

# Detection and Navigation of Unmanned Vehicles in Wooded Environments Using Light Detection and Ranging Sensors

Zhiwei Zhang,<sup>1</sup> Jyun-Yu Jhang,<sup>2</sup> and Cheng-Jian Lin<sup>3\*</sup>

<sup>1</sup>School of Electronic Information Engineering, Hebei University of Technology, Tianjin 300401, China

<sup>2</sup>Department of Computer Science and Information Engineering,

National Taichung University of Science and Technology, Taichung 404, Taiwan

<sup>3</sup>Department of Computer Science & Information Engineering, National Chin-Yi University of Technology, Taichung 411, Taiwan

(Received July 3, 2023; accepted October 31, 2023)

**Keywords:** unmanned vehicle, LiDAR sensor, navigation control, artificial potential field algorithm

With the advancement of automatic navigation, navigation control has become an indispensable core technology in the movement of unmanned vehicles. In particular, research on navigation control in outdoor wooded environments, which are more complex, less controlled, and more unpredictable than indoor environments, has received widespread attention. To realize the movement control and obstacle avoidance of unmanned vehicles in unknown environments, in this study, we use light detection and ranging (LiDAR) sensors to sense the surrounding environment. By plane meshing the point cloud reflected from LiDAR, we can instantly establish feasible regions. At the same time, using the artificial potential field algorithm, a stable obstacle avoidance and navigation path is planned for use in an unknown environment. In an actual woods navigation experiment to evaluate our proposed LiDAR detection method, we used an independently developed unmanned vehicle with Ackermann steering geometry. Experimental results indicate that the proposed method can effectively detect obstacles. The accuracy requirement is within 30 cm from the navigation target, and the experimental results show that the average navigation success rate of the proposed method is as high as 85%. The experimental results demonstrate that the system can stably and safely navigate in scenarios with different unknown environments.

## 1. Introduction

One of the main goals of robotics research is to develop unmanned vehicles that can navigate autonomously in unknown environments.<sup>(1–3)</sup> In the autonomous navigation of unmanned vehicles, the perception of the environment is a crucial factor. Therefore, high-precision and high-reliability sensors are of great significance for unmanned vehicles.<sup>(4)</sup> Ortiz *et al.*<sup>(5)</sup> discussed the advantages and disadvantages of various external sensors. The camera is the most common sensor used to simulate the image perceived by the human eye. It can obtain detailed color and texture features from the environment, but it is easily affected by light and weather.<sup>(6)</sup>

---

\*Corresponding author: e-mail: [cjlin@ncut.edu.tw](mailto:cjlin@ncut.edu.tw)  
<https://doi.org/10.18494/SAM4688>

Radar has the advantages of long detection range and small size, but it has the disadvantage of being expensive. Moreover, the data must be preprocessed before the information obtained from the radar can be seen.<sup>(7)</sup> To avoid the above problems, Roriz *et al.*<sup>(8)</sup> used light detection and ranging (LiDAR) sensors for intelligent transportation systems. The advantages of LiDAR are its long detection distance and wide viewing angle. The data obtained by its sensing environment is also very accurate and not easily affected by changes in lighting. Although the amount of input data is huge, the stability of LiDAR exceeds that of other sensors.

Research on autonomous navigation is mainly divided into two types: map-based navigation<sup>(9–10)</sup> and mapless navigation.<sup>(11–14)</sup> For map-based navigation, simultaneous localization and mapping (SLAM) using LiDAR is a common method used to traverse the entire map, construct a global map, and plan a feasible path. Its disadvantage is that every time a new map is employed, it is necessary to rebuild the map. Therefore, its general applicability is poor. A mapless navigation method is commonly used to navigate through machine learning methods.<sup>(15)</sup> This method uses sensors to sense information about the surrounding environment. On the basis of the perceived information, machine learning technology is used to learn how to avoid obstacles and navigate. Commonly used mapless navigation learning methods are divided into supervised learning<sup>(16,17)</sup> and reinforcement learning.<sup>(18–20)</sup>

The supervised learning navigation and obstacle avoidance method involves collecting training data through sensors and labeling the correct answers, which are used to navigate and avoid obstacles. Hua *et al.*<sup>(21)</sup> proposed a supervised visual navigation method based on RGB-D sensors. They input the image into their proposed two-stage RGB-D semantic segmentation network to segment the feasible area and plan obstacle avoidance angles to achieve visual navigation. However, a disadvantage of the supervised navigation system is that the time cost of collecting data in the early stage is relatively high, and changes in the image light and bad weather will also affect the stability, making it easy to obtain incorrect results for unlearned data. The navigation and obstacle avoidance method of reinforcement learning<sup>(22)</sup> is usually used for problems where training data is not available. During the learning process, the robot interacts with the environment. Reward and punishment mechanisms are used to evaluate the robot's movement quality so that the robot gradually learns the desired movement direction. Lin *et al.*<sup>(23)</sup> used a type-2 fuzzy neural network with an improved particle swarm algorithm to solve the problem of the traditional particle swarm algorithm easily falling into a local optimal solution. Although reinforcement learning can successfully achieve navigation and obstacle avoidance, it may suffer as a result of differences between the simulated environment and the actual environment, resulting in poor performance in actual obstacle avoidance. Therefore, some scholars use fuzzy controllers and allow experts to define fuzzy rules for navigation with obstacle avoidance.<sup>(24)</sup> Lin *et al.*<sup>(25)</sup> proposed a wall-following strategy based on a fuzzy controller and defined fuzzy rules. They used the center of gravity method to resolve fuzzification to obtain the angle of navigation obstacle avoidance. However, because the fuzzy rules are designed by experts, the objects in the environment are too small to be detected.

In addition, some researchers<sup>(26,27)</sup> have used sensors to sense the surrounding environment to obtain information about obstacles in real time, and LiDAR sensors have been used to construct obstacles. The point cloud clustering method<sup>(26)</sup> is often used to cluster obstacle

information, but the number of point clouds is huge and the method has high time and computing costs. Therefore, the detection of feasible areas has been proposed, with the remaining point clouds considered as obstacles. Castejon *et al.*<sup>(27)</sup> proposed a threshold setting for the feasible areas of an outdoor area based on the traversal ability of the unmanned vehicle. They used LiDAR to obtain feasible areas and achieved good results in actual outdoor verification. However, the commonly used local path planning algorithms include the fuzzy logic method,<sup>(28)</sup> dynamic window method,<sup>(29)</sup> and artificial potential field method.<sup>(30)</sup> Among these methods, the artificial potential field method requires the least calculation, has the simplest model, and is the most widely used. Li *et al.*<sup>(31)</sup> proposed an improved artificial potential field method for local path planning, in which the gravitational and repulsive functions in the traditional artificial potential field method are modified. The obtained results show that the safe distance is increased and that the local minima can be bypassed to successfully reach the destination.

In this study, a LiDAR-sensor-based detection and navigation system for unmanned vehicles in feasible areas in wooded environments is proposed. It uses LiDAR to construct obstacle information in real time and artificial potential field algorithms to make obstacle avoidance plans. Finally, the proposed method is applied to various terrain scenarios. The major contributions of this study are as follows.

