

Research Article

A Real-Time Obstacle Avoidance Method for Autonomous Vehicles Using an Obstacle-Dependent Gaussian Potential Field

Jang-Ho Cho ¹, Dong-Sung Pae ¹, Myo-Taeg Lim ¹ and Tae-Koo Kang ²

¹School of Electrical Engineering, Korea University, Seoul, Republic of Korea

²Department of Information and Telecommunication Engineering, Sangmyung University, Cheonan, Republic of Korea

Correspondence should be addressed to Myo-Taeg Lim; mhim@korea.ac.kr and Tae-Koo Kang; tkkang@smu.ac.kr

Received 31 January 2018; Accepted 9 July 2018; Published 23 August 2018

Academic Editor: Jose E. Naranjo

Copyright © 2018 Jang-Ho Cho et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A new obstacle avoidance method for autonomous vehicles called *obstacle-dependent Gaussian potential field* (ODG-PF) was designed and implemented. It detects obstacles and calculates the likelihood of collision with them. In this paper, we present a novel attractive field and repulsive field calculation method and direction decision approach. Simulations and the experiments were carried out and compared with other potential field-based obstacle avoidance methods. The results show that ODG-PF performed the best in most cases.

1. Introduction

Autonomous vehicle systems need path planning, indoor and outdoor localization, obstacle avoidance, object detection, and classification of cars, humans, pets, and traffic signs, signals, etc.

Path planning is one of the fundamental technologies in automotive vehicles. There are two categories in path planning, namely, global path planning and local path planning. Global path planning needs information from a geographic information system (GIS) and global localization, while local path planning needs relative position and obstacle avoidance. There are many algorithms for both global and local path planning. In global path planning, graphical methods such as the Voronoi algorithm [1–4], the cell decomposition algorithm [5–7], and fast marching algorithm [4, 8, 9] are used to develop a graph and then find the shortest path to the goal.

Obstacle avoidance is one of the essential technologies in local path planning and one of the critical technologies that guarantees human and vehicle safety. For decades, many researches have been performed on this theme and many methods have been developed, a few of which have been implemented in real systems. In order to avoid collision with obstacles, a robot needs not only to detect obstacles but also

to recalculate the detouring path and to steer itself toward a safe and efficient path in real time.

Khatib [10] formulated the concept of artificial potential field method (PFM) for implementing an obstacle avoidance system. This method consisting of an attractive field and a repulsive field is simple and computationally light. The two different fields are calculated and summed up, after which the robot follows the direction of the calculated vector. Avoiding obstacles is achieved by the repulsive field and heading toward the goal by the attractive field. The concept of artificial potential fields can also be used as a global path planner [11].

The bug algorithm was introduced by Lumelsky and Stepanov [12], in which a vehicle runs along the wall of an obstacle whenever it meets one. If the direction toward the goal is available again, the vehicle leaves the obstacle and goes toward the goal.

Borenstein and Koren [13] presented a different method to avoid obstacles called the vector field histogram method (VFH), and Ulrich and Borenstein [14] improved the method by considering the robots size and choosing a safe and efficient angle (VFH+: later upgraded to VFH* [15]).

The dynamic window approach (DWA) using the vehicle's dynamics was proposed in [16]. This method consists of three velocity-angular velocity windows: the first window is the vehicles' minimum to maximum velocity-angular velocity

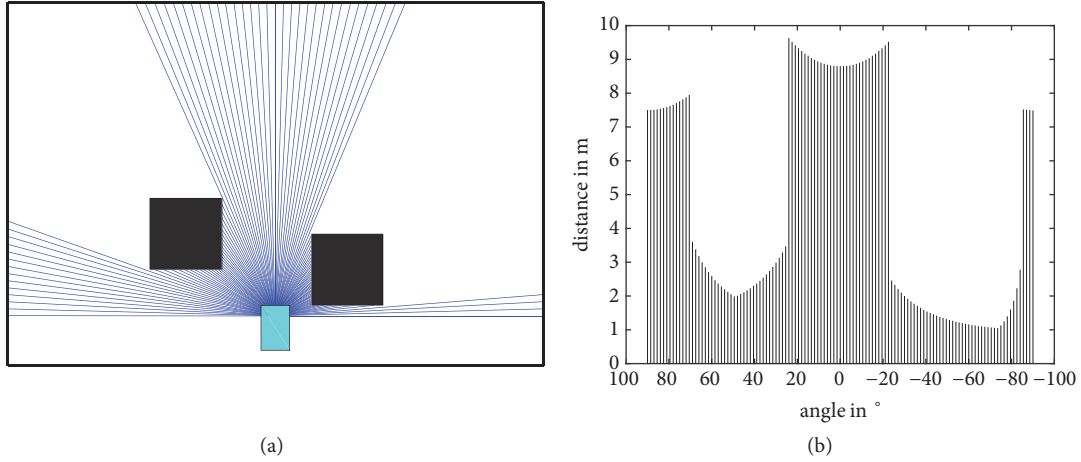


FIGURE 1: (a) Illustration of the laser range finder and (b) corresponding sensor data (distance versus angle).

window, the second is an obstacle-free zone in the first window, and the last one is the available velocity-angular velocity window feasible from the current velocity-angular velocity pair. In the intersection of the three windows, DWA chooses a velocity-angular velocity pair maximizing the objective function.

