

# WiFi Indoor Positioning and Tracking Algorithm Based on Compressive Sensing and Sage-Husa Adaptive Kalman Filter

Yingjie Sun<sup>1</sup>, Yi Zhong<sup>2\*</sup>, Congwei Hu<sup>3</sup>, Ao Xiong<sup>1</sup>, Hu Zhao<sup>1</sup>

<sup>1</sup>College of Railway Transportation, Hunan University of Technology, Zhuzhou, China

<sup>2</sup>Jiangsu Shengtong Electric New Energy Technology Co., Ltd., Changsha, China

<sup>3</sup>Huazi Technology Co., Ltd., Changsha, China

Email: 17863276205@163.com, \*28732654@qq.com

**How to cite this paper:** Sun, Y.J., Zhong, Y., Hu, C.W., Xiong, A. and Zhao, H. (2024) WiFi Indoor Positioning and Tracking Algorithm Based on Compressive Sensing and Sage-Husa Adaptive Kalman Filter. *Open Journal of Applied Sciences*, 14, 379-390. <https://doi.org/10.4236/ojapps.2024.142026>

**Received:** January 18, 2024

**Accepted:** February 24, 2024

**Published:** February 27, 2024

Copyright © 2024 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## Abstract

Aiming at the problem that the positioning accuracy of WiFi indoor positioning technology based on location fingerprint has not reached the requirements of practical application, a WiFi indoor positioning and tracking algorithm combining adaptive affine propagation (AAPC), compressed sensing (CS) and Kalman filter is proposed. In the off-line phase, AAPC algorithm is used to generate clustering fingerprints with optimal clustering effect performance; In the online phase, CS and nearest neighbor algorithm are used for position estimation; Finally, the Kalman filter and physical constraints are combined to perform positioning and tracking. By collecting a large number of real experimental data, it is proved that the developed algorithm has higher positioning accuracy and more accurate trajectory tracking effect.

## Keywords

WiFi Indoor Positioning, Cluster, Signal Recovery, Trajectory Tracking

## 1. Introduction

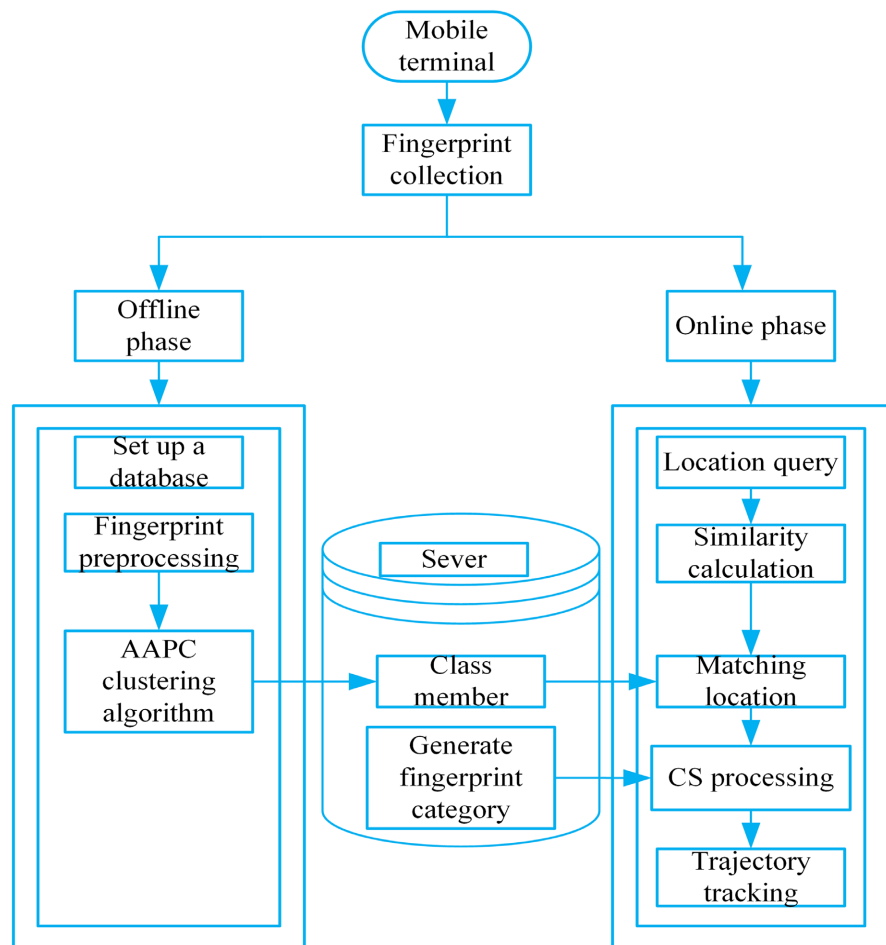
With the rapid development of Internet technology and the large-scale popularization and application of WiFi technology, people's demand for the accuracy, efficiency and ease of use of indoor positioning technology is increasing, so there are more and more indoor positioning methods [1] [2]. Among them, WiFi indoor positioning technology based on location fingerprint has been widely concerned and applied due to its low cost and universality [3].