1. We propose a LiDAR-sensor-based detection and navigation system for unmanned vehicles.
2. A novel artificial potential field algorithm is used in the proposed system for planning a stable obstacle avoidance and navigation path in an unknown environment.
3. Experimental results indicate that the proposed LiDAR detection method can achieve an accuracy rate of 99.8% in a wooded environment.
4. The accuracy requirement is within 30 cm from the navigation target, and the experimental results show that the average navigation success rate of the proposed method reaches 85%.

The remainder of the paper is organized as follows. Section 2 presents the proposed LiDAR-sensor-based detection and navigation system. Section 3 introduces the experimental results of detection and navigation in a wooded environment. Section 4 provides the conclusions.

## 2. Materials and Methods

This section introduces the detection and navigation of unmanned vehicles in feasible areas in a wooded environment using LiDAR sensors, as shown in Fig. 1. In the system, we input the LiDAR point cloud and the target direction into the embedded system (Jetson AGX Xavier), and perform navigation and obstacle avoidance path planning using the artificial potential field algorithm. Finally, the rotation angle of the unmanned vehicle is output.

### 2.1 Embedded systems

For feasible area detection in an outdoor wooded environment, movement cannot be controlled by returning to the server under limited resources. Therefore, embedded systems are installed on unmanned vehicles. After the embedded system completes the calculation, it issues instructions to control the unmanned vehicle, thus eliminating the bandwidth and transmission

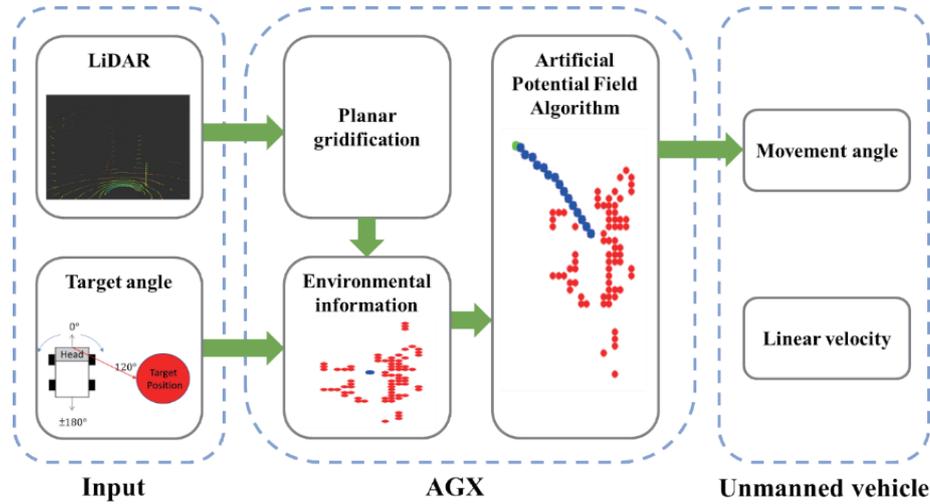


Fig. 1. (Color online) Overall architecture of system.

delays due to the transmission of large amounts of data transmission. In this study, the Jetson AGX Xavier embedded edge computing system was selected, which is a compact and energy-saving system with a hardware size of  $100 \times 87$  mm, a 512-core NVIDIA Volta™ GPU, 8 MB of L2 temporary storage, 4 MB of L3 temporary storage, and an 8-core ARM v8.2 64-bit CPU. The development kit uses 32 GB of double data rate synchronous dynamic random access memory (LPDDR4x).

## 2.2 LiDAR sensors

We use LiDAR sensors to sense environmental changes in real time. The VLP-16 Puck 3D LiDAR model shown in Fig. 2(a) is selected as our LiDAR. The horizontal measurement angle is  $360^\circ$ . The 16-line 3D LiDAR can reflect the spatial structure of the woods. The detection range of the LiDAR sensor is 50 cm to 100 m, as shown in Fig. 2(b). The image of the actual environment and the LiDAR scanning imaging image are shown in Figs. 2(c) and 2(d), respectively. In this study, we only consider the distances within 5 m, which is sufficient to make immediate judgments for obstacle avoidance.

## 2.3 Feasible area detection

In this subsection, we mainly explain how to use point clouds reflected from obstacles to detect feasible areas. The detection process can be divided into three steps: (1) determination of the height range of obstacle point clouds; (2) point cloud plane meshing; (3) obstacle expansion.

### 2.3.1 Determination of height range of obstacle point clouds

Since a large amount of point cloud information is collected by LiDAR, we use filtering to simplify the point cloud and reduce the number of subsequent calculations. The formula describing the filtering is

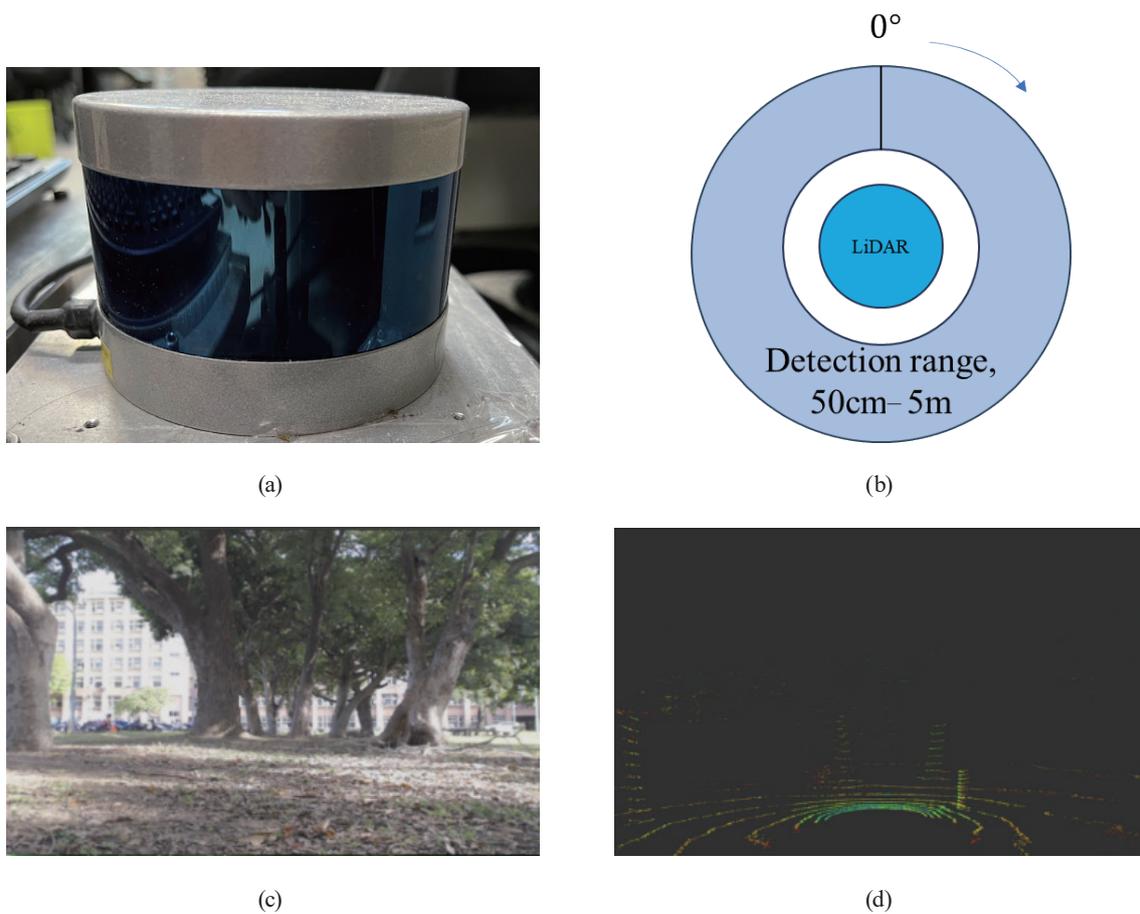


Fig. 2. (Color online) (a) VLP-16 LiDAR sensor, (b) LiDAR detection range, (c) image of actual environment, and (d) LiDAR scanning imaging image.

