
Application of adaptive vector projection iterative algorithm based on least square in ultra-wideband positioning

Penghui Wang, Zengzeng Lian, M. Amparo Núñez-Andrés, Andres Calabia, Yalin Tian, Mengqi Wang, Zhe Yue, and Hongtao Mu

Penghui Wang, School of Surveying and Land Information Engineering, Henan Polytechnic University, 454003, Jiaozuo, China.(email: 212204020037@home.hpu.edu.cn).

Zengzeng Lian, School of Surveying and Land Information Engineering, Henan Polytechnic University, 454003, Jiaozuo, China.(email: zengzenglian@hpu.edu.cn).

M. Amparo Núñez-Andrés, Department of Civil and Environmental Engineering, Universitat Politècnica de Catalunya-BarcelonaTech, 08034, Barcelona, Spain. (e-mail: m.amparo.nunez@upc.edu).

Andres Calabia, School of Surveying and Land Information Engineering, Henan Polytechnic University, 454003, Jiaozuo, China, and Department of Physics and Mathematics, University of Alcala, 28801, Alcalá de Henares, Spain. (e-mail: a.calabia@uah.es)

Yalin Tian, School of Surveying and Land Information Engineering, Henan Polytechnic University, 454003, Jiaozuo, China.(e-mail: 212204010028@home.hpu.edu.cn).

Mengqi Wang, School of Surveying and Land Information Engineering, Henan Polytechnic University, 454003, Jiaozuo, China.(email: 212204010021@home.hpu.edu.cn).

Zhe Yue, School of Surveying and Land Information Engineering, Henan Polytechnic University, 454003, Jiaozuo, China.(e-mail: yuezhe@hpu.edu.cn).

Hongtao Mu, Henan Keyu Information Technology CO., LTD., 450001, Zhengzhou, China.(e-mail:332948883@qq.com).

ABSTRACT

Ultra-wideband technology has been extensively used for indoor positioning, but there are problems such as random errors in ranging and asymmetric topological grouping of base stations. To reduce the influence of random error on positioning accuracy, this paper proposes a least square-adaptive vector projective iteration (LS-AVPI) algorithm. First, the range error equation is established based on the vector projection angle relationship, and the learning rate is introduced into least square correction in the iteration. Then, the parameters such as the projection angle are optimized by the sparrow search algorithm based on the root mean square error, and the data set is used to train the convolutional neural network (CNN). Next, ranging information and least square solution are fed into the CNN to obtain the optimal parameters, and the parameters are input into the positioning model to obtain the label position. In the static experiment, LS-AVPI achieves 50 %, 30.4 %, 54.1 %, and 32.8 % higher accuracy than the constrained weighted least square, iterative least square, least square, and two-step weighted least square methods, respectively. In the dynamic experiment, LS-AVPI achieves 47 %, 31.1 %, 47.4 %, and 31.7 % higher accuracy than other algorithms, respectively.

Keywords: ultra-wideband; random error; vector projection; sparrow search algorithm; convolutional neural network

Introduction

During the urbanization process, there are changes in people's lifestyles and needs. According to statistics, individuals spend 70% to 90% of their lifetime indoors¹. With the development of network technology, the integration of the Internet of Things (IoT) with daily life has become a prominent trend. In this context, the availability of location information for individuals, equipment, and products is a crucial prerequisite for realizing IoT applications. From parking navigation to library book navigation and hospital department guidance, there is a wide demand for indoor positioning technology. Though global satellite positioning systems such as the Global Positioning System (GPS) and the BeiDou Navigation Satellite System (BDS)^{2,3} can provide users with reliable real-time location information, they are still faced with various challenges in indoor positioning applications.

Numerous positioning technologies are available, such as WiFi, Bluetooth, Zigbee, Radio Frequency Identification (RFID), Ultra-Wideband (UWB), visual, audio, 5G, etc.⁴⁻¹⁰. Among them, UWB technology stands out owing to its characteristics including high bandwidth, strong anti-interference capability, high security, low power consumption, and low device cost¹¹. It has been widely used in fields such as military, medical, and rescue operations, making it a research hotspot in the indoor positioning domain^{12,13}.

In the research on UWB positioning, most coordinate calculation methods rely on distance measurements. However, due to environmental interference, there are random errors in the distance information measured by UWB. To reduce the impact of measurement errors on positioning accuracy, references 14-17 processed the noise in the measured signals by using denoised data. Therefore, improving the accuracy of distance information helps to achieve a better positioning solution.

To enhance positioning accuracy, scholars have introduced deep learning into the field of indoor positioning¹⁸. References 19-22 utilized deep learning-related algorithms to classify data features or calculate key values for fusion algorithms. This approach effectively improves signal classification and positioning accuracy and provides a new direction for parameter optimization.

Many intelligent optimization algorithms have been developed, such as the Sparrow Search Algorithm (SSA)²³, Grey Wolf Optimizer (GWO)²⁴, and Whale Optimization Algorithm (WOA)²⁵. Based on the research of literature 26-32, the use of an intelligent optimization algorithm can improve the key value or fitness optimization, and the optimal value affects the subsequent positioning solution. Intelligent optimization algorithms and deep learning can form a connection, providing an idea for optimization.

Since the ranging information can be expressed by various equations and relationships, this paper proposes a least square-adaptive vector projective iteration (LS-AVPI) algorithm. The ranging error equation is established based on the geometric relationship of vector projection. Referring to the research of literature 33, SSA is employed to optimize the initial angle and other variables with the optimal root mean square error (RMSE), and the data set is established to train the convolutional neural network (CNN). Then, the ranging information and the least square results are fed into the network for parameter prediction. Finally, the coordinates of the label are obtained by using least square-vector projective iteration (LS-VPI).

Principle

UWB Ranging Principle

The experiments in this paper utilize the DWM1000 localization module to obtain the distance

information between the base station and tag through double-sided two-way ranging (DS-TWR), which can reduce the excessive measurement error due to the clock offset³⁴. DS-TWR is based on single-sided two-way ranging (SS-TWR) with one additional base station communication. Its ranging principle is shown in Figure 1.

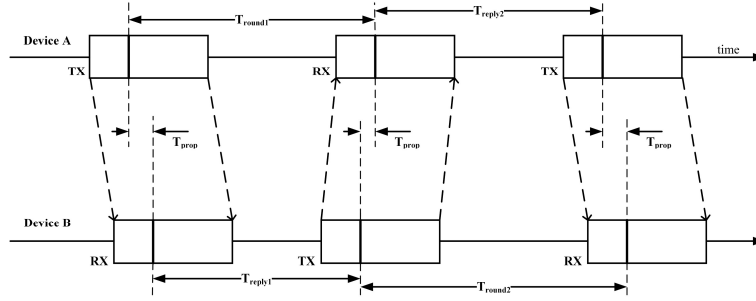


Figure 1 The ranging principle of DS-TWR

Device A initiates the first ranging message proactively, device B responds, and then device A receives the data and immediately returns the data to obtain four time differences: T_{round1} , T_{reply1} , T_{round2} , and T_{reply2} . Signal flight time can be calculated from the above information, see equation (1).

$$T_{prop} = \frac{(T_{round1} \square T_{round2} - T_{reply1} \square T_{reply2})}{(T_{round1} + T_{round2} + T_{reply1} + T_{reply2})} \quad (1)$$

Let the electromagnetic wave propagation speed in the air be c and the distance between equipment A and equipment B be D_{AB} . Then, we have

$$D_{AB} = c \square T_{prop} \quad (2)$$

Trilateral Positioning Principle

The trilateral positioning algorithm takes the base station as the center of the circle, the distance from the base station to the tag as the radius to make a circle, and the tag position is the intersection of three circles. The positioning principle is illustrated in Figure 2.

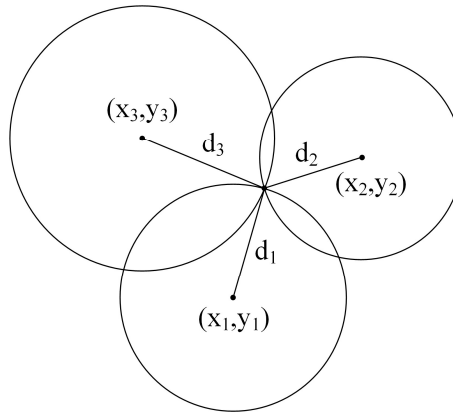


Figure 2 Trilateral positioning principle

Let the coordinates of the base stations be $(x_i, y_i), i = 1, 2, 3$. The distance from the base station

to the tag is denoted by d_i , and the tag position is represented by (x, y) . Then, the system of equations for determining the tag position can be constructed, as shown in equation (3).

$$\begin{cases} d_1 = \sqrt{(x_1 - x)^2 + (y_1 - y)^2} \\ d_2 = \sqrt{(x_2 - x)^2 + (y_2 - y)^2}, \quad i = 1, 2, 3 \\ d_3 = \sqrt{(x_3 - x)^2 + (y_3 - y)^2} \end{cases} \quad (3)$$

Equation (3) can be solved by the Least Square (LS) method, which transforms the system of equations into an LS matrix form, as shown in equation(4).

$$A = \begin{bmatrix} 2(x_2 - x_1) & 2(y_2 - y_1) \\ 2(x_3 - x_1) & 2(y_3 - y_1) \end{bmatrix}, b = \begin{bmatrix} d_1^2 - d_2^2 - K_1^2 + K_2^2 \\ d_1^2 - d_3^2 - K_1^2 + K_3^2 \end{bmatrix}, i = 1, 2, 3 \quad (4)$$

$K_i = \sqrt{x_i^2 + y_i^2}$, and let $X = [x, y]^T$. Then, the LS solution of the trilateral localization algorithm can be obtained as

$$X = (A^T A)^{-1} A^T b \quad (5)$$

Least Square-Adaptive Vector Projection Iteration

LS-AVPI first searches for the initial angle, learning rate, and threshold by using SSA; then, it combines the search results with the LS results to form a dataset, and employs this dataset to train the CNN to predict the new data. Finally, the output parameters and distance data are input to the LS-VPI to obtain the final position of the tag.

Least Square

The LS-AVPI algorithm takes the LS solution as the initial iterative position. The equation in the case of multiple base stations is given below: The base station changes from 3 to M in the trilateral positioning algorithm.

$$\begin{cases} d_1 = \sqrt{(x_1 - x)^2 + (y_1 - y)^2} \\ d_2 = \sqrt{(x_2 - x)^2 + (y_2 - y)^2}, \quad i = 1, 2, 3 \dots M \\ \vdots \\ d_i = \sqrt{(x_i - x)^2 + (y_i - y)^2} \end{cases} \quad (6)$$

The system of equations is transformed into an LS matrix form in equation (7).

$$A = \begin{bmatrix} 2(x_2 - x_1) & 2(y_2 - y_1) \\ 2(x_3 - x_1) & 2(y_3 - y_1) \\ \vdots & \vdots \\ 2(x_i - x_1) & 2(y_i - y_1) \end{bmatrix}, b = \begin{bmatrix} d_1^2 - d_2^2 - K_1^2 + K_2^2 \\ d_1^2 - d_3^2 - K_1^2 + K_3^2 \\ \vdots \\ d_1^2 - d_i^2 - K_1^2 + K_i^2 \end{bmatrix}, i = 1, 2, 3 \dots M \quad (7)$$

$K_i = \sqrt{x_i^2 + y_i^2}$, and let $X = [x, y]^T$. Then, the LS results can be obtained using equation (5).

