Preliminaries

2.1 Problem Formulation

In a UWB positioning system, each anchor can measure the distance from each tag to itself via some radio ranging approaches, e.g., ToF, ToA, and so on. The measured distance can be expressed as Equation 1:

$$d_{measure} = d_{real} + noise, \tag{1}$$

where the noise part is assumed as a white Gaussian noise generally.

Further, we suppose that there are n anchors at known locations $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ and the sole tag at unknown position (x, y). Then we can collect a set of n samples at a time. Obviously, in a noise-free environment, these samples will satisfy the following equations:

$$\begin{aligned} d_1 &= \sqrt{(x-x_1)^2 + (y-y_1)^2}, \\ d_2 &= \sqrt{(x-x_2)^2 + (y-y_2)^2}, \\ d_3 &= \sqrt{(x-x_3)^2 + (y-y_3)^2}, \\ &\ldots \\ d_n &= \sqrt{(x-x_n)^2 + (y-y_n)^2}, \end{aligned} \tag{2}$$

in which d_i is the distance from the tag to $anchor_i$.

The unknown position of the tag, (x, y), can be determined if $n \geq 3$. Particularly, if $n = 3$, the only three anchors should not be in a line. It is the classical trilateration issue for a 2D plane. Here, we illustrate the ideal example with Figure 1, that is, the solution in a 2D plane is the intersection of three circles centered at three anchors for $n = 3$. Note that if the issue is extended to a 3D space, all above conditions regarding the number of anchors should plus one. For instance, the unknown position of the tag in a 3D space, (x, y, z), can be determined if the number of anchors is greater than or equal to four.

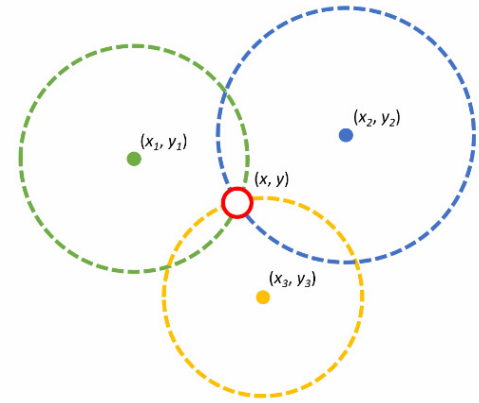
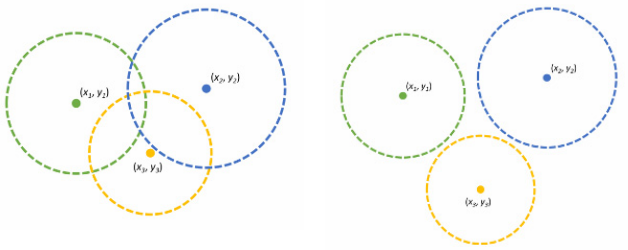


Figure 1. An ideal example with a sole solution of the positioning issue in a 2D plane

Unfortunately, these circles may not intersect at a sole point while we consider uncertain measurement in the real environment. This means that there may be more than one solution, i.e., an area (Figure 2(a)), or no solution, i.e., no intersected point (Figure 2(b)), in a practical scenario.



(a) An example with more than one solution in a 2D plane

(b) an example without any solution in a 2D plane

Figure 2. Examples

2.2 Time of Flight (TOF)

Consider the scenario in a UWB positioning system, the tag transmits a message, which contains the synchronization information that the message is sent at time t_0. Then the anchor receives the message at time t_1, the flight time of the message is $\Delta T = t_1 - t_0$. Particularly, the speed of light in vacuum, commonly

denoted C, is a universal physical constant. It is exactly defined as $C = 299792458\ m/s$. Then the speed of light in the atmosphere is lower due to many variables. Here, the distance between the above mentioned tag and anchor can be estimated:

$$dis\tan ce = C * \Delta T, \quad (3)$$

this is the simplest TOA method for estimating the distance [20-22].

However, the key is the clock of each tag and anchor in the UWB positioning system must be strictly synchronized, otherwise the error of the measuring range would be very large due to the velocity of electromagnetic wave. For example, even the clock error is only $1\mu s$, the range error will be about $300m$. Obviously, it is impracticable for general purposes that may only with tiny, cheap and power constrained devices.

TOF is a modified method for dealing properly with the clock issue of TOA. We illustrate an example of the fundamental TOF method introduced in [20] with Figure 3. In order to estimate the distance between a tag and an anchor, the tag first sends $message_0$ to the anchor at t_0. Then upon receiving $message_0$ at t_1, the anchor makes a response, $message_1$, to the tag at t_2. Also, upon receiving $message_1$ at t_3, the tag replies $message_2$ to the anchor at t_4. At t_5, the anchor receives $message_2$ and then responds $message_3$ to the tag at t_6. Finally, upon receiving $message_3$ at t_7, we can obtain the measuring range by the following formulation:

$$\Delta T = \frac{(t_3 - t_0) - (t_2 - t_1) + (t_7 - t_4) - (t_6 - t_5)}{4}, \quad (4)$$

where $(t_3 - t_0)$, $(t_2 - t_1)$, $(t_7 - t_4)$, and $(t_6 - t_5)$ are used to remove the effects of clock errors. Lastly, it is divided by 4 to get a one-way trip time ΔT. This has been proven that it will succeed to eliminate the clock error [20-22].