$$PC_{Reserve} = PC_{original} \mid PC_z \in [H_{min}, H_{max}], \quad (1)$$

where  $PC_{original}$  is the 3D point cloud before filtering and  $PC_{Reserve}$  is the remaining 3D point cloud after filtering. In this study, only point clouds whose height ( $PC_z$ ) falls between the greatest height ( $H_{max}$ ) and the least height ( $H_{min}$ ) are retained.  $H_{max}$  and  $H_{min}$  depend on the height and spanning capability of the unmanned vehicle. In this study, we set  $H_{max}$  and  $H_{min}$  to 50 and 20 cm, respectively.

### 2.3.2 Point cloud plane meshing

The amount of point cloud data is still very large even after filtering out non-obstacles. We propose a point cloud plane meshing method to effectively process the point cloud data. As shown in Fig. 3, the point cloud is divided into a  $50 \times 50$  flat grid space based on distance. On the basis of the radius of 5 m detected by LiDAR, the plane grid space can be divided into a

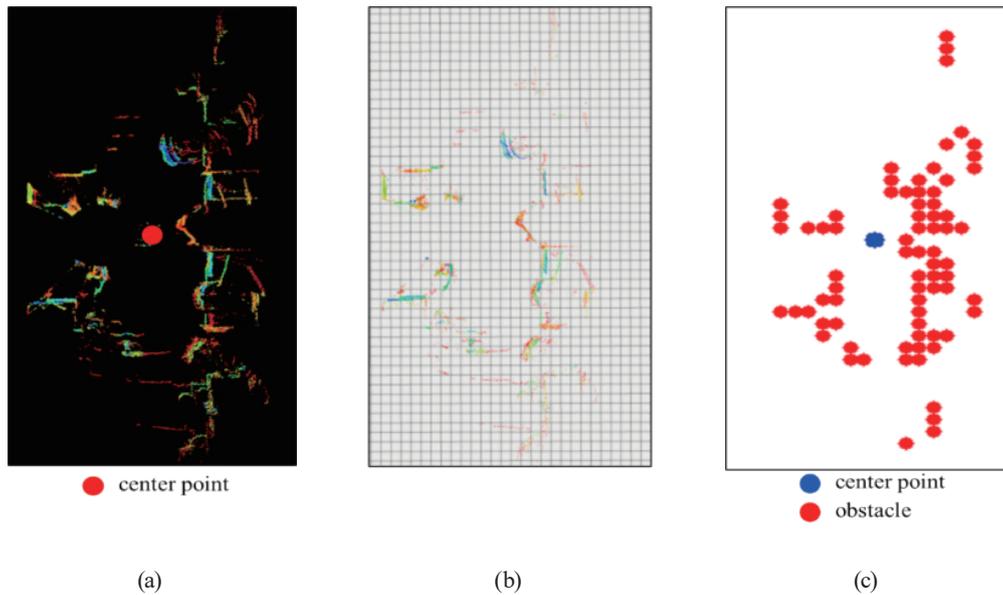


Fig. 3. (Color online) VLP16 LiDAR sensor: (a) LiDAR detection range, (b) image of actual environment, and (c) LiDAR scanning image.

square of  $20 \times 20 \text{ cm}^2$ . The 3D point cloud data is mapped into a flat grid space in accordance with the distance and is presented quantitatively. Owing to the characteristics of LiDAR, only when the laser hits an obstacle is it reflected back to form a point cloud. Therefore, the number of point clouds can reflect the feasible areas within the planar grid.

As shown in Fig. 3(c), the determination of feasible areas and obstacles is based on the number of point clouds, where the following formula is used.

$$Grid_{(i,j)} = \begin{cases} 0, & \text{if } Grid_{(i,j)} < P_{th} \\ 1. & \text{if } Grid_{(i,j)} > P_{th} \end{cases} \quad (2)$$

In this study, the point cloud number threshold ( $P_{th}$ ) is set to 15. When the number of point clouds in the plane grid is less than the threshold, that part of the grid is judged to be a feasible area, otherwise it is an infeasible area or obstacle. Therefore, unmanned vehicles can use the planar grid to instantly obtain the feasible areas of the surrounding environment.

### 2.3.3 Obstacle expansion

Owing to the blind spot of the LiDAR sensor, it cannot detect obstacles within a radius of 50 cm from the center of the sensor. To avoid the risk of collision due to this limitation, obstacles are expanded to be larger than their actual size to ensure that the unmanned vehicle does not fail to detect them, as shown in Fig. 4.

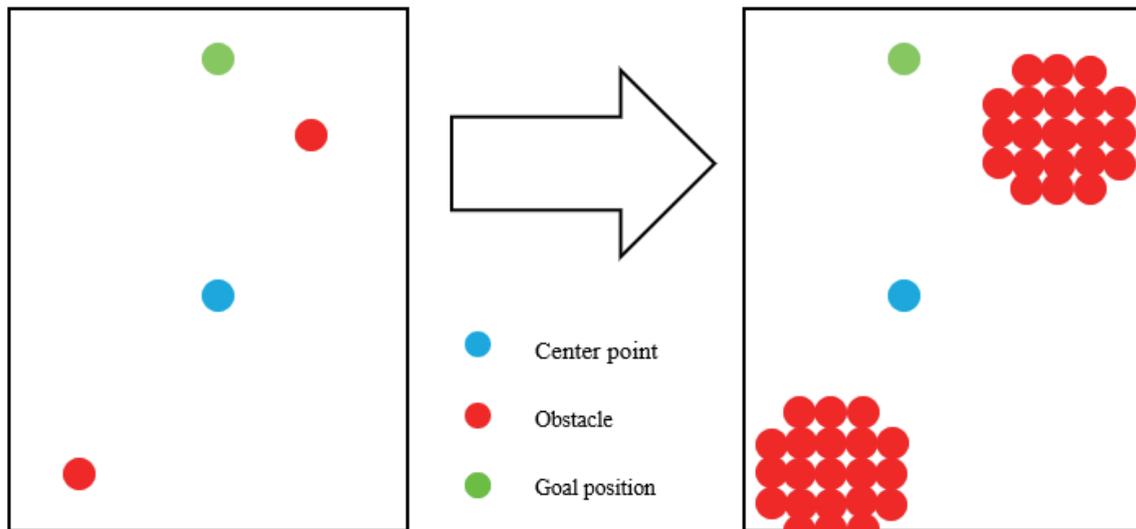


Fig. 4. (Color online) Expansion of obstacles to avoid limitation of blind spot of LiDAR sensor.

## 2.4 Creation of target information

To obtain the position of a target, the angle between the front of the unmanned vehicle and the target must be calculated, as shown in Fig. 5. Subsequently, the following formulas are used to convert the target angle into 2D  $(x, y)$  coordinates:

$$x = r(\cos\theta), \quad (3)$$

$$y = r(\sin\theta), \quad (4)$$

where  $x$  and  $y$  represent the coordinates of the target point, and  $r$  denotes the distance between the target and the unmanned vehicle.