The WiFi indoor positioning system based on location fingerprint is divided into two stages: offline and online. The main task of offline phase is data acquisition and preprocessing. In order to shorten the positioning time, clustering processing is usually required after the database is constructed. Common clustering algorithms include K-means [4], Affinity Propagation Clustering (APC) [5]. The main task of the online phase is to process the real-time received signal data through the positioning algorithm to estimate the target position. At present, location fingerprint matching algorithms are mainly divided into deterministic methods [6] [7], probabilistic methods [8] [9], machine learning methods [10] [11] and Compressed Sensing (CS) [12] [13]. Compared with other methods, the deterministic method is more lightweight and consumes less time while ensuring the positioning accuracy. The deterministic method is based on the K-nearest Neighbor algorithm (KNN) [6]. In the online positioning phase, the algorithm matches the currently received signal strength value with the k points with the highest similarity in the offline database. The final positioning result is determined by the center of these points. On the basis of this method, a Weighted K-nearest Neighbor algorithm (WKNN) [7] is proposed by giving the weight to k points, which further improves the positioning accuracy. In order to solve the problem of fixed k value, the Self-adaptive Weight K-nearest Neighbor (SAWKNN) [14] method was proposed, which can adaptively select k value according to the specific situation. According to the user's motion state Restricted Weight K-nearest Neighbors (RWKNN) [15] are proposed. Although deterministic methods have been widely studied and discussed in previous studies, they still face the following problems: 1) How to reduce positioning time consumption; 2) How to ensure both positioning speed and accuracy; 3) How to track continuous motion states in indoor positioning. The Adaptive Affine Propagation Clustering (AAPC) algorithm was introduced to reduce time consumption. To improve positioning accuracy, a method combining CS with online positioning algorithms was used. After positioning, the Sage-Husa adaptive Kalman filter algorithm [16] was used to track user trajectories. Numerous experimental results have shown that this method significantly improves the localization performance of indoor positioning algorithms.

## 2. System Model

### 2.1. Location System Framework Based on Location Fingerprint

As shown in **Figure 1**, in the offline process, the terminal device will collect fingerprint data at the preset sampling point and preprocess it, and then use AAPC algorithm to divide the obtained fingerprint into different fingerprint members and corresponding fingerprint database. The online phase will also be divided into two processes—"positioning" and "tracking". The positioning stage is divided into two stages: coarse positioning and fine positioning. In the coarse positioning stage, when the mobile terminal sends a positioning query request, the positioning engine will determine which feature set its data belongs to by calculating



**Figure 1.** System block diagram.

the similarity between the location data corresponding to the request and the related signals in the reference database. In the process of fine positioning, “CS + positioning algorithm” is used to obtain the position estimation. Finally, in the tracking phase, a method combining Kalman filter and physical constraints is used to track the trajectory.

## 2.2. Positioning Related Mathematical Expression

In order to ensure the accuracy and reliability of the experimental data, when the mobile terminal reaches the reference point, it needs to collect several data of an AP in four directions, East, West, North and South (40 in the article), and process the data obtained in each direction to obtain the average value of the signal strength of an AP in each direction corresponding to the mobile terminal at this point, using  $r_{ss}_{i,j}^d$ . Where  $d$  represents the direction of the mobile terminal,  $i$  represents the AP number corresponding to the data, and  $j$  represents the location of the reference point where the mobile terminal is located. Then the mathematical expression of the average signal strength of all APS corresponding to this point is shown in formula (1), and  $l$  represents the number of APS that can be detected by the mobile terminal at this reference point.

$$rss_j^d = [rss_{1,j}^d, rss_{2,j}^d, \dots, rss_{i,j}^d, \dots, rss_{L,j}^d] \tag{1}$$

So the data of multiple APS detected at the  $r$ -th test point can be expressed as:

$$rss_r = [rss_{1,r}, rss_{2,r}, \dots, rss_{i,r}, \dots, rss_{L,r}] \tag{2}$$

Then in the coarse positioning process, the similarity between the calculated signal strength and the class representation can be expressed as:

$$sim_{r,c} = |rss_r - ch_c|^1 \tag{3}$$

where the number of clusters is  $C$ ,  $Ch_c$  represents the class representation.