Anderson et al. [17] used model predictive control (MPC) to develop an optimal control-based path planning method that uses a relatively new control method to predict vehicle movement. The main drawback of this approach is that it is basically a nonlinear optimal control problem, and even when the problem can be linearized, finding the optimal path in real time is not easy. This is highly dependent on CPU performance.

The nearness diagram (ND) method was proposed by Minguez and Montano [18], who classified real situations into five categories, each with a different decision-making procedure. Later on, this method was upgraded by several other researchers [19–21].

The follow-the-gap method (FGM) was developed by Sezer and Gokasan [22], in which the angle of maximum gap is found and then the weighted average of the angle and the angle toward the goal are calculated.

Park et al. [23] devised the advanced fuzzy potential field method (AFPFM) as an improved algorithm of conventional PFM by using fuzzy membership and TS fuzzy inference [24]. In this method, the attractive field of AFPFM is the same as that of the conventional PF, but the repulsive field calculation is carried out using TS fuzzy inference.

Many kinds of sensors are used to detect obstacles such as sonar, laser range finder, LIDAR, stereo vision, and 3D depth sensor. In this study, we used a laser range finder and introduced a new method to avoid obstacles that we call the obstacle-dependent Gaussian potential field (ODG-PF) method, which we compared with PFM, FGM, and AFPFM.

This article consists of 6 sections. Section 1 is the introduction. Algorithms for comparison with ODG-PF are mentioned in Section 2. Section 3 describes ODG-PF in

detail. The results of simulations are in Section 4. Corresponding outdoor experiments carried out are in Section 5. The conclusion is in Section 6.

2. Related Work

All of the following methods use a range sensor. The blue lines in Figure 1(a) are infrared laser beams used to measure the distance. As depicted in Figure 1(b), the data from laser range finder are distances corresponding to predefined angles in the sensor.

These distance data are manipulated differently in different algorithms.

2.1. The Conventional Potential Field Method. Essentially, potential field-based methods use an artificial potential field consisting of an attractive field and a repulsive field. The attractive field attracts the vehicle toward the goal while the repulsive field repels the vehicle from obstacles. The following equations depict conventional potential field calculations:

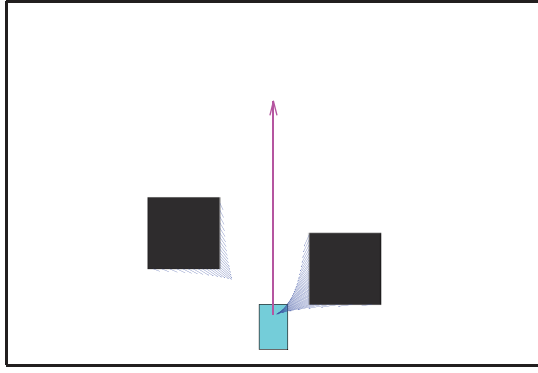
$$\mathbf{f}_{total} = \mathbf{f}_{att} + \mathbf{f}_{rep}, \quad (1)$$

$$\mathbf{f}_{att} = k_{att} \frac{\mathbf{r}_{goal} - \mathbf{r}}{|\mathbf{r}_{goal} - \mathbf{r}|}, \quad (2)$$

$$\mathbf{f}_{rep} = \begin{cases} -k_{rep} \sum_{i=1}^n \left(\frac{1}{d_i} - \frac{1}{d_{max}} \right) \mathbf{s}_i, & \text{if } d_i < d_{max} \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

where $\mathbf{s}_i = (\mathbf{r} - \mathbf{o}_i)/|\mathbf{r} - \mathbf{o}_i|$.

Equation (1) shows the total field calculation, while the attractive field that “attracts” the vehicle toward the goal is calculated by (2), where \mathbf{r}_{goal} is the position vector of the goal point and \mathbf{r} is the position vector of the vehicle. Equation (3) is the calculation for the repulsive field that pushes the vehicle away from obstacles in which \mathbf{o}_i is the position vector of each



→ Attractive Field
→ Repulsive Field

FIGURE 2: Attractive field and repulsive field in PFM.

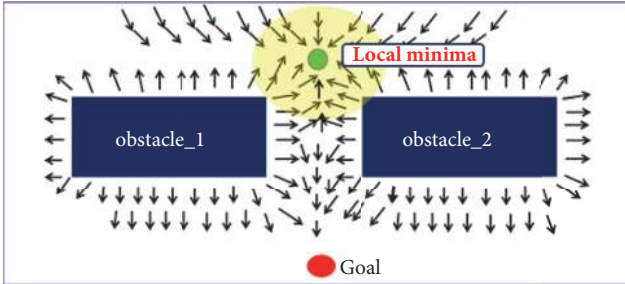


FIGURE 3: Local minima problem of conventional PFM.

obstacle. Figure 2 shows the attractive field and the repulsive field.