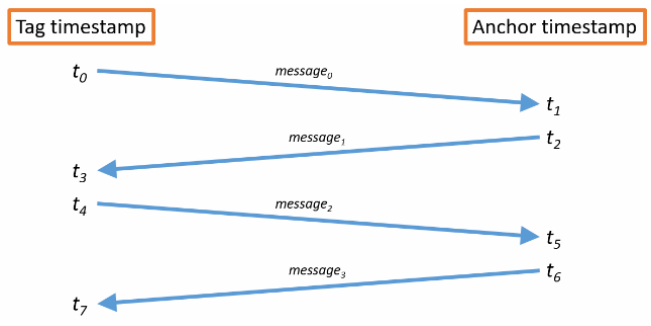


Figure 3. A fundamental TOF method with timestamp

2.3 A classical Solution

According to above discussion, we have known that with uncertain measurement in the real environment, circles may not intersect at a sole point. Namely, there may be more than one solution, i.e., an area, or no solution, i.e., no intersected point. One feasible solution for the scenario with more than one solution is that after obtaining these solutions, the average of them is the unique and optimal solution. We illustrate this idea by Figure 4. Foremost, suppose that we have a sole tag and three anchors. In accordance with the classical trilateration algorithm, we will obtain six possible solutions due to the erroneous measurement, i.e., (x'_{12}, y'_{12}), (x'_{13}, y'_{13}), (x'_{23}, y'_{23}), (x''_{12}, y''_{12}), (x''_{13}, y''_{13}), and (x''_{23}, y''_{23}). Then the average of them will be the unique and optimal solution, i.e., (x, y).

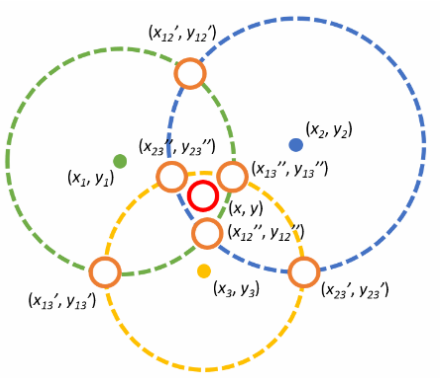


Figure 4. A feasible solution for UWB positioning with uncertain measurement in a 2D plane

On the other hand, for the scenario without any intersected point, i.e., no solution, we can further incorporate some assistant algorithms into the classical trilateration for the purpose of figuring at least one solution, e.g., digit-by-digit algorithm and so on.

2.4 Regression Analysis

Regression analysis is for estimating the relationship between a dependent variable and one or more independent variables [23-25]. Linear regression is the most common form, in which we can find a linear line that fits the data most closely. Then we can make a prediction by computing a weighted sum of the input features and the bias term, which is shown in the following equation:

$$\hat{y} = \theta_n x_n + \theta_{n-1} x_{n-1} + \cdots + \theta_1 x_1 + \theta_0, \quad (5)$$

where $\hat{y}$ is the prediction, n the number of features, x_i the i^{th} feature, and θ_j the j^{th} parameter.

Next, polynomial regression is a form of regression analysis as well. The relationship between the dependent variable and the independent variables is presented as a n^{th} degree polynomial in independent variables for $n > 1$ [23]. More specifically, if the data is more complex than a straight line, we can add powers of each feature as new features and then train this model on these extended features. This manner is called polynomial regression [25].

On the other hand, based on regression analysis, some researches perform both variable selection and regularization to enhance the interpretability and prediction accuracy, e.g., Lasso regression and Ridge

regression. Namely, both of them they can reduce the model complexity and prevent overfitting. Thus, we can regularize and optimize regression analysis by the following form:

$$\theta_n := \arg\min_\theta [\frac{1}{2}\Sigma(\hat{y}_i - y_i)^2 + R(\theta)], \quad \textbf{(6)}$$

where $R(\theta)$ is the regularization. Then it can be extended to distinct regularization forms, i.e., Lasso regression (Equation 7) and Ridge regression (Equation 8).

$$\theta_n := \arg\min_\theta [\frac{1}{2}\Sigma(\hat{y}_i - y_i)^2 + \lambda \| \theta \|_1], \quad \textbf{(7)}$$

$$\theta_n := \arg\min_\theta [\frac{1}{2}\Sigma(\hat{y}_i - y_i)^2 + \lambda \| \theta \|^2], \quad \textbf{(8)}$$

3 Our UWB Positioning Procedure

In the real environment, the uncertain measurement of time and/or distance may bring incorrect positioning information. Generally speaking, the noise part is assumed as a white Gaussian noise. In this research, we reconsider the UWB positioning issue by incorporating machine learning approaches with uncertain measurement. Our main concept is that the white Gaussian noise may not fit the uncertain measurement well in practice. Thus we use some machine learning approaches for overall consideration instead of using a deterministic model to evaluate the uncertainty. Particularly, several forms of regression analysis will be presented in the paper, i.e., linear regression, i^{th} order polynomial regression, Lasso regression, and Ridge regression. Notice that in our pseudocode, we will use the term, regression analysis, to express them for the clear description. Also, in the experiments, we just need to replace the corresponding part with another for implementing different forms of regression analysis.