Least Square-Vector Projection Iteration

In UWB positioning calculations, the accuracy of distance measurements has a significant impact on the positioning results. However, eliminating ranging errors is challenging. Therefore,

this paper employs a projection method to establish a measurement error equation and utilizes an iterative approach to determine the tag's position.

In two-dimensional space, as shown in Figure 3(a), $\overline{AB} = \mathbf{a}$ and $\overline{CD} = \mathbf{b}$. The process of projecting vector \mathbf{a} onto vector \mathbf{b} is as follows: Through the start point A and the endpoint B of \overline{AB} , make the vertical line of the line where \overline{CD} is located, and the vertical foots are A_1, B_1 respectively. Then, $\overline{A_1B_1}$ is the projection vector of vector \mathbf{a} onto vector \mathbf{b} . Move the start point to the same point O as shown in Figure 3 (b), then $\overline{OM} = \mathbf{a}, \overline{ON} = \mathbf{b}$, and $MM_1 \perp ON$ at point M_1 . Then, $\overline{OM_1}$ is the projection vector of vector \mathbf{a} onto vector \mathbf{b} .

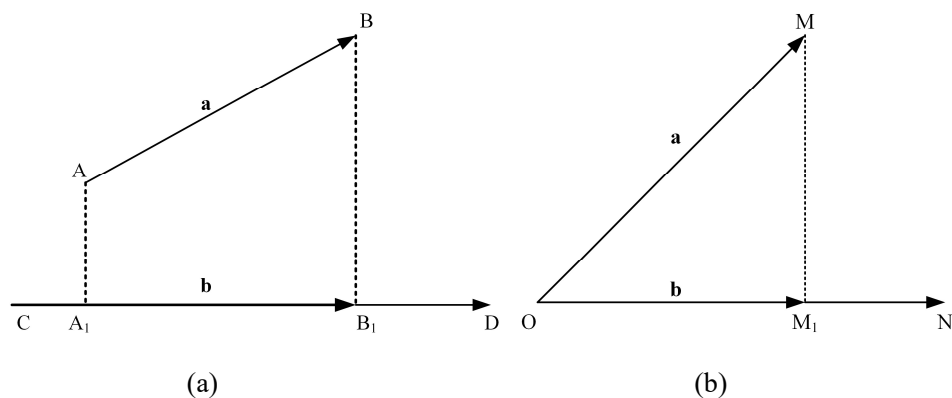


Figure 3 Spatial projection

Make a vertical line from M_1 to OM , $M_1P \perp OM$ at point P in Figure 4, and $\theta_1 = \theta_2$. If the coordinates of points O, M are known, then the length from point O to point M can be expressed as:

$$d_{OM} = d_{OM_1} \cos \theta_1 + d_{MM_1} \sin \theta_2 \quad (8)$$

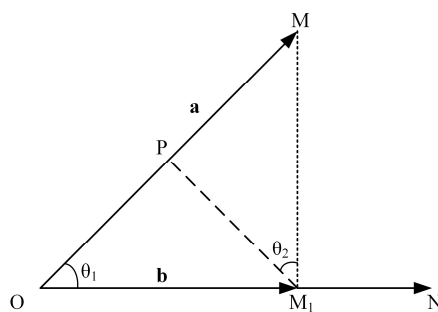


Figure 4 The projection of an auxiliary line

Let $d_{OM_1} = x_i, d_{MM_1} = y_i$, equation (8) can be rewritten as $d_{OM} = x_i \cos \theta_1 + y_i \sin \theta_2$. Let the coordinates of the base station be $(x_i, y_i), i = 1, 2, 3 \dots M$ and the tag coordinates be (x, y) .

Then, $\Delta x = x - x_i$, $\Delta y = y - y_i$. The distance d_i from the base station to the tag can be expressed as:

$$d_i = \Delta x \cos \theta_i + \Delta y \sin \theta_i \quad (9)$$

The error equation is formed based on the plumb line distance obtained from vector projection and DS-TWR measurement distance:

$$\varepsilon_i = \left| (x - x_i) \cos \theta_i + (y - y_i) \sin \theta_i \right| - d_c \quad (10)$$

In equation (10), ε_i represents the ranging error, which is the difference between the distance obtained by the projection method and the observed distance, and θ_i represents the angle of the corresponding projection vector. d_c represents the observed distance from the base station to the tag using DS-TWR, and $i = 1, 2, 3 \dots M$. The derivatives of equation (10) with respect to x, y, θ_i are obtained as follows:

$$\begin{cases} \frac{\partial \varepsilon_i}{\partial x} = \text{sign}((x - x_i) \cos \theta_i + (y - y_i) \sin \theta_i) \cos \theta_i \\ \frac{\partial \varepsilon_i}{\partial y} = \text{sign}((x - x_i) \cos \theta_i + (y - y_i) \sin \theta_i) \sin \theta_i \\ \frac{\partial \varepsilon_i}{\partial \theta_i} = \text{sign}((x - x_i) \cos \theta_i + (y - y_i) \sin \theta_i) [(y - y_i) \cos \theta_i - (x - x_i) \sin \theta_i] \end{cases} \quad (11)$$

In equation (11), the sign is the current sign, and by the property of projection, $\frac{\partial \varepsilon_i}{\partial x}$ is the rate of change of the error function in the x-direction, $\frac{\partial \varepsilon_i}{\partial y}$ is the rate of change of the error function in the y-direction, and $\frac{\partial \varepsilon_i}{\partial \theta_i}$ is the rate of change of the error function in the direction of θ_i . Due to the large number of base stations, for ease of computation, the rates of change in all directions are summed. However, after summation, the accumulation of angles in $\frac{\partial \varepsilon_i}{\partial \theta_i}$ may lead to excessively large updates. Therefore, the mean of $\frac{\partial \varepsilon_i}{\partial \theta_i}$ is taken.

$$\begin{cases} \Delta x_0 = \text{sum}(\frac{\partial \varepsilon_i}{\partial x}) \\ \Delta y_0 = \text{sum}(\frac{\partial \varepsilon_i}{\partial y}) \\ \Delta \theta_0 = \frac{\text{sum}(\frac{\partial \varepsilon_i}{\partial \theta_i})}{n} \end{cases} \quad (12)$$

Next, the iterative equations are formed to find the parameters associated with the position update and to control the degree of change in $\Delta x_0, \Delta y_0, \Delta \theta_0$. Meanwhile, the learning rate α is introduced to update the change parameters in the position iteration.

$$\begin{cases} x_k = x_{k-1} - \alpha \Delta x_0 \\ y_k = y_{k-1} - \alpha \Delta y_0 \\ \theta_k = \theta_{k-1} - \alpha \Delta \theta_0 \end{cases} \quad (13)$$

In equation (13), k denotes the current moment, and $k-1$ denotes the previous moment.

During the solving process, the initial position of the tag in equation (13) can be obtained by using LS in equation (5). The iteration process terminates when the preset number of iterations or iteration threshold is reached, and the final coordinates are output. The flowchart of LS-VPI is demonstrated in Figure 5.

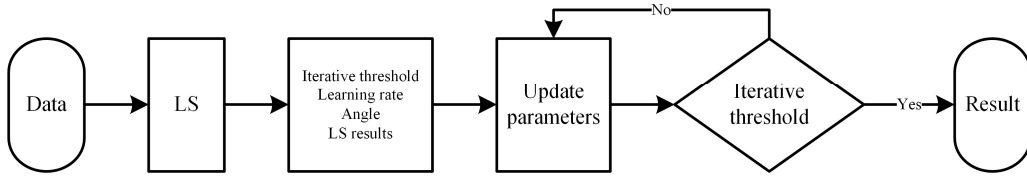


Figure 5 The flowchart of LS-VPI

Sparrow Search Algorithm

SSA is mainly developed based on the behavior of the sparrow population to solve the optimization problem of the objective function. The population foraging for food is divided into two categories: finders and followers. The finders and followers change dynamically throughout the foraging process, and when a follower is in a position where there is more food, it is updated as a finder. The whole process is represented as follows:

$$X = \begin{bmatrix} x_1^1 & x_1^2 & \cdots & \cdots & x_1^a \\ x_2^1 & x_2^2 & \cdots & \cdots & x_2^a \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_m^1 & x_m^2 & \cdots & \cdots & x_m^a \end{bmatrix} \quad (14)$$

where x_i^k ($i=1,2,3,\dots,m; k=1,2,3,\dots,a$) represents the position of the sparrow, i.e., the k -th dimensional component of the i -th sparrow.

$$F_X = \begin{bmatrix} f([x_1^1 & x_1^2 & \cdots & x_1^a]) \\ f([x_2^1 & x_2^2 & \cdots & x_2^a]) \\ \vdots \\ f([x_m^1 & x_m^2 & \cdots & x_m^a]) \end{bmatrix} \quad (15)$$

where $f([x_i^1 \ x_i^2 \ \cdots \ x_i^a])(i=1,2,3,\cdots,m)$ is the fitness value of the i -th sparrow. F_X represents the location distribution of the sparrow population.

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^t \exp\left(-\frac{i}{\alpha \eta}\right) & \text{if } R_a < ST \\ X_{i,j}^t + Q \cdot L & \text{if } R_a \geq ST \end{cases} \quad (16)$$

where $X_{i,j}^t (i=1,2,3,\cdots,m; k=1,2,3,\cdots,a)$ is the k -th dimensional component of the i -th sparrow at the t -th iteration. $\alpha \in [0,1]$ is a random number generated from a uniform distribution. η is a constant representing the maximum number of iterations. $R_a \in [0,1]$ is a random number generated from a uniform distribution, and it represents the warning threshold. $ST \in [0.5,1]$ is a constant representing the safety threshold. Q is a random number generated from a standard normal distribution, and L represents a $1 \times a$ matrix, with all elements equal to 1. The locations of the finders are updated based on the current warning results.

The positions of the followers are updated as follows:

$$X_{i,j}^{t+1} = \begin{cases} Q \exp\left(\frac{X_{worse} - X_{i,j}^t}{t^2}\right) & \text{if } i > \frac{n}{2} \\ X_p^{t+1} + |X_{i,j} - X_p^{t+1}| \cdot A^+ \cdot L & \text{if } i \leq \frac{n}{2} \end{cases} \quad (17)$$

where X_p represents the current optimal position occupied by the finders, and X_{worse} represents the current globally worst position. A represents a $1 \times a$ matrix, where each element is randomly assigned a value of 1 or -1, and $A^+ = A^T (AA^T)^{-1}$. When $i \leq \frac{n}{2}$, it indicates that the i -th follower with a lower fitness has not obtained food and needs to fly to another location to forage for more energy.