## 2.5 Artificial potential field

In the artificial potential field method, the environment is transformed into a virtual energy field. In this virtual energy field, the goal point exerts an attractive force on the unmanned vehicle, whereas obstacles generate a repulsive force. Combining the attractive and repulsive fields results in a resultant force field that determines the path of the unmanned vehicle, allowing it to navigate around obstacles and reach the target point, as shown in Fig. 6.

The artificial force acting at the position within the unmanned vehicle's space can be calculated as

$$F(p) = -\nabla G(p), \quad (5)$$

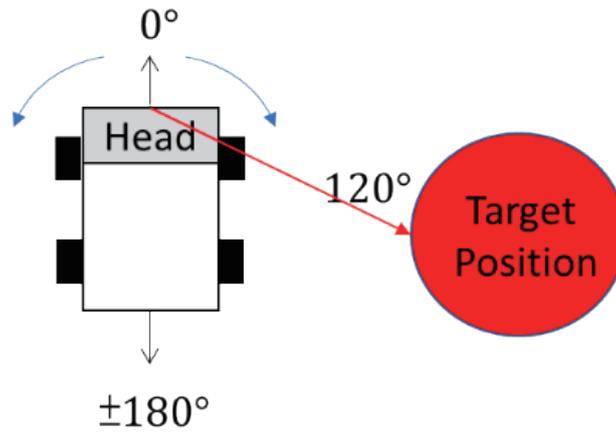


Fig. 5. (Color online) Angle between target and front of unmanned vehicle.

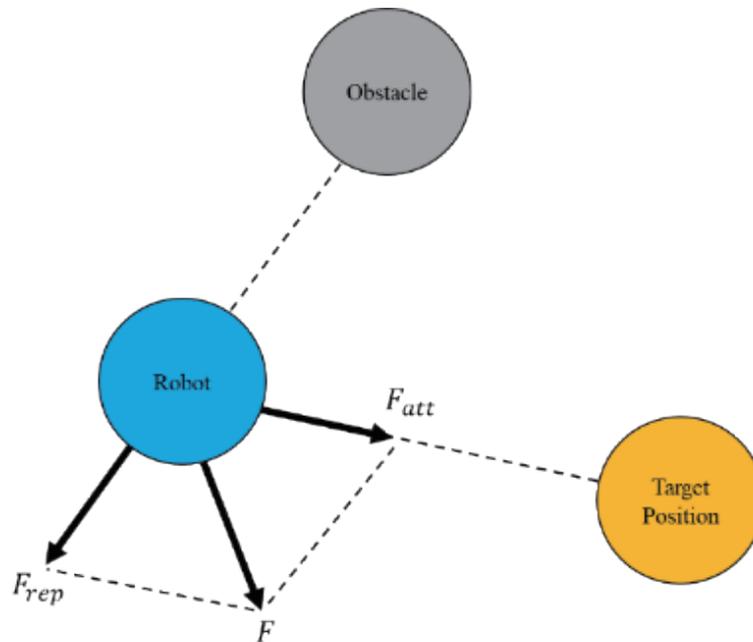


Fig. 6. (Color online) Combination of the attractive and repulsive fields in artificial potential field.

where  $F(p)$  is the resultant force and  $\nabla G(p)$  is the gradient vector of  $G$  at the unmanned vehicle position  $p = (x, y)$ . The unmanned vehicle experiences the effect of the potential field. This artificial potential field is computed by summing the attractive force  $F_{att}(p)$  from the goal and the repulsive force  $F_{rep}(p)$  from each obstacle:

$$G(p) = F_{att}(p) + F_{rep}(p), \quad (6)$$

$$F_{att}(p) = \frac{1}{2} \alpha p - p_g, \quad (7)$$

where  $\alpha$  is the scaling factor of the attractive force and  $p - p_g$  is the Euclidean distance between the unmanned vehicle position  $p$  and the target position  $p_g$ . The repulsive field function  $F_{rep}(p)$  is

$$F_{rep}(p) = \begin{cases} \frac{1}{2} \beta \left( \frac{1}{D(p, p_o)} - \frac{1}{D_{th}} \right)^2, & \text{if } D(p, p_o) \leq D_{th} \\ 0, & \text{if } D(p, p_o) > D_{th} \end{cases} \quad (8)$$

where  $\beta$  is the scaling factor of the repulsive forces,  $D(p, p_o)$  is the distance between the unmanned vehicle and the obstacle,  $D_{th}$  is the threshold of the range of the effect of the obstacle on the unmanned vehicle. In this study, this threshold is set to 12 grids. That is, the repulsive force is generated only when the obstacle appears within the threshold.

In this study, the artificial potential field algorithm is employed to plan obstacle avoidance navigation paths. Figure 7 shows path planning using the artificial potential field method. The red points represent obstacles, which exert repulsive forces, detected by the autonomous vehicle. The green point denotes the target, which exerts an attractive force. By constructing the resultant of the attractive and repulsive forces in the environment, the unmanned vehicle can move toward the target while avoiding obstacles. The blue points represent the path planned by the artificial potential field algorithm, which is also a representation of the resultant force.

### 3. Experimental Results

This experiment is explained in three parts. In Sect. 3.1, we describe the Ackermann unmanned vehicle designed in this study. In Sect. 3.2, we evaluate the performance of the proposed method through three virtual scenarios. In Sect. 3.3, we place the unmanned vehicle in a real environment to validate its performance. The parameter settings of this experiment are shown in Table 1, which includes the LiDAR detection range, the number and size of grids, the point cloud threshold, the obstacle height, and the expansion coefficient.

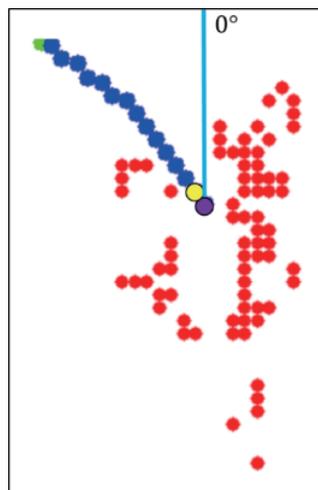


Fig. 7. (Color online) Path planning using artificial potential field method.

Table 1  
Experimental parameter settings.

LiDAR detection range	50 cm–5 m
Number of grids	50 × 50
Grid size	20 × 20 cm <sup>2</sup>
Point cloud threshold	>15
Obstacle height range	20–50 cm
Obstacle expansion coefficient	10

### 3.1 Newly developed Ackermann steering system for unmanned vehicle

The developed unmanned vehicle adopts the Ackermann steering system and has the advantages of high stability of movement and small turning radius. Two important control parameters in unmanned vehicle control are linear velocity (m/s) and angular velocity (rad/s). The unmanned vehicle detects information in the environment through LiDAR, and the information is input to the AGX edge computing device for inference. Finally, the linear and angular velocities are output to control the unmanned vehicle to achieve the functions of navigation and obstacle avoidance. The length, width, and height of the designed unmanned vehicle are 67, 40, and 35 cm, respectively, as shown in Fig. 8.

