An Indoor Geolocation Algorithm based on CSI and Affine Propagation Clustering

Lei Liu 1,2,3, a, Shuwei Zhang1, 2, 3, *
1Institute of Oceanographic Instrumentation, Qilu University of Technology (Shandong Academy of Sciences), Qingdao 266061, China
2Shandong Provincial Key Laboratory of Ocean Environmental Monitoring Technology, Qingdao 266061, China
3National Engineering and Technological Research Center of Marine Monitoring Equipment, Qingdao 266061, China
*Corresponding author e-mail: zswcn2001@qul.edu.cn, *liulei_private@163.com

Abstract. An indoor geolocation algorithm based on WLAN CSI and affine propagation clustering is proposed in this paper. In this algorithm, the wireless network adaptor Intel 5300 and 802.11n CSI Tools is used to collect the CSI data, and the PCA algorithm is used to reduce the dimension of the CSI data from 30 to 3, based on which the CSI fingerprint database is established. And the affine propagation clustering algorithm is used to group the reference nodes into clusters and then the WKNN is adopted to realize precise indoor geolocation. Experiment results show that the algorithm in this paper has a better performance in the location accuracy, 74% of the testing results has the accuracy of 1m, and 95% of the testing results has the accuracy of 2m.

1. Introduction

Location based service is a hot topic in recent years. GPS is an well-established outdoor location technology, which has been applied into many areas of our daily life. But GPS signals can’t penetrate through walls of the buildings, so GPS based location technologies can’t be applied into the indoor environments.

For indoor geolocation, RSSI based scheme is often adopted, such as the location method in the references [1~5]. But RSSI signal is not stable in the indoor dynamic environments, eg. people coming and going, which may change the level of RSSI abruptly. So RSSI based indoor location systems can’t achieve high location accuracy.

An indoor geolocation algorithm based on WLAN CSI is proposed in this paper. In this algorithm, the wireless network adaptor Intel 5300 is adopted, and Linux and 802.11n CSI Tools is used to collect the CSI data, then the PCA algorithm is used to reduce the dimension of the CSI data, based on which the CSI fingerprint database is established. And finally, the affine propagation and WKNN is used to realize precise indoor geolocation. Experiment results show that the algorithm in this paper has better performance than the RSSI based algorithms.
2. CSI data

2.1. OFDM and CSI

The Orthogonal Frequency Division Multiplexing (OFDM) technology is used in WLAN for the data modulation [6, 7]. OFDM is a digital multi-carrier modulation scheme. In OFDM, the original data is split into groups with n bits (1 bit -BPSK, 2 bits -QPSK, 4 bits -16QAM, or 6 bits -64QAM), and these groups of data will be modulated to a large number of closely spaced orthogonal subcarriers that are transmitted in parallel. The OFDM modulation is realized by IFFT and the demodulation is realized by IDFT, from which we can conclude the equivalent frequency model of OFDM as the following:

\[
R_n = H_n S_n + N_n \tag{1}
\]

Here, \( R_n \) is the received signal, \( S_n \) is the transmitted signal, \( H_n \) is the frequency response on the n sub-carriers, and \( N_n \) is the AWGN on the sub-carriers. So we have

\[
H_n = \frac{R_n}{S_n} = H_n + \frac{N_n}{S_n} \tag{2}
\]

From above, it is clear that we can use Wi-Fi to obtain the frequency channel response, and here the amplitude frequency channel response is the so-called CSI data. CSI data is superior to the RSSI data, so nowadays researcher prefer the CSI to RSSI for the indoor location purpose, and a lot of location algorithms were developed [8, 9].

In real practice, we can obtain CSI data from the 30 sub-carriers by CSI tools, and here, the value of \( n \) in equation (2) is 30. And three antennas are used to receive the CSI data, and one antenna to transmit data, that is, the CSI data we received is a matrix with the dimension of 3×1×30.