In equation (17), the conditional judgment of early warning is introduced. The random generation of a part of the boundary in the population realizes reconnaissance and early warning, providing reliable safety information for population foraging. The location update is represented as follows:

$$X_{i,j}^{t+1} = \begin{cases} X_{best}^t + \beta |X_{i,j}^t - X_{best}^t| & \text{if } f_i > f_g \\ X_{i,j}^t + K \left| \frac{X_{i,j}^t - X_{worst}^t}{f_i - f_w + \delta} \right| & \text{if } f_i = f_g \end{cases} \quad (18)$$

where X_{best} denotes the current global optimal position, β is a random number generated from a standard normal distribution to control the step size, $K \in [-1, 1]$ is a random number generated from a uniform distribution. f_i represents the fitness value of the current sparrow individual; f_g, f_w are the fitness values of the current global best and worst positions among sparrows, and their relationship affects the distribution of populations. δ is an infinitesimal constant used to avoid the situation where $f_i - f_w$ equals 0. K controls the direction and step size of the sparrow's movement during this process.

As the core component of the SSA, the fitness function is the RMSE between the optimal tag position and the real position in LS-VPI.

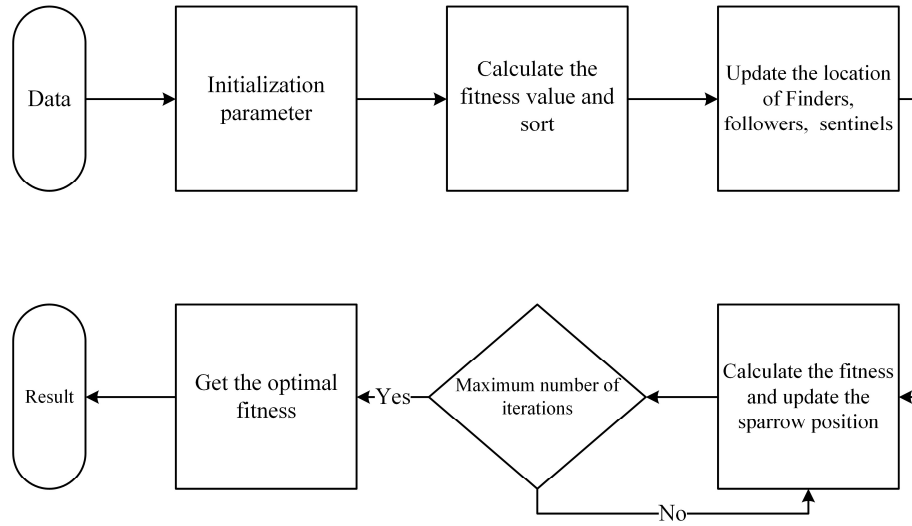


Fig. 6 SSA flowchart

For LS-VPI, the parameters include the initial angle, the number of iterations, the learning rate, and the conditional threshold. Among them, the number of iterations can be set as a constant. The optimal values of the other three parameters need to be searched. The optimization dimension is 3, so a equals 3 in equation (14).

Convolutional Neural Network

CNN is a deep feedforward neural network with local connection and weight sharing. It has the ability of representation learning, which is composed of the input layer, the convolution layer, the pooling layer, the fully connected layer, and the output layer. Its weight-sharing network structure is similar to the biological neural network, which reduces the complexity of the network model and reduces the number of weights. Owing to its excellent input offset invariance and the ability to learn complex models, CNN can eliminate the need for preprocessing input data and exporting low-dimensional input vectors, followed by the convolutional and pooling layers.

The role of the convolutional layer is to perform a convolution operation on the input data, which can be considered as a filtering process. The convolution kernel is a window filter, and a custom-sized convolution kernel is used as the sliding window to convolve the input data during the network training process. The purpose of the convolution operation is to extract different features

of the input, and it consists of many convolutional units, each of which has its local input-receiving domain. It comprises several individual artificial neurons that are vertically organized into planes with different weights to perform multiple feature extractions for the same input. The weights of a unit are shared with all units in the layer, and all units in that layer generate a set of outputs that are passed to the next layer.

There are two important parameters in the convolutional layer, which are bias and activation. The function of the bias vector is to perform a simple linear addition on the convolution data, i.e., the convolution data is added to the data in the bias vector. To increase the nonlinear ability of the network, the activation function is employed to activate the data. After each convolutional layer, a linear correction unit layer is usually introduced. The Rectified Linear Unit (ReLU) is selected as the activation function in this layer. The expression is as follows:

$$f(x) = \max(0, x) \quad (19)$$

If $x > 0$, the function takes the value of x ; if $x < 0$ the function takes a value of 0. The ReLU function has a good fitting effect in the neural network, and the operation is simple, which significantly improves the operation efficiency of the machine. After the linear correction unit layer is the pooling layer. The pooling layer has the advantages of feature dimension reduction, translation invariance, preventing over-fitting to a certain extent, and facilitating optimization.

The structure of the CNN chosen in this paper is as follows :

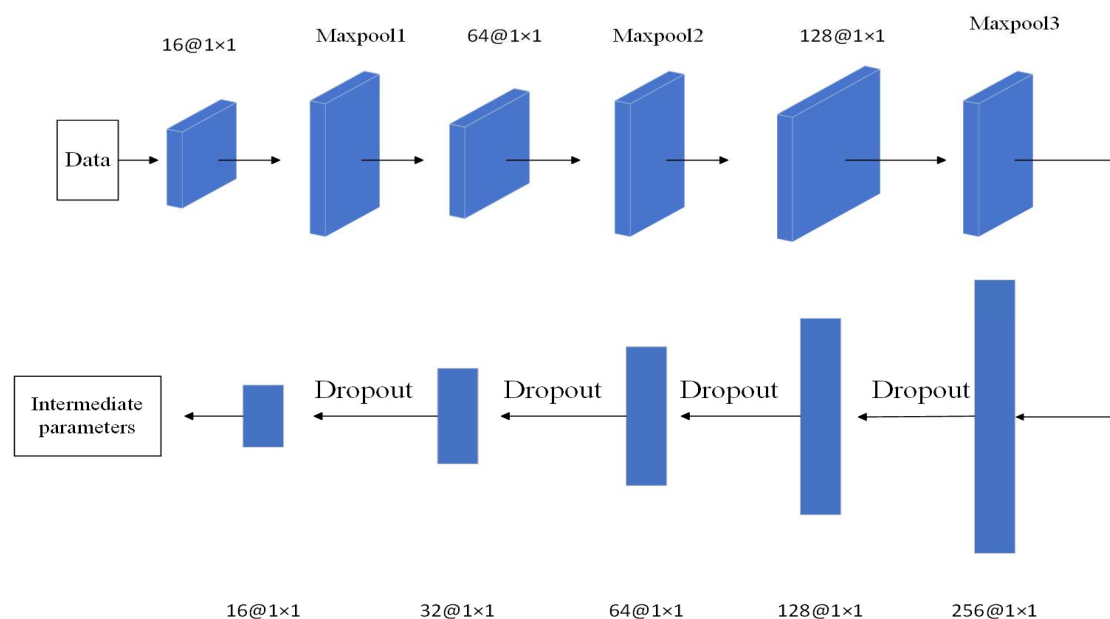


Figure 7 The structure of the CNN

To sum up, the implementation process of the LS-AVPI algorithm consists of two stages: the database preparation stage and the solution stage. The database preparation stage is to adaptively solve the optimal initial angle, learning rate α and threshold. Firstly, SSA is employed to determine the corresponding initial angle, learning rate, and threshold search according to the known point information in the training set. The ranging information, LS results, and the corresponding optimal parameters are combined into the training set, which is then imported into the CNN. The ranging information and LS results are input, and the optimal initial angle, learning rate, and threshold are output for training. The solution stage is to initialize the optimal parameters and the

number of iterations predicted in the adaptive stage and take the least square estimation solution as the basic solution into the iterative calculation to output the final coordinates. The operation flow of the LS-AVPI algorithm is shown in Figure 8:

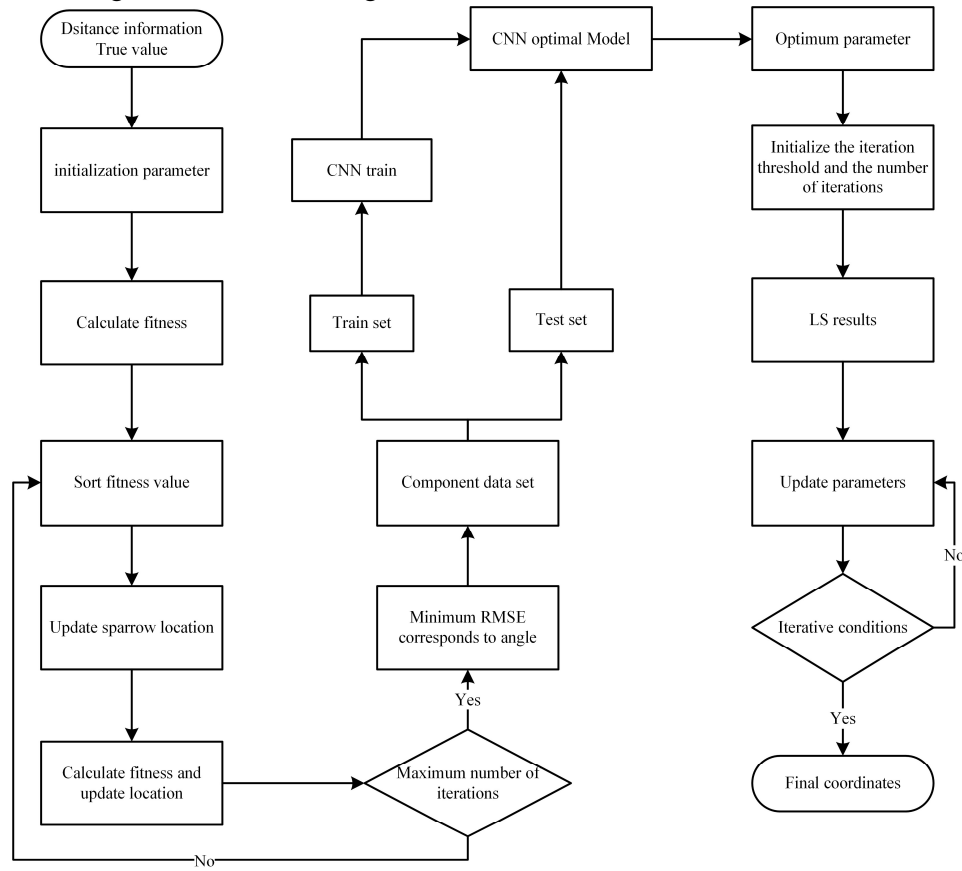


Figure 8 The flowchart of the LS-AVPI algorithm

Results

The test site is a square area of $8.5\text{m} \times 8.5\text{m}$, and the actual coordinate data is measured by the total station. The total station is installed outside the test site, and a custom coordinate system is established. Six base stations are set up in the experimental area. The base stations are xx (229.87 cm, 215.4 cm), 00 (755.85 cm, 901.99 cm), 01 (1493.26 cm, 506.79 cm), 02 (938.74 cm, -186.51 cm), 03 (1219.2 cm, 174.54 cm), and 04 (474.51 cm, 552.1 cm). The location distribution is shown in Figure 9.