Now we begin to elaborate our algorithm, which is UWB positioning procedure based on regression analysis and presented in Figure 5. Foremost, in the collecting phase, for a reference point, we initialize the locations of anchor $anchor_i$ and the sole tag *tag* (lines 1-2). Then we collect the real distance between $anchor_i$ and *tag*, $real\_dis(anchor_i, tag)$, as well as the corresponding distance provided by the UWB positioning system, $sys\_dis(anchor_i, tag)$ (line 4). Namely, we can obtain the actual distance and its inaccurate version provided by the UWB positioning system as the training data. Particularly, we collect x pairs of distance information once for $anchor_i$ and *tag*, both are located at specific locations (line 3). Obviously, if there are y anchors and z reference points in a scenario, we should collect xyz pairs of distance information as the training data.

Algorithm: UWB positioning algorithm for *tag*

collecting phase:

(01) **initialize** the position of $anchor_i$:
$anchorx_i$, $anchory_i$;

(02) **initialize** the position of *tag*:
tagx, *tagy*;

(03) **for** $1 \le x \le n$ do

(04) $TD_i \leftarrow$ (**sys_dis**($anchor_i$, *tag*), **real_dis**($anchor_i$, *tag*));

learning phase:

(05) **refine** TD_i;

(06) $reg\_mod_i \leftarrow$ **train**(TD_i);

implementation phase:

upon receiving $distance_i$ **from** $anchor_i$**:**

(07) $new\_distance_i \leftarrow$ **predictor**(reg_mod_i, $distance_i$);

upon obtaining enough distance information:

(08) *tagx, taxgy* $\leftarrow$ **trilateration**(all latest existing $new\_distance_i$);

Figure 5. Our UWB positioning algorithm

After the collecting phase, the procedure enters the learning phase. In this phase, the training data TD_i will be refined first (line 5). More specifically, we sort the data in TD_i from large to small and remove the largest 10% and smallest 10%, this may help us remove some extreme unreasoned data. Then the residual data is averaged and stored back to TD_i. Finally, we utilize the regression analysis to train a model for each anchor (line 6). Here is an interesting and important property needed to be addressed. According to our experimental results, even with the same real distance between the tag and different anchors, these anchors with the same specification in the UWB positioning system would report very distinct distance values. The intuition is that the environment states between the tag and the anchors are different, and there may be some hardware component tolerances of these anchors, thus it is better to train different models for distinct pairs of anchors

and tags.

On the other hand, in the implementation phase, upon receiving the distance information $distance_i$ from the anchor $anchor_i$, we will utilize the corresponding predictor $predictor(reg\_mod_i, distance_i)$ to obtain the new distance information $new\_distance_i$ (line 7). Notice that according to the above discussion, we will have a unique model for each pair of anchor and tag. Hence, by checking the source of the distance information, the corresponding model can be applied to optimize the distance information.

Finally, upon receiving enough distance information at a specific time interval, i.e., at least three distance values from distinct anchors, we can apply a trilateration algorithm to estimate the location of the tag, i.e., tag_x and tag_y (line 8).

4 Experimental Results

In this section, we start to illustrate the experiments detailedly. The experimental settings are listed below:

- 3 scenarios are considered in our experiments, i.e., classroom, parking garage, and parking lot (Figure 6).
- 4 anchors are set at fixed positions.
- Varied number of static reference points are used for collecting the training data in a scenario, i.e., 8, 12, and 16.
- For each reference point, we have collected 100 continuing distance values.

(a) classroom (b) parking garage (c) parking lot

Figure 6. Three scenarios are considered in our experiments

- The detailed locations of all anchors and reference points are shown in Figure 7.

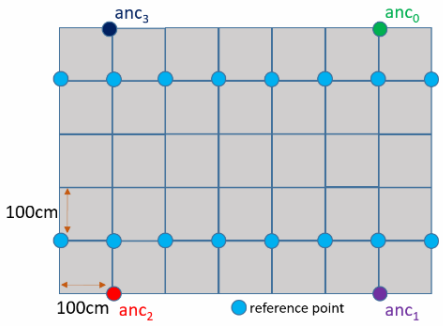


Figure 7. The detailed locations of all anchors and reference points in our experiments

- The testing data are collected by 2 kinds of trajectories in each scenario.
- 5 forms of regression analysis will be applied to enhance the testing data, i.e., linear regression (degree 1), 5^{th} order polynomial regression (degree 5), 10^{th} order polynomial regression (degree 10), 10^{th} order polynomial regression with Lasso (degree 10 with Lasso), and 10^{th} order polynomial regression with Ridge (degree 10 with Ridge).
- The UWB wireless transceiver module is DWM1000 [26].
- Scikit-learn [27] is utilized to implement all machine learning approaches in the experiments.

The main purpose of the design of three scenarios in the experiments is to verify our method can be applied to distinct environments with various types of density of obstacles. After that, the experimental results are shown in following figures. Furthermore, the statistic presented in tables reveal further information about these experiments.

First, according to the experimental results shown in Figure 8, Figure 9, and Figure 10, which are parts of statistic in three scenarios. In these figures, we present the relationships between the actual values and corresponding values provided by the UWB system. Particularly, the measured data for the same actual value, 412cm, from different anchors and scenarios are provided. Evidently the measurement errors are not normal distributions.