In the process of fine positioning, we calculate  $rss_r$  and the final positioning result obtained by matching the similarity between each member in the class.

### 3. Proposed Algorithm

#### 3.1. AAPC Clustering Algorithm

Using compressive sensing algorithm to process signals first requires clustering processing of the signals [17]. Here, we use AAPC clustering algorithm [18], after the location fingerprint is collected, AAPC clustering algorithm is used to generate different fingerprint categories.

$$\left. \begin{aligned} p_e &= p_n, C_n > C_t \\ p_s &= p_n, C_n < C_t \\ p_{n+1} &= 0.5 \cdot (p_e + p_s) \end{aligned} \right\} \tag{4}$$

Formula (4) shows that AAPC uses dichotomy to dynamically change the bias parameter  $p$  to get the clustering result, and the clustering number is  $[2, \sqrt{N}]$ .  $p_e$  and  $p_s$  represents the starting and ending values of the dichotomy respectively.  $N$  is the number of iterations,  $C_t$  is the number of clusters generated. When  $C_t = C_b$ , the clustering result  $Ch_c$  is obtained,  $N$  is the total number of reference points.

#### 3.2. Compressed Sensing Location

Because the location of the mobile terminal at a certain time in the positioning area is unique with the corresponding best reference point, the current user's location can be expressed as a 1 sparse vector, so the problem of indoor positioning can be transformed into how to recover the sparse vector. Sparse vector  $a$  is shown in formula (5):

$$a = [0, \dots, 0, 1, 0, \dots, 0]^T \tag{5}$$

According to the principle of compressed sensing  $rss_r$  can be expressed as:

$$rss_r = c_{cn} \cdot a + \varepsilon \tag{6}$$

In formula (6),  $c_{cn}$  represents the best matching class generated by AAPC clustering algorithm using formula (3), and  $\varepsilon$  represents the environmental noise in practical applications. Multiply both sides of the equation by the perception matrix  $P$ , where  $P$  is the AP selection matrix:

$$Y = P \cdot r_{ss_r} = P \cdot c_{cn} \cdot a + \varepsilon \quad (7)$$

Due to the certain spatial correlation between  $P$  and  $c_{cn}$  represents the best matching class generated by AAPC clustering algorithm using for the non correlation condition of the sensing matrix is not satisfied. Therefore, the perception matrix  $P$  and  $c_{cn}$  are orthogonalized. Let  $Z = MY$ , and matrix  $A$  be an orthogonalized matrix. Therefore, the problem is solved by minimizing the  $l_1$  norm in formula (8).

$$\hat{a} = \arg \min \|a\|_1, \text{ s.t. } Z = A \cdot a + M \cdot \varepsilon \quad (8)$$

$\| \cdot \|$  is the norm of vector  $l_1$ , formula (9) represents the set of values in a that are greater than the threshold value, denoted by  $D$ .

$$D = \{n | a(n) > \lambda\} \quad (9)$$

The final position estimate is:

$$LOC(x, y) = \frac{\sum_{n \in D} a(n)(x_n, y_n)}{\sum_{n \in D} a(n)} \quad (10)$$

### 3.3. WKNN Positioning

The WKNN positioning method ensures the positioning accuracy of the algorithm to a certain extent when the external conditions of the experiment meet the requirements, and due to its simplicity, it has become a relatively classic and commonly used algorithm in indoor positioning. The principle is as follows: first, select the  $K$  best matched candidate reference point positions, and then use these  $K$  positions to weighted sum up to obtain the final position estimation as shown in formula (11):

$$LOC(x, y) = \frac{\sum_{l=1}^K LOC(x_l, y_l)}{\sum_{l=1}^K \omega_l} \quad (11)$$

### 3.4. Principle of Joint Positioning Method

Let  $a$  solve for non-zero positions as  $\hat{a}$ , In an ideal environment, when there is little or no environmental noise,  $\hat{a}$  can be obtained. However, in practical application environments, the impact of environmental noise is inevitable, resulting in a value of  $\hat{a}$ , not being 1, and the degree of deviation is positively correlated with the size of environmental noise. Based on this conclusion, the accuracy of the localization results obtained by the CS algorithm can be determined. Take threshold  $\delta = 0.1$ , determine the size relationship between  $\hat{a}$  and threshold. When  $\hat{a}$  is less than the threshold, it can be considered that the deviation from 1 is small, that is, the CS algorithm is less affected by environmental noise; on the contrary, when  $\hat{a}$  is greater than the threshold, the environmental impact is greater. Therefore, the joint localization method proposed in the article adopts the CS algorithm results when the environmental noise impact is small,

because the signal reconstruction effect of CS is better at this time; When subjected to significant environmental noise, the results of online positioning algorithms are more accurate. So the final position estimation is shown in formula (12):

$$LOC(x, y) = \begin{cases} \frac{\sum_{l=1}^K LOC(x_l, y_l)}{\sum_{l=1}^K \omega_l} & |\dot{a} - 1| > \delta \\ \frac{\sum_{n \in D} \mathbf{a}(n)(x_n, y_n)}{\sum_{n \in D} \mathbf{a}(n)} & |\dot{a} - 1| \leq \delta \end{cases} \quad (12)$$