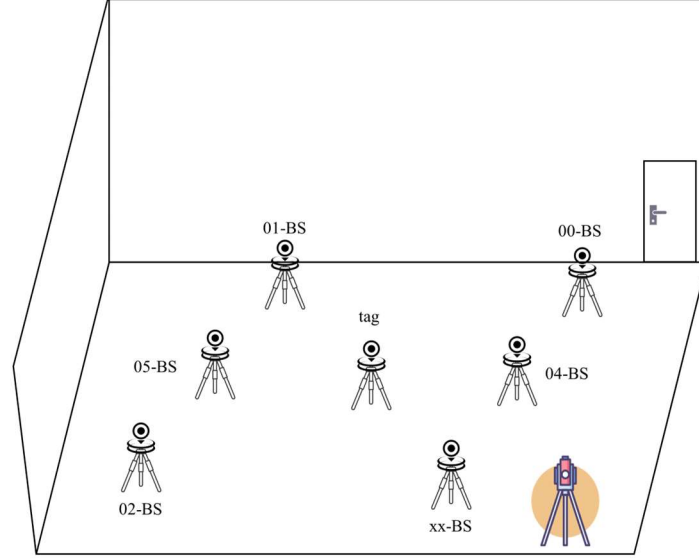


Figure 9 The measurement diagram

Static measurement is performed first. Ten positions are randomly placed in the measurement area, and six base stations are observed at the same time to obtain the distance information from the tag to the base station. The true coordinates of points near the label are measured. The measurement results of the nearby points are marked as the training set, and the measurement results of the 10 points are marked as the test set to complete the collection of static experimental data sets. The position of the point is illustrated in Figure 10.

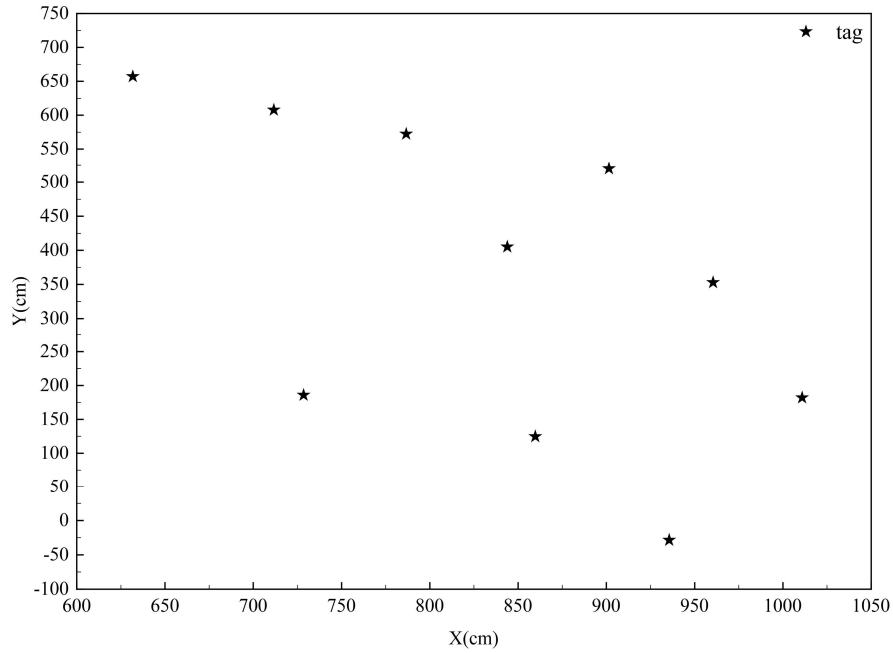


Figure 10 The position distribution of the ten points

To verify the effectiveness of the algorithm when the base station adopts a non-standard rectangular topology, the data of four base stations are selected from six base stations to enrich the data set. The algorithms chosen for comparison in this experiment include constrained weighted least square (CWLS), iterative least square (ILS), LS, and two-step weighted least square (TSWLS). Among them, LS and TSWLS are classic time-of-arrival algorithms, CWLS is greatly affected by the weight, while LS-AVPI also involves iterative operations. The weighted matrix is a unit matrix,

the number of iterations is 1000, and the experimental accuracy evaluation metric is RMSE. The results are demonstrated in Figure 11 and Figure 12.

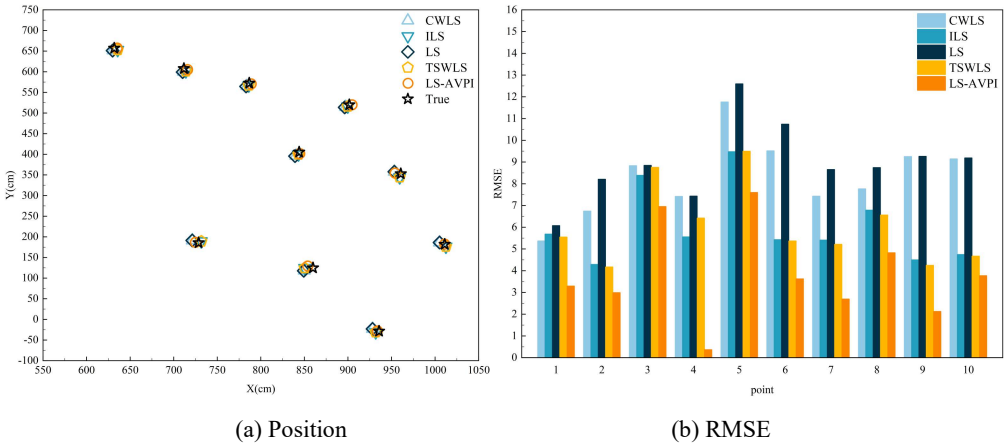


Figure 11 The position of the ten points and comparison of RMSE

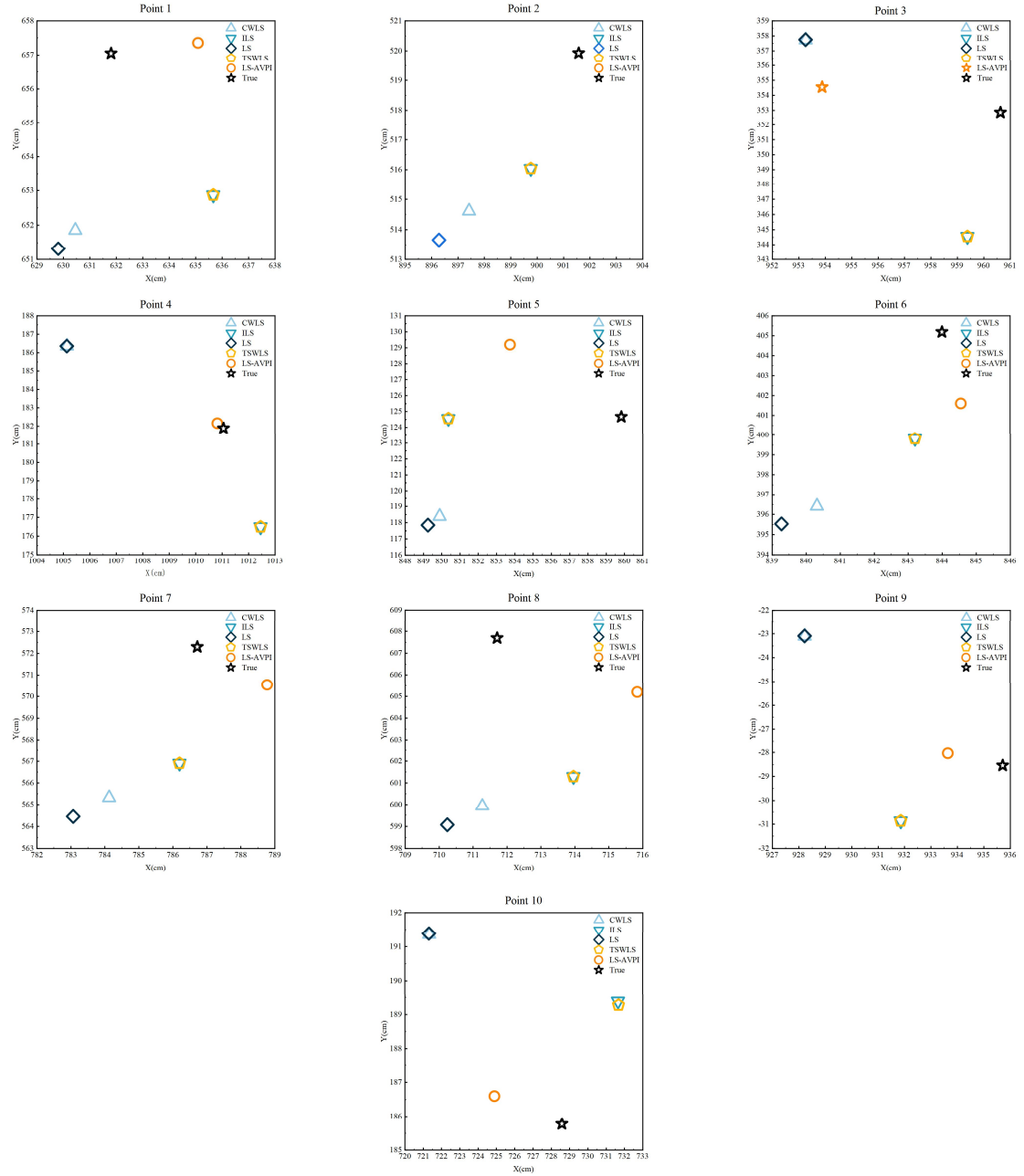


Figure 12 Comparison of a single point position

Figure 11(a) shows the positioning results of each algorithm. Specifically, the results of each algorithm are distributed near the true value, and there is no large offset. The positioning results of LS-AVPI are closer to the true value. At (859.84cm, 124.66cm), the results of CWLS, ILS, LS, and TSWLS are slightly farther than the true value. The positioning results of LS have the largest deviation, and they are all located on the left side of the true value. However, LS-AVPI corrects LS results nearest to the true value, so it obtains better positioning results than other algorithms. Figure 11(b) shows the comparison of the RMSE of each algorithm. The RMSE of LS-AVPI is the smallest among all the comparison algorithms. The detailed data are listed in Table 1.