On the basis of Figure 11, Figure 12, and Figure 13, which illustrate the straight trajectory experiments with 16 reference points in the classroom, parking lot, and parking lot respectively. It is obvious that the results of traditional trilateration and 10^{th} order polynomial regression with Ridge are always within the last two ranks no matter how the density of obstacles is. Hence, we can conclude that in our method, except Ridge regression, all regression models are efficient to refine the raw data for obtaining more accurate positioning information.

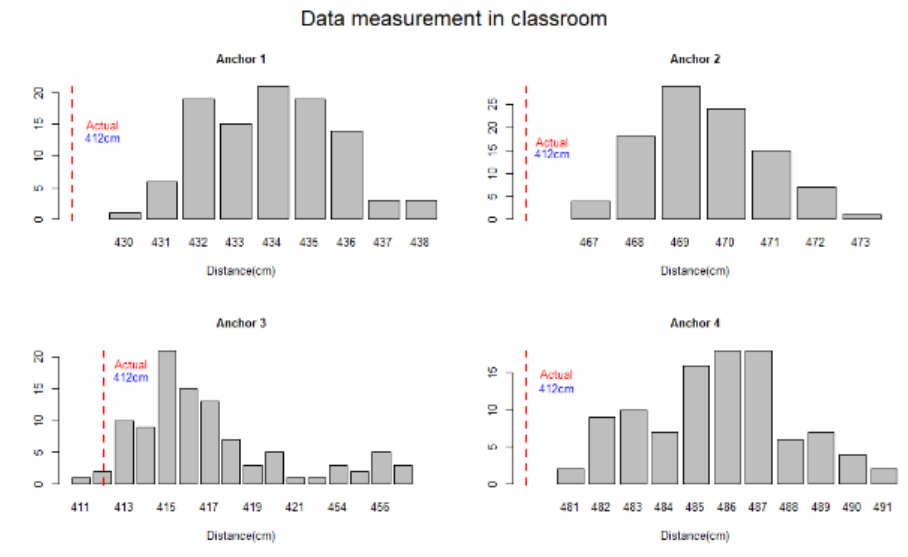


Figure 8. Data measurement in the classroom

Figure 9. Data measurement in the parking garage

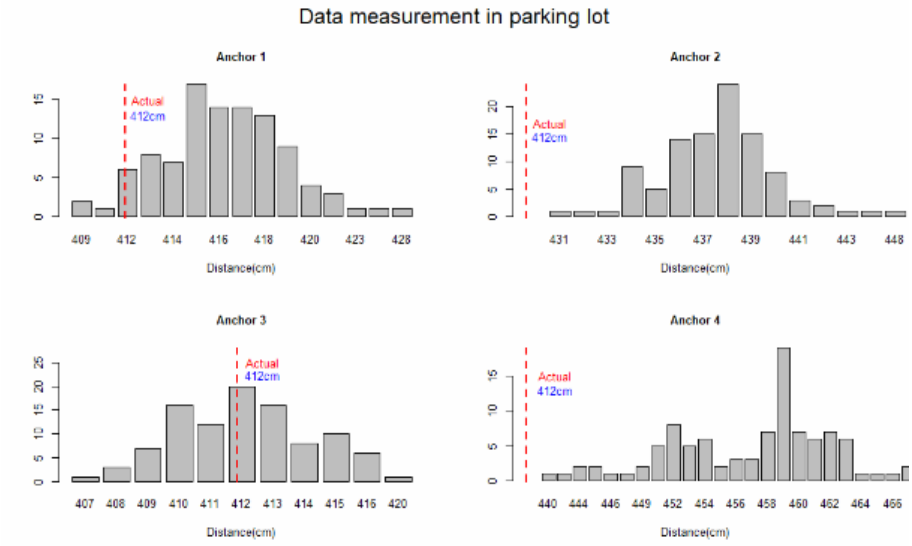


Figure 10. Data measurement in the parking lot

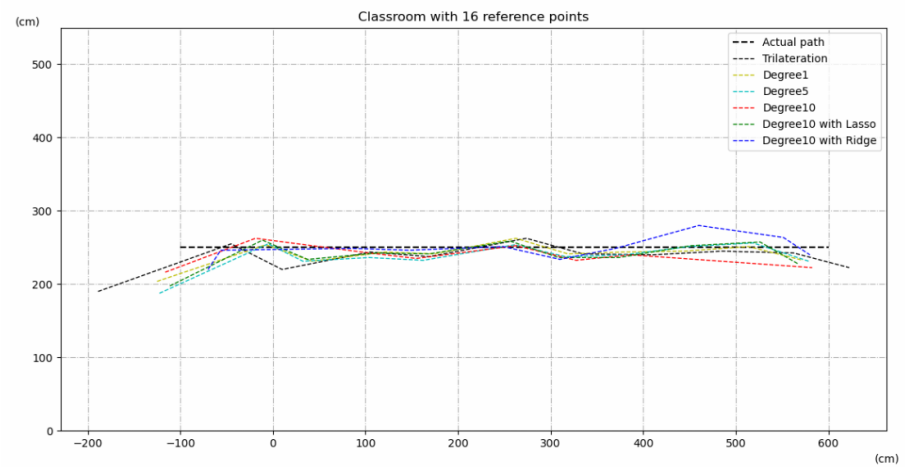


Figure 11. The straight trajectory experiment with 16 reference points in the classroom