As depicted in Figure 3, the conventional potential field method has a drawback in that it becomes easily stuck at local minima [25].

2.2. The Follow-the-Gap Method. FGM is not based on an artificial potential field, but its guidance angle decision is somewhat similar. Besides, its obstacle detecting and gap finding methods are somewhat like our ODG-PF.

The main idea of FGM is to find the maximum gap in front of the vehicle and calculate the middle angle of the gap (θ_{gap}) first. From the received data, we know where the obstacles are and we should enlarge the obstacles to a degree with regard to the width of the vehicle. By enlarging the obstacles, we can easily know whether a gap is good for the vehicle to go through or not. This method also attains the angle toward the goal for which the direction decision is carried out using the weighted average of the angle of maximum gap and the angle toward the goal.

In the example in Figure 1, the maximum gap is in $-14^\circ \sim 18^\circ$ range and the medium angle (2°) becomes θ_{gap} as depicted in Figure 4. The guide angle is chosen by calculating the weighted average of θ_{gap} and θ_{goal} as follows:

$$\theta = \frac{(\alpha/d_{min})\theta_{gap} + \theta_{goal}}{\alpha/d_{min} + 1}, \quad (4)$$

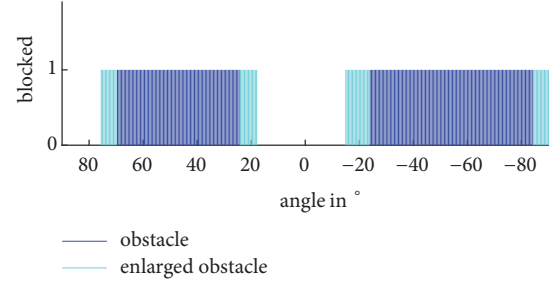


FIGURE 4: Finding the maximum gap from sensor data.

where d_{min} is the minimum distance from all of the distance data from the range sensors and α is a value that we should decide to avoid an obstacle and travel efficiently toward the goal. If it is small, the vehicle will travel more directly toward the goal but will be more likely to collide with obstacles whereas if it is large, the vehicle will travel more safely but the path toward the goal will not be so efficient. If the vehicle is near an obstacle, d_{min} will be small and $(\alpha/d_{min})\theta_{gap}$ will become greater than θ_{goal} , so the vehicle will travel more toward θ_{gap} rather than toward θ_{goal} . If the vehicle is far away from any obstacles, d_{min} will become large and $(\alpha/d_{min})\theta_{gap}$ will become relatively small, so the vehicle will travel more directly toward θ_{goal} .

2.3. The Advanced Fuzzy Potential Field Method. The repulsive force calculation in AFPFM is different from that in conventional PFM. The following equation shows the repulsive force calculation in AFPFM:

$$f_{rep} = -k_{rep} \left(\sum_{i=1}^n \frac{\varepsilon + d_i}{d_{max}} \right) \cdot \frac{\sum_{p=1}^n w_p (d_{max}/(\varepsilon + d_p)) (1/d_p - 1/d_{max}) \mathbf{s}_p}{\sum_{p=1}^n w_p}, \quad (5)$$

where d_{max} is the maximum range of the distance sensor, ε is the safety term considering the vehicle's outer hull, k_{rep} is the repulsive field coefficient with basically the same meaning in that of PFM, and w_p is a weight that, in TS fuzzy, is the minimum value of three membership functions u_{d_p} , u_{ϕ_p} , and u_{ψ_p} . The detailed calculation processes are explained in [23].

3. The Obstacle-Dependent Gaussian Potential Field

Here, we introduce ODG-PF. The main idea behind this method is that, after receiving distance data from the range sensor(s) (as shown in Figure 1(b)), we consider only the objects that are within the threshold range (2 m, for example), enlarge the obstacles with regard to the vehicle's width, and construct a Gaussian (repulsive) potential field from them. Next, we calculate the attractive field from the yaw angle information from an inertial measurement unit (IMU). The total field is made of these two fields, and, from it, we choose the angle with the minimum total field value. Figure 5 is the flowchart of the whole process.

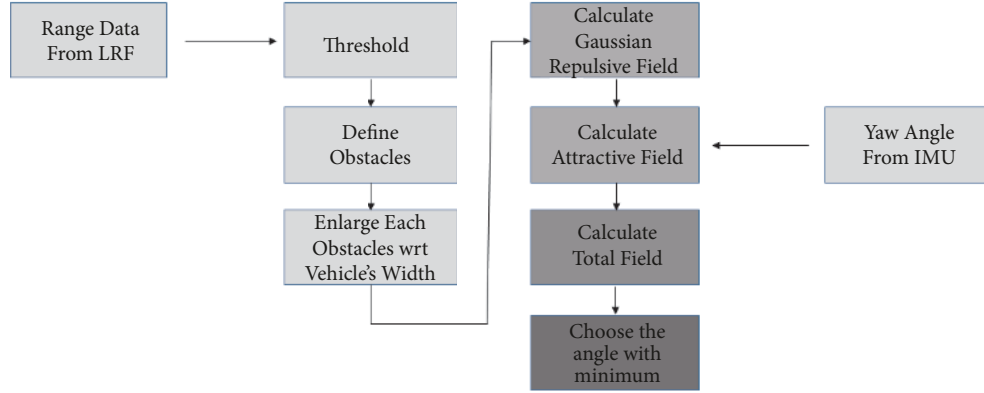


FIGURE 5: Flowchart of ODG-PF.