Table 1 The RMSE of each algorithm

(cm)

algorithm \ point	CWLS	ILS	LS	TSWLS	LS-AVPI
1	5.4	5.7	6.1	5.6	3.3

2	6.7	4.3	8.2	4.2	3
3	8.8	8.4	8.8	8.8	7
4	7.4	5.6	7.4	6.4	0.4
5	11.8	9.5	12.6	9.5	7.6
6	9.5	5.4	10.7	5.4	3.6
7	7.4	5.4	8.7	5.2	2.7
8	7.8	6.8	8.7	6.6	4.8
9	9.2	4.5	9.3	4.2	2.1
10	9.1	4.7	9.2	4.7	3.8

The average root mean square errors of CWLS, ILS, LS, TSWLS, and LS-AVPI are 8.3 cm, 6 cm, 9 cm, 6.1 cm, and 3.8 cm, respectively. At the fourth point, the minimum root mean square error of LS-AVPI reaches 0.4 cm, which is 94.6 %, 92.9 %, 94.6 %, and 93.8 % higher than other algorithms. In terms of the overall average root mean square error, the accuracy of LS-AVPI is improved by 54.2 %, 36.7 %, 57.8 %, and 37.7 % respectively compared with other algorithms. Five points parallel to the y-axis of the survey area are selected, and three sets of positioning data are selected for each point. The results are presented in Figure 13.

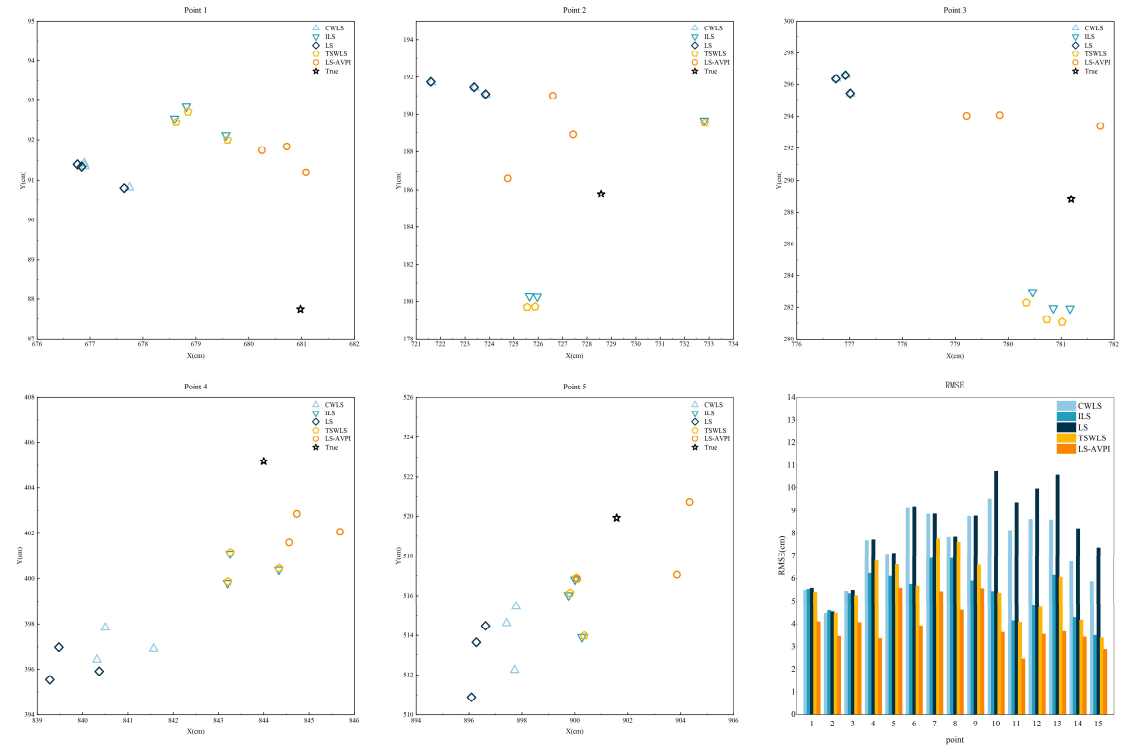


Figure 13 Comparison of three-times measurement positioning

In Figure 13, Points 1 to 5 are the three-times positioning positions of each algorithm at each point. Due to the deviation of the distance data of the equipment positioning, the positioning results of CWLS, ILS, LS, and TSWLS all exhibit certain regularity and are distributed on the same side of the real value. Since LS-AVPI is developed based on the LS results, it presents the same distribution as the LS in the positioning of Point 1 and Point 4. However, in the overall distribution, LS-AVPI positioning results are more evenly distributed. LS-AVPI has a smaller RMSE overall. The results are presented in Table 2.

Table 2 The RMSE of each algorithm measured three times.

(cm)

algorithm	CWLS	ILS	LS	TSWLS	LS-AVPI
-----------	------	-----	----	-------	---------

point											
1	5.5		5.5		5.6		5.4		4.1		
2	4.5	5.2	4.6	5.1	4.5	5.2	4.5	5	3.5	3.9	
3	5.5		5.3		5.5		5.2		4.1		
4	7.7		6.2		7.7		6.8		3.4		
5	7.1	8	6.1	6	7.1	8	6.6	6.4	5.6	4.3	
6	9.2		5.8		9.2		5.7		3.9		
7	8.8		6.9		8.6		7.7		5.4		
8	7.8	8.4	6.9	6.6	7.8	8.4	7.6	7.3	4.6	5.2	
9	8.7		6		8.8		6.6		5.6		
10	9.5		5.4		10.7		5.4		3.6		
11	8.1	8.7	4.1	4.8	9.4	10	4.1	4.7	2.4	3.2	
12	8.6		4.8		10		4.7		3.5		
13	8.6		6.2		10.6		6.1		3.7		
14	6.7	7.1	4.3	4.7	8.2	8.7	4.2	4.6	3.4	3.3	
15	5.9		3.5		7.4		3.4		2.9		

In Table 2, for each algorithm, the first column shows the RMSE, and the second column shows the average of the three results at the same point. The average RMSE values of CWLS, ILS, LS, TSWLS, and LS-AVPI are 7.5 cm, 5.4 cm, 8.1 cm, 5.6 cm, and 4 cm, respectively. Compared with other algorithms, LS-AVPI has 46.7 %, 26 %, 50.6 %, and 28.6 % higher accuracy, respectively.

The dynamic test involves a rectangle and a straight line, and the positioning sampling is performed every two steps. The straight line selects two routes, which are parallel to the x-axis and the y-axis, respectively. When traveling along the marked rectangular route, the establishment of the data set is the same as that of the static experiment. The points on the route are those of the test set, and the points measured on both sides of the route are the test set. The positioning results of the rectangular route are shown in Figure 14.

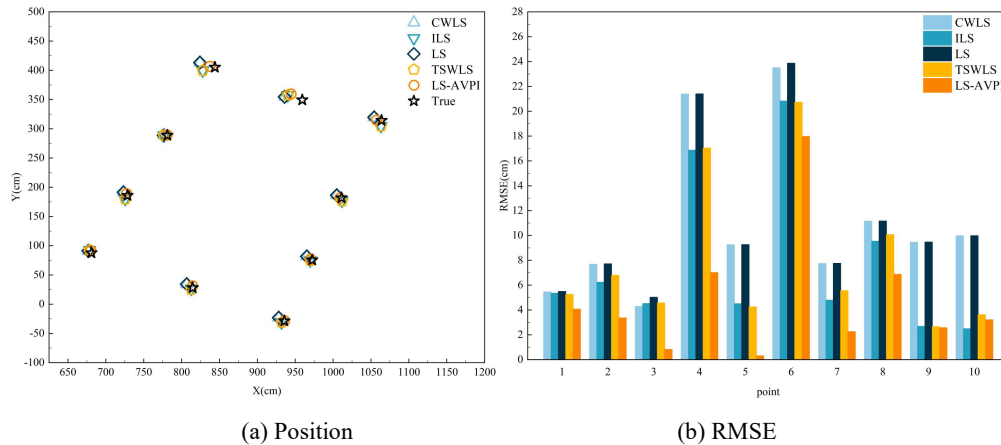


Figure 14 The positioning results of the rectangular route

Since the measuring area is an oblique rectangle, the measuring points are inclined. Figure 14(a) shows the positioning results of each algorithm. Compared with LS, CWLS, ILS, and TSWLS, the positioning result of LS-AVPI is closer to the true value position. Because the upper half of the rectangle is closer to the base station laid horizontally in the measuring area, the data will be disturbed, and the random error will increase, but LS-AVPI still obtains a better positioning effect

than other algorithms. Figure 14(b) shows the comparison of the RMSE, and the results show that LS-AVPI is superior to other algorithms. Figure 15 illustrates the result of single-point positioning. Combined with Table 3, the positioning advantages of LS-AVPI can be seen more intuitively. The average RMSE values of CWLS, ILS, LS, TSWLS, and LS-AVPI are 11 cm, 7.8 cm, 11.1 cm, 8 cm, and 4.8 cm, respectively. At the 3rd and 5th points, the RMSE of LS-AVPI reaches 0.8 cm and 0.3 cm, respectively, and the positioning accuracy is higher. Compared with CWLS, ILS, LS, and TSWLS, the positioning accuracy is increased by 81.4 %, 82.2 %, 84 %, and 82.6 %, respectively. In the overall mean square error comparison, the accuracy of LS-AVPI is improved by 56.4 %, 38.5 %, 56.8 %, and 40 % respectively compared with other algorithms.