### 3.2 Navigation simulated in virtual scenarios

We demonstrated the proposed navigation algorithm in three virtual scenarios. We also compared the proposed method with a reinforcement learning controller<sup>(23)</sup> and a fuzzy controller<sup>(25)</sup> to verify its performance. Three different virtual scenarios, a maze map, an irregular obstacle map, and a sharp-corner map, were constructed for simulation and validation, as shown in Fig. 9.

The results of the navigation simulation for the virtual scenarios are shown in Fig. 10. For Scenario A, the reinforcement learning and fuzzy controllers required a longer time and path than the proposed method to reach the destination, because the navigation mechanism used movement along this obstacle. In Scenarios B and C, all three methods successfully reached the destination. However, the proposed artificial potential field method relies on the combined resultant force calculated from attraction and repulsion. As an unmanned vehicle moves, such forces continuously change, resulting in larger path oscillations and less smooth trajectories during movement. The movement distance and time of the path for each scenario and method are shown in Table 2. Although the proposed method has a less smooth path, it is superior to the other methods in terms of moving distance and time.

### 3.3 Unmanned vehicle navigation in real environments

To evaluate the obstacle avoidance and navigation capabilities of the proposed Ackermann unmanned vehicle, and verify the stability of the proposed method, we performed experiments in two different outdoor wooded environments.

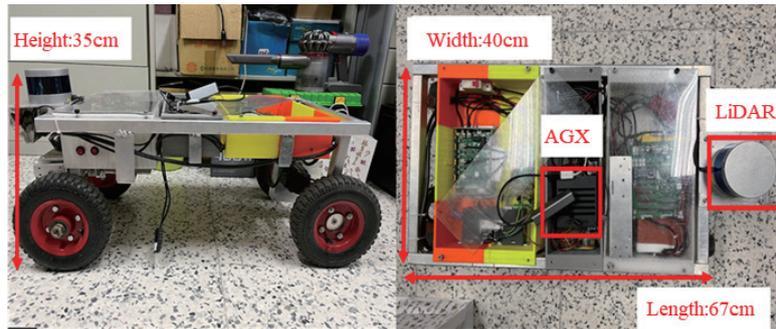


Fig. 8. (Color online) Our developed Ackermann unmanned vehicle.

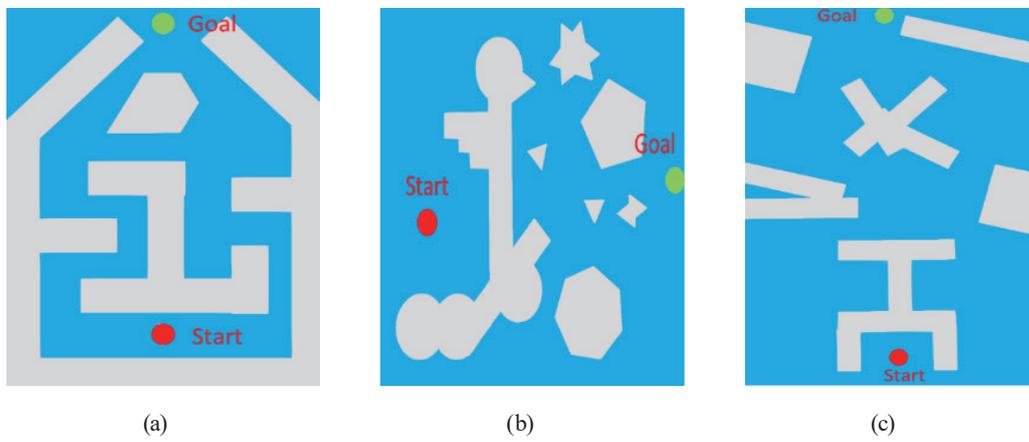


Fig. 9. (Color online) Virtual verification environment: (a) maze map, (b) irregular obstacle map, (c) and sharp-corner map.

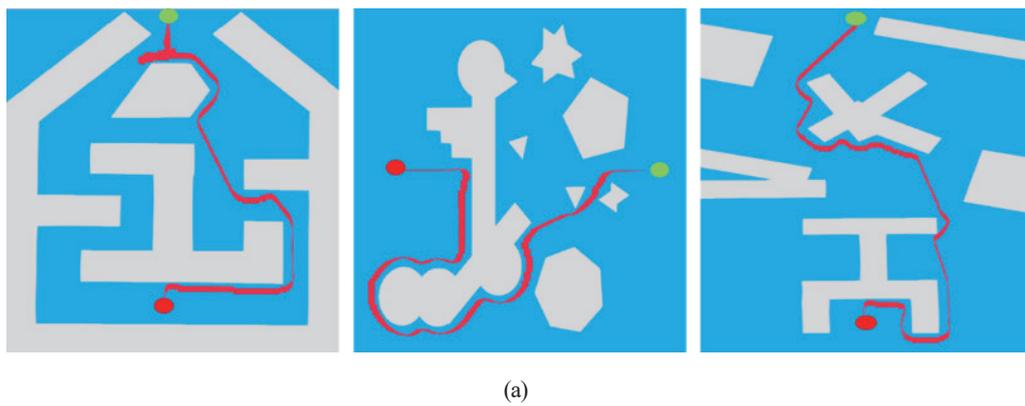


Fig. 10. (Color online) Results of navigation simulation: (a) reinforcement learning controller, (b) fuzzy controller, and (c) proposed method.

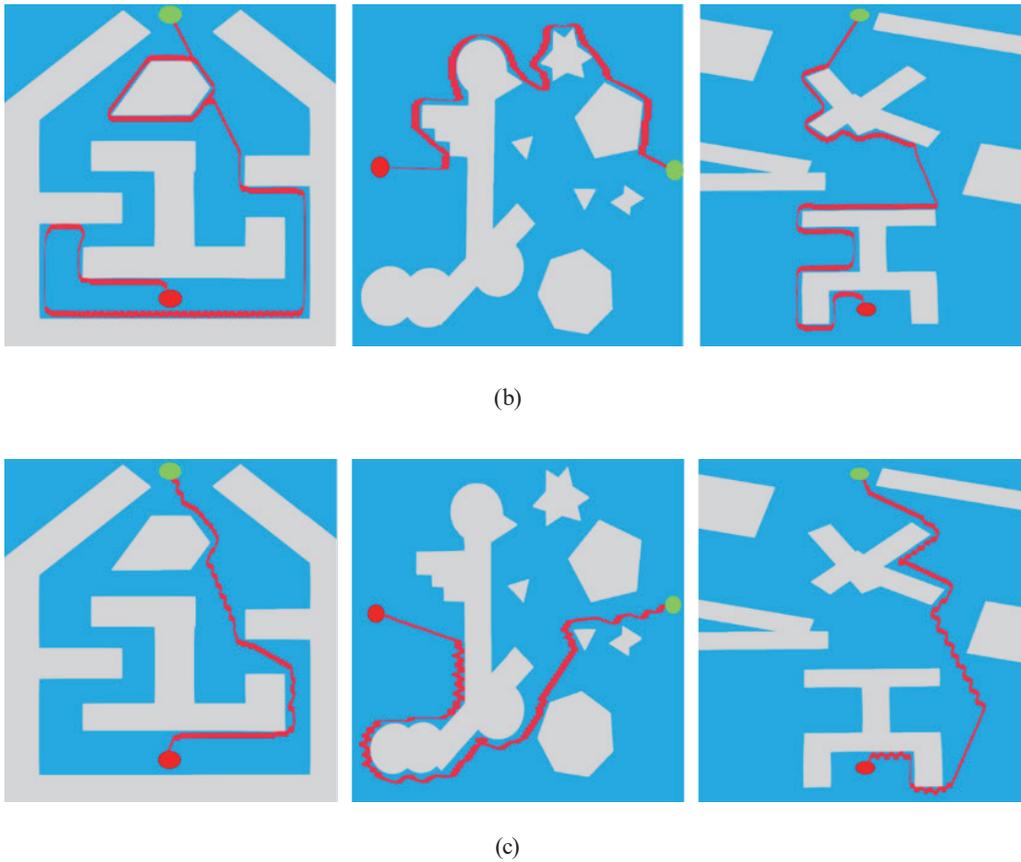


Fig. 10. (Continued) (Color online) Results of navigation simulation: (a) reinforcement learning controller, (b) fuzzy controller, and (c) proposed method.