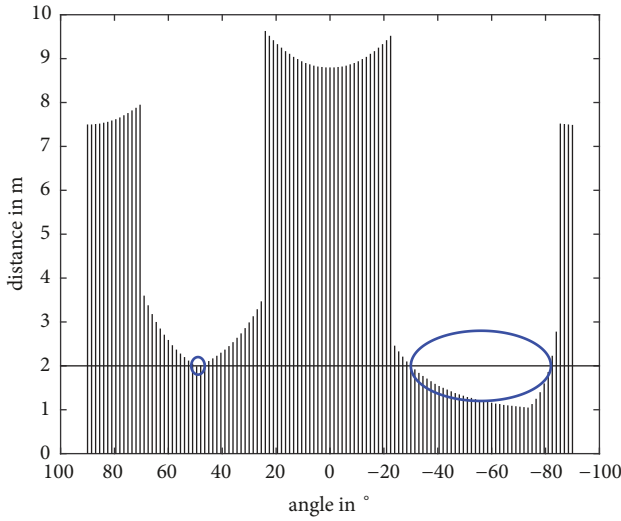


FIGURE 6: Sensor data and threshold distance.

If some data are spatially continuous within the threshold, they are recognized as an obstacle. Ellipses in Figure 6 show that there are two obstacles. Subsequently, we calculate the average distance to each obstacle (d_k) and the angle occupied by it. In Figure 6, the average distances to obstacles 1 and 2 are 1.99 and 1.36 m and the corresponding angles (ϕ_k) occupied by the obstacles are 1.5 and 51.5°, respectively.

In real systems, we should consider the vehicle's width, and, in many algorithms, the vehicle's width or size is added to the obstacle's size and the vehicle's size is shrunk to zero. Figure 7 shows the obstacle enlarging process.

If we consider the vehicle's width, we need to recalculate angle Φ_k as

$$\Phi_k = 2\sigma_k = 2 \cdot \text{atan2} \left(d_k \tan \frac{\phi_k}{2} + \frac{w_{robot}}{2}, d_k \right), \quad (6)$$

where d_k is the average distance to the k_{th} obstacle.

As described in (6), Φ_k is obtained from each obstacle's average distance and the occupied angle of obstacle. The bright bars added to the left and right sides of the dark bar in

Figure 7 are the enlargement of the obstacle after considering the vehicle's width. If more safety is needed, we can add more margin to the vehicle's width.

Gaussian likelihood functions (repulsive fields) of the obstacles are calculated as

$$f_k(\theta_i) = A_k \exp \left(-\frac{(\theta_k - \theta_i)^2}{2\sigma_k^2} \right) \quad (7)$$

where θ_k is the center angle of each obstacle. This is a function of θ_i . The subscript i means that the i_{th} data segment of the sensor data ($i = 1 \dots 361$) and it is the sequence number of each angle. Each Gaussian likelihood function becomes a component of the repulsive field. The coefficient A_k should be set in order that the Gaussian likelihood of each obstacle fully embraces the angle occupied by each obstacle. Subsequently,

$$\tilde{d}_k = A_k \exp \left(-\frac{(\theta_k - (\theta_k \pm \sigma_k))^2}{2\sigma_k^2} \right) = A_k \exp \left(-\frac{1}{2} \right). \quad (8)$$

Thus,

$$A_k = \tilde{d}_k \exp \left(\frac{1}{2} \right), \quad (9)$$

where $\tilde{d}_k = d_{max} - d_k$. d_{max} is the maximum detection range of the range sensor and σ_k is half of the angle occupied by the k_{th} obstacle.

In Figure 8(a), the angles occupied by and the Gaussian likelihoods of the obstacles are shown. This explains that the Gaussian likelihood of each obstacle fully embraces the occupied angle at \tilde{d}_k .

We obtain the repulsive field by summing up all of the Gaussian likelihood functions of obstacles into one function as

$$f_{rep}(\theta_i) = \sum_{k=1}^n A_k \exp \left(-\frac{(\theta_k - \theta_i)^2}{2\sigma_k^2} \right). \quad (10)$$