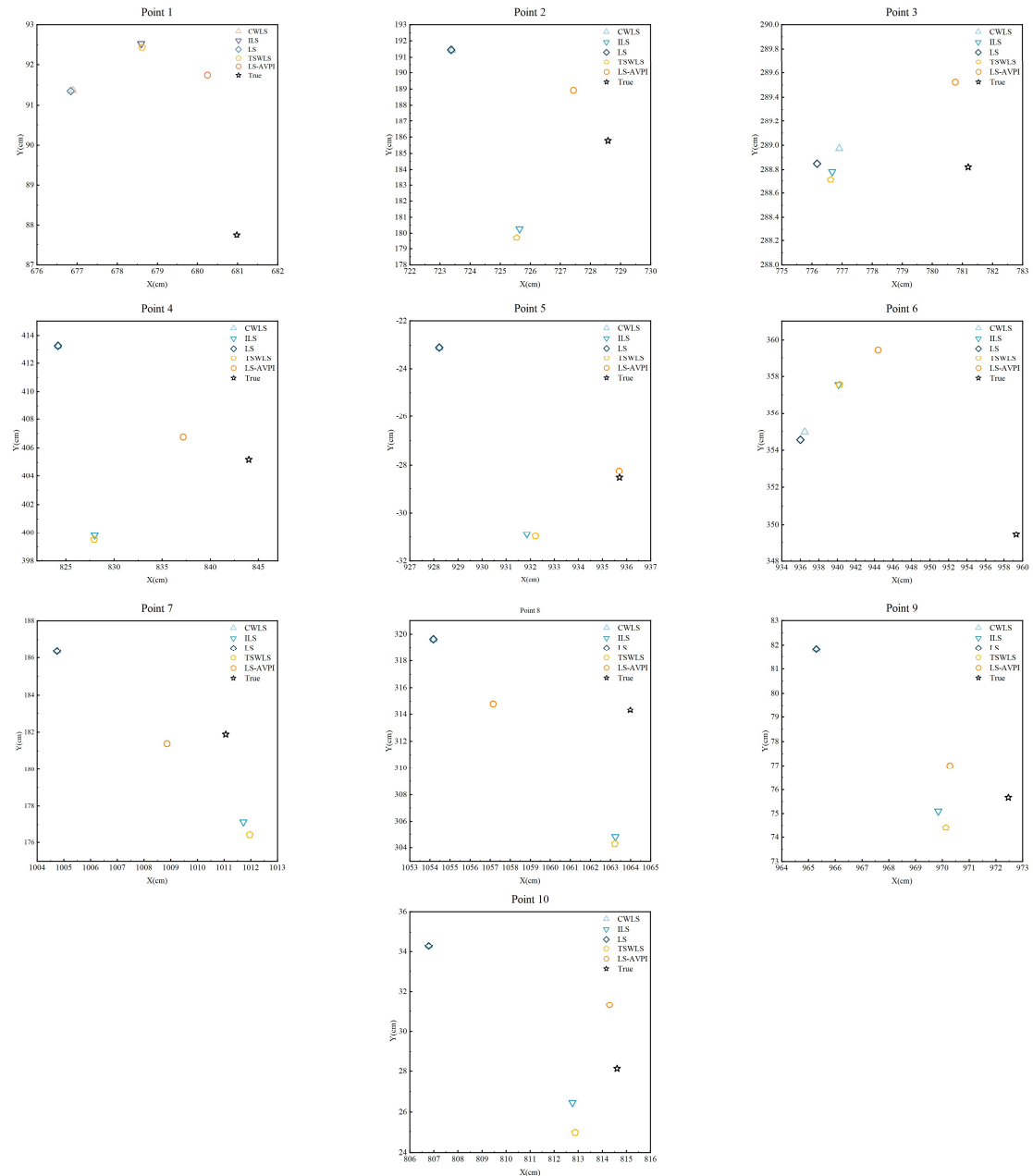


Figure15 The single-point positioning results of the rectangular route

Table 3 The RMSE of each algorithm for the rectangular route (cm)

algorithm	CWLS	ILS	LS	TSWLS	LS-AVPI
-----------	------	-----	----	-------	---------

point					
1	5.4	5.3	5.5	5.2	4.1
2	7.7	6.2	7.7	6.8	3.4
3	4.3	4.5	5	4.6	0.8
4	21.4	16.9	21.4	17	7
5	9.2	4.5	9.3	4.2	0.3
6	23.5	20.8	23.9	20.7	18
7	7.7	4.8	7.8	5.6	2.2
8	11.1	9.5	11.2	10.1	6.9
9	9.5	2.7	9.5	2.7	2.6
10	10	2.5	10	3.6	3.2

In the straight part, the transverse direction is parallel to the x-axis direction of the survey area. The results are as follows. In Figure 16, each small rectangular area shows the positioning results of the five algorithms. The results of CWLS, ILS, LS, and TSWLS show aggregation characteristics, and their positioning performance tends to be consistent, while the results of LS-AVPI tend to stay away from the aggregation area and close to the true values, thereby reducing the error and improving the accuracy. Combined with Table 4 and Figure 17, the maximum RMSE of most column points is greater than 15 cm, and the maximum value is 23.9 cm. After the treatment of LS-AVPI, the maximum RMSE is 18 cm, which is 23.4 %, 13.5 %, 24.7 %, and 13 % higher than that of CWLS, ILS, LS, and TSWLS, respectively. Overall, the mean RMSE values of CWLS, ILS, LS, TSWLS, and LS-AVPI are 19.6 cm, 17.8 cm, 19.9 cm, 17.9 cm, and 13 cm, respectively. Compared with other algorithms, the accuracy of LS-AVPI is improved by 33.7 %, 27 %, 34.7 %, and 27.4 %, respectively.

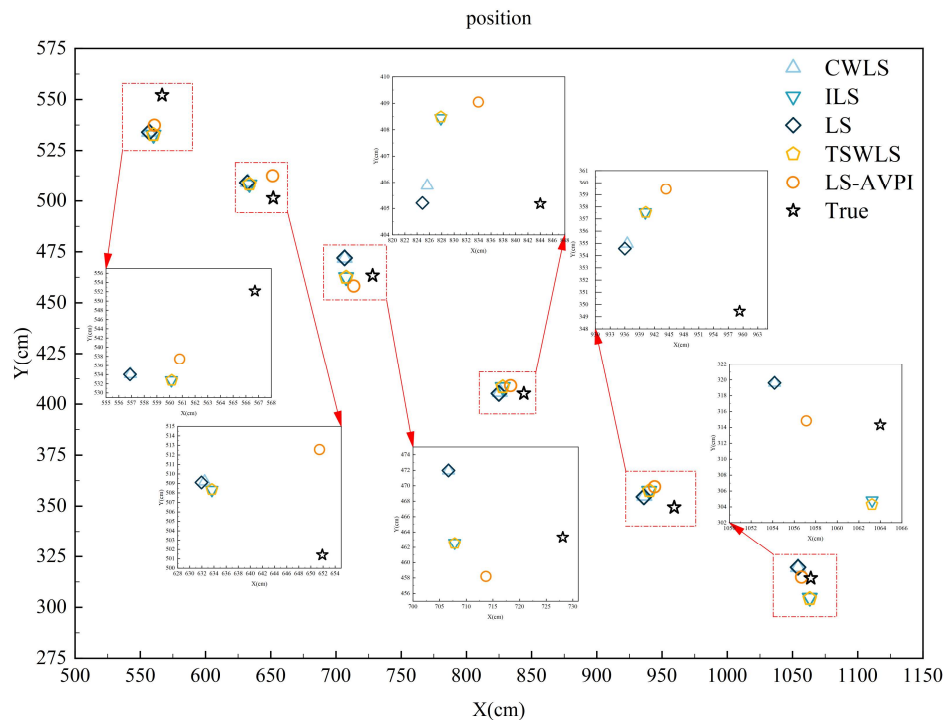


Figure 16 The positioning results of the parallel X-axis route

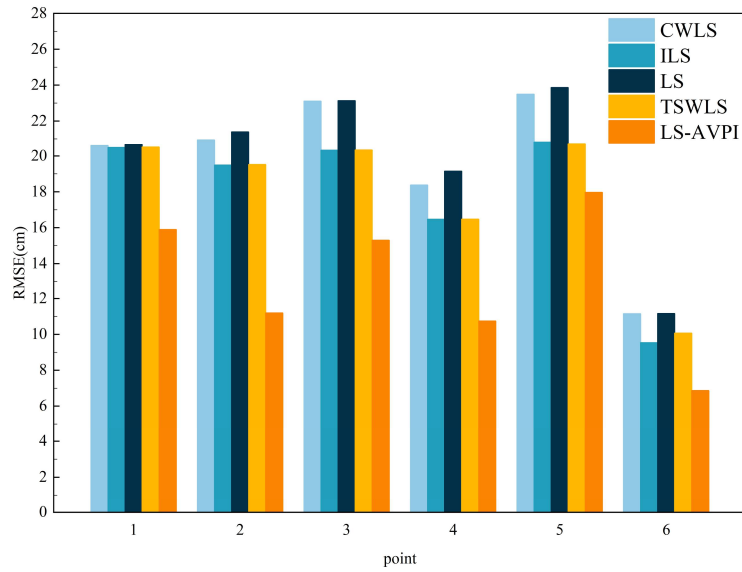


Figure 17 The positioning accuracy of the parallel X-axis route

Table 4 The RMSE of each algorithm in parallel X-axis route (cm)

algorithm point	CWLS	ILS	LS	TSWLS	LS-AVPI
1	20.6	20.5	20.6	20.5	15.9
2	20.9	19.5	21.4	19.5	11.2
3	23.1	20.3	23.1	20.3	15.3
4	18.4	16.5	19.1	16.5	10.7
5	23.5	20.8	23.9	20.7	18
6	11.1	9.5	11.2	10.1	6.9

The longitudinal route is a straight line parallel to the y-axis of the survey area. The results are presented in Figure 18, Figure 19, and Figure 20.

