IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 20, 2019, accepted January 23, 2020, date of publication January 30, 2020, date of current version March 13, 2020.

*Digital Object Identifier 10.1109/ACCESS.2020.2970503*

# ICCIU: A New Real-Time Lane-Level Positioning Algorithm

**WENXIN TENG, YANHUI WANG, BIBO YU, AND JIAHAO LIU**

3D Information Collection and Application Key Lab of Education Ministry, Capital Normal University, Beijing 100084, China
Beijing State key Laboratory Incubation Base of Urban Environmental Processes and Digital Simulation, Capital Normal University, Beijing 100048, China

Corresponding author: Yanhui Wang (huiwangyan@sohu.com)

This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB0505400, in part by the National Natural Science Foundation of China under Grant 41771157, in part by the Great Wall Scholars Program under Grant CIT&TCD20190328, in part by the Key Research Projects of National Statistical Science of China under Grant 2018LZ27, in part by the Young Yanjing Scholar Project of Capital Normal University, and in part by the Academy for Multidisciplinary Studies, Capital Normal University.

**ABSTRACT** Previous real-time lane-level positioning algorithms for in-vehicle navigation systems have problems of inaccurate positioning and insufficient robustness. Therefore, this paper proposes a new lane-level positioning algorithm integrating Improved filter method, Curve Circle method and Improved Unet (ICCIU) method. Specifically, in ICCIU algorithm, we design improved filter method to improve the accuracy of original position by combining three filters to enhance image features, propose curve circle method to achieve real-time curve positioning by introducing two movement indicators to get precise curve position and design a light improved Unet method that integrates residual thought and cascading thought to detect the lane, and integrates two new parameters to get more accurate position by reducing the horizontal position errors The experiment results show that evaluation indicators of the improved Unet method are over 20% higher than those of the existing algorithms, the running time of single point positioning is about 28ms and lane-level accuracy is over 96% used by ICCIU, which demonstrates the pretty performance both in feasibility and efficiency of the new algorithm.

**INDEX TERMS** Lane-level positioning, improved filter method, curve circle, improved Unet, ICCIU.

## I. INTRODUCTION

With the rapid popularization of 5G communication technology and the progress of Intelligent Transportation System (ITS), Autonomous driving has become a hot topic in current research area. The vehicle's self-positioning algorithm is directly related to whether automatic driving can provide accurate real-time response or not. Therefore, how to design an accurate, real-time and light mobile lane-level positioning algorithm is still a problem deserving of study.

Many scholars have studied various lane-level positioning algorithms, including: (1) Road structure [1]–[4]. These algorithms obtain lane-level position by using road structure and monocular camera to increase the accuracy of single GPS. These algorithms have high efficiency but lack the curve positioning method, and the accuracy needs to be improved. (2) Graphics [5], [6]. The algorithms used lane detection methods (e.g., Hough [7]–[9], RANSAC [7]–[9], Least Squares [12]–[14]. lane line feature [11], [15]–[17]) to obtain lane lines, and then applied high GPS to achieve lane-level positioning. Unfortunately, they rely heavily on the precision of lane detection and rarely have methods to reduce the horizontal position errors. (3) Sensor fusion [18]–[21]. These methods used sensor fusion such as infrared, LIDAR and radar to get lane-level location and employed the difference between lane line and background to achieve lateral position correction. Although these methods have high precision and can complete all-weather and all-day positioning, the cost and way of these positioning methods restrict existing vehicles to apply these algorithms. (4)Deep learning [22]–[27]. These algorithms use deep learning model (e.g., Unet) and image features to get lane-level position. For example, Unet has good accuracy in lane detection which can be positioned without restrictions but doesn't have good efficiency because it has too many parameters. These algorithms put the rightmost lane as the original location which will cause positioning errors when car navigation system starts at other lane. Although these algorithms have been used to improve positioning accuracy, little attention has been paid to reduce horizontal position errors.

The associate editor coordinating the review of this manuscript and approving it for publication was Yanli Xu.

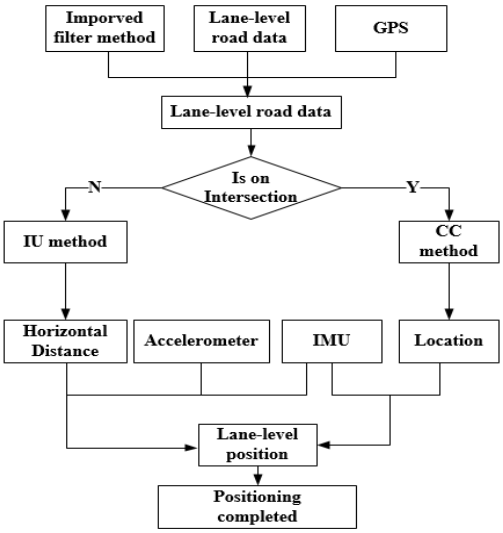



**FIGURE 1.** Overall process of ICCIU algorithm.

In this context, we propose a new real-time lane-level positioning algorithm for vehicle positioning by integrating

Improved filter method, Curve Circle method and Improved Unet method (referred to as ICCIU). Our main contribution is to improve the lane-level accuracy and efficiency by solving the problem of corner positioning and reducing horizontal position errors. Firstly, we design improved filter method to get initial position and propose Curve Circle (CC) method to achieve corner positioning. Secondly, we develop Improved Unet (IU) method to reduce horizontal position errors by employing new downsampling to enhance the feature and pruning network to improve efficiency by reducing redundant parameters. Finally, we conduct some experiments to verify accuracy and timeliness of the algorithm.

## II. METHOD

To respond to the lane-level algorithm's problems in terms of lacking curve positioning and low accuracy in real-time positioning for in-vehicle navigation systems, this paper proposes an ICCIU algorithm which integrates Improved Filter method, Curve Circle (CC) method and Improved Unet (IU) method to improve the lane-level positioning precision and efficiency. Specifically, in this algorithm, improved filter method integrates three filters (such as Select Saturation (SelectS), Sobelx, YWmask) to ensure that the initial location can be positioned accurately and quickly. CC method is firstly proposed to achieve corn er positioning by integrating centripetal acceleration and angular velocity. Light improved Unet method is designed to improve positioning accuracy by reducing horizontal position errors.