This is also a function of angle θ_i . The attractive field is calculated as

$$f_{att}(\theta_i) = \gamma |\theta_{goal} - \theta_i|. \quad (11)$$

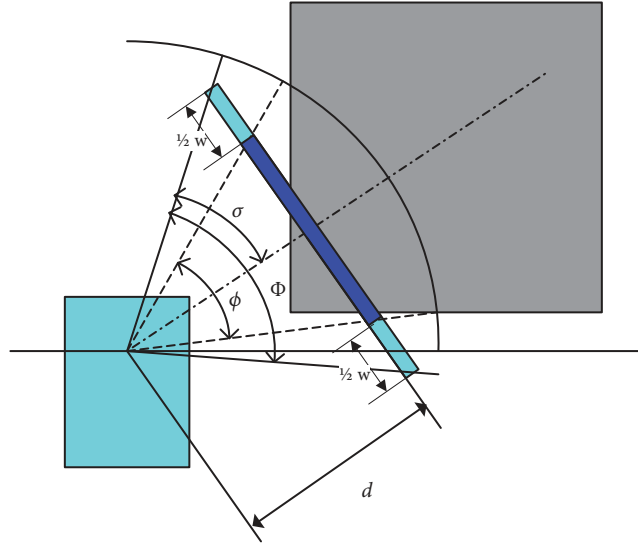


FIGURE 7: Enlarging the angle occupied by obstacles.

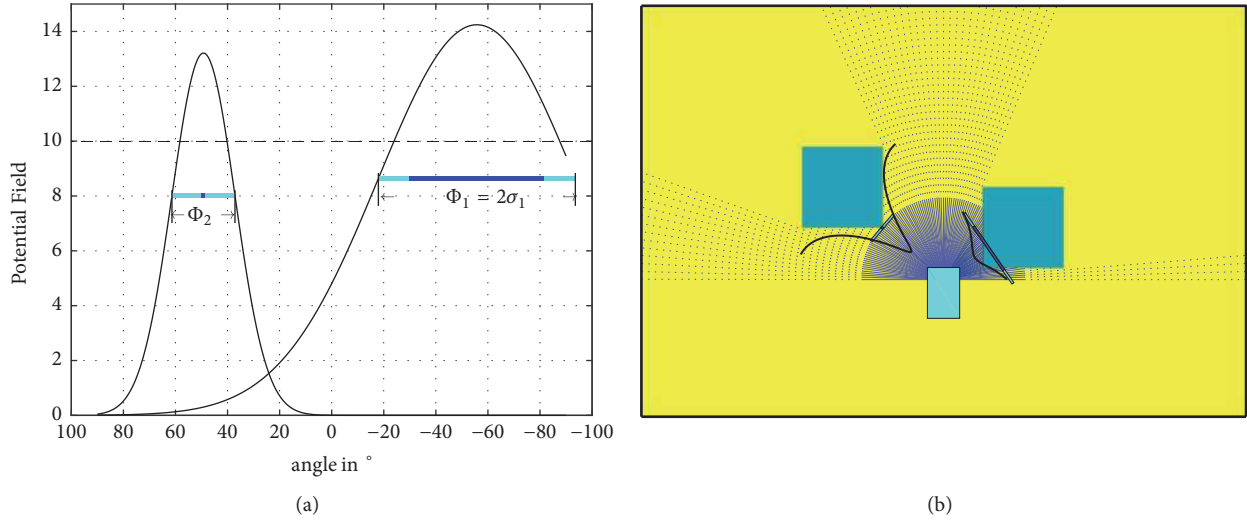


FIGURE 8: (a) Gaussian likelihood of each obstacle and (b) illustration with vehicle and obstacles.

The total field is calculated by adding these two fields. Since these two fields are functions of θ_i , the total field is also a function of θ_i :

$$f_{total}(\theta_i) = f_{rep}(\theta_i) + f_{att}(\theta_i). \quad (12)$$

In (11), γ is the value that we should choose (it is like k_{att} in PFM and AFPFM). If γ is too small, the vehicle will avoid obstacles but the path will be inefficient whereas if it is too large, the vehicle will be more likely to collide with obstacles. The γ value was chosen as 5.0 by executing several simulations. We used this predetermined value in all the simulations and experiments. The calculation example of the fields is shown in Figure 9. Finally, the angle minimizing the total field is chosen as the heading angle. The small circle in Figure 9 points out this angle.

This method has some features that are not like in classical potential field-based methods, i.e., used by Ge and Cui [26]. First, instead of putting the range sensor data into equation, ODG-PF defines obstacles (occupied angle, average distance, etc.) from range sensor data and then calculates the repulsive fields of the obstacles. Second, the repulsive field and the attractive field of ODG-PF are functions of angle; that is to say, they are not vectors. In the classical potential field, the direction is decided by the direction of the total vector of the repulsive field and the attractive field. Thus, ODG-PF is very robust that even if there are small changes in the environment, the trajectory is not so much affected by them. Mathematically, the main difference between PFM and ODG-PF is that the attractive field and the repulsive field (as well as the total field) are angle functions while all of the fields in PFM are vector values. The direction decision is made by