### 3.5. Sage-Husa Adaptive Kalman Filter

The traditional Kalman filter [19] includes two parts: the prediction process and the correction process. Based on the minimum mean square error, the optimal state of the current time state of the system is estimated through the observation data and the previous time state of the system. But in the process of indoor positioning, the noise in the environment is uncontrollable, and the noise matrix  $\mathbf{q}_k$  and  $\mathbf{r}_k$  is usually not zero, which will affect the accuracy of traditional Kalman filter prediction results. Aiming at the disadvantage of traditional Kalman filter, this paper uses sage Husa adaptive Kalman filter algorithm to realize the user's trajectory tracking. Sage Husa adaptive Kalman filter is a filtering method with the function of restraining filter divergence. It adds the idea of dynamic statistical estimation in the process of Kalman filter prediction and correction. On the one hand, it uses the observation value to constantly correct the prediction value, and on the other hand, it estimates and corrects the unknown system model parameters and noise statistical parameters. The formula is shown in (13)-(22).  $\mathbf{P}$  is the variance of state estimation error and  $\mathbf{K}$  is the gain matrix.

$$\hat{\mathbf{x}}_k^- = \mathbf{F}\hat{\mathbf{x}}_{k-1}^- + \mathbf{q}_{k+1} \quad (13)$$

$$\mathbf{P}_k^- = \mathbf{F}\mathbf{P}_{k-1}^- \mathbf{F}^T + \mathbf{Q}_{k-1} \quad (14)$$

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}^T (\mathbf{H}\mathbf{P}_k^- \mathbf{H}^T + \mathbf{R}_k)^{-1} \quad (15)$$

$$\mathbf{e}_k = \mathbf{Z}_k - \mathbf{H}\hat{\mathbf{x}}_k^- - \mathbf{r}_{k-1} \quad (16)$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k \mathbf{e}_k \quad (17)$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}) \mathbf{P}_k^- \quad (18)$$

Add a forgetting factor to adaptively adjust the noise matrix:

$$\mathbf{q}_k = (1 - d_k) \mathbf{q}_{k-1} + d_k (\hat{\mathbf{x}}_k - \mathbf{F}\hat{\mathbf{x}}_{k-1}^-) \quad (19)$$

$$\mathbf{Q}_k = (1 - d_k) \mathbf{Q}_{k-1} + d_k (\mathbf{K}_k \mathbf{e}_k \mathbf{e}_k^T \mathbf{K}_k^T + \mathbf{P}_k - \mathbf{F}\mathbf{P}_{k-1}^- \mathbf{F}^T) \quad (20)$$

$$\mathbf{r}_k = (1 - d_k) \mathbf{r}_{k-1} + d_k (\mathbf{Z}_k - \mathbf{H}\hat{\mathbf{x}}_k^-) \quad (21)$$

$$\mathbf{R}_k = (1 - d_k) \mathbf{R}_{k-1} + d_k (\mathbf{e}_k \mathbf{e}_k^T - \mathbf{H}\mathbf{P}_k^- \mathbf{H}^T) \quad (22)$$

In the formula  $d_k = (1-b)(1-b^{k+1})$ ,  $b$  is the forgetting factor, with a value range of (0, 1).