The overall process of the ICCIU algorithm is shown in Fig.1. Firstly, ICCIU employs GPS and improved filter method to get original location. Then, if vehicle is on the intersection, the CC method is used to achieve curve positioning, else if the vehicle drives on the straight road, IU method is applied to get more accurate position based on improved filter method results. At last, IMU and accelerometer are used to correct lane-level position.

### A. IMPROVED FILTER METHOD

Traditional lane-level algorithms putting the rightmost lane as the original location, which will cause positioning errors when car navigation system starts at the other lane. Therefore, this paper designs an improved filter method to get accurate original location by integrating Hough Transform [28] and filter method. Through different filtering experiments, from the result show in Fig.2, we find that lane lines have obvious characteristics in the width direction, saturation and yellow-white color.

Therefore, this paper introduces *Sobelx* standing for Sobel filtering on the image width, *SelectS* standing for filtering image on saturation and *YWmask* standing for filter image on yellow and white color as three parameters, and proposes improved filter method that integrates three parameters (*Sobelx*, *SelectS* and *YWmask*). In addition, the camera is used to get initial position at the center of the vehicle behind the front windshield. The improved filter method is as follows and also shown in Fig.3.

(1) Selecting Region Of Interesting(ROI). Perform inverse perspective transformation on image and record the images as original image. We convert RGB original image to HSL image and normalize image pixel values to 0∼1, then select saturation in image to get binary the image which is recorded as *SelectS*.

(2) Binarizing the selected image. Determined the number of lane lines as $l_i$ (i = 1,2,3,4…) by counting the number of effective peaks. Then, select yellow and white color to filter the image, record the result as a combination *YWmask*.

(3) The binaries image is subjected to Sobel filtering processing along the image width direction. The filter is 3*3 and also show in formula 1. Then, filter the image and record it as *Sobelx*.

$$\text{Filter} = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{1}$$

To improve detection efficiency, we normalize the three image (*SelectS, YWmask* and *Sobelx*) to 0 and 1. To get the fused image f, should follow these principles:

Assuming *pixel1*, *piexl2*, *piexl3* and *piexlf* are the pixel value of the same position in the images of *SelectS,YWmask, Sobelx* and *f*.

Rule 1: If $pixel1 = pixel2 = pixel3 = S$ ($S = 0$ or 1), we set pixelf = s.

Rule 2: If one of them (*pixel1*, *piexl2* and *piexl3*) are 1, we set pixel3 = 1.

(4) Multi-window is taken on the image f to extract the lane line. Set the upper left corner of the image as the origin of the coordinates. And then, we use the least squares method [29] to fit the pixels($x_i$, $y_i$) in each search box to get lane line set

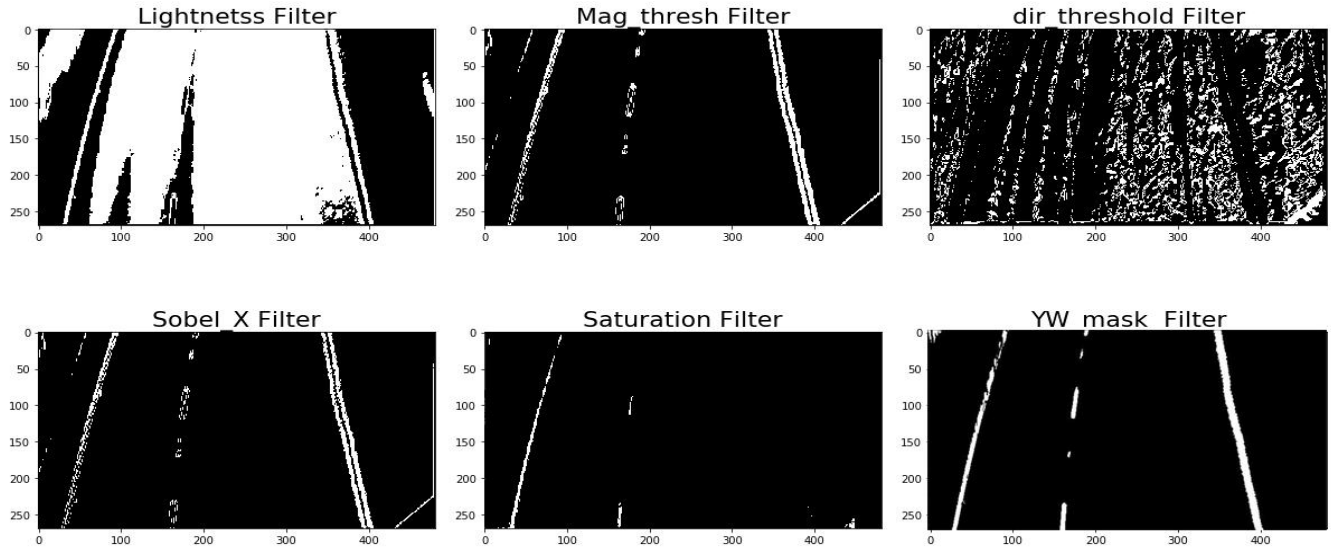



**FIGURE 2.** Results of different filter method.

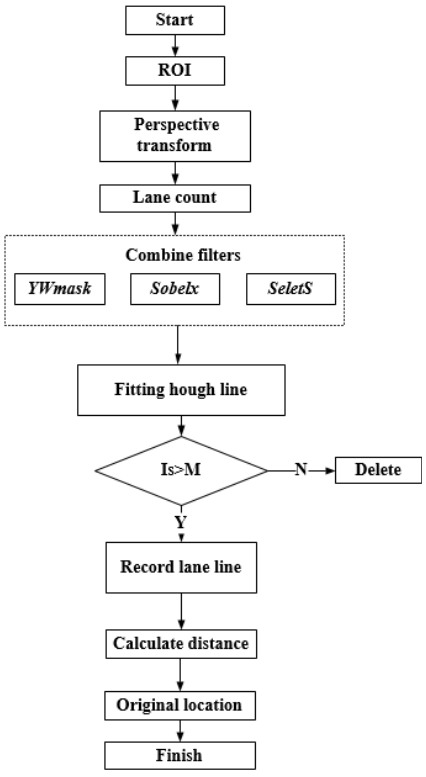



**FIGURE 3.** Overall process of improved filter method.

$l_i$ ($i$ = 1,2,3,4…). The lane line on the far left of the image is marked as $l_1$. Assuming that the linear equation of $l_i$ is:

$$y = ax + b \tag{2}$$

Least squares method requires that $D$ is the smallest when $a,b$ is the smallest. $D$ is defined as follow:

$$D = \sum_{i=1}^{n} d_i^2 = \sum_{i=1}^{n} (y_i - a - b_i)^2 \tag{3}$$

Then, it can get the $a,b$.

$$a = \frac{xy_{ave} - x_{ave}y_{ave}}{(x^2)_{ave} - (x_{ave})^2}, \quad b = y_{ave} - ax_{ave} \tag{4}$$

where,

$$xy_{ave} = \frac{1}{n}\sum_{i=1}^{n} x_i y_i^2, \quad x_{ave} = \frac{1}{n}\sum_{i=1}^{n} x_i^2$$

$$y_{ave} = \frac{1}{n}\sum_{i=1}^{n} y_i^2, \quad (x^2)_{ave} = \frac{1}{n}\sum_{i=1}^{n} x_i^{2^2}.$$

Then, setting the threshold M to remove the noisy line. M is determined according to the length of lane line, space and the proportion of the image. Removing the lane line if its line spacing is less than the threshold M.

(5) Calculating the distance $d_i$(i = 1,2,3,4…) between image center coordinate points $p_c$ $(x,y)$ and lane line set. If $d_i$ and $d_{i+1}$ are the two smallest distance values, the vehicle is on $l_i$.

Fig.4 is the result of the improved filter method proposed in this paper. (a) is the original image, (b) is the inverse perspective image, and (c) is the result of combining three filter binaries image. From (d) we can see that all lane lines are correctly detected.

### B. CURVED CIRCLE METHOD

Considering that the vehicle is perpendicular to the stop line when turning, the center of the arc of the vehicle turning is on the extension of stop line. And the radius of the vehicle (R) turning is not only related to the angular velocity (a) but also the driving speed (v), the Curve Circle (CC) method

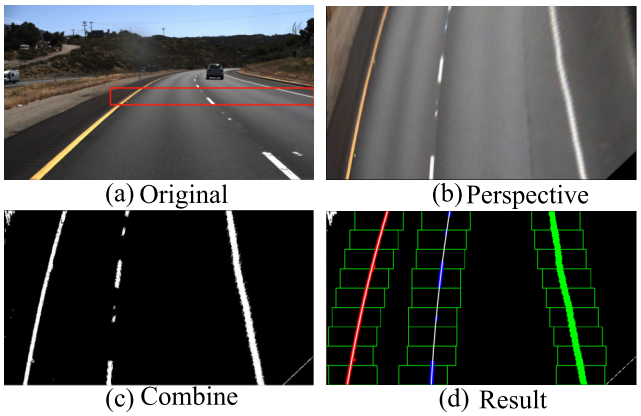



**FIGURE 4.** Results of improved filter method.

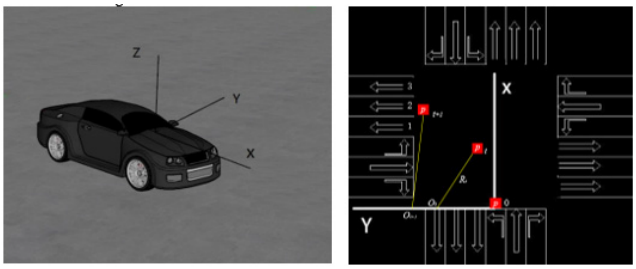



**FIGURE 5.** Vehicle coordinate system.

integrates velocity (a) and the driving speed (v) velocity to construct a circle that can map matching while vehicle turning. The Curve Circle method is as follows.

The reading of the three-axis acceleration sensor in the stereo coordinate system is $X_{acc} = \{x_{acc}, y_{acc}, z_{acc}\}$ , angular velocity reading is $w_{ang} = \{w_x, w_y, w_z\}$ , and gyroscope reading is $\{x_{pg}, y_{pg}, z_{pg}\}$ . As shown in Fig.5, taking left-turning of the two-way and six-lane vehicle $p$ as an example, the coordinate system is established with the road intersection $L_0$ $(x_0, y_0)$ where the vehicle $p$ is located. According to vehicular dynamics, the radius of the curve circle (R) is met:

$$ma = m\frac{V^2}{R}, \tag{5}$$

$$R = \frac{a}{w^2}, \tag{6}$$

where, $a$ is centripetal acceleration, $w$ is turning angular velocity.

The turning radius $(R_t)$ of the vehicle at $t$ time can be obtained according to the formula 6, and the curve circle is established with as the center and $R_t$ as the radius get location. The real vehicle position $L_t$ $(x_t, y_t)$ at t time in the geographic coordinate system is:

$$x_t = R_t sin(\theta) + x_0, \tag{7}$$

$$y_t = R_t(1 - \sqrt{1 - sin^2\theta} + y_0, \tag{8}$$

where, $\theta = \sum_{i=0}^{i=t} |w_i|$.

