A Vehicle Location Algorithm Based on Community Discovery

Yang Yang¹, Fei Lin²*, Dingguo Yu³, Chen Yang⁴ and Qiang Lin¹

¹Institute of Intelligent Media Technology, Communication University of Zhejiang, Hangzhou 310018, China
²College of computer Science, Hangzhou Dianzi University, Hangzhou 310018, China
³Department of Media Engineering, Communication University of Zhejiang, Hangzhou 310018, China
⁴Accounting college, Hangzhou Dianzi University, Hangzhou 310018, China

*Corresponding author e-mail: linfei@hdu.edu.cn

Abstract. The group behavior of the vehicle can accurately describe the movement law of the vehicle in the vehicle network. A vehicle node with strong GPS signal in the Internet of Vehicles can use its communication capability and its neighbor positional relationship to calculate the position of a vehicle with a weak GPS signal located within a certain distance range and having the same or similar driving destination. The algorithm divides the vehicles with the same or similar destination destinations into the same community based on the destination of the vehicle, and then calculates the position of the vehicle that needs to be located, and corrects the calculation result. The experimental results show that the proposed algorithm reduces the positioning error by 15%-35% compared with similar methods. Under different traffic conditions, the stability of the algorithm is improved by more than 25%.

1. Introduction
With the popularization of vehicle self-organizing networks [1] and the increasing interconnection of vehicles [2], research on vehicle positioning methods has been increasing in recent years. Herrera-Quintero et al [3] proposed a sensor-based vehicle positioning technology. Tension et al [4] proposed a cooperative positioning technology based on frequency-shot recognition and proposed an indoor navigation algorithm. Chen Shijun et al [5] proposed a cellular positioning method to improve the base station layout optimization algorithm for search. So now most vehicle positioning still depends on the GPS system. [6] GPS system features high precision, fast speed, strong confidentiality and simple operation, [7] but it also has some problems: first, in the blind zone, it cannot get accurate position, such as underground garage or GPS signal on the viaduct will be weak. Even disappeared; [8] Secondly, some external factors, such as weather, will affect the positioning results; [9] Finally, not all vehicles are equipped with GPS system hardware units. Based on the above situation, this paper uses the vehicle network and vehicle group behavior to calculate the position of vehicles that cannot rely on GPS positioning.
2. Location algorithm based on community discovery

This paper first proposes a positioning algorithm based on community discovery, which can calculate the vehicle node in the vehicle network, but the vehicle node that cannot be located due to the loss of GPS signal, its approximate location in a certain period of time. The algorithm can be divided into two cases according to whether the node vector \( \text{vector}_i \) of the vehicle \( c_i \) contains relevant historical destination information within a certain time period \( T \):

2.1. No relevant historical destination information. \( \text{vector}_i = \emptyset \)

If there is no relevant historical destination information in the node vector \( \text{vector}_i \) of the vehicle node \( c_i \) in the time period \( T \), the vehicle position is calculated into two types of cases:

2.1.1. Vehicle node \( c_i \) has no location information during time period \( (T-1) \). The vehicle node \( c_i \) sends a \( \text{packet}_\text{request} \) to the neighboring vehicle in the range of distance \( L_d \) to request location feedback at a certain frequency. After receiving the \( \text{packet}_\text{request} \), the neighbor node adds the packet to the packet at the time of \( T \) to the \( \text{packet}_\text{response} \) returns. After receiving the data packet, the vehicle node \( c_i \) extracts the location information and stores it in a temporary location set \( S_{\text{location}}^T \), that is, as shown in the formula (1):

\[
S_{\text{location}}^T = \{(c_1, \text{lng}_1^T, \text{lat}_1^T), \{c_2, \text{lng}_2^T, \text{lat}_2^T\}, \ldots, \{c_n, \text{lng}_n^T, \text{lat}_n^T\}\}
\]

(1)

Where \( \{\text{lng}_i^T, \text{lat}_i^T\} \) indicates that the longitude of the vehicle node \( c_i \) is \( \text{lng}_i^T \) and latitude \( \text{lat}_i^T \) in the time period \( T \). Therefore, when the vehicle node \( c_i \) receives the data \( \text{packet}_\text{response} \), and the signal strength is stored in a temporary signal set \( S_{\text{signal}}^T \), which is also shown in formula (2):

\[
S_{\text{signal}}^T = \{(c_1, \text{intensity}_1), \{c_2, \text{intensity}_2\}, \ldots, \{c_n, \text{intensity}_n\}\}
\]