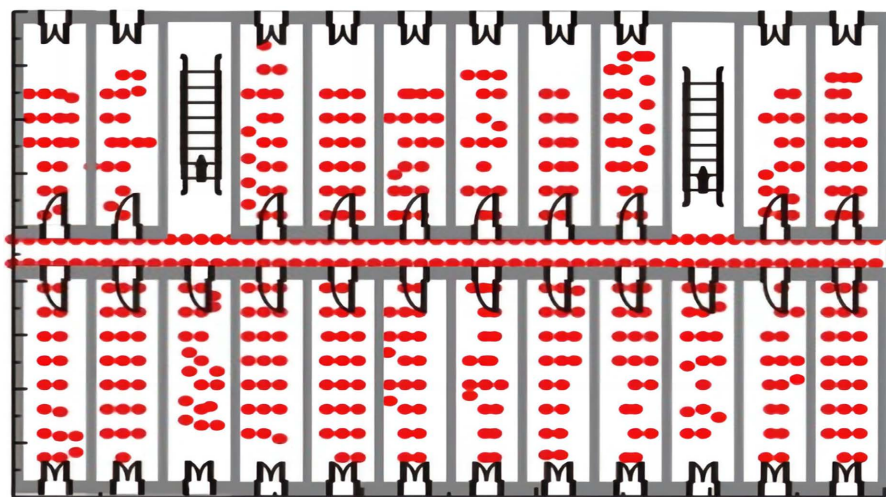
## 4. Experimental Results and Analysis

### 4.1. Experimental Environment

The experimental environment used in this paper is shown in **Figure 2**. There are 538 reference points (RPS) marked by red dots. The experimental environment covers an area of 50 \* 18 square meters, and the number of available APS is 339. During the experiment, when an AP signal cannot be detected, the default value is -100 dbm. The parameters of the experimental equipment are shown in **Table 1**.

### 4.2. Experimental Results of CS Algorithm

In this paper, the box chart is used to display the error distribution of positioning results. Each box chart contains six values: the upper edge (1.5 times), the upper quartile (75%), the mean value (represented by a small square in the figure), the median value, the lower quartile (25%), and the lower edge (1.5 times). In the box graph, we can clearly see the average and edge distribution of the positioning error of the algorithm. As shown in **Figure 3**, this paper compares the positioning accuracy of WKNN, SAWKNN and RWKNN algorithms before and after CS processing, and it can be seen that the positioning accuracy of the positioning algorithm processed by CS algorithm has been improved to a certain extent. The upper quartile of the three algorithms has decreased from 1.45 M,



**Figure 2.** Location area plan.

**Table 1.** Equipment parameters.

Equipment model	WLAN	CPU	Memory
Lenovo Legion R7000 P2021	Realtek RTL8852AE WiFi 6	AMD Ryzen5 5600H	16 GB

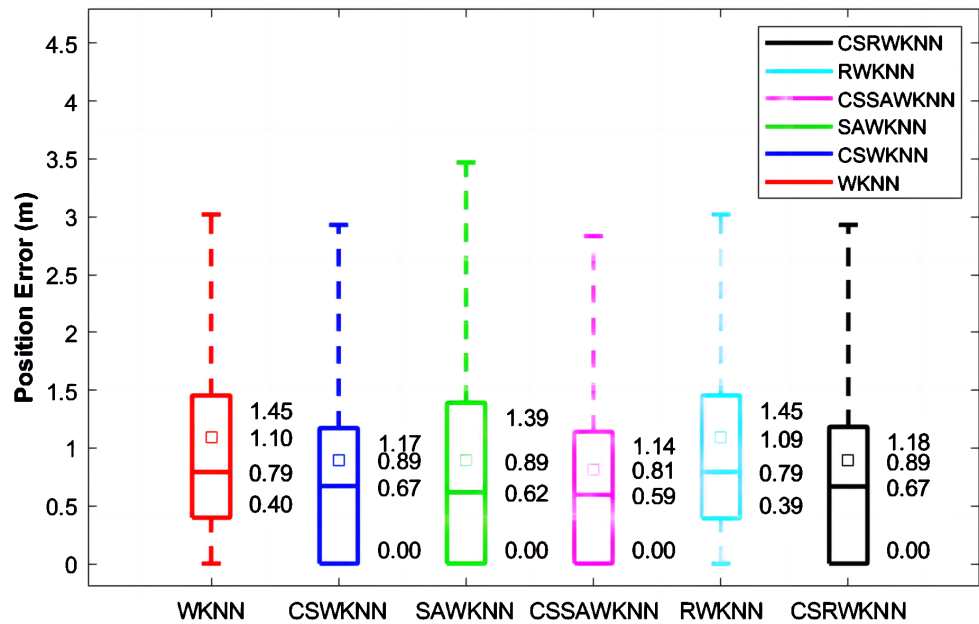


Figure 3. Comparison of accuracy of various algorithms after CS processing.

1.39 M and 1.45 M to 1.17 M, 1.14 M and 1.18 M respectively, while the lower quartile is 0, which means that the overall larger error points of the algorithm are decreasing and the smaller error points are increasing. The positioning accuracy is improved by 23%, 10% and 22% respectively.