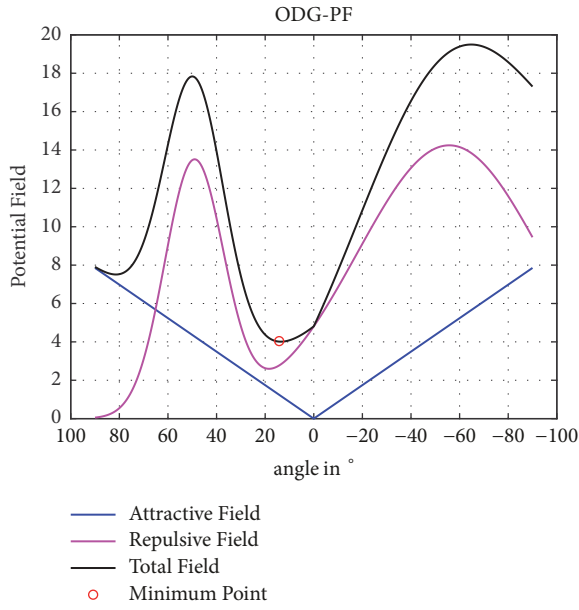


FIGURE 9: ODG-PF potential fields.

finding the angle with the minimum value of the total field rather than by calculating the arctangent of a vector.

4. Simulations

4.1. Settings. Positions of obstacles in the simulations and the vehicle were as in Figure 10. The settings were the same as that of the real experiments. The obstacles were set in a zigzag formation and the vehicle was in front of them. To minimize the effect of the wall, the positions of obstacles and the vehicles were not near the walls on either side.

We compared our novel ODG-PF method to conventional PFM, FGM, and AFPFM. PFM and AFPFM use artificial potential field and it was useful to compare them with the ODG potential field. FGM uses gaps between obstacles and the obstacle dependency of ODG-PF is comparable. All of the algorithms used in this study decide on the direction angle from the received range sensor data. From the sensor data in Figure 1(b), we were able to know where the maximum gap in FGM was and to calculate the repulsive fields of PFM, AFPFM, and ODG-PF.

Each method had some parameters that needed to be defined. For PFM and AFPFM, we fixed the repulsive field coefficient (k_{rep}) at 0.16 and carried out simulations with them while varying the attractive field coefficient (k_{att}) for simplicity.

For FGM, we needed to choose an appropriate α in (4), and so we set it to 0.5 m, which meant that angles toward the goal and the maximum gap had the same weights when the minimum distance from the obstacles and the vehicle was 0.5 m.

Actually, we should choose the maximum sensor distance for PFM and AFPFM. Our range sensor can detect object as far as 10m, but if we changed the d_{max} value in PFM and AFPFM, we would have had to change all of the fuzzy

inference rules in AFPFM. For a fair comparison, we left it unchanged.

We needed to define two things for ODG-PF. One is the threshold distance which we set to 2.0 m in order to be easily adaptable to any low-speed, low-resolution sensors. Another parameter we needed to decide on was γ , which is like k_{att} in PFM and AFPFM but, as described in (7), is used to apply a penalty proportional to the deviation from the angle toward the goal. We set γ to 5.0, which was obtained by both simulations and experiments. ODG-PF is not very sensitive to γ , and, experimentally, it can be set to 4.0~6.0.

θ_{goal} , the angle toward the goal in all the algorithms in this study, was fixed. That is to say, there was no goal position but the fixed angle and the purpose of the obstacle avoidance algorithms in this study was to do their best to keep the fixed θ_{goal} without colliding with anything.

Thus, we needed to change the vector of the attractive field described in (2) to

$$\mathbf{f}_{att} = k_{att} \begin{pmatrix} \cos(\theta_{goal} - \theta_{current}) \\ \sin(\theta_{goal} - \theta_{current}) \end{pmatrix}. \quad (13)$$

4.2. Simulation Results under a Static Environment. We carried out simulations with the algorithms using the same settings shown in Figure 10, the results of which are shown in Figure 11. Figure 11(a) exhibits the results of the PFM simulations. As predicted, k_{att} played a key role in the trade-off between safety and efficiency, and PFM did not perform efficiently in any of the cases. As already compared in [23], PFM was very sensitive to the repulsive field and the trajectories were easily affected even when the vehicle was relatively far away from the obstacles. A large k_{att} value made it less sensitive.

FGM performed better than PFM in that it tried to find the safest angle, as shown in Figure 11(b) and tended to follow the maximum gap if the vehicle came close to the obstacles, as described in (4). Anyway, FGM will be in danger of a collision if an obstacle is between the angle of the maximum gap (θ_{gap}) and the angle toward the goal (θ_{goal}). AFPFM was better than PFM in all cases, but when k_{att} was large, there were collisions, as shown in Figure 11(c). The value of k_{att} should be much larger (by 10 times, for simplicity) than PFM. The reason for this is that there is a coefficient in calculating repulsive field and there are also weights of TS Fuzzy in AFPFM which are a maximum of 15 times larger than those in PFM when an obstacle is placed close to the front of the vehicle.

The simulation of ODG-PF performed the best in all cases. We performed the same experiments several times and there were no collisions. It always avoided the obstacles successfully, and its heading angle finally returned to θ_{goal} , as shown in Figure 11(d).

PFM did not attain good results and the final deviation from the initial y position was the largest. When we set k_{att} to a large value, the deviation became smaller but, as shown in Figure 11(a), the risk of collision also became larger.