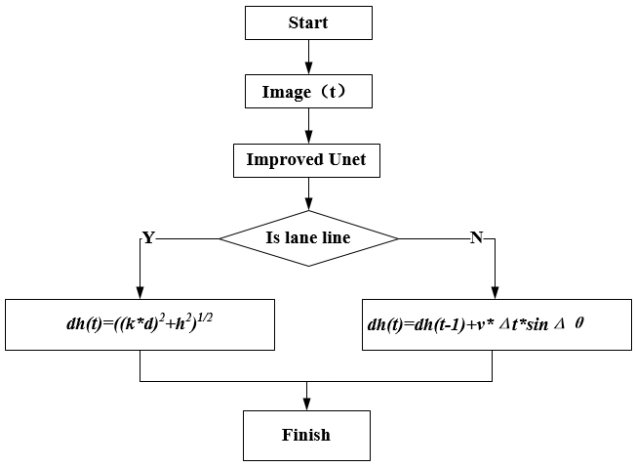



**FIGURE 6.** Overall process of IU method.

Similarly, the coordinates of the position of the right turn of the vehicle are:

$$x_t = R_t sin(\theta) + x_0, \tag{9}$$

$$y_t = y_{t-1} - R_t(1 + cos(\sum_{i=0}^{i=t} |w_i|)), \tag{10}$$

### C. IMPROVED UNET METHOD

We design IUnet to detect the accurate lane lines and integrate height and width to obtain horizontal position. The overall process of improved Unet method is shown in Fig.6.

#### 1) IMPROVED UNET STRUCTURE

Traditional Unet network has a good effect on static medical image segmentation, but it does not have good efficiency, especially in real-time image segmentation. Under this backgroud, this paper designs an improved Unet(IUnet) which applies residual block to replace traditional plain network to enhance downsampling features, and combines shallow features with deep rough features to improve the accuracy of image segmentation. Besides, in order to improve model efficiency, we design experiments to reduce the network parameters. After a large number of experiments, our IUnet including a total of 585,665 parameters, 7 million fewer than the traditional Unet network. The IUnet is defined as follows and shown in Fig.7.

The IUnet consists of 16 layers of convolution, including four downsamplings and four upsamplings processes. To improve real-time performance of the model, we set the input size to 80*160*3. The IUnet is as follows:

(1) Convolution function. The network has 16 convolution layers that 1∼8 layers are for downsampling convolutions and 9∼16 upsampling convolutions (the first, fourth, fifth, and eighth layers are convolution layers, and the second, third, sixth, and seventh layers are residuals). In differential convolution, 9∼16 layers are transpose convolutions layers, which kernel_size = (3,3), strides = (1,1),

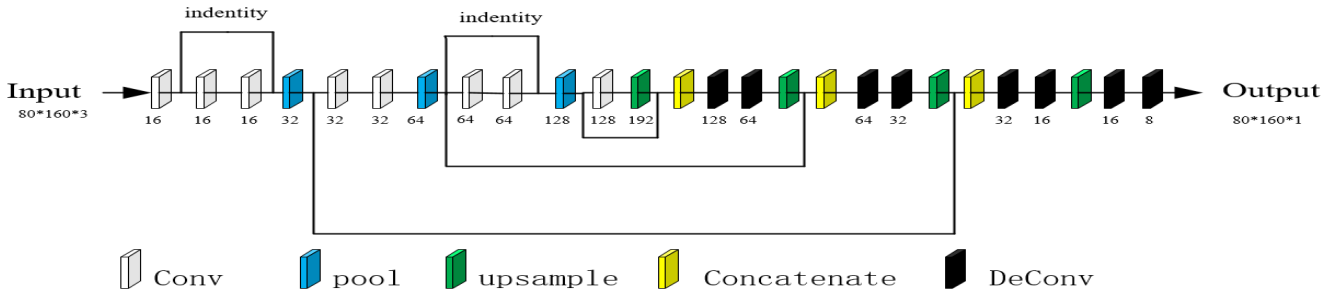
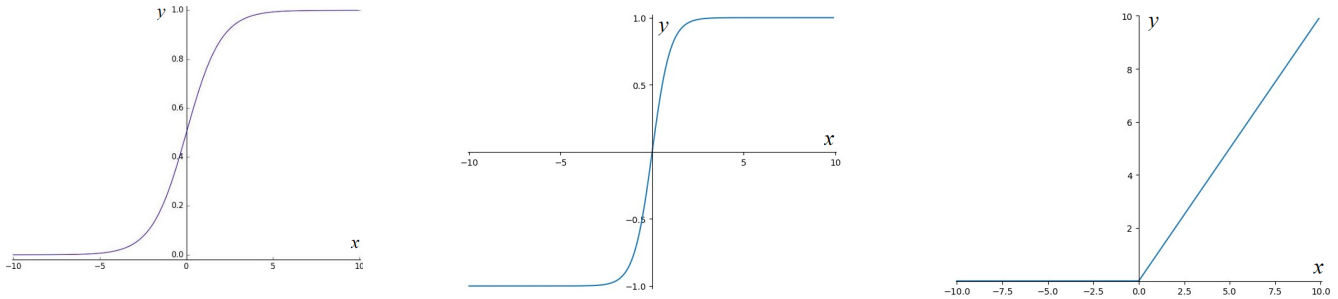



**FIGURE 7.** IUnet network structure diagram.




(a) *sigmoid* (b) *tanH* (c) *relu*

**FIGURE 8.** Common activation function.

activation = 'leakey_relu', and padding='same'. Each convolution layer is followed by a Batch Normalization layer.

(2) Activation function. Activation function. Common activation functions are sigmoid, tanH, and relu activation functions and shown in Fig.8.

*Sigmoid* activation function can make a big difference in output because of the slight change of weight. Therefore, it is easy to judge whether the network converges through output. But the activation function is easy to saturate. While retaining the advantages of *sigmoid* activation function , *tanH* activation function has a faster convergence rate than sigmoid. It is easier to make the network converge because its mean value is 0. However, the problem of this function is that when input value is large, the directional propagation gradient is close to 0, which makes it difficult for the network to converge, and the amount of calculation is also large. Therefore, it is not conducive to improve the efficiency of the algorithm. Compared with *sigmoid* and *tanH* functions, *relu* function has small calculation and can better deal with the problem of over-fitting. However, the large sparsity will make neurons of the model not work, resulting in a decrease of the effective capacity of the model. Therefore, this article selects the *leakey_relu* activation as activation function. leakey_relu activation function is as follows:

$$f(x) = max(ax, x) \tag{11}$$

(3) Optimization function. This paper selects adaptive moment estimation (Adam) as optimization function. Compared with the SGD optimization method with fixed learning rate, Adam can adaptively adjust the learning rate of each parameter through the first-order and second-order matrix estimation of the gradient. Adam takes up less memory than SGD and is more advantageous for sparsely stable targets. Therefore, Adam is more suitable for real time extracting lane lines than SGD.