### 4.3. Sage-Husa Adaptive Kalman Filtering.

Combined with the static positioning results of the user, the current position information of the user is used as the prediction value of the adaptive Kalman filter, and the observation value is the result of the algorithm proposed in this paper, that is,  $LOC'(x_r, y_r)$  is used as the input of the filtering system, in which the forgetting factor  $B$  is 0.5, and  $\dot{x}_k, \dot{y}_k, x_k, y_k$  is used to represent the displacement and velocity on the X and Y axes respectively. In the ideal state, the motion state is uniform linear motion, so there is the state equation (23), which shows the state transition matrix  $F$ . The observation vector is shown in formula (24), where  $Z_{k1}$  and  $Z_{k2}$  represent the observation values in the x-axis and y-axis respectively. Equation (24) gives the expression of the observation equation, which shows that the observation matrix  $H$ , the positioning interval  $T$  in the experiment is 1s,  $q_k$  and  $r_k$  initial value of is 0. Formulas (25) and (26) are the initial values of  $Q_k, R_k$  respectively.

$$\begin{bmatrix} \dot{x}_k \\ \dot{y}_k \\ x_k \\ y_k \end{bmatrix} = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x}_{k-1} \\ \dot{y}_{k-1} \\ x_{k-1} \\ y_{k-1} \end{bmatrix} + W_k, W_k \sim (0, Q) \tag{23}$$

$$\begin{bmatrix} Z_{k1} \\ Z_{k2} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{x}_k \\ \dot{y}_k \end{bmatrix} + V_k, V_k \sim (0, R) \tag{24}$$



$$Q_k = 50 \cdot \begin{bmatrix} \frac{T^3}{3} & 0 & \frac{T^2}{2} & 0 \\ 0 & \frac{T^3}{3} & 0 & \frac{T^2}{2} \\ \frac{T^2}{2} & 0 & T & 0 \\ 0 & \frac{T^2}{2} & 0 & T \end{bmatrix} \tag{25}$$

$$R_k = \begin{bmatrix} 20 & 0 \\ 0 & 20 \end{bmatrix} \tag{26}$$

Figures 4-6 show the trajectory comparison of three algorithms, WKNN, SAWKNN, and RWKNN, processed by Sage-Husa adaptive Kalman filtering. The red line represents the actual trajectory, and the black line represents the trajectory tracking trajectory. The error in trajectory optimization can be divided into two categories: normal error and abnormal error. For normal errors, adaptive Kalman filtering can be directly applied for optimization, as it follows the laws of physical motion. For abnormal errors, physical constraints need to be added first to ensure that the positioning point does not collide with obstacles, while limiting the maximum speed to avoid excessive displacement. After

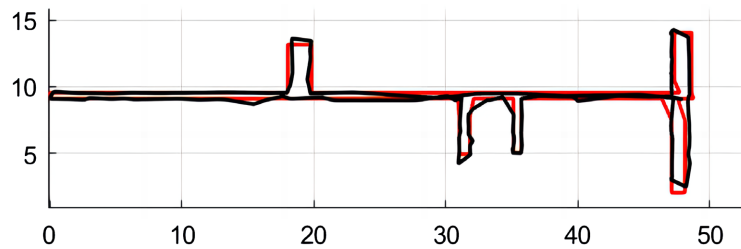


Figure 4. WKNN algorithm Sage-Husa adaptive Kalman filter trajectory tracking.

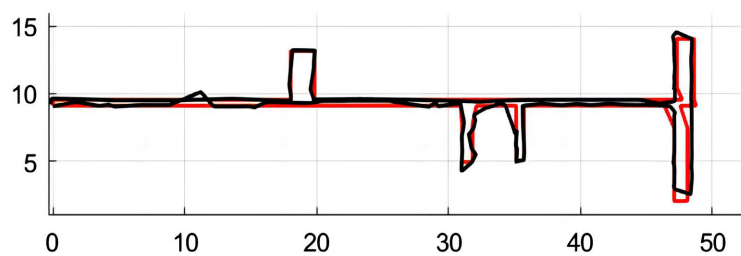


Figure 5. SAWKNN algorithm Sage-Husa adaptive Kalman filter trajectory tracking.