2.2. CSI data acquisition

In order to collect the CSI data, wireless network adaptor Atheros 9390 or Intel 5300 is needed. Ubuntu operating system and AP devices based on the protocol 802.11n are also necessary. Using the laptop with the OS Ubuntu, we ping the IP of the router, so the laptop with wireless network adapter Intel 5300 can receive the data transmitted from the router, and the CSI data of the physical layer. These data are the original data of the wireless channel and they are in the DAT format. And after the data is processed using the C compile, they can be displayed in matlab as Fig.1.
(1) $N_{rx}$ is the number of the receiving antennas, and $N_{tx}$ is the number of the transmitting antennas. Here, $N_{rx}$ is 1 and $N_{tx}$ is three in this paper.

(2) $rssi_{a}, rssi_{b}, rssi_{c}$ is the RSSI of the three receiving antennas.

(3) $csi$ is a matrix with the dimension $N_{rx} \times N_{tx} \times 30$, the number of sub-carriers is 30.

Fig. 2 is the SNR of the CSI of the three receiving antennas.

![Fig. 2 A set of samples of CSI](image)

3. CSI data preprocessing and the fingerprint database

3.1. CSI data preprocessing

As we all know, the indoor environment is not stable, so the CSI data we collected in the same position will be inconsistent sometimes as shown in the Fig. 3. For example, if there is someone walking around when we are collecting the CSI data of a specific position, we can’t get the stable CSI data.

![Fig. 3 CSI data with outliers](image)
So we need some preprocessing method to exclude the unreasonable samples. In this paper, we use the $3\sigma$ rule in CSI data preprocessing, in which the standard deviation $\sigma$ and expectation $\mu$ of the CSI samples collected in a specific reference node is calculated. Then the sample errors beyond the range $[\mu-3\sigma, \mu+3\sigma]$ will be taken as gross error, so such kind of errors will be replaced by the sample expectation $\mu$ as shown in Fig.4. Based on these steps, the stable CSI fingerprint is established.

And then the CSI data needs to be normalized to the range [0-1] by the equation (3):

$$H'_i = \frac{H_i - \text{Min}(H)}{\text{Max}(H) - \text{Min}(H)}$$ (3)

Here, $H_i$ is the original data value, $H'_i$ is the normalized data value, and $\text{Max}(H)$ and $\text{Min}(H)$ is the maximum and minimum value of the CSI data.

3.2. PCA of CSI

After the preprocessing of the CSI data, and here the dimension of every CSI data is 30, and it’s complicated to handle vectors with the dimension 30, so we use the PCA method to extract the principle components of the CSI firstly. PCA is an orthogonal transform, it projected the original vectors into independent orthogonal vectors, and only few of the independent vectors contributes much which is called as the principle components [10]. So in this condition, we can only keep the main components, that is to keep the principle features, by which the vector dimension can be greatly reduced. This is the main idea of PCA.

The steps using PCA to reduce the vector dimension and to extract the principle components is as the following:

Step 1: prepare the dataset $F$ including all the CSI samples:

$$F = [F_1, F_2, \cdots F_N]^T$$ (4)

Here, $F_i = \{f_{1i}, f_{2i}, \cdots f_{30i}\}, \ i = 1,2,\cdots N$ is a CSI sample with the amplitude frequency response of the 30 sub-carriers, $N$ is the number of samples.

Step 2: calculate the mean value $\mu_i$ of all the $F_i$s, subtract $\mu_i$ from $F_i$, then we can get a new matrix $A$ as the following:

$$A = (F_1 - \mu_1, F_2 - \mu_2, \cdots F_N - \mu_N)^T,$$

and $\mu_i = \frac{1}{N} \sum_{i=1}^{N} F_i$ (5)

Step 3: calculate the covariance matrix of $A$:

$$B = \frac{1}{N} AA^T$$ (6)

Step 4: calculate the eigen value and vectors using SVD:

$$BV = \lambda V$$ (7)

Here, $\lambda$ is the eigen value of matrix $B$, vector $V$ is the eigen vectors of related to $\lambda$. 
Step 5: Rank the eigen value from large to small, and keep the largest $p$ eigen values, the largest $\lambda_i$s, and the corresponding eigen vectors $V_i$, which accounts for 99% of the total contribution\cite{48} of all the components.