(4) Loss function. The network output image size is 80*160*1. The network uses a loss function to evaluate the difference between the network output value and the true value. The experimental results show that compared with other loss functions, MSE has better image extraction effect. Therefore, this paper selects MSE as the loss function. The MSE is defined as follow:

$$MSE = \frac{1}{M}\sum_{m=1}^{M}(y_m - \hat{y}_m)^2, \tag{12}$$

where, $y_m$ is output value and $\hat{y}_m$ true value.

### 2) IMPROVED UNET METHOD

In order to get horizontal distance($d_h$ ), IU method installs cameras under the rearview mirror and transmits the image to IUnet extraction lane lines. Fig.12 is an example of obtaining horizontal distance of the right side ($d_{hr}$) of vehicle.

Assuming that the height of feature point a is $h$ and lane line($l$) is detected by IUnet on the right line of vehicle, we select the feature point $a(x, y)$ on the vehicle of the image.

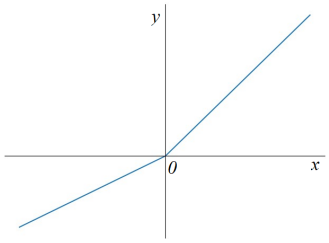



**FIGURE 9.** Leakey_*relu* activation function.

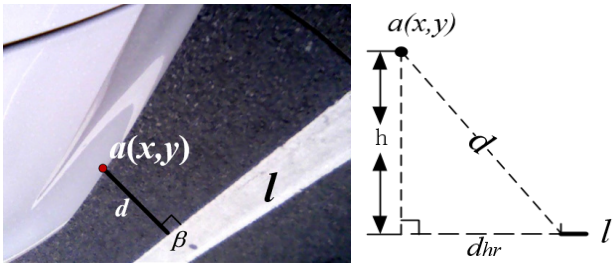



**FIGURE 10.** Example of obtaining horizontal distance.

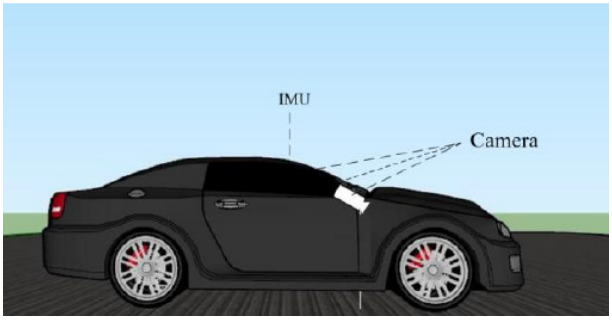



**FIGURE 11.** Overview of experimental equipment.

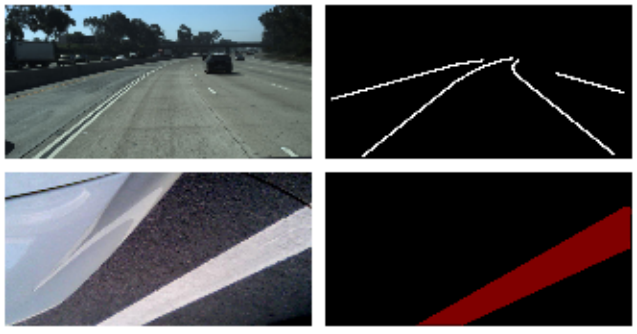


**FIGURE 12.** Tusimple database and self-annotated database.

$d_{hr}$ is defined as follow:

$$d_{hr} = \sqrt{(k * d)^2 + h^2}, \tag{13}$$

where, $k$ is ratio of the distance on the graph to the actual distance, $d$ is the shortest distance from lane line to point $a$.

If no lane lines are detected by IUnet, we calculate the distance of current frame by using the distance of previous frame and IMU. The distance is calculated as follows:

$$d_{hr} = d_{hr-1} + v * \Delta t * sin\Delta\theta, \tag{14}$$

where, $v$ is current vehicle speed, $\Delta t$ is time between two consecutive frames images, $\Delta\theta$ is the change of vehicle's driving angle, $d_{hr-1}$ is the last frame distance between lane line and vehicle.

**TABLE 1.** Confusion matrix.

| | Positive | Negative |
|---|---|---|
| True | True Positive(TP) | True Negative(TN) |
| False | False Positive(FP) | False Negative(FN) |

## III. EXPERIMENT & RESULTS

The ICCIU positioning algorithm proposed in this paper has been experimental in real experiments. The experimental designs and results are as follows.