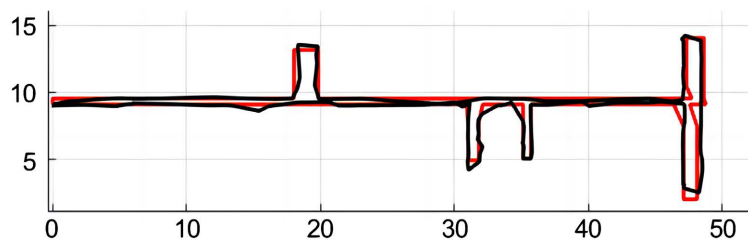
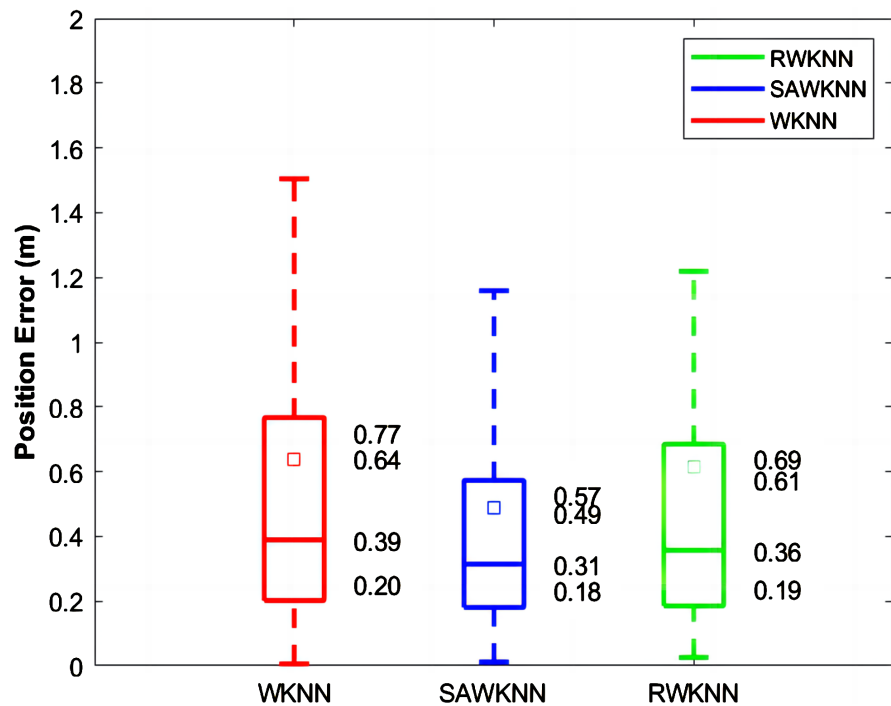


Figure 6. RWKNN algorithm Sage-Husa adaptive Kalman filter trajectory tracking.



**Figure 7.** Comparison of accuracy of various algorithms after CS processing

experiments, the ideal trajectory tracking effect was ultimately achieved.

**Figure 7** shows the distribution of positioning errors of WKNN, SAWKNN, and RWKNN algorithms after Sage Husa adaptive Kalman filtering. The average positioning errors are 0.65 m, 0.49 m, and 0.61 m, respectively, further improving the positioning accuracy.

## 5. Conclusion

The indoor positioning algorithm in the article has been processed by the CS algorithm, and it has been found that the positioning accuracy has been significantly improved. The average positioning accuracy of the WKNN, SAWKNN, and RWKNN algorithms has increased by 23%, 10%, and 22%, respectively. Kalman filtering is a classic algorithm for trajectory tracking, and incorporating Kalman filtering in indoor positioning applications is of great significance. In order to meet the noise problem in real environments, a method combining indoor positioning algorithms with Sage Husa adaptive Kalman filtering is proposed. The experiment confirms that this method can effectively solve the problem of dynamic trajectory positioning and significantly improve the positioning accuracy. Future work will further study the compressed sensing algorithm to improve the sparse signal recovery method to improve the anti-interference ability and positioning accuracy under different environmental noise, especially when the signal conditions change greatly or the environment is complex. Combined with other sensor data, a multimodal fusion localization algorithm is designed to further improve the overall positioning effect and reliability through complementary advantages.

## Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

## References

- [1] Geok, T.K., Aung, K.Z., Aung, M.S., *et al.* (2021) Review of Indoor Positioning: Radio Wave Technology. *Applied Sciences*, **11**, Article 279. <https://doi.org/10.3390/app11010279>
- [2] Zafari, F., Gkelias, A., Leung, K.K. (2019) A Survey of Indoor Localization Systems and Technologies. *IEEE Communications Surveys & Tutorials*, **21**, 2568-2599. <https://doi.org/10.1109/COMST.2019.2911558>
- [3] Shang, S. and Wang, L.X. (2022) Overview of WiFi Fingerprinting-Based Indoor Positioning. *IET Communications*, **16**, 725-733. <https://doi.org/10.1049/cmu2.12386>
- [4] Pinto, B., Barreto, R. and Souto, E. (2021) Robust RSSI-Based Indoor Positioning System Using K-Means Clustering and Bayesian Estimation. *IEEE Sensors Journal*, **21**, 24462-24470. <https://doi.org/10.1109/JSEN.2021.3113837>
- [5] Subedi, S., Gang, H.S., Ko, N.Y., Hwang, S.S. and Pyun, J.Y. (2019) Improving Indoor Fingerprinting Positioning with Affinity Propagation Clustering and Weighted Centroid Fingerprint. *IEEE Access*, **7**, 31738-31750. <https://doi.org/10.1109/ACCESS.2019.2902564>
- [6] Zhang, H., Wang, Z., Xia, W., Ni, Y. and Zhao, H. (2022) Weighted Adaptive KNN Algorithm with Historical Information Fusion for Fingerprint Positioning. *IEEE Wireless Communications Letters*, **11**, 1002-1006. <https://doi.org/10.1109/LWC.2022.3152610>
- [7] Tran, H.Q. and Ha, C. (2020) High Precision Weighted Optimum K-Nearest Neighbors Algorithm for Indoor Visible Light Positioning Applications. *IEEE Access*, **8**, 114597-114607. <https://doi.org/10.1109/ACCESS.2020.3003977>
- [8] Vanhie-Van Gerwen, J., Geebelen, K., Wan, J., Joseph, W., Hoebeke, J. and De Poorter, E. (2022) Indoor Drone Positioning: Accuracy and Cost Trade-Off for Sensor Fusion. *IEEE Transactions on Vehicular Technology*, **71**, 961-974. <https://doi.org/10.1109/TVT.2021.3129917>
- [9] Yadav, R.K., Bhattarai, B., Gang, H.-S. and Pyun, J.-Y. (2019) Trusted K Nearest Bayesian Estimation for Indoor Positioning System. *IEEE Access*, **7**, 51484-51498. <https://doi.org/10.1109/ACCESS.2019.2910314>
- [10] Chen, Z., Zou, H., Yang, J., Jiang, H. and Xie, L. (2019) WiFi Fingerprinting Indoor Localization Using Local Feature-Based Deep LSTM. *IEEE Systems Journal*, **14**, 3001-3010. <https://doi.org/10.1109/JSYST.2019.2918678>
- [11] Own, C.-M., Hou, J. and Tao, W. (2019) Signal Fuse Learning Method with Dual Bands WiFi Signal Measurements in Indoor Positioning. *IEEE Access*, **7**, 131805-131817. <https://doi.org/10.1109/ACCESS.2019.2940054>
- [12] Gligorić, K., Ajmani, M., Vukobratović, D. and Sinanović, S. (2018) Visible Light Communications-Based Indoor Positioning via Compressed Sensing. *IEEE Communications Letters*, **22**, 1410-1413. <https://doi.org/10.1109/LCOMM.2018.2833550>
- [13] Gong, X., Liu, J., Yang, S., Huang, G. and Bai, Y. (2022) A Usability-Enhanced Smartphone Indoor Positioning Solution Using Compressive Sensing. *IEEE Sensors Journal*, **22**, 2823-2834. <https://doi.org/10.1109/JSEN.2021.3137327>

- [14] Hu, J., Liu, D., Yan, Z. and Liu, H. (2018) Experimental Analysis on Weight  $\{K\}$   $\delta$ -Nearest Neighbor Indoor Fingerprint Positioning. *IEEE Internet of Things Journal*, **6**, 891-897. <https://doi.org/10.1109/IJOT.2018.2864607>
- [15] Chen, G., Guo, X., Liu, K., Li, X. and Yang, J. (2022) RWKNN: A Modified WKNN Algorithm Specific for the Indoor Localization Problem. *IEEE Sensors Journal*, **22**, 7258-7266. <https://doi.org/10.1109/JSEN.2022.3155902>
- [16] Liu, X., Zhou, B. and Huang, P. (2016) Kalman Filter-Based Data Fusion of Wi-Fi RTT and PDR for Indoor Localization. *IEEE Sensors Journal*, **21**, 8479-8490. <https://doi.org/10.1109/JSEN.2021.3050456>
- [17] Feng, C., Au, W.S.A., Valaee, S. and Tan, Z. (2011) Received-Signal-Strength-Based Indoor Positioning Using Compressive Sensing. *IEEE Transactions on Mobile Computing*, **11**, 1983-1993. <https://doi.org/10.1109/TMC.2011.216>
- [18] Hu, J., Liu, H. and Yan, Z. (2019) Adaptive Affinity Propagation Algorithm Based on New Strategy of Dynamic Damping Factor and Preference. *IEEE Transactions on Electrical and Electronic Engineering*, **14**, 97-104. <https://doi.org/10.1002/tee.22792>
- [19] Hu, J. and Hu, C. (2023) A WiFi Indoor Location Tracking Algorithm Based on Improved Weighted K Nearest Neighbors and Kalman Filter. *IEEE Access*, **11**, 32907-32918. <https://doi.org/10.1109/ACCESS.2023.3263583>