Table 2  
Navigation performance in three virtual scenarios.

Method	Virtual scenario	Distance (m)	Time (s)
Proposed method		220.5	20.5
Reinforcement learning controller <sup>(23)</sup>	A	263.5	24.7
Fuzzy controller <sup>(25)</sup>		579.2	46.6
Proposed method		289.5	27.3
Reinforcement learning controller <sup>(23)</sup>	B	271.7	25.6
Fuzzy controller <sup>(25)</sup>		245.6	22.6
Proposed method		312.2	29.1
Reinforcement learning controller <sup>(23)</sup>	C	332.9	31.3
Fuzzy controller <sup>(25)</sup>		370.4	34.2

### 3.3.1 Wooded environment A

Details of wooded environment A are shown in Fig. 11. In this environment, there are many slender trees with a spacing of 1 to 1.2 m between them. We chose a destination that was surrounded by trees to verify that the proposed method can effectively navigate around obstacles and reach the destination. The navigation path of the unmanned vehicle is shown in Fig. 12,



Fig. 11. (Color online) Wooded environment A.

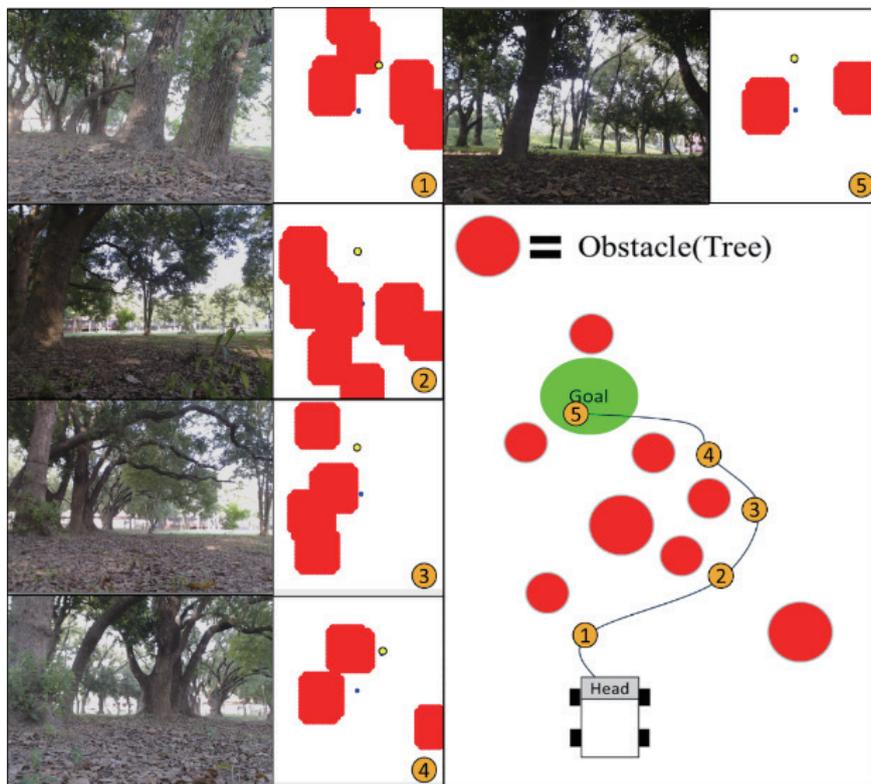


Fig. 12. (Color online) Navigation path of unmanned vehicle in wooded environment A.

where the blue point represents the current position of the unmanned vehicle and the yellow point represents the next position planned using the artificial potential field method. It can be observed that the unmanned vehicle successfully avoided obstacles and reached the destination.

### 3.3.2 Wooded environment B

Wooded environment B also contains many side-by-side and sparse trees, as shown in Fig. 13. In this experiment, we set a more distant destination to evaluate the stability of long-distance navigation. The path of the navigation experiment is shown in Fig. 14. Initially, the unmanned vehicle proceeded directly toward the destination as there were no obstacles in its vicinity. When the unmanned vehicle encountered obstacles, it avoided them and navigated through the trees to reach the destination.



Fig. 13. (Color online) Wooded environment B.

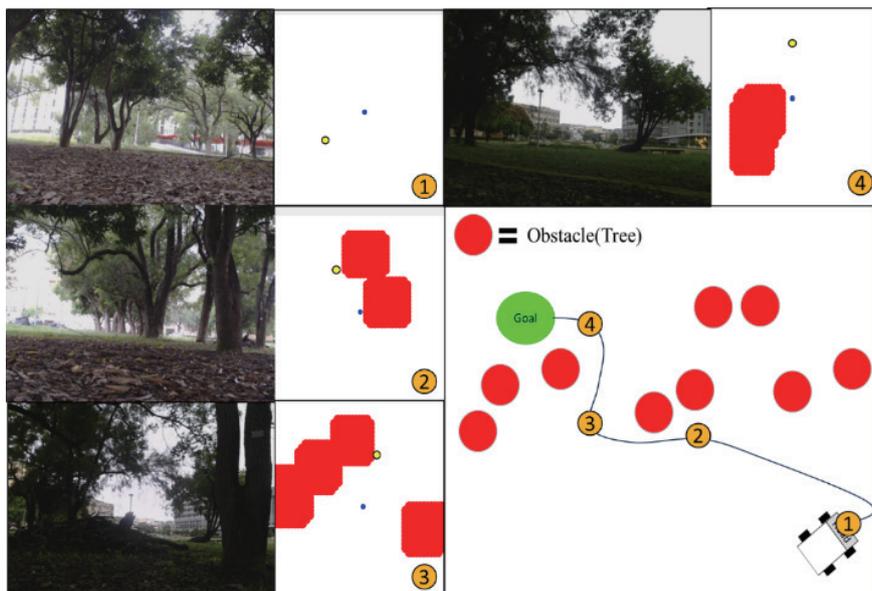


Fig. 14. (Color online) Navigation path of unmanned vehicle in wooded environment B.

### 3.3.3 Experimental evaluation of stability

We next conducted navigation experiments in wooded environments A and B, with five trials for each environment. The initial and target points were the same for each trial, and successful navigation was defined as reaching the destination without colliding with obstacles and remaining within a 30 cm margin of error. The results are shown in Table 3. It can be observed that the error from the destination for the proposed method remained within 20 cm. However, in wooded environment B, the distribution of trees is less ordered, leading to misjudgments by the unmanned vehicle regarding feasible areas and collisions with obstacles.