### A. EXPERIMENT ENVIRONMENT

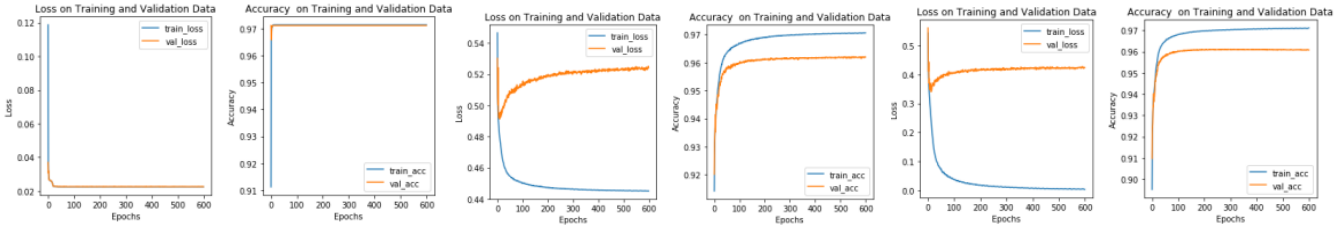
To verify the timeliness and accuracy of algorithm in this paper, we combine C# programming, python, keras 2.2.4 and ArcGIS Engine 10.2 to design some experiments under Windows 10 operating system. All tests were run on a computer with CPU: intel(R)Core (TM)i9-9900K @3.6 GHz×8 multithreaded processors, GPU: GEFORCE RTX 2080Ti 11G and 32G RAM. Meanwhile, we choose road of Rizhao as experimental area to complete the matching experiment. The experimental area is two-way four-lane. We use android mobile device to derive positioning data, and road data is downloaded from OpenStreetMap (OSM). Figure.11 shows that our experimental car which equipped with 3 cameras and an automotive grade IMU 3D space sensor. The camera's field of view that used to get initial position is 117° and resolution is 1280*720 pixels. The IMU static measurement accuracy is 0.05°, the heading dynamic measurement accuracy is 1°, and the roll and pitch measurement accuracy is 0.1°. The resolution of camera that used to reduce the horizontal position errors is 866*486 pixels. In addition, GPS positioning interval is 2S and cars collects data at the average speed of 35 Km/h. Furthermore, according to the experimental area road condition, we set M at 50 pixels.

### B. ALGORITHM EVALUATION INDEXES

To evaluate the performance of algorithm, the following indexes are designed to evaluate the accuracy and efficiency of algorithm in various aspects, according to the binary confusion matrix shown in table 1.

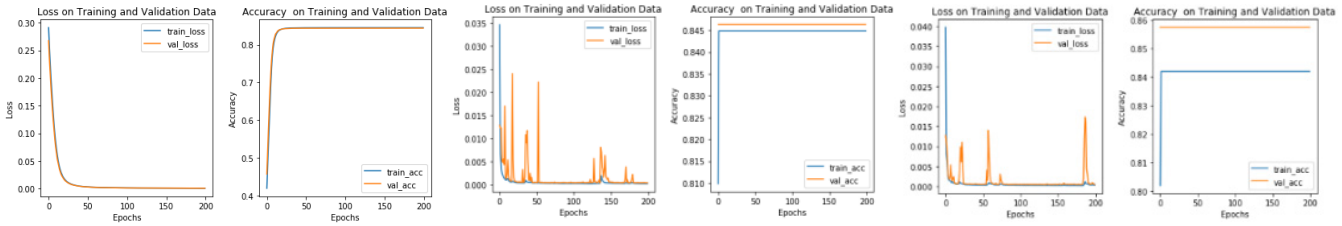
(1) Accuracy: This index is the degree of approximation between predicted result and real value. The larger this index is, the higher the accuracy of the model will be. Accuracy is defined as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{15}$$




(a) IUnet (b) Unet (c) ResNet-34-Unet

**FIGURE 13.** Network training Tusimple data sets.




(a) IUnet (b) Unet (c) ResNet-34-Unet

**FIGURE 14.** Network training Tusimple data sets.

(2) Recall: This index is the proportion of all correct samples that are correctly identified, which can reflect the model's ability to classify positive samples. The higher this index is, the stronger the classification ability is. Recall is defined as follows:

$$Recall = \frac{TP}{TP + TN}, \tag{16}$$

(3) F1_score: This index can reflect the classification ability of the model comprehensively. The higher the F1_score is, the better the classification effect of the modified model will be. F1_score is defined as follows:

$$F1\_core = \frac{2 * Precision * Recall}{Precision + Recall}, \tag{17}$$

where, $Precision = \frac{TP}{TP+FP}$.

(4) Average time: The index reflects the average time used to complete the positioning of single frame image in the matching process. It can reflect the timeliness of the algorithm in practical applications. The smaller the index is, the higher efficiency of the network.

(5) Lane-level accuracy: This index is the ratio of the length of correct lane-level position to the total matching length. The higher the index is, the higher accuracy of the algorithm.

### C. EXPERIMENTAL RESULTS

#### 1) IUNET METHOD PERFORMANCE

To verify the accuracy and efficiency of the IU method in this paper, we combine python 3.6.5 programming and keras 2.2.4 to design some experiments under windows 10 operating system. To compare the performance with Unet and ResNet-34-Unet, Tusimple database and self-annotated database were selected to divide training sets and verification sets according to 8/2 as shown in fig.12. The training parameters are shown in table 2.

**TABLE 2.** Training paramenters.

| | Tusimple database | self-annotated database |
|---|---|---|
| bach_size | 16 | 16 |
| epoch | 600 | 200 |
| Verification set`s count | 572 | 92 |

(1) Experiment in Tusimple data

Fig.13 shows the performance of different network models on the Tusimple dataset. From the figure we can see that IUnet training accuracy and verification accuracy are both over 97% and the accuracy is higher than the accuracy of the other two networks. Unet and Resnet-34-Unet are overtrained after 30 epochs, but IUnet still has a good performance that demonstrates IUnet has a better learning ability.

(2) Experiment in self-annotated data

Fig. 14 is the performance of different network models on self-annotated data sets. As can be seen from the figure, IUnet has better stability and higher accuracy compared with Unet and ResNet-34-Unet, indicating that IUnet more accurate in lane-line extraction.

(3) Experiment in self-annotated verification data

Fig. 15 is the results of recall and F1_score of each network model on the self-annotated verification set. According to the figure, the average recall and F1_score of IUnet are about 93.90% and 96.64%. Compared with 69% and 62.41%,

**TABLE 3.** Results of each network model.

| | IUnet | Unet | ResNet-34-Unet |
|---|---|---|---|
| Total parameters | 585,665 | 7,765,857 | 9,186,417 |
| Accuracy | 85.03% | 84.2% | 84.03% |
| Recall | 93.90% | 69% | 72.8% |
| F1_score | 96.64% | 62.41% | 75.91% |
| Average time | 28ms | 34ms | 45ms |


**FIGURE 15.** Recall and F1_score of each network model.

72.8% and 75.91% of the two other network models, our method is more precise and robust.