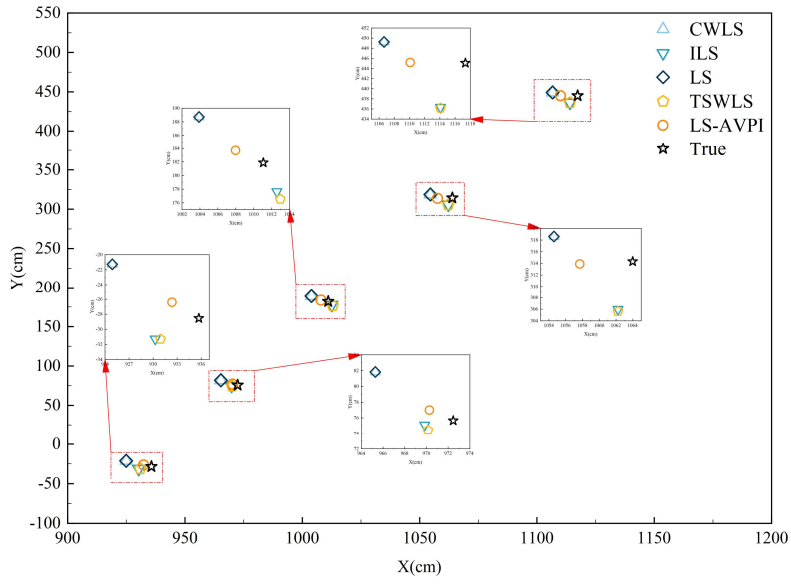


Figure 18 The positioning results of the parallel X-axis

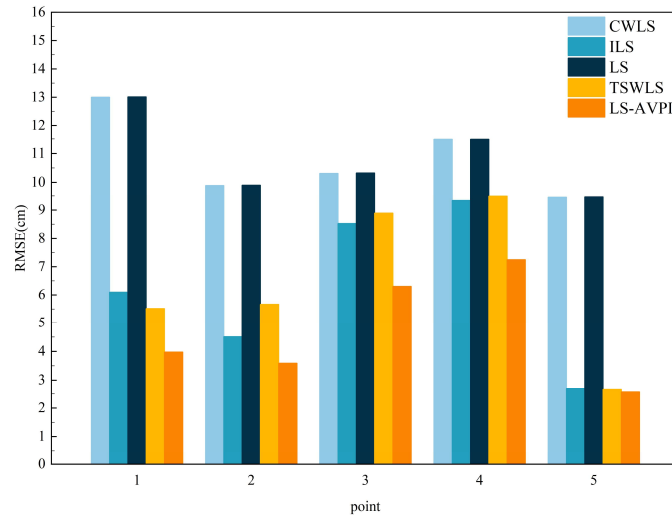


Figure 19 The positioning results in the longitudinal X-direction route

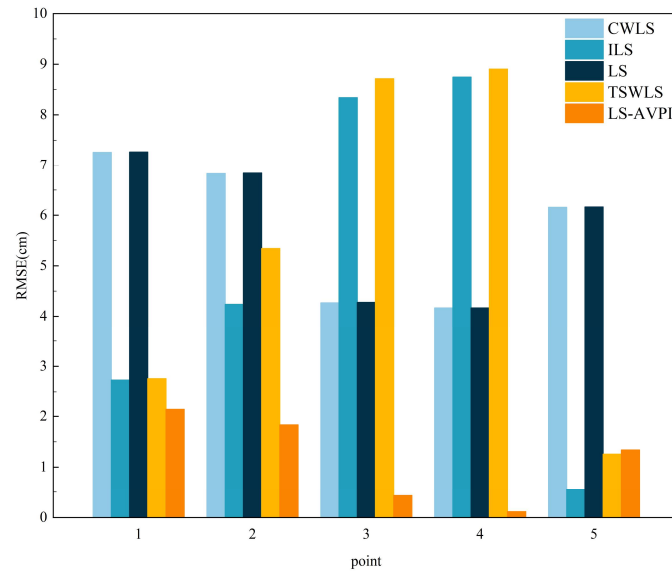


Figure 20 The positioning results in the longitudinal Y-direction route

In the comparison of the position of the points, the positioning results of CWLS, ILS, LS, and TSWLS are still concentrated and relatively scattered near the true value, but they do not show high accuracy in a single direction. Combined with Figure 20, LS-AVPI shows high accuracy in the Y-axis direction, so its positioning accuracy in the Y-axis direction is not affected as much as possible, and only the error in the X-axis direction should be concerned. Combined with Table 5, the RMSE of LS positioning is approximately 10 cm, which is significantly worse than that of ILS and TSWLS. After LS-AVPI treatment, the maximum RMSE is 7.2 cm, which is 37.4 %, 22.6 %, 37.4 %, and 24.2 % higher than that of CWLS, ILS, LS, and TSWLS. Generally, the average RMSE values of CWLS, ILS, LS, TSWLS, and LS-AVPI are 10.8 cm, 6.2 cm, 10.8 cm, 6.4 cm, and 4.7 cm, respectively. The accuracy of LS-AVPI is 56.5 %, 24.2 %, 56.5 %, and 26.6 % higher than other algorithms.

Table 5 The RMSE of each algorithm for the longitudinal route (cm)

algorithm point	CWLS	ILS	LS	TSWLS	LS-AVPI
1	13.0	6.1	13.0	5.5	4.0
2	9.9	4.5	9.9	5.7	3.6
3	10.3	8.5	10.3	8.9	6.3
4	11.5	9.3	11.5	9.5	7.2
5	9.5	2.7	9.5	2.7	2.6

The above has shown the dynamic point test of the algorithm. Only by comparing a certain route or scatter, it does not reflect the overall performance. The following compares all the data in the static experiment and the dynamic experiment respectively. The results are demonstrated in Figures 21 to 24.

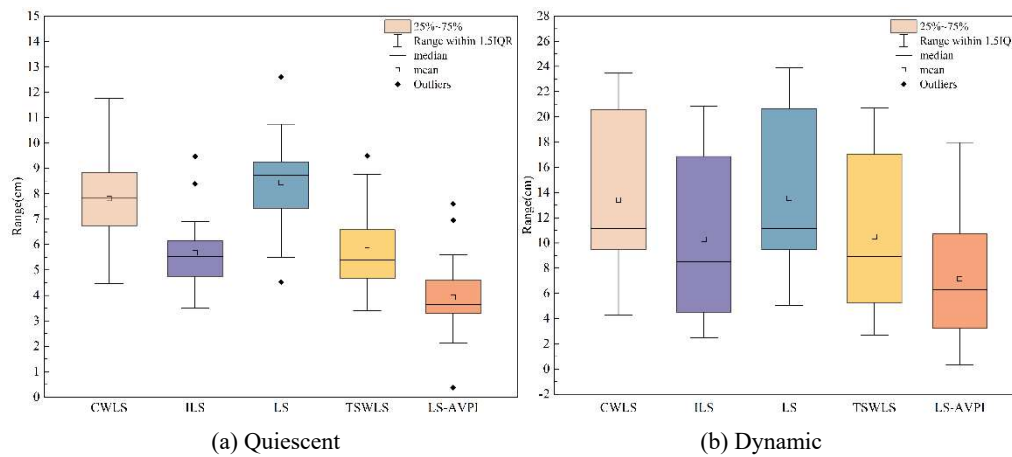
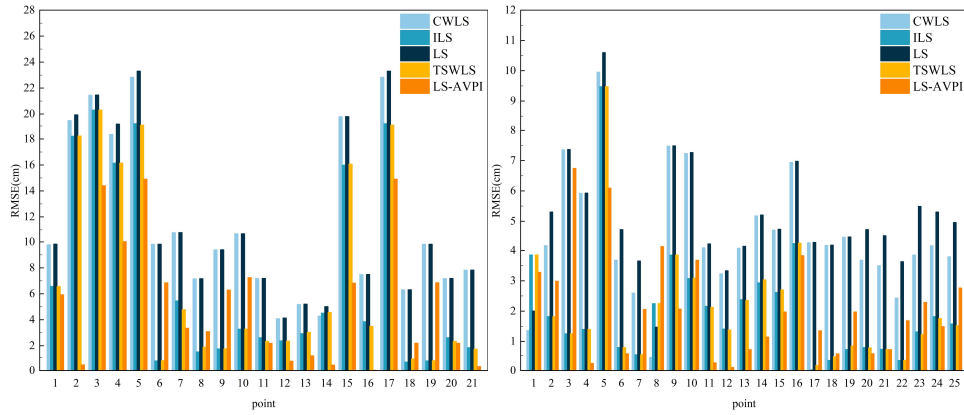


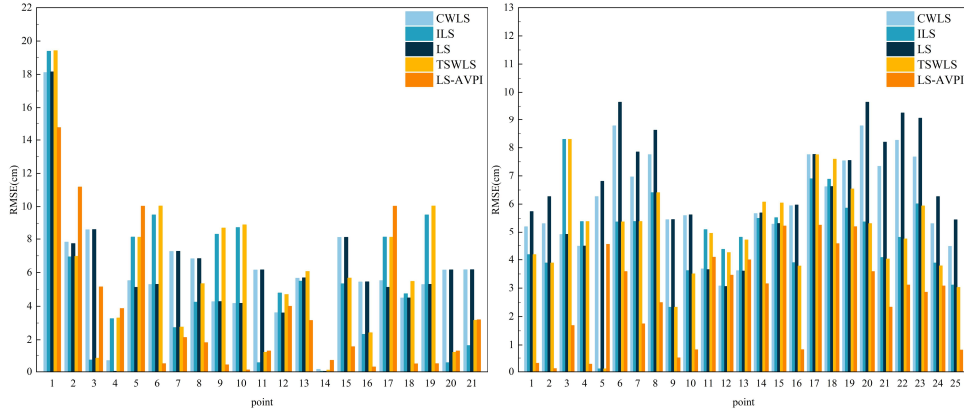
Figure 21 Static and dynamic error distribution



(a) Quiescent

(b) Dynamic

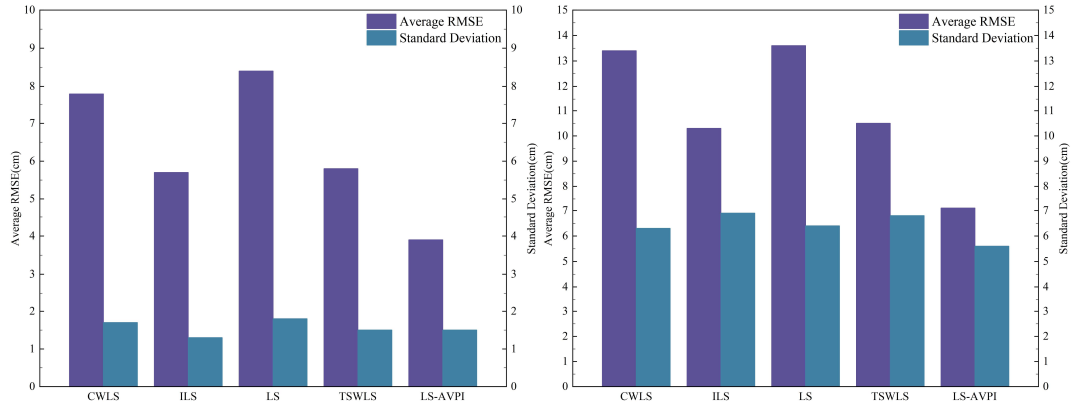
Figure 22 Static and dynamic error distribution in the X direction



(a) Quiescent

(b) Dynamic

Figure 23 Static and dynamic error distribution in the Y direction



(a) Quiescent

(b) Dynamic

Figure 24 Static and dynamic RMSE and standard deviation

In Figure 21, the mean and median of the positioning results of LS-AVPI are smaller than those of other algorithms. Combined with Figure 22, Figure 23, and Figure 24, the label is less disturbed in static experiments, and the standard deviations of the five algorithms are similar. The average RMSE values of CWLS, ILS, LS, TSWLS, and LS-AVPI are 7.8 cm, 5.6 cm, 8.5 cm, 5.8 cm, and 3.9 cm, respectively. The average RMSE of LS-AVPI is lower than that of CWLS, ILS, LS, and TSWLS. Compared with other algorithms, the accuracy is improved by 50 %, 30.4 %, 54.1 %, and 32.8 %, respectively. The standard deviation of LS-AVPI is smaller than that of CWLS, ILS, LS, and TSWLS, and the results exhibit a small fluctuation. The average RMSE values of CWLS, ILS,

LS, TSWLS, and LS-AVPI are 13.4 cm, 10.3 cm, 13.5 cm, 10.4 cm, and 7.1 cm, respectively. Compared with other algorithms, the accuracy of LS-AVPI is improved by 47 %, 31.1 %, 47.4 %, and 31.7 % respectively. Combined with the positioning accuracy in the X and Y directions in the static and dynamic states in Table 5, the LS-AVPI algorithm has the smallest error in each state and direction. In general, LS-AVPI performs better than CWLS, ILS, LS, and TSWLS.

Table 6 Errors in the X and Y directions

(cm)

algorithm error	CWLS	ILS	LS	TSWLS	LS-AVPI
Static X direction	4.5	2.1	5	2.1	2.1
Static Y direction	6.1	4.8	6.5	4.9	2.7
Dynamic X direction	11.5	7.2	11.7	7.1	5.3
Dynamic Y direction	6	5.5	5.9	5.8	3.6

Discussion

The LS-AVPI algorithm is proposed in this paper to improve the localization accuracy of UWB by using the distance estimation method and addressing the random error in UWB localization with asymmetric topological placement of base stations.