(2)

\[
\begin{align*}
\text{lng}_i^T &= \frac{\sum_{j=1}^{n}(\text{lng}_j^T \cdot \text{intensity}_j) - \sum_{j=1}^{n} \text{intensity}_j \cdot \langle \text{lng}_j^T \rangle}{\sum_{j=1}^{n} \text{intensity}_j} \\
\text{lat}_i^T &= \frac{\sum_{j=1}^{n}(\text{lat}_j^T \cdot \text{intensity}_j) - \sum_{j=1}^{n} \text{intensity}_j \cdot \langle \text{lat}_j^T \rangle}{\sum_{j=1}^{n} \text{intensity}_j}
\end{align*}
\]

(3)

Where \( \{\text{lng}_i^T, \text{lat}_i^T\} \) represents that during the time period \( T \), the vehicle node \( c_i \) perceives that the signal strength from the neighbor node \( c_j \) is \( \text{intensity}_j \). Then, according to the temporary location set \( S_{\text{location}}^T \) and the temporary signal set \( S_{\text{signal}}^T \), the location information \( \{c_i, \text{lng}_i^T, \text{lat}_i^T\} \) of the vehicle node \( c_i \) can be calculated. That is, it is shown in formula (3).

2.1.2. Vehicle node \( c_i \) has location information in time period \( (T-1) \). For this case, this paper first obtains the average vehicle speed \( \text{avg}_i(T-1) \) of the vehicle node \( c_i \) in the time period \( (T-1) \), and then the average vehicle speed \( \text{avg}_i(T) \) of the current time period \( T \) jointly calculate the average speed \( \text{avg}_i(T^*) \) of the vehicle in these two time periods, as shown in formula (4):

\[
\text{avg}_i(T^*) = \frac{\text{avg}_i(T-1) + \text{avg}_i(T)}{2}
\]

(4)

\[
\begin{align*}
\text{lng}_i^T &= \text{lng}_i^{T-1} \left(1 + 2 \arcsin \left( \frac{\text{avg}_i(T^*)}{2R} \right) \sin \theta \right) \\
\text{lat}_i^T &= \text{lat}_i^{T-1} \left(1 + 2 \arcsin \left( \frac{\text{avg}_i(T^*)}{2R} \right) \cos \theta \right)
\end{align*}
\]

(5)
Obtain the traveling direction $d_{T-1}$ of the vehicle in the time period $(T-1)$, and the angle between the direction and the earth rotation direction $d_{earth}$ is $\theta$, then if the earth radius is represented by $R$, and due to the time period $(T-1)$ The distance traveled by the vehicle node to the time period $T$ is short, so the earth surface curvature is ignored, and the position information of the vehicle node $c_i$ in the time period $T$ can be obtained by solving the straight line distance $(c_i, lng_i, lat_i)$, as shown in formula (5).

2.2. Relevant historical destination information. vector$_i \neq \emptyset$

If within the time period $T$, there is relevant historical destination information in the node vector $vector_i$ of the vehicle node $c_i$, and the current vehicle node has only one destination $l$, then all the vehicle node identifier IDs accompanying the destination $l$ are stored to Temporary set $S_{label-l} = \{c_1, c_2, ..., c_n\}$. Then, the position of the vehicle node $c_i$ is calculated based on other vehicle node position information accompanying the destination in the VANET. First, in order to detect the neighboring vehicle node around the vehicle node $c_i$, the vehicle node $c_i$ broadcasts a probe request message probe request to the surroundings, and if the vehicle node $c_i$ receives the message probe response returned from the neighbor vehicle node, the neighbor vehicle is extracted from the message. The node identifier ID is stored in the temporary set $S_{temp-neighbor} = \{c_1, c_2, ..., c_N\}$, where $N < n$, and then the intersection of $S_{label-l}$ and $S_{temp-neighbor}$ is obtained. That is, as shown in formula (6):

$$S_{inter} = S_{label-l} \cap S_{temp-neighbor} = \{c_{inter-1}, c_{inter-2}, ..., c_{inter-n}\}$$  \hspace{1cm} (6)

The vehicle node $c_i$ broadcasts to all neighbor nodes a message location request requesting its location information, which contains all elements in $S_{inter}$. When the neighbor node receives the location request, it determines whether its own identifier ID exists in the set $S_{inter}$. If it exists, it represents that it belongs to the neighbor node that can be detected around the vehicle node $c_i$, and also represents the same destination as the vehicle node $c_i$. It can locate the vehicle node $c_i$, and then encapsulate its own location information into the message location response. Returned to the vehicle node $c_i$. After receiving the message sent by the neighboring vehicle node, the vehicle node $c_i$ extracts its location information, stores it in the set $S_{location_t}$, and calculates the location of the vehicle $c_i$ by using the corresponding method of formula (3).