The LiDAR sensor of the unmanned vehicle has blind-spot regions with a radius of 50 cm, meaning that obstacles in this area cannot be effectively detected. In this study, we utilized seven different directional sensing angles, as illustrated in Fig. 15, to ensure that the navigation process avoided blind-spot regions. The sensing distances for each direction during navigation were also recorded and are shown in Fig. 16. As also shown in the figure, the sensing distance for each angle remained at least 1 m. This indicates that during obstacle avoidance in the two real-world environments, the unmanned vehicle moved within a safe range.

To verify the navigation performance of the proposed method and other methods, we used the RGB-D semantic segmentation network,<sup>(21)</sup> the reinforcement learning controller,<sup>(23)</sup> and the fuzzy controller<sup>(25)</sup> for comparison. As shown in Table 4, the performance of the image-based RGB-D semantic segmentation network method was unsatisfactory in wooded environments A and B; the complexity of the environments made it challenging for it to effectively differentiate obstacles. For the reinforcement learning and fuzzy controller methods, the number of sensing

Table 3  
Results of experimental stability evaluation.

	Target distance	Number of trials	Number of successful trials	Success rate (%)	Average error (m)	Moving time (s)
Wooded environment A	7 m ahead, 1 m to right	5	5	100	0.13	11.3
Wooded environment B	2 m behind, 19 m to left	5	4	80	0.29	25.3

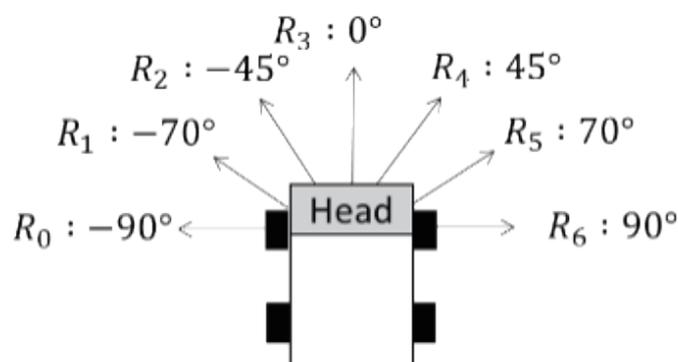


Fig. 15. Sensing angles in seven different directions for unmanned vehicle.

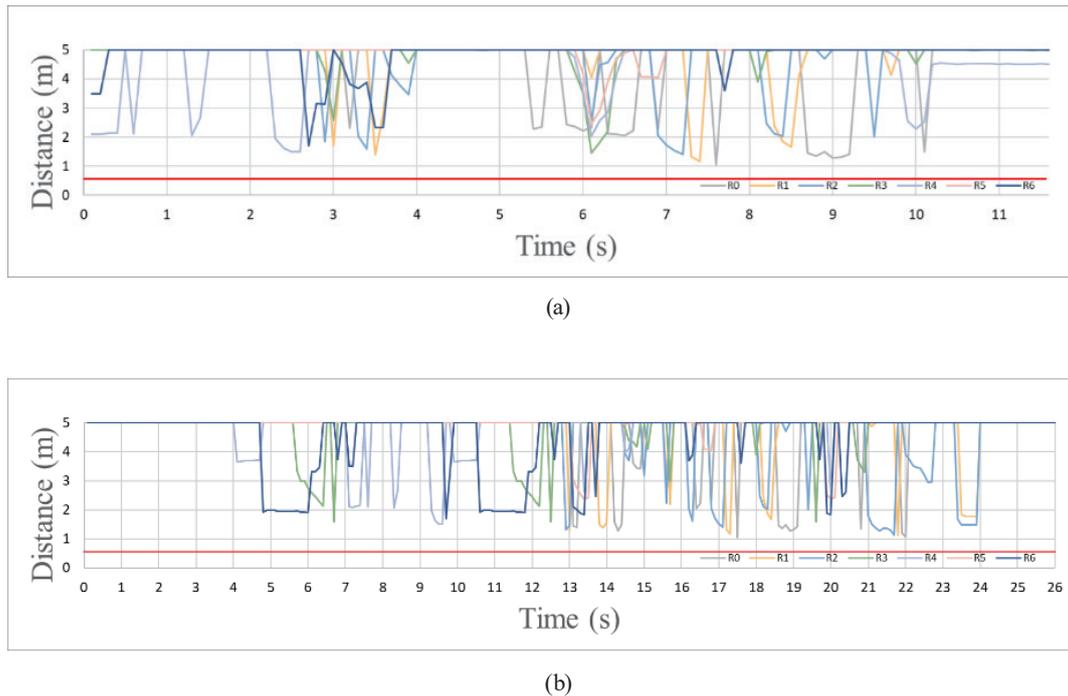


Fig. 16. (Color online) Obstacle distance sensing: (a) wooded environment A and (b) wooded environment B.

Table 4  
Performances of proposed and other methods.

Wooded environment A	Number of trials	Number of successful trials	Moving time (s)
RGB-D semantic segmentation network <sup>(21)</sup>	5	4	12
Reinforcement learning controller <sup>(23)</sup>	5	3	13
Fuzzy controller <sup>(25)</sup>	5	3	15
Proposed method	5	5	11.3
Wooded environment B	Number of trials	Number of successful trials	Moving time (s)
RGB-D semantic segmentation network <sup>(21)</sup>	5	3	26
Reinforcement learning controller <sup>(23)</sup>	5	4	27
Fuzzy controller <sup>(25)</sup>	5	3	27
Proposed method	5	4	25.3

angles was limited to four, which resulted in the inability to accurately detect slender trees during navigation and obstacle avoidance. The proposed method achieved the highest success rate in the navigation and obstacle avoidance experiments, effectively navigating through both wooded environments to reach the destination.

#### 4. Conclusions

We presented a novel and widely applicable method for navigation with obstacle avoidance in outdoor wooded environments. The proposed method can be directly applied to unmanned vehicles without an extensive training time. The designed feasible area detection method

effectively processes large volumes of LiDAR point cloud data, enabling real-time navigation control. In various experimental environments, the proposed method successfully completed navigation tasks and reached specified destinations with errors relative to the targets consistently within 30 cm. The average navigation success rate was 85%. The experimental results demonstrated that the proposed navigation method is suitable for complex outdoor wooded environments.