In this paper, the LS-AVPI algorithm is proposed cutting the angle into the UWB positioning settlement. Then, the error equation is established according to the projection relationship, and it is adjusted by iteration. The positioning model is employed as the fitness function of SSA. Considering the update of the fitness function, the measured true value is taken. For the process of estimating parameters with known true values, the measurement is fixed-point, a more stable CNN is used for learning, and the prediction results are fed into the iterative model to calculate the final positioning results. Experimental results show that LS-AVPI is superior to the other four algorithms in positioning accuracy, mainly in the single-axis direction. Since the learning rate is utilized to correct the single-axis error in the iteration, the positioning result is more inclined to the high precision of the single-axis, and the result in other axes is slightly corrected to ensure the improvement of its overall accuracy. Because the training data contains the distance and LS estimation position information of the asymmetric topology network structure, the model has better positioning processing capability.

However, our method has certain defects. Due to the large number of algorithms involved, the early adaptive phase of the algorithm has a high demand for computing resources. Meanwhile, LS-AVPI relies on the SSA to find the optimal parameters. When there is a large error in the establishment of the database, it will affect the final positioning results.

UWB performs well in this experiment, but considering the environmental factors and real-time applications, our future work will optimize the fitness function in the LS-AVPI algorithm, to reduce the demand for computing resources and eliminate the need for neural network training for positioning. Meanwhile, it is necessary to perform non-line-of-sight experiments and modifications on the algorithm to enable it to achieve positioning in non-line-of-sight situations.

Conclusion

In this paper, a positioning model is constructed by vector thought, and the model parameters

are optimized by SSA. The optimal data is matched with the distance and position data to establish the data set. Then, CNN is used to predict the parameters, and the final positioning result is solved by LS-VPI. In the experiment, LS-AVPI achieves higher accuracy than CWLS, ILS, LS, and TSWLS. The error is minimized by correcting the position in the uniaxial direction, and it is suitable for the base station to form an asymmetric topology network.

(1) In the static experiment, the average RMSE values of CWLS, ILS, LS, TSWLS, and LS-AVPI are 7.8 cm, 5.6 cm, 8.5 cm, 5.8 cm, and 3.9 cm, respectively, showing 50 %, 30.4 %, 54.1 %, and 32.8 % higher accuracy than other algorithms.

(2) In the dynamic experiment, the average RMSE values of CWLS, ILS, LS, TSWLS, and LS-AVPI are 13.4 cm, 10.3 cm, 13.5 cm, 10.4 cm, and 7.1cm, showing 47 %, 31.1 %, 47.4 %, and 31.7 % higher accuracy than other algorithms.

Reference

1. CHEN Ruizhi, CHEN Liang. Indoor Positioning with Smartphones:The State-of-the-art and the Challenges[J]. *Acta Geodaetica et Cartographica Sinica*, 2017, 46(10): 1316-1326.
2. Xu Jingcheng, Lian Zengzeng, Dong Jiaqi, et al. Research on anti-multipath error of BDS based on WPT decomposition and reconstruction algorithm[J]. *Science Technology and Engineering*, 2022, 22(35): 15477-15484.
3. Cahyadi, M. N., Asfihani, T., Mardiyanto, R. & Erfianti, R. Performance of GPS and IMU sensor fusion using unscented Kalman filter for precise i-Boat navigation in infinite wide waters. *Geodesy and Geodynamics* **14**, 265-274 (2023).
4. Xu, F., An, N. K. & Zhiyuan, L. WiFi Access Points Line-of-Sight Detection for Indoor Positioning Using the Signal Round Trip Time. *Remote Sensing* **14** (2022).
5. Xin, Z. H. *et al.* The location tracking and intelligent street lighting control system based on the RFID and Zigbee for energy efficiency. *Int. J. Elec. Eng. Educ.*, 11, doi:10.1177/0020720920944441 (2020).
6. Zhuang, Y. *et al.* Bluetooth Localization Technology: Principles, Applications, and Future Trends. *IEEE Internet Things J.* **9**, 23506-23524, doi:10.1109/jiot.2022.3203414 (2022).
7. Dong, J. Q., Lian, Z. Z., Xu, J. C. & Yue, Z. UWB Localization Based on Improved Robust Adaptive Cubature Kalman Filter. *Sensors* **23**, 18, doi:10.3390/s23052669 (2023).
8. Chaoyong, Y. *et al.* A Novel Deep Learning Approach to 5G CSI/Geomagnetism/VIO Fused Indoor Localization. *Sensors* **23** (2023).
9. CHEN Ruizhi, QIAN Long, NIU Xiaoguang, XU Shihao, CHEN Liang, QIU Chao. Fusing acoustic ranges and inertial sensors using a data and model dual-driven approach[J]. *Acta Geodaetica et Cartographica Sinica*, 2022, 51(7): 1160-1171.
10. Wenqi, Q., Qinghua, Z., Jianye, L., Yixue, L. & Yan, H. A new monocular vision simultaneous localization and mapping process for high-precision positioning in structured indoor environments. *Measurement Science and Technology* **33** (2022).
11. DONG Jiaqi, LIAN Zengzeng, XU Jingcheng, WEI Fengyuan. Noise reduction of Chan sequential adjustment combination algorithm in ultra wide band positioning[J]. *Bulletin of Surveying and Mapping*, 2023, 0(1): 95-100, 140.
12. DONG Jiaqi, LIAN Zengzeng, XU Jingcheng, LU Xinghao. Research on fusion Kalman filter algorithm in UWB positioning. *Science of Surveying and Mapping* **47**, 10-17, doi:10.16251/j.cnki.1009-2307.2022.05.002 (2022).
13. Long, K., Kong, D. F. N., Zhang, K., Tian, C. & Shen, C. A CSI-Based Indoor Positioning System Using Single UWB Ranging Correction. *Sensors* **21**, doi:10.3390/s21196447 (2021).
14. Cheng, J. H., Yu, P. P. & Huang, Y. R. Application of Improved Kalman Filter in Under-Ground Positioning System of Coal Mine. *Ieee Transactions on Applied Superconductivity* **31**, doi:10.1109/tasc.2021.3101751 (2021).
15. Xu, Y., Shmaliy, Y. S., Ahn, C. K., Tian, G. H. & Chen, X. Y. Robust and accurate UWB-based indoor robot localisation using integrated EKF/EFIR filtering. *Iet Radar Sonar and Navigation* **12**, 750-756, doi:10.1049/iet-rsn.2017.0461 (2018).
16. Cong, F., Hong, Z., Lin, X. & Li, H. Peak Ratio Iteration-Based Leading-Edge Detection Algorithm in UWB Localization. *Wireless Personal Communications*, doi:10.1007/s11277-023-10517-x (2023).

-
17. Wu, J. K., Zhang, Z. Q., Zhang, S. L., Kuang, Z. W. & Zhang, L. P. UWB Positioning Algorithm Based on Fuzzy Inference and Adaptive Anti-NLOS Kalman Filtering. *Applied Sciences-Basel* **12**, doi:10.3390/app12126183 (2022).
 18. Wang, X. Y., Gao, L. J., Mao, S. W. & Pandey, S. CSI-Based Fingerprinting for Indoor Localization: A Deep Learning Approach. *Ieee Transactions on Vehicular Technology* **66**, 763-776, doi:10.1109/tvt.2016.2545523 (2017).
 19. Poulose, A. & Han, D. S. UWB Indoor Localization Using Deep Learning LSTM Networks. *Applied Sciences-Basel* **10**, doi:10.3390/app10186290 (2020).
 20. Sung, S. M., Kim, H. & Jung, J. I. Accurate Indoor Positioning for UWB-Based Personal Devices Using Deep Learning. *IEEE Access* **11**, 20095-20113, doi:10.1109/access.2023.3250180 (2023).
 21. Jiang, C. H. *et al.* UWB NLOS/LOS Classification Using Deep Learning Method. *IEEE Commun. Lett.* **24**, 2226-2230, doi:10.1109/lcomm.2020.2999904 (2020).
 22. Pei, Y., Chen, R. Z., Li, D. R., Xiao, X. W. & Zheng, X. Y. FCN-Attention: A deep learning UWB NLOS/LOS classification algorithm using fully convolution neural network with self-attention mechanism. *Geo-Spat. Inf. Sci.*, 20, doi:10.1080/10095020.2023.2178334 (2023).
 23. Xue, J. & Shen, B. A novel swarm intelligence optimization approach: sparrow search algorithm. *Systems Science & Control Engineering* **8**, 22-34, doi:10.1080/21642583.2019.1708830 (2020).
 24. Mirjalili, S., Mirjalili, S. M. & Lewis, A. Grey Wolf Optimizer. *Advances in Engineering Software* **69**, 46-61, doi:https://doi.org/10.1016/j.advengsoft.2013.12.007 (2014).
 25. Mirjalili, S. & Lewis, A. The Whale Optimization Algorithm. *Advances in Engineering Software* **95**, 51-67, doi:https://doi.org/10.1016/j.advengsoft.2016.01.008 (2016).
 26. Zhang, H. *et al.* A Multi-Strategy Improved Sparrow Search Algorithm for Solving the Node Localization Problem in Heterogeneous Wireless Sensor Networks. *Applied Sciences-Basel* **12**, doi:10.3390/app12105080 (2022).
 27. Tlili, S., Mnasri, S. & Val, T. UWB time of flight-based indoor IoT localisation solution with deep learning optimised by meta-heuristics. *International Journal of Sensor Networks* **44**, doi:10.1504/ijsn.2024.136695 (2024).
 28. Gou, P. Z., He, B. & Yu, Z. Y. A Node Location Algorithm Based on Improved Whale Optimization in Wireless Sensor Networks. *Wireless Communications & Mobile Computing* **2021**, doi:10.1155/2021/7523938 (2021).
 29. Zhao, J. L., Li, C. X. & Wang, L. Wireless Position Method for Indoor Personnel Based on Improved Neural Network by Chaotic Whale Optimization Algorithm. *Journal of Nanoelectronics and Optoelectronics* **17**, 427-435, doi:10.1166/jno.2022.3231 (2022).
 30. LI Jiming, ZHANG Jianhui, WANG Xiaohan, WANG Shanpei, ZHAO Meng, CHENG Xuezheng. Downhole TDOA positioning method with improved whale optimization algorithm and Taylor series. *Experimental Technology and Management* **39**, 30-36+54, doi:10.16791/j.cnki.sjg.2022.12.005 (2022).
 31. Tian, Y. L. *et al.* Application of a long short-term memory neural network algorithm fused with Kalman filter in UWB indoor positioning. *Scientific Reports* **14**, doi:10.1038/s41598-024-52464-y (2024).
 32. Dong, J. Q., Lian, Z. Z., Xu, J. C. & Yue, Z. An Improved Adaptive Sparrow Search

-
- Algorithm for TDOA-Based Localization. *Isprs International Journal of Geo-Information* **12**, doi:10.3390/ijgi12080334 (2023).
33. LI Yali, WANG Shuqin, CHEN Qianru, WANG Xiaogang. Comparative Study of Several New Swarm Intelligence Optimization Algorithms[J]. *Computer Engineering and Applications*, 2020, 56(22): 1-12.
34. KUANG Bin CHEN Fengran SUN Maomao ZENG Xianfeng JING Hui. Dynamic Tracking Method of Indoor Trolley Facing Ultra-wideband. *Journal of Chongqing Jiaotong University (Natural Science)* **34**, 163-169 (2020).