Step 6: the original sample matrix is projected to the $p$ eigen vectors, then we will have the new sample matrix $H' = (H_1', H_2', \cdots, H_n')^T$, where $H_i' = \{r_1, r_2, \cdots, r_p\}$, $i = 1, 2, \cdots, N$. So the dimension of the data sample is deduced from 30 to $p$.

Fig. 4 is an example of the PCA result of one CSI sample matrix:

![PCA result of one CSI sample matrix](image)

From Fig.4, we can see that the contribution of the first three components is 95.6%, 2.7% and 0.9%, and the contribution of the other 27 components is almost zero. And we did a lot of experiments based on large amount of CSI sample data, we conclude from the testing results that the contribution of the first three components is no less than 99%, so in this paper, we only keep the first three components of the PCA matrix as the principle components. That is, we reduce the dimension of the CSI data from 30 to 3, and here PCA is an efficient method to extract features.

3.3. CSI Fingerprint database

After the CSI data collection of all the reference nodes in the testing area, in each node, several datasets will be collected, and then PCA will be carried out. The PCA results and the corresponding position in coordinates of every reference point will be saved as a fingerprint record in the fingerprint database.

And the CSI of the $i$th reference node is represented as the following:

$$H_i = \begin{bmatrix}
\tilde{h}_{1,1} & \tilde{h}_{1,2} & \cdots & \tilde{h}_{1,30} \\
\tilde{h}_{2,1} & \tilde{h}_{2,2} & \cdots & \tilde{h}_{2,30} \\
\tilde{h}_{3,1} & \tilde{h}_{3,2} & \cdots & \tilde{h}_{3,30}
\end{bmatrix}$$

(8)

Here, $\tilde{h}_{i,j}$ is the average amplitude of the $j$th carrier in the $i$th antenna, and $i$ is the number of the receiving antennas, and because the network adapter we use has 3 receiving antennas, so here, $i = 1, 2, 3$, $j = 1, 2, \cdots, 30$, and the original number of sub-carriers is 30, it is reduced to 3 after the PCA processing. So, the CSI data related to a specific reference node is a matrix with a dimension of 3*3.

Based on above, we set up the CSI fingerprint database as the following:

$$F = \begin{bmatrix}
csi_{i_{1,1}} & csi_{i_{1,2}} & \cdots & csi_{i_{1,n}} \\
csi_{i_{2,1}} & csi_{i_{2,2}} & \cdots & csi_{i_{2,n}} \\
\vdots & \vdots & \ddots & \vdots \\
csi_{i_{m,1}} & csi_{i_{m,2}} & \cdots & csi_{i_{m,n}}
\end{bmatrix}$$

(9)
In which, $\text{CSI}_{m,n}$ is the final CSI fingerprint data of the $N$th AP after the PCA processing.

4. Affine propagation clustering algorithm

After the fingerprint database is set up, we all know that the wireless channel related to the same area is similar, and so are the fingerprints, so if the fingerprints can be grouped into different clusters by clustering algorithm firstly, it will help to improve the efficiency of the fingerprint matching process. Therefore, the affine propagation clustering algorithm is adopted in this paper to realize the fingerprints clustering.

The affine propagation clustering algorithm avoids the setting of the clustering centers in the algorithm initialization which is the precondition of the classic K-mean algorithm [10]. In the affine propagation clustering algorithm, every reference node is considered as a potential cluster center, and the final optimal cluster centers are deduced by the communication between the reference nodes. And the input of the affine propagation clustering algorithm is the similarity matrix of the reference nodes, that is matrix $S$, in which $s(i, j)$ is the similarity of the node $i$ and node $j$, and it can also be considered as how much the $j$th node fits in the $i$th cluster center. Here each reference node has the same possibility to be the potential cluster center, and the bias parameter can be described as the following:

$$p = \lambda \cdot \text{median} \{s(i, j), i, j = 1, 2, \ldots, N, i \neq j\}$$ (10)

And here $\lambda$ is decided by experimental tests in the lab, it is used to decide the number of the clusters.