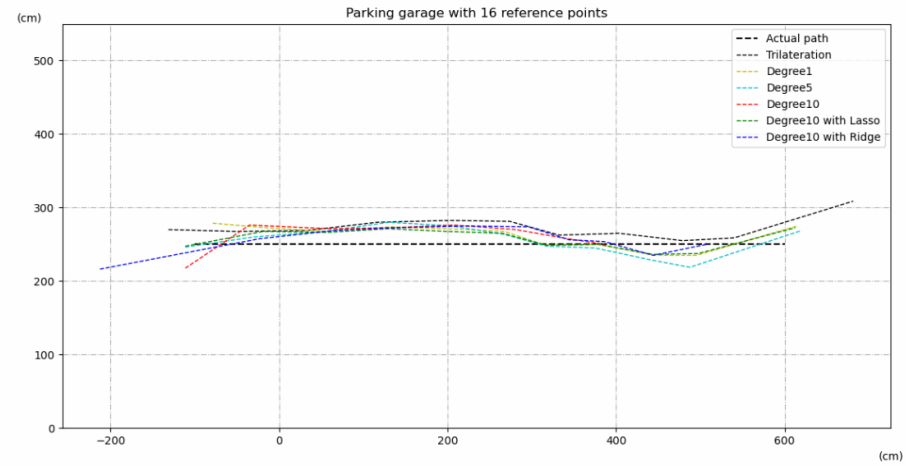


Figure 12. The straight trajectory experiment with 16 reference points in the parking garage

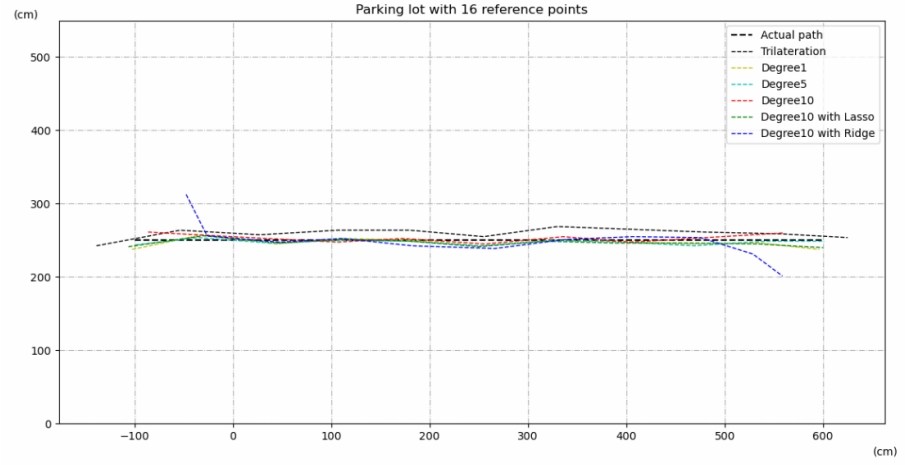


Figure 13. The straight trajectory experiment with 16 reference points in the parking lot

In Figure 14, Figure 15, and Figure 16, a multi-turn trajectory is applied to the next experiments with 16 reference points. Also, the results of traditional trilateration and 10^{th} order polynomial regression with Ridge are still within the last two ranks. This is consistent with the conclusion in the previous paragraph. Namely, most regression models are suitable for our design in various environments.

Next, we use the statistic presented in Table 1, Table 2, and Table 3 to present the effects of distinct number of reference points. Here, the scenarios with straight trajectory in the classroom, parking garage, and parking lot are considered.

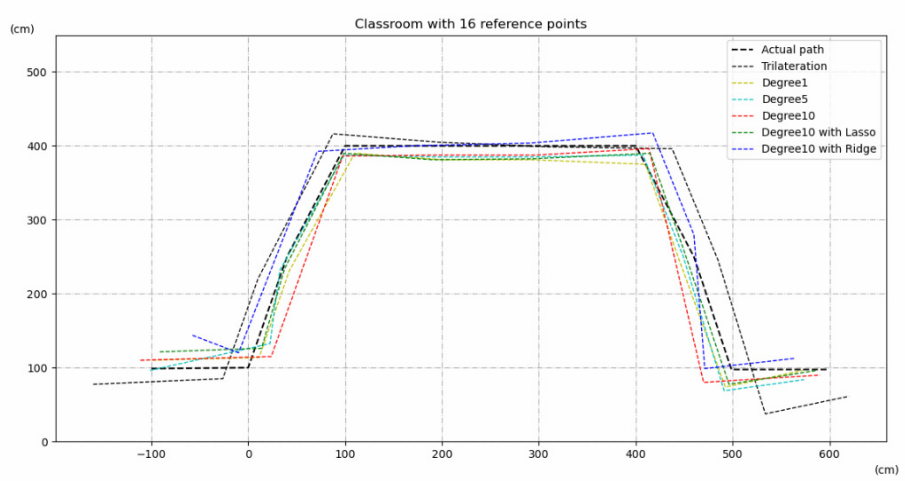


Figure 14. The multi-turn trajectory experiment with 16 reference points in the classroom