Table 3 is the results of different network models. It can be seen from the table that the total number of training parameters in IUnet model is far less than the number of parameters in Unet and ResNet-34-Unet. In terms of Accuracy, IUnet has higher accuracy than the other two networks. As for Recall and F1_score, IUnet is 30% higher than Unet and 20% higher than resnet-34-unet, indicating that IUnet has higher accurate and robust than the other two networks. The average time of IUnet single frame image positioning is 28ms, which is much faster than other two network models. Both the better performance of IUnet in precision and efficiency is verified in the experiments.

### 2) ICCIU ALGORITHM PERFORMANCE

As shown in Fig.16, we select Yingbin road, Donggang District and Rizhao City in China as the experiment area to examine ICCIU algorithm. Yingbin Road is a two-way four-lane road.

**TABLE 4.** Positioning results multipleareas.

| NO. | Approximate traveled distance(m) | Average time(ms) | Lane-level accuracy(%) |
|---|---|---|---|
| 1 | 5580 | 28 | 98.21% |
| 2 | 1952 | 24.8 | 97.10% |
| 3 | 6469.9 | 26.4 | 97.81% |
| 4 | 3359 | 25.9 | 96.12% |

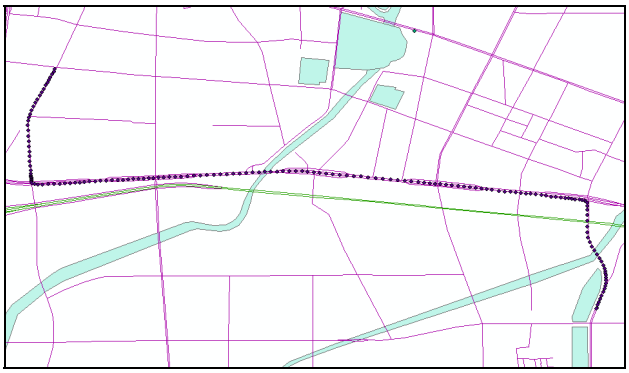


**FIGURE 16.** Overview of study area.

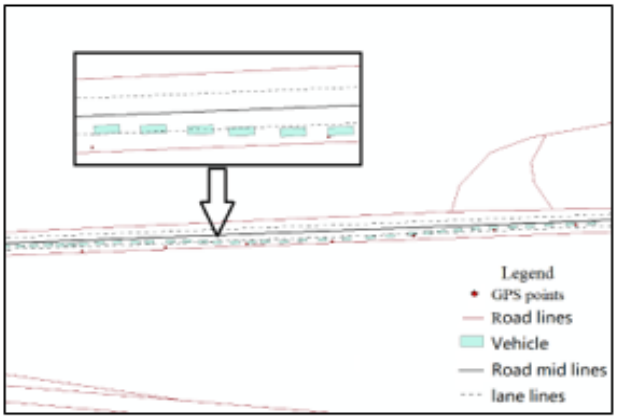



**FIGURE 17.** Lane-level positioning result using ICCIU algorithm.

Fig.17 is the experimental result of ICCIU algorithm for intercepting an approximately 380-meter-long road. There is

a total of 330 frames in this area. The following figure displays the lane-level positioning results in part of areas.

To verify the reliability of the algorithm, Shangdong Middle road, Yanhai highway and No. 204 National high way in Rizhao city (No.2, 3 and 4, as shown in figure 18) were selected for testing to compare with above-mentioned experiment (No.1). The results are shown in table 4. As the table shows, limited by the blurring of lane markings, the accuracy of NO.3 area decreases but is still higher than 96%, which indicates that ICCIU is very robust.

## IV. CONCLUSION & DISCUSSION

Aiming at the lane-level algorithm's problems in terms of lacking curve positioning and inaccuracy in real-time positioning for in-vehicle navigation systems, this paper proposes an ICCIU algorithm which integrates improved filter method to improve original location accuracy, proposes curve circle method to overcome the deficiency of the previous algorithms of positioning corner and designs IU method to improve lane-level precision by reducing horizontal position errors. The experiments show that IUnet has higher accuracy than the two other networks. In terms of Recall and F1_score, IUnet exceeds 93%, which is 30% higher than the score of Unet, 20% higher than the score of ResNet-34-Unet.The running time of singe frame image positioning is about 28ms and the lane-level accuracy is over 96% used by ICCIU algorithm, which demonstrates the higher performance of our algorithm both in feasibility and efficiency, and responds to the requirements of lane-level real-time positioning for in-vehicle navigation system. However, the accuracy of the algorithm depends on the clarity of the lane line marker, we will discuss the high-precision algorithm of lane extraction in next paper and use more data to evaluate the performance of ICCIU algorithm. Furthermore, this study mainly focused on 2-D road network and gave little attention to the positioning of 3-D road network such as overpasses, which will be our next study interest.

## REFERENCES

[1] Y. F. Yu, H. J. Zhao, J. S. Cui, and H. B. Zha, "Road structural feature based monocular visual localization for intelligent vehicle," *Acta Autom. Sinica*, vol. 43, no. 5, pp. 725–734, 2017.

[2] D. P. Ding, "Visual positioning technology to be based on pavement structure," *Construct. Machinery Technol. Manage.*, vol. 5, pp. 69–72, 2017.

[3] M. Wu, J. M. Hao. H. Fu, Y. Gao, and H. Zhang, "A stereo visual odometry poseoptimization method via flow-decoupled motion field model," *Acta Geodaetica et Cartograph. Sinica*, vol. 48, no. 4, pp. 460–472, 2019.

[4] Y. R. Zheng, J. Z. Yuan, and H. Z. Liu, "Method of Real Time Intersection Location Based on Monocular Vision for Intelligent Vehicle," *Comput. Eng.*, vol. 43, no. 9, pp. 288–299, 2017.

[5] M. M. Atia, A. R. Hilal, C. Stellings, E. Hartwell, J. Toonstra, W. B. Miners, and O. A. Basir, "A low-cost lane-determination system using GNSS/IMU fusion and HMM-based multistage map matching," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 11, pp. 3027–3037, Nov. 2017.