The communication between the different nodes includes the degree of the attraction and affiliation, and the degree of the attraction $r(i, j)$ is the information transmitted from the $i$th node to the $j$th node, and it is used to represent how much the $i$th node is attracted by the $j$th node, and the possibility of the $j$th node to be the $i$th cluster center.

$$r(i, j) = s(i, j) - \max_{j \neq j} \{a(i, j') + s(i, j')\}, i \neq j$$ (11)

Affiliation $a(i, j)$ is the information transmitted from the $j$th node to the $i$th node, which is used to represent the possibility of the $j$th node to be the $i$th cluster center from the view of the $j$th node:

$$a(i, j) = \min \left\{0, r(j, j) + \sum_{i \neq i} \max \left[0, r(i', j)\right]\right\}$$ (12)

Here, $a(j, j) = \sum_{i \neq i} \max \left\{0, r(i', j)\right\}$, is always positive representing the information of auto-affiliation.

We use iteration to realize the algorithm, the attraction and affiliation is updated in every iteration step, and if the summation of the attraction and the affiliation is more than the threshold, the reference node will be considered as the cluster center, if not, the iteration will be continued until it hits the algorithm convergence or the iteration steps is more the maximum.

The iteration steps is as the following:
Step 1. Set all the elements of the affiliation matrix and attraction matrix to 0, set 
\[ s(j,j) = p, \]
so the equation (11) can be simplified as:
\[ r(i,j) = s(i,j) - \max_{j' \neq j} \{ s(i,j') \} \]  
(13)

Step 2. Updates the matrix \( r \) and matrix \( a \), and the damping factor \( \gamma \) is used here to prevent the evolution from divergence, so the iteration can be described as:
\[ r_n(i,j) = (1 - \gamma) \times \left( s(i,j) - \max_{j' \neq j} \{ a(i,j') + s(i,j') \} \right) + \gamma \times r_{n-1}(i,j) \]  
(14)
\[ a_n(i,j) = (1 - \gamma) \times \left[ 0, r(j,j) + \sum_{i' \neq i,j} \max_{r(i',j')} \{ 0, r(i',j') \} \right] + \gamma \times a_{n-1}(i,j) \]  
(15)

In order to decide the value of the damping factor, we did a lot of tests, and we found that when \( \gamma = 0.9 \) we have the best performance, so \( \gamma \) is set to 0.9 in this paper.

Step 3. Set the vector \( c \), \( c(i) = a(i,i) + r(i,i) \), and if \( c(i) > 0 \) then the \( i \)th node is considered as the cluster center.

Step 4. Check the convergence, if the algorithm hits the convergence, then every reference node will be classified as one specific cluster center, and the affine propagation algorithm finishes. If not, then updates the matrix \( r \) and \( a \), and the iteration finishes when the iteration steps is more than the maximum.

After the algorithm is finished, all the cluster centers is stored in the set \( E \), and for reference node \( j \in E \), all the reference nodes in the cluster \( j \) are stored in data set \( C_j \). And the clustering results when the parameter \( p \) is set to different values is shown in the following figures.

Fig.5 The clustering effect when \( p = -0.06 \)
From the clustering results, we can see that the number of clusters is decided by the value of $p$, so a proper value of $p$ should be decided in order to achieve good clustering.

4.1. **Online location**

4.1.1. **Rough location.** In the rough locating step, we calculate the distance between the live sample and the cluster centers:

$$s(r,j) = -\|H_i - H_j\|^2, \forall j \in E$$ (16)

We choose the smallest $N s(r,j)$, and take them as the possible areas the testing object can be in. $H_i$ is the CSI fingerprint data of the node $i$, $H_j$ is the cluster center of cluster $j$.

4.1.2. **Precise location.** After we obtained the $n$ possible cluster centers, we use WKNN to get the final location result. And here, the weight value related to the different Aps should be different, because those
Aps closer to the testing object influence the accuracy of the location more. So we use equation (20) to calculate the weight values:

\[ w_i = \frac{1}{\sum_{j=1}^{k} \frac{1}{d_j}} \]

Here \( d_i \) is the distance between the testing object and the \( i \)th cluster center.