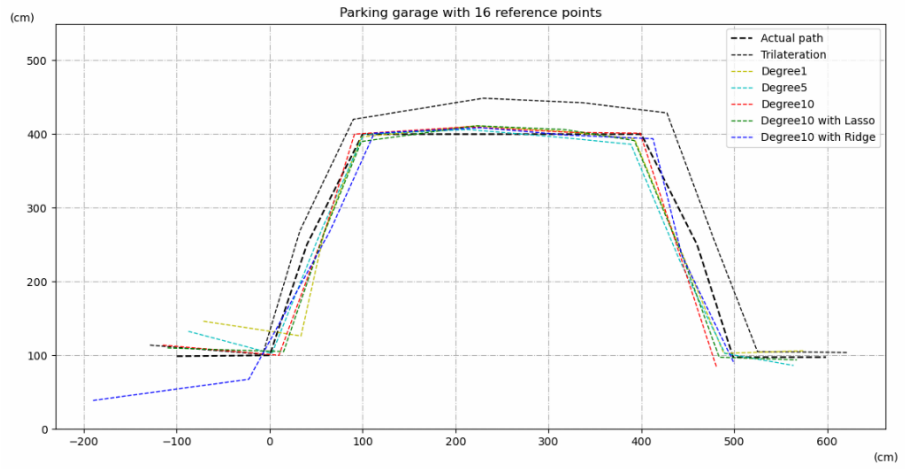


Figure 15. The multi-turn trajectory experiment with 16 reference points in the parking garage

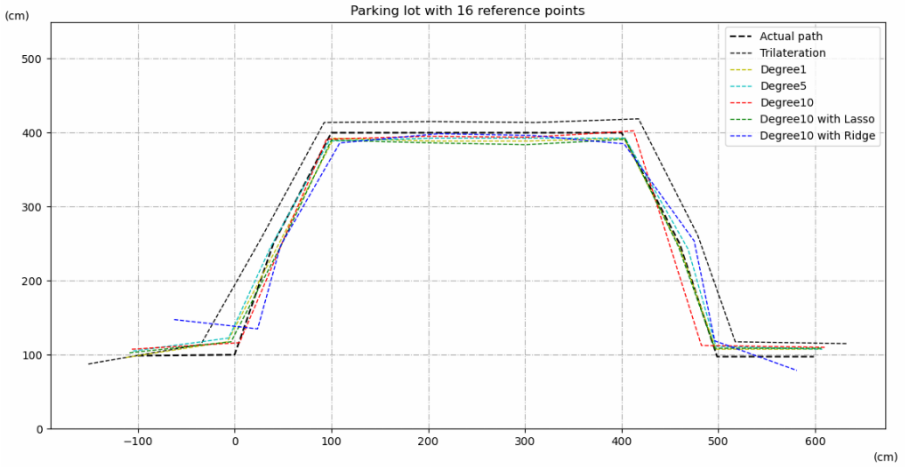


Figure 16. The multi-turn trajectory experiment with 16 reference points in the parking lot

According to positioning errors of traditional trilateration, we can know that if the experimental space is with higher density of obstacles, the mean of erroneous value and its standard deviation are apparent larger, i.e., classroom and parking garage. In most experimental scenarios, the 10^{th} order polynomial regression with Lasso shows its capability to refine the raw data for obtaining more accurate positioning information. No matter what the scenario is, it always works well.

Table 1. The experimental statistics for straight trajectory with 16 reference points

Straight trajectory with 16 ref. Points	Degree 1		Degree 5		Degree 10		Degree 10 with Lasso		Degree 10 with Ridge		Trilateration	
	Average	Standard deviation	Average	Standard deviation	Average	Standard deviation	Average	Standard deviation	Average	Standard deviation	Average	Standard deviation
Classroom	20.2836	11.9156	22.1407	15.1550	21.0319	9.96278	20.0327	13.6711	24.3999	9.97418	31.9212	25.6048
Parking garage	25.4044	9.69335	24.1325	13.3784	27.4300	10.7134	20.7484	8.6298	36.7143	30.4450	33.2029	22.9433
Parking lot	6.2382	3.9173	6.0746	3.3964	16.9626	11.099	6.07359	2.9024	23.7445	24.0259	21.3663	8.5036

Note. The unite of measurement of all values presented in this table is centimeter.

Table 2. The experimental statistics for straight trajectory with 12 reference points

Straight trajectory with 12 ref. Points	Degree 1		Degree 5		Degree 10		Degree 10 with Lasso		Degree 10 with Ridge		Trilateration	
	Average	Standard deviation	Average	Standard deviation	Average	Standard deviation	Average	Standard deviation	Average	Standard deviation	Average	Standard deviation
Classroom	16.8768	11.3931	17.4069	12.7829	31.262	43.0145	21.7137	16.5288	106.319	34.8461	31.9212	25.6048
Parking garage	20.9251	10.9132	28.6610	18.7989	24.8696	20.0192	24.7616	8.7620	28.5981	22.4078	33.2029	22.9433
Parking lot	6.0028	5.1993	20.4319	33.1106	28.3685	37.3573	6.4124	3.3183	24.1693	24.8790	21.3663	8.5036

Note. The unite of measurement of all values presented in this table is centimeter.