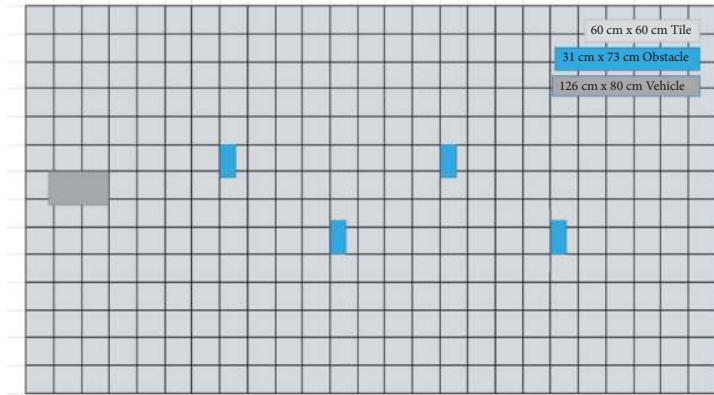


FIGURE 10: Mobile robot and obstacle settings.

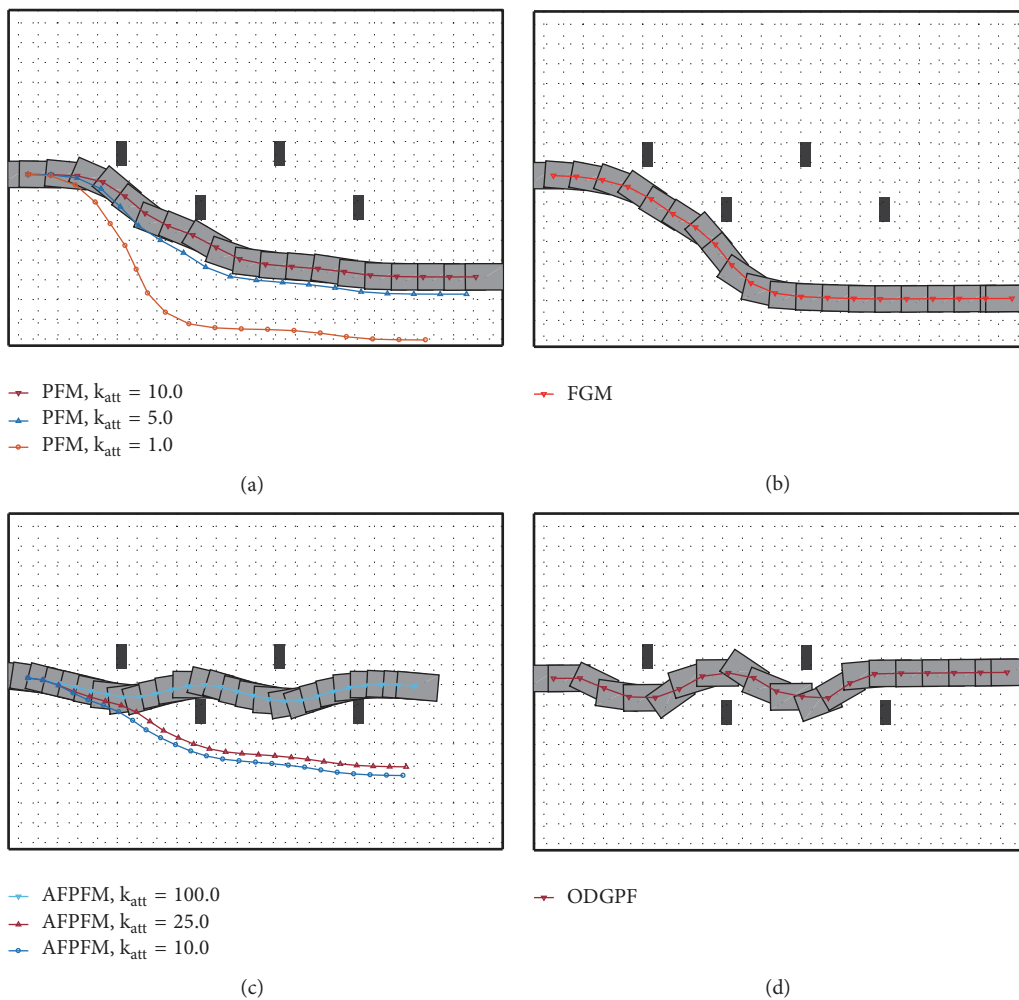


FIGURE 11: Simulation results under static environment: (a) PFM, (b) FGM, (c) AFPFM, and (d) ODG-PF.

The results for FGM were not too bad. When we set α to a smaller value, this caused collisions, and larger values caused bigger deviations.

The results of AFPFM show that it is dependent on k_{att} as PFM is, but they also show that this algorithm could

be improved. In Figure 11(c), we can imagine that there would be an optimal k_{att} (or membership function) value that efficiently avoids obstacles without much deviation in simulation, but the optimal k_{att} value would vary on a case-by-case basis.