If within the time period $T$, the node $vector_i$ of the vehicle node $c_i$ has relevant historical destination information, but the current vehicle node does not only have one destination $l$, then it is necessary to select a most suitable destination to the vehicle. Node $c_i$ is positioned. First, the vehicle node $c_i$ calculates the straight line $distance_j$ between the neighboring node $c_j$ and its own according to the signal strength when the neighbor node sends the message, and sets $n$ as the number of neighbor nodes, and then stores all of them into the temporary set, that is, As shown in formula (7):

$$S_{distance}^{distance} = \{distance_1, distance_2, ..., distance_j, ..., distance_n\}$$  \hspace{1cm} (7)

Then according to the direction of the signal transmission when the neighbor message is received according to the time period $(T-1)$ and the next time period $T$, calculate the angle inclination$_i$ generated by the traveling direction of the vehicle node $c_j$ and $c_i$, and set a total of $n$ neighbor vehicles, that is, As shown in formula (8):

$$S_{inclination}^{inclination} = \{inclination_1, inclination_2, ..., inclination_j, ..., inclination_n\}$$  \hspace{1cm} (8)

Next the vehicle node $c_i$ calculates the time period of the data reception time from the current positioning time according to the timestamp of the last received node $c_j$ data packet and the time stamp of the current positioning time. $time_j$ and set $n$ as the number of neighbor nodes. And then save it all in a temporary collection, as shown in formula (9):
Finally, the number of neighboring vehicle nodes that can communicate in the current time period in each destination cluster accompanying the vehicle node \( c_i \) is counted, and all of them are stored in the temporary set, that is, as shown in formula (10):

\[
S_{\text{communication}} = \{ \text{communication}_{c_1}, \text{communication}_{c_2}, ..., \text{communication}_{c_m} \}
\]

Where \( cs \) represents all destination clusters and \( m \) represents destination overlap, which is \( L_o(c_i, T) = m \). Through the above calculation, the most suitable destination for the vehicle node \( c_i \) can be calculated according to formula (11).

\[
\max_{1 \leq \mu \leq m} \sum \frac{\text{distance}_{time_\tau}(\pi - \text{inclination}_\tau)}{\sum_{\tau=1}^n \text{distance}_{time_\tau}}
\]

The above algorithm can obtain the companion destination that is most suitable for positioning, and then acquire the location of the vehicle node based on this destination.

3. Positioning result correction algorithm

In a certain period of time \( T \) obtains the strength of the current GPS signal of the vehicle node \( c_i \) \( \text{strength}_\tau \), and then calculates the position error \( \text{error}_\tau \) when the signal strength is \( \text{strength}_\tau \) according to the relationship between the GPS signal strength and the position error. Then use GPS to obtain the current real-time location of the vehicle node \( c_i \) \( \text{location}_{\text{obtain}}(T) \), and then solve the current real-time location \( \text{location}_{\text{calculate}}(T) \) by the LCVN algorithm proposed in the previous section. Finally, calculate the distance between the two positions, which can be solved by the method of norm, that is, as shown in formula (12):

\[
\text{spacing}_\tau = \| \text{location}_{\text{calculate}}(T) - \text{location}_{\text{obtain}}(T) \|
\]

If the distance \( \text{spacing}_\tau \) between the location information \( \text{location}_{\text{calculate}}(T) \) and the location information \( \text{location}_{\text{obtain}}(T) \) is less than or equal to the location error \( \text{error}_\tau \), that is, as shown in the formula (13):

\[
\text{spacing}_\tau \leq \text{error}_\tau
\]

It is considered that the LCVN algorithm proposed in this paper is accurate and does not need to be corrected. However, if the distance \( \text{spacing}_\tau \) between the location information \( \text{location}_{\text{calculate}}(T) \) and the location information \( \text{location}_{\text{obtain}}(T) \) is greater than the location error \( \text{error}_\tau \), it is as shown in formula (14):

\[
\text{spacing}_\tau > \text{error}_\tau
\]

Then use the following steps to fix the positioning results:

(1) The information of the destination accompanying the vehicle in the time period \( T \) is taken out from the vector \( \text{vector}_i \) of the vehicle node \( c_i \), and this information is stored in the temporary set, that is, as shown in the formula (15):

\[
S_{\text{label}} = \{ (c_1, T), (c_2, T), ..., (c_n, T) \}
\]
(2) Iteratively deletes the elements \((c, \tau)\) from the set \(S_{\text{label}}\), and then uses the LCVN algorithm proposed in this paper to calculate the real-time position of the \(c_i\) of the vehicle node after each element is deleted, and calculates the value of \(\text{spacing}_\tau\) at this time;