Table 3. The experimental statistics for straight trajectory with 8 reference points

Straight trajectory with 8 ref. Points	Degree 1		Degree 5		Degree 10		Degree 10 with Lasso		Degree 10 with Ridge		Trilateration	
	Average	Standard deviation	Average	Standard deviation	Average	Standard deviation	Average	Standard deviation	Average	Standard deviation	Average	Standard deviation
Classroom	20.3424	12.4985	27.9931	21.7890	45.1859	28.7674	24.4625	14.0421	77.6283	26.9217	31.9212	25.6048
Parking garage	25.9862	10.5394	28.3872	15.9474	35.8955	22.9026	21.3894	10.8132	118.1067	74.9323	33.2029	22.9433
Parking lot	7.4927	3.2589	27.2722	24.5840	47.9591	60.5230	6.8070	3.9129	117.8494	88.6764	21.3663	8.5036

Note. The unite of measurement of all values presented in this table is centimeter.

On the other hand, the linear regression is alsways within the top two ranks in the scenarios with less reference points, i.e., 8 and 12. However, in the scenarios with the most reference points, i.e., 16, its improvement cannot compare to the 10^{th} order polynomial regression with Lasso and 5^{th} order polynomial regression.

Consider the number of reference points in all scenarios, we can observe that a scenario with more reference points is always more accurate and stable than the same scenario with less reference points. Hence we can conclude that the larger the number of reference points is, the more accurate and stable the system is.

Last but not least, although some form of regression analysis has the better performance in some particular scenario, it is the worse in some other scenarios with different conditions. For example, the 5^{th} order polynomial regression has the better performance in the parking garage with 16 reference points, but the worse performance in the same scenario with 12 reference points. Hence, it is better to choose a stable form of regression analysis for safety sake if we intend to apply our method to a new and unknown environment.

5 Conclusion and Future Work

In this paper, we reconsider the UWB positioning issue by incorporating machine learning approaches with uncertain measurement in the real environment. The experimental results show some interesting properties in the real environment. First, if the regression models are applied to refine the raw data before applying to the classical trilateration, we will have more accurate positioning information. After that, even with the same real distance between the tag and different anchors, these anchors with the same specification in the UWB positioning system would report very distinct distance values. Hence, it is better to train different models for distinct pairs of anchors and tags.

For the future work, we intend to integrate heterogeneous machine learning approaches into classical algorithms to refine the raw data for getting more accurate and stable positioning information. Similarly, using distinct kinds of technologies with different electromagnetic waves to realize and optimize the corresponding positioning issues is also interesting, e.g., 5G cellular network, WiFi, Dedicated Short Range Communications (DSRC) and so on.

Acknowledgments

This research was supported by Ministry of Science and Technology, Taiwan, R.O.C. under grant 109-2221-E-035-067-MY3, 109-2622-H-035-001-, and 108-2221-E-035-063-.

References

[1] T.-T. Lu, S.-C. Yeh, G. Fan, Indoor Positioning Systems for Different Mobile Terminal Devices, *Journal of Internet Technology*, Vol. 19, No. 6, pp. 1811-1821, November, 2018.

[2] S. Dionisio-Ortega, L. O. Rojas-Perez, J. Martinez-Carranza, I. Cruz-Vega, A Deep Learning Approach towards Autonomous Flight in Forest Environments, *2018 International Conference on Electronics, Communications, and Computers*, Cholula, Puebla, Mexico, 2018, pp. 139-144.

[3] V. Maximov, O. Tabarovsky, Survey of Accuracy Improvement Approaches for Tightly Coupled ToA/IMU Personal Indoor Navigation System, *Proceedings of International Conference on Indoor Positioning and Indoor Navigation*, Montbeliard-Belfort, France, 2013, pp. 1-4.

[4] C.-C. Chang, J. Tsai, P.-C. Lu, C.-A. Lai, Accuracy Improvement of Autonomous Straight Take-off, Flying Forward, and Landing of a Drone with Deep Reinforcement Learning, *International Journal of Computational Intelligence Systems*, Vol. 13, No. 1, pp. 914-919, June, 2020.

[5] X. Li, Y. Wang, K. Khoshelham, UWB/PDR Tightly Coupled Navigation with Robust Extended Kalman Filter for NLOS Environments, *Mobile Information Systems*, Vol. 2018, Article No. 8019581, December, 2018.

[6] T. Arsan, M. M. N. Hameez, A Clustering-Based Approach for Improving the Accuracy of UWB Sensor-Based Indoor Positioning System, *Mobile Information Systems*, Vol. 2019, Article No. 6372073, Sepember, 2019.

[7] A. Poulose, O. S. Eyobu, M. Kim, D. S. Han, Localization Error Analysis of Indoor Positioning System Based on UWB Measurements, *2019 Eleventh International Conference on Ubiquitous and Future Networks*, Zagreb, Croatia, 2019, pp. 84-88.

[8] Y. Zhou, C. L. Law, Y. L. Guan, F. Chin, Indoor Elliptical Localization Based on Asynchronous UWB Range Measurement, *IEEE Transactions on Instrumentation and Measurement*, Vol. 60, No. 1, pp. 248-257, January, 2011.

[9] S. S. Saab, Z. S. Nakad, A Standalone RFID Indoor Positioning System Using Passive Tags, *IEEE Transactions on Industrial Electronics*, Vol. 58, No. 5, pp. 1961-1970, May, 2011.