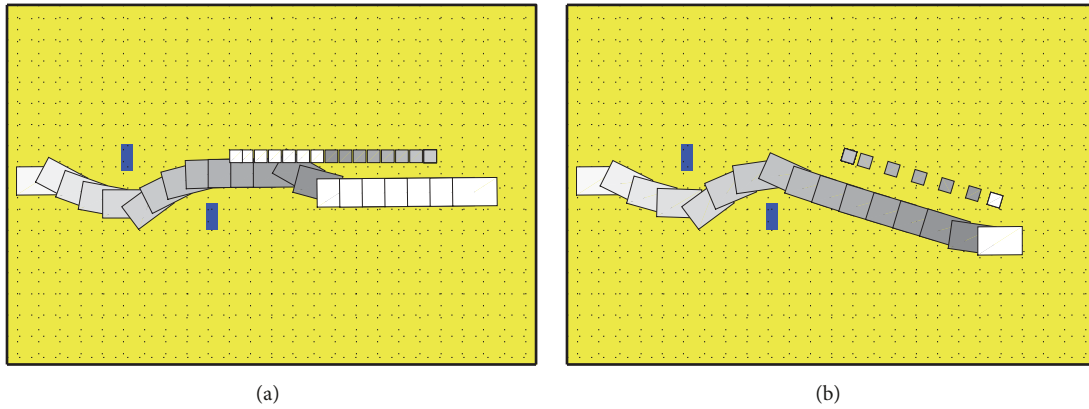


FIGURE 12: Simulation results under dynamic environments: (a) Scenario 1. (b) Scenario 2.

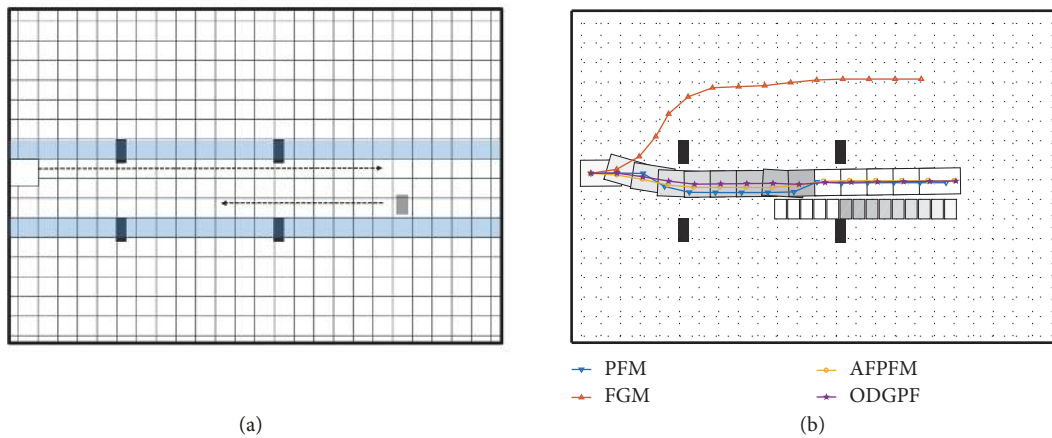


FIGURE 13: Dynamic environment Scenario 3: (a) scenario and (b) simulation results.

4.3. *Simulation Results under a Dynamic Environment.* Another merit of ODG-PF is that it can successfully avoid obstacles without any adjustment even when they are moving. Any obstacle avoidance method cannot avoid all moving obstacles because of the vehicle's physical limitations such as velocity, angular velocity, radius of rotation, sensor measurement range, etc. The success of avoiding moving obstacles depends heavily on the relative velocity between the moving obstacle and the vehicle, but it also depends on the obstacle avoidance algorithm.

We designed three moving obstacle scenarios. In Scenarios 1 and 2, we used the first two static obstacles used in the previous experiments and a moving obstacle, but the trajectory of the moving obstacle in Scenario 1 was different from that of Scenario 2.

Figure 12 shows the scenarios and simulations of moving obstacle avoidance in Scenarios 1 and 2. As shown in Figure 12(a), the vehicle using ODG-PF successfully avoided the moving obstacle. The colors of the vehicle and the moving obstacle became darker as they became closer and after they met (the point at which they were closest), their color returned to white. By using different colors, the reader can

easily distinguish the movements of the vehicle and the moving obstacle with regard to time steps.

In Scenario 2, the moving obstacle approached from the left side of the vehicle with the same velocity as the vehicle (after the vehicle had gone past the second static obstacle). The vehicle using ODG-PF moved as if it was going along with the obstacle.

To compare the algorithms more strictly, we devised another experimental scenario. Figure 13(a) shows the static obstacles and the direction of the vehicle and the moving obstacle.

The simulation results of this moving obstacle scenario show that vehicles with PFM, AFPFM, and ODG-PF were able to avoid both the static obstacles and the moving obstacle. However, when we saw the details of the simulation results for PFM and AFPFM, their movements were somewhat unstable that the vehicle moved in a zigzag fashion.

5. Experiments

5.1. *Hardware Specifications and Setup.* Figure 14 shows the configuration of the system.