## References

- 1 S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-E. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. V. Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney: *J. Field Robot.* **23** (2006) 9. <https://doi.org/10.1002/rob.20147>
- 2 U. Nunes, C. Laugier, and M. M. Trivedi: *IEEE Trans. Intell. Transp. Syst.* **10** (2009) 3. <https://doi.org/10.1109/TITS.2009.2037205>
- 3 A. Broggi, L. Bombini, S. Cattani, P. Cerri, and R. I. Fedriga: 2010 IEEE Intelligent Vehicles Symp. (IEEE, 2010) 500. <https://doi.org/10.1109/IVS.2010.5548026>
- 4 S. Baruah, P. Lee, P. Sarathy, and M. Wolf: *Proc. IEEE (IEEE, 2020)* 1196–1207. <https://doi.org/10.1109/JPROC.2020.2978661>
- 5 F. M. Ortiz, M. Sammarco, L. H. M. K. Costa, and M. Detyniecki: *IEEE Trans. Intell. Veh.* **8** (2023) 1. <https://doi.org/10.1109/TIV.2022.3182218>
- 6 V. R. Kumar, C. Eising, C. Witt, and S. K. Yogamani: *IEEE Trans. Intell. Transp. Syst.* **24** (2023) 4. <https://doi.org/10.1109/TITS.2023.3235057>
- 7 O. Schumann, C. Wohler, M. Hahn, and J. Dickmann: 2017 Sensor Data Fusion: Trends Solutions Applications (IEEE, 2017) 1. <https://doi.org/10.1109/SDF.2017.8126350>
- 8 R. Roriz, J. Cabral, and T. Gomes: *IEEE Trans. Intell. Transp. Syst.* **23** (2022) 7. <https://doi.org/10.1109/TITS.2021.3086804>
- 9 Y. Liu and J. Miura: *IEEE Access* **9** (2021) 23772. <https://doi.org/10.1109/ACCESS.2021.3050617>
- 10 J. He, M. Li, Y. Wang, and H. Wang: *IEEE Sensors J.* **23** (2023) 12. <https://doi.org/10.1109/JSEN.2023.3270534>
- 11 A. Devo, G. Mezzetti, G. Costante, M. L. Fravolini, and P. Valigi: *IEEE Trans. Robot.* **36** (2020) 5. <https://doi.org/10.1109/TRO.2020.2994002>
- 12 I. Kim, S. H. Nengroo, and D. Har: 2021 5th Int. Conf. on Electronics, Communication and Aerospace Technology (IEEE, 2021) 148. <https://doi.org/10.1109/ICECA52323.2021.9675949>
- 13 B. Tian, V. A. Shim, M. Yuan, C. Srinivasan, H. Tang, and H. Li: 2013 IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IEEE, 2013) 1562. <https://doi.org/10.1109/IROS.2014.6942923>
- 14 G. Oriolo, G. Ulivi, and M. Vendittelli: *IEEE Trans. Syst. Man Cybern. Syst. B* **28** (1998) 3. <https://doi.org/10.1109/3477.678626>
- 15 Y. Tang, C. Zhao, J. Wang, C. Zhang, Q. Sun, W. X. Zheng, W. Du, F. Qian, and J. Kurths: *IEEE Trans. Neural Netw. Learn. Syst.* (2022) 1. <https://doi.org/10.1109/TNNLS.2022.3167688>
- 16 V. R. Nagarajan and P. Singh: 2021 8th Int. Conf. Smart Computing and Communications (IEEE, 2021) 275. <https://doi.org/10.1109/ICSCC51209.2021.9528162>
- 17 L. M. Lorigo, R. A. Brooks, and W. E. L. Grimsou: *Proc. 1997 IEEE/RSJ Int. Conf. Intelligent Robot and Systems. Innovative Robotics for Real-World Applications (IEEE, 1997)* 373–379. <https://doi.org/10.1109/IROS.1997.649086>
- 18 K. Lobos-Tsunekawa, F. Leiva, and J. Ruiz-del-Solar: *IEEE Robot. Autom. Lett.* **3** (2018) 4. <https://doi.org/10.1109/LRA.2018.2851148>
- 19 Z. Liu, Q. Liu, L. Tang, K. Jin, H. Wang, M. Liu, and H. Wang: *IEEE Trans. Autom. Sci. Eng.* **19** (2022) 4. <https://doi.org/10.1109/TASE.2021.3114327>
- 20 W. Zhang, N. Liu, and Y. Zhang: *IEEE Robot. Autom. Lett.* **6** (2021) 2. <https://doi.org/10.1109/LRA.2021.3061305>
- 21 M. Hua, Y. Nan, and S. Lian: 2019 IEEE/CVF Int. Conf. Computer Vision Workshop (IEEE, 2019) 886. <https://doi.org/10.1109/ICCVW.2019.00117>
- 22 B. R. Kiran, I. Sobh, V. Talpaert, P. Mannion, A. A. A. Sallab, S. Yogamani, and P. Pérez: *IEEE Trans. Intell. Transp. Syst.* **23** (2022) 6. <https://doi.org/10.1109/TITS.2021.3054625>
- 23 C. J. Lin, B. H. Chen, and J. Y. Jhang: *IEEE Access* **11** (2023) 107917. <https://doi.org/10.1109/ACCESS.2023.3315741>

- 24 G. Feng: IEEE Trans. Fuzzy Syst. **14** (2006) 5. <https://doi.org/10.1109/TFUZZ.2006.883415>
- 25 C. J. Lin, M. Y. Chang, K. H. Tang, and C. K. Huang: Sens. Mater. **35** (2023) 781. <https://doi.org/10.18494/SAM4120>
- 26 J. Duan and A. Valentyna: 2015 Int. Conf. Control, Automation and Information Sciences (IEEE, 2015) 137. <https://doi.org/10.1109/ICCAIS.2015.7338649>
- 27 C. Castejon, D. Blanco, and L. Moreno: IEEE Trans. Syst. Man Cybern. Syst. Part A: Syst. Humans **38** (2008) 1. <https://doi.org/10.1109/TSMCA.2007.904786>
- 28 M. Dirik, A. F. Kocamaz, and E. Dönmez: 2017 25th Signal Processing and Communications Applications Conf. (IEEE, 2017) 1. <https://doi.org/10.1109/SIU.2017.7960644>
- 29 L. Xiang, X. Li, H. Liu, and P. Li: IEEE Open J. Intell. Transp. Syst. **3** (2022) 1. <https://doi.org/10.1109/OJITS.2021.3137931>
- 30 P. Wu, F. Gao, and K. Li: IEEE Access **10** (2022) 4359. <https://doi.org/10.1109/ACCESS.2022.3141075>
- 31 Y. Li, B. Tian, Y. Yang, and C. Li: 2022 IEEE 6th Information Technology and Mechatronics Engineering Conf. (IEEE, 2022) 91–94. <https://doi.org/10.1109/ITOEC53115.2022.9734712>

## About the Authors



**Zhiwei Zhang** received her B.S. and Ph.D. degrees in electronic engineering from Hebei University of Technology, Tianjin, China. She is currently a lecturer of the School of Electronic Information Engineering, Hebei University of Technology, Tianjin, China. Her research interests include computer vision, pattern recognition, and machine learning. ([zhangzhiwei@hebut.edu.cn](mailto:zhangzhiwei@hebut.edu.cn))



**Jyun-Yu Jhang** received his B.S. and M.S. degrees from the Department of Computer Science and Information Engineering, National Chin-Yi University of Technology, Taichung, Taiwan, in 2013 and 2015, respectively, and his Ph.D. degree from the Institute of Electrical and Control Engineering in 2021. He is currently an assistant professor of the Department of Computer Science & Information Engineering, National Taichung University of Science and Technology, Taichung, Taiwan. His research interests include fuzzy logic theory, type-2 neural fuzzy systems, evolutionary computation, machine learning, computer vision, and their applications. ([jyjhang@nuct.edu.tw](mailto:jyjhang@nuct.edu.tw))



**Cheng-Jian Lin** received his B.S. degree in electrical engineering from Ta Tung Institute of Technology, Taipei, Taiwan, R.O.C., in 1986 and his M.S. and Ph.D. degrees in electrical and control engineering from National Chiao Tung University, Taiwan, R.O.C., in 1991 and 1996, respectively. He is currently a chair professor of the Department of Computer Science and Information Engineering, National Chin-Yi University of Technology, Taichung, Taiwan, R.O.C. His research interests are machine learning, pattern recognition, intelligent control, image processing, intelligent manufacturing, and evolutionary robots. ([cjlin@nuct.edu.tw](mailto:cjlin@nuct.edu.tw))