(3) Calculate the element \((c_{\text{min}}, \tau)\) of the collection when \(\text{spacing}_\tau\) gets the minimum value, delete the set element \((c_{\text{min}}, \tau)\) from the set \(S_{\text{label}}\), and the set is updated to the new set \(S_{\text{label}}'\) and calculate this. The position information of the time \(\text{location calculate}(\tau)\) and the distance information at this time \(\text{spacing}_\tau'\); if \(\text{spacing}_\tau' \leq \text{error}_\tau\), the algorithm stops; otherwise repeat the above steps (1)-(4) until the stop condition is met.

4. Experimental results and analysis

This paper will verify the performance of the localization algorithm based on the destination partitioning results through a series of simulation experiments. In this experiment, a \(4000 \times 3000 m^2\) area in Hangzhou will be selected as the experimental area. As shown in Figures 1 and 2, the vehicle nodes will be randomly scattered on the main road in the area, as shown in Figure 3. Both the Package exchange and the Neighbor information can be exchanged between the nodes. Figure 4 represents an enlarged schematic view of one of these roads. Before running the location-based partitioning result-based positioning algorithm (LCVN), the vehicle nodes in the VANET have already performed the destination cluster division by the matching algorithm based on the vehicle node destination overlap (MALO), that is, the current time period \(T\), at VANET. The accompanying vector \(\text{vector}_i\) of any vehicle node \(c_i\) in the middle contains the corresponding destination cluster information. In the case of geolocation of any vehicle node \(c_i\), the vehicle GPS device cannot provide data in the current time period \(T\), and the vehicle position can only be acquired by the LCVN algorithm.

4.1. No relevant historical destination information, \(\text{vector}_i = \emptyset\)

First, the error analysis is an important indicator to measure the accuracy of a positioning algorithm. Second, because the current location time \(T\), the vehicle nodes in the VANET need to have two positions, one is the real position \(\text{Rel}_i(\text{lng}_\text{rel}, \text{lat}_\text{rel})\), the other is the calculation position obtained by the PPLR algorithm \(\text{Cal}_i(\text{lng}_\text{cal}, \text{lat}_\text{cal})\). The general error can be directly regarded as the difference \(\delta\) between the real position \(\text{Rel}_i(\text{lng}_\text{rel}, \text{lat}_\text{rel})\) and the calculated position \(\text{Cal}_i(\text{lng}_\text{cal}, \text{lat}_\text{cal})\). It can't directly deal with \(\delta\) as error. Instead, it takes one to balance the \(\delta\) value interval \(\tau\), and then after 300 times of positioning, it is worth Positioning Error every 10 times. The physical length of a vehicle that is common in life is about 4.5m. In this paper, the length of the vehicle of 2~4 segments (9~18m) is selected as the balance interval, as shown in formula (16)(17):

**Known:**

\[
\{ \delta = \|\text{Rel}_i(\text{lng}_\text{rel}, \text{lat}_\text{rel}) - \text{Cal}_i(\text{lng}_\text{cal}, \text{lat}_\text{cal})\| \}
\]

\[9 \leq \tau \leq 18\]

**Result:**

\[
\text{Positioning Error} = \begin{cases} 0, \delta \leq \sigma \\ \sqrt{(\text{lng}_\text{rel} - \text{lng}_\text{cal})^2 + (\text{lat}_\text{rel} - \text{lat}_\text{cal})^2} - \tau, \delta > \sigma \end{cases}
\]

(17)
4.2. Comparative Experiment

In this paper, the GPS-free algorithm proposed by Khattab A is used. Using the DDGPS algorithm proposed by M Rohani as a second set of comparative experiments, the GNSS algorithm proposed by De Ponte Muller is used. The algorithm uses the communication between the car and the car to convey the information that the radar sensor is incorrectly located due to factors such as buildings. Improve overall positional accuracy. [12] First, randomly broadcast 100 vehicle nodes on the Experimental Area to simulate the current traffic conditions. Observe the two indicators of each algorithm in this case: the positioning error trend and the positioning error value. Finally, 500, 1000 and 2000 vehicle nodes were planted separately, simulating the comparison of good traffic conditions and poor neutralization. The experimental results are shown in Figures 5-8:

1. As the number of operations increases, the error continues to decrease, and the decline has a slowing trend;
2. When there are few vehicle nodes, there are fewer neighboring vehicle nodes that can be used in the same destination cluster, and the positioning error is large. As the number of vehicle nodes increases, the positioning error decreases, but when the number increases to the road saturation state, the intervention of the abnormal node causes the positioning error to increase.
3. DDGPS has a lower positioning error when the amount of computation is higher, but the algorithm has greater error volatility than the PPLR algorithm.
The size of the PECT can be measured by the standard deviation $\sigma_{\text{PECT}}$ of Positioning Error. The size of the PEV can be measured by the average $\mu_{\text{PECT}}$ of Positioning Error. This paper draws the results shown in Figure 9. Further conclusions can be drawn from the experimental results box plot:
Figure 9. Positioning errors for four algorithms at 100, 500, 1000, and 2000 vehicle counts

- The upper limit of LCVN algorithm is lower than that of similar algorithms, and the median of comparison is lower than that of similar algorithms.
- The LCVN algorithm positioning error average $\rho_{PECT}$ is the smallest when the number of vehicle nodes is 100-2000, and the accuracy is better than the similar algorithm;
- The LCVN algorithm positioning error standard deviation $\sigma_{PECT}$ is the smallest when the number of vehicle nodes is 100-2000, and the robustness is better than the similar algorithm.

5. Conclusion
This paper specifically describes a vehicle location algorithm based on community discovery, including a vehicle node location algorithm and a location result correction algorithm. In this paper, the vehicle node is first given the accompanying label of the driving destination, and then the neighboring vehicle node is used to perform the positioning calculation on the location of the current vehicle node according to the technology that the vehicle nodes can communicate with each other. This paper also innovatively proposes a positioning result correction algorithm based on the calculation results for positioning correction of inaccurate positions. In this paper, the simulation data is located, and the accuracy of the positioning algorithm is analyzed from the perspective of positioning error. The experimental results show that the proposed positioning algorithm can achieve better results.

Acknowledgments
This work was financially supported by the National Natural Science Foundation of China (No. 61602141) and the Key Research and Development Program of Zhejiang Province, China (Grant No.2019C03138).
References

[1] Yuan Y, Xiao B, Wu G. Multi-channel-based Sybil Attack Detection in Vehicular Ad Hoc Networks using RSSI [J]. IEEE Transactions on Mobile Computing, 2019, 1 (99):1-1.

[2] Lai M, Yang H, Yang S. Cyber-physical logistics system-based vehicle routing optimization [J]. Journal of Industrial & Management Optimization, 2017, 10 (3):701-715.

[3] Herrera-Quintero L F, Vega-Alfonso J C, Banes K B A. Smart ITS Sensor for the Transportation Planning Based on IoT Approaches Using Serverless and Microservices Architecture [J]. IEEE Intelligent Transportation Systems Magazine, 2018, 10 (2):17-27.

[4] Zhang Li, Xu Wei, Xia Chao. Design of AGV navigation system based on RFID positioning[J]. Electronic World, 2017, 1 (5):67-69.

[5] Chen Shijun, Wang Huiqiang, Wang Yuanyuan. A Base Station Selection Optimization Method for Indoor Positioning[J]. Computer Science, 2018, 45 (10):115-119.

[6] Ou C H, Wu B Y, Cai L. GPS-free vehicular localization system using roadside units with directional antennas [J]. Journal of Communications and Networks, 2019, 21 (1):12-24.

[7] Xin Z, Li J, Yan X, et al. Robust Adaptive Cubature Kalman Filter and Its Application to Ultra-Tightly Coupled SINS/GPS Navigation System [J]. Sensors, 2018, 18 (7):2352.

[8] Moritoki N, Watanabe K, Nagai I. A method for measuring the position of quadrotors using a tether winder [C]. 2016 16th International Conference on Control, Automation and Systems (ICCAS). IEEE, 2016.

[9] Shi Jinhui, Tang Xiaorong, Zhang Wei. The Influence of Haze Weather on GPS Measurement Accuracy [J]. 2017 (12):11-12.

[10] Khattab A, Fahmy Y A, Wahab A. High Accuracy GPS-Free Vehicle Localization Framework via an INS-Assisted Single RSU [J]. International Journal of Distributed Sensor Networks, 2015, 2015 (1):1-16.

[11] Rohani M, Gingras D, Gruyer D. A Novel Approach for Improved Vehicular Positioning Using Cooperative Map Matching and Dynamic Base Station DGPS Concept [J]. IEEE Transactions on Intelligent Transportation Systems, 2015, 17 (1):230-239.

[12] De Ponte Müller, Fabian, Munoz Diaz E, Rashdan I. Cooperative Infrastructure-based Vehicle Positioning [C]. Vehicular Technology Conference. IEEE, 2016:3004-3010.