[10] Y. Zhuang, J. Yang, Y. Li, L. Qi, N. El-Sheimy, Smartphone-Based Indoor Localization with Bluetooth Low Energy Beacons, *Sensors*, Vol. 16, No. 5, Article No. 596, May, 2016.

[11] M. Uradzinski, H. Guo, X. Liu, M. Yu, Advanced Indoor Positioning Using Zigbee Wireless Technology, *Wireless Personal Communications*, Vol. 97, No. 4, pp. 6509-6518, December, 2017.

[12] A. Poulose, O. S. Eyobu, D. S. Han, An Indoor Position-Estimation Algorithm Using Smartphone IMU Sensor Data, *IEEE Access*, Vol. 7, pp. 11165-11177, January, 2019.

[13] H. Xing, Y. Zhao, Y. Zhang, Y. Chen, 3D Trajectory Planning of Positioning Error Correction Based on PSO-A* Algorithm, *Computers, Materials and Continua*, Vol. 65, No. 3, pp. 2295-2308, 2020.

[14] X. Guo, S. Shao, N. Ansari, A. Khreishah, Indoor Localization Using Visible Light Via Fusion of Multiple Classifiers, *IEEE Photonics Journal*, Vol. 9, No. 6, Article No. 7803716, December, 2017.

[15] Y. Zhuang, Z. Syed, J. Georgy, N. El-Sheimy, Autonomous Smartphone-Based WiFi Positioning System by Using Access Points Localization and Crowdsourcing, *Pervasive and Mobile Computing*, Vol. 18, pp. 118-136, April, 2015.

[16] C. Peng, Z. Yu, Modeling Analysis for Positioning Error of Mobile Lidar Based on Multibody System Kinematics, *Intelligent Automation and Soft Computing*, Vol. 25, No.4, pp. 827-835, 2019.

[17] M.-F. Tsai, T.-N. Pham, B.-C. Hu, F.-R. Hsu, Improvement in UWB Indoor Positioning by Using Multiple Tags to Filter Positioning Errors, *Journal of Internet Technology*, Vol. 20, No. 3, pp. 677-688, May, 2019.

[18] L. Zhao, X. Chen, J. Cheng, L. Yu, C. Lv, J. Wang, New Three-Dimensional Assessment Model and Optimization of Acoustic Positioning System, *Computers, Materials and Continua*, Vol. 64, No. 2, pp. 1005-1023, 2020.

[19] W. Li, Z. Chen, X. Gao, W. Liu, J. Wang, Multimodel Framework for Indoor Localization Under Mobile Edge Computing Environment, *IEEE Internet of Things Journal*, Vol. 6, No. 3, pp. 4844-4853, June, 2019.

[20] F. Dong, C. Shen, J. Zhang, S. Zhou, A TOF and Kalman filtering joint algorithm for IEEE802.15.4a UWB Locating, *2016 IEEE Information Technology, Networking, Electronic and Automation Control Conference*, Chongqing, China, 2016, pp. 948-951.

[21] W. Lee, V. Patanavijit, Performance Evaluation of TOA Estimation for Ultra-Wideband System under AWGN and IEEE 802.13a Channel Model, *2011 Seventh International Conference on Signal-Image Technology and Internet-Based System*, Dijon, France, 2011, pp. 237-244.

[22] M. Liu, L. Cheng, K. Qian, J. Wang, J. Wang, Y. Liu, Indoor Acoustic Localization: A Survey, *Human-centric Computing and Information Sciences*, Vol. 10, Article No. 2, January, 2020.

[23] A. Illukkumbura, *Introduction to Regression Analysis*, Amazon Digital Services, 2020.

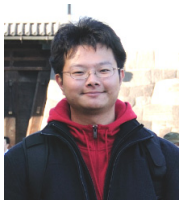
[24] R. S. Sutton, A. G. Barto, *Reinforcement Learning: an Introduction*, The MIT Press, 2018.

[25] A. Geron, *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, O'Reilly Media, 2019.

[26] DWM1000 Module – Decawave, https://www.decawave.com/product/dwm1000-module/.

[27] scikit-learn:machine learning in Python - scikit-learn 0.23.2 documentation, https://scikit-learn.org/stable/index.html.

Biographies



Che-Cheng Chang received his Ph.D. degree in department of electrical engineering from National Chung Hsing University, Taiwan in 2015. Then he joined the department of information engineering and computer science, Feng Chia University, Taiwan, as an assistant professor in 2018. His current research interests include autonomous vehicles, machine learning, and distributed computing.

Hong-Wen Wang received his B.S. degree in department of information engineering and computer science, Feng Chia University, Taiwan in 2021. Then he starts his graduate study at National Taipei University, Taipei, Taiwan. His current research interests include computer vision and machine learning.

Yu-Xiang Zeng received his B.S. degree in department of information engineering and computer science, Feng Chia University, Taiwan in 2021. Then he starts his graduate study in institute of smart industry and green energy at National Chiao Tung University, Tainan, Taiwan. His current research interests include pattern recognition and deep learning.

Jin-Da Huang received his B.S. degree in department of information engineering and computer science, Feng Chia University, Taiwan in 2021. His current research interests include intelligent transportation system and development of cross-domain system. He currently prepares for exam of MS degree in information engineering.