[6] R. Raymond, T. Morimura, T. Osogami, and N. Hirosue, "Map matching with hidden Markov model on sampled road network," in *Proc. 21st Int. Conf. Pattern Recognit.*, Nov. 2012, pp. 2242–2245.

[7] U. Ozgunalp and S. Kaymak, "Lane detection by estimating and using restricted search space in Hough domain," *Procedia Comput. Sci.*, vol. 120, pp. 148–155, Jan. 2017.

[8] A. Elsawy, M. Abdel-Mottaleb, and M. Abou Shousha, "Segmentation of corneal optical coherence tomography images using randomized Hough transform," *Proc. SPIE*, vol. 10949, Mar. 2019, Art. no. 109490U.

[9] M. C. L. Yuan, Y. C. Xu, and Y. L. Li, "Lane line extraction-based lateral positioning method for intelligent vehicle," *J. Mil. Transp. Univ.*, vol. 20, no. 10,pp. 38–43, 2018.

[10] Z. Zhu, Q. Xiong, J. Chen, F. Zhang, X. Liu, and G. Yao, "Intrinsic parameter calibration of line-scan cameras using RANSAC algorithm," in *Proc. Int. Conf. Inf. Syst. Comput. Aided Educ. (ICISCAE)*, Jul. 2018, pp. 416–425.

[11] M. Lehtomäki, A. Kukko, L. Matikainen, J. Hyyppä, H. Kaartinen, and A. Jaakkola, "Power line mapping technique using all-terrain mobile laser scanning," *Autom. Construct.*, vol. 105, Sep. 2019, Art. no. 102802.

[12] G. Shapira and T. Hassner, "Fast and accurate line detection with GPU-based least median of squares," *J. Real-Time Image Process.*, vol. 5, pp. 1–13, Oct. 2018.

[13] R. Fan and N. Dahnoun, "Real-time stereo vision-based lane detection system," *Meas. Sci. Technol.*, vol. 29, no. 7, May 2018, Art. no. 074005.

[14] J Rabe, M. Meinke, M. Necker, and C. Stiller, "Lane-level map-matching based on optimization," in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2016, pp. 1155–1160.

[15] Y. Xing, C. Lv, L. Chen, H. Wang, H. Wang, D. Cao, E. Velenis, and F.-Y. Wang, "Advances in vision-based lane detection: Algorithms, integration, assessment, and perspectives on ACP-based parallel vision," *IEEE/CAA J. Automatica Sinica*, vol. 5, no. 3, pp. 645–661, May 2018.

[16] G. Liu, S. Li, and W. Liu, "Lane detection algorithm based on local feature extraction," in *Proc. Chin. Autom. Congr.*, Nov. 2013, pp. 59–64.

[17] H. Xuan, H. Liu, J. Yuan, and Q. Li, "Robust lane-mark extraction for autonomous driving under complex real conditions," *IEEE Access*, vol. 6, pp. 5749–5765, 2018.

[18] W. Li, Y. Guan, L. Chen, and L. Sun, "Millimeter-wave radar and machine vision-based lane recognition," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 32, no. 05, Jan. 2018, Art. no. 1850015.

[19] L. Li, W. Luo, and K. C. P. Wang, "Lane marking detection and reconstruction with line-scan imaging data," *Sensors*, vol. 18, no. 5, p. 1635, May 2018.

[20] R. Toledo-Moreo, D. Betaille, and F. Peyret, "Lane-level integrity provision for navigation and map matching with GNSS, dead reckoning, and enhanced maps," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 1, pp. 100–112, Mar. 2010.

[21] W. Yanwen, Z. Nan, Z. Tao, and Y. Wei, "Research of lane detection and tracking methods based on multi-sensor fusion," *Appl. Res. Comput.*, vol. 35, no. 2, p. 60, 2018.

[22] Y. Zihao and S. Rui, "Lane recognition method based on fully convolution neural network and conditional random fields," *Opto-Electron. Eng.*, vol. 46, no. 2, pp. 37–48, 2019.

[23] D. Neven, B. De Brabandere, S. Georgoulis, M. Proesmans, and L. Van Gool, "Towards end-to-end lane detection: An instance segmentation approach," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 286–291.

[24] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2015.

[25] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3431–3440.

[26] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2015, pp. 234–241.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[28] D. H. Ballard, "Generalizing the Hough transform to detect arbitrary shapes," *Pattern Recognit.*, vol. 13, no. 2, pp. 111–122, Jan. 1981.

[29] Y. Li, C. Lin, and W. Zhang, "Improved sparse least-squares support vector machine classifiers," *Neurocomputing*, vol. 69, nos. 13–15, pp. 1655–1658, Aug. 2006.

**WENXIN TENG** received the bachelor's degree in GIS from the University of Jinan, Jinan, China, in 2013, and the master's degree from the 3D Information Collection and Application Key Lab of Education Ministry, Capital Normal University, Beijing, China.

His major field was GIS, ITS, and vehicle navigation and image. He is interested in solving the map matching problems through deep learning.

**YANHUI WANG** was born in Henan, China. She received the bachelor's and master's degrees in aerospace engineering from Central South University, in 1998 and 2001, respectively, and the Ph.D. degree in LBS from the China University of Mining and Technology, Beijing, in 2005.

From July 2005 to December 2006, she worked as a Lecturer at the College of Resources Environment and Tourism, Capital Normal University, where she has been an Associate Professor, since January 2007. Her research fields are GIS application and LBS. She is interested in solving complex real-world positioning problems.

**BIBO YU** is currently pursuing the degree in geographic information science. His research has been concerned with the method and application of GIS in traffic scenes, traffic big data, and deep learning. He is interested in solving problems in lane-level positioning and high-precision map in automatic drive.

**JIAHAO LIU** received the B.E. degree in geographic information system from Hebei Agricultural University, Baoding, China, in 2018. He is currently pursuing the M.E. degree in geographic information system with the Beijing State Key Laboratory Incubation Base of Urban Environmental Processes and Digital Simulation, Capital Normal University, Beijing. His research interests include LBS, ITS, and deep learning and its application.