Then we can use the weights to estimate the final location:

\[ (x, y) = \sum_{i=1}^{k} w_i (x_i, y_i) \]  

5. Experiments

We collect data using the method described in

And we use a lab in robot center J11 of SDUST, with the length 16m, width 8m, and there are a lot of desks and chairs in it, the deployment is shown in Fig. 6

![Fig. 6 The experimental layout](image)

In Fig. 6, the 30 red points are the reference nodes, the 4 star shapes are the location of the AP used in this paper and they are fixed on tables with the height 1.2m. At each reference node, the CSI data file includes 600 stable data packages which is from the data collection with a period of 10 minutes. Based on the CSI data file and the PCA processing, the fingerprint of each reference node is obtained, so the fingerprint database is created.

Based on the fingerprint database, a lot of testing work was carried out. And we calculated the CDF of the algorithm proposed in this paper and the traditional algorithm without clustering, the CDF curve is shown in Fig.7. From Fig.7, it is shown that the algorithm has a better performance in the location
accuracy, 74% of the testing results has the accuracy of 1m, and 95% of the testing results has the accuracy of 2m.

6. Conclusion
An indoor geolocation algorithm based on WLAN CSI and affine propagation clustering is proposed in this paper. In this algorithm, we use the wireless network adaptor Intel 5300 and 802.11n CSI Tools to collect the CSI data, and the PCA algorithm to reduce the dimension of the CSI data from 30 to 3, based on which the CSI fingerprint database is established. And we use the affine propagation clustering algorithm to group the reference nodes into clusters and then the WKNN realize precise indoor geolocation. A lot of testing work is carried out in the lab, and the CDF curve of the algorithm in this paper and the traditional algorithm without clustering is given. The CDF curves show that the algorithm in this paper has a better performance in the location accuracy, 74% of the testing results has the accuracy of 1m, and 95% of the testing results has the accuracy of 2m.

References
[1] Demirbas M, Song Y. An RSSI-based Scheme for Sybil Attack Detection in Wireless Sensor Networks [C]// World of Wireless, Mobile and Multimedia Networks, 2006. WoWMoM 2006. International Symposium on a. IEEE, 2006:5 pp.-570.
[2] Bahl P, Padmanabhan V N. RADAR: an in-building RF-based user location and tracking system [C]// INFOCOM 2000. Nineteenth Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE. IEEE Xplore, 2000:775-784 vol.2.
[3] Lang V, Gu C. A Locating Method for WLAN based Location Service [C]// IEEE International Conference on E-Business Engineering. IEEE Computer Society, 2005:427-432.
[4] Pan B; Ma Y; Ren H; He Y; Wang Y; Lv X; Liu D; Ji L; Yu B; Wang Y; Chen YE; Pennathur S; Smith JD; Liu G; Zheng L. A Novel Approach to Indoor RSSI Localization by Automatic Calibration of the Wireless Propagation Model [C]// Vehicular Technology Conference, 2009. Vtc Spring 2009. IEEE. IEEE, 2009:1-5.
[5] Gholoobi A, Stavrou S. RSS Based Localization Using a New WKNN Approach [C]// International Conference on Computational Intelligence, Communication Systems and
Networks. IEEE, 2015:27-30.

[6] Paul T, Ogunfunmi T. Wireless LAN Comes of Age: Understanding the IEEE 802.11n Amendment [J]. Circuits & Systems Magazine IEEE, 2008, 8 (1): 28-54

[7] Nee R V, Prasad R. OFDM for Wireless Multimedia Communications [M]// OFDM for wireless multimedia communications. Artech House, 2000

[8] Cai Xiong Research on Indoor WiFi Location Technology based on CSI [D]. XIDIAN UNIVERSITY, 2015.12

[9] Wang Kai. Range-based Lightweight Fingerprint Indoor Localization Using CSI [D]. XIDIAN UNIVERSITY, 2014

[10] Song Kun, Li Lijuan, Zhao Yingkai. Affinity propagation clustering algorithm based on principle components analysis [J]. Computer Engineering and Applications, 47 (34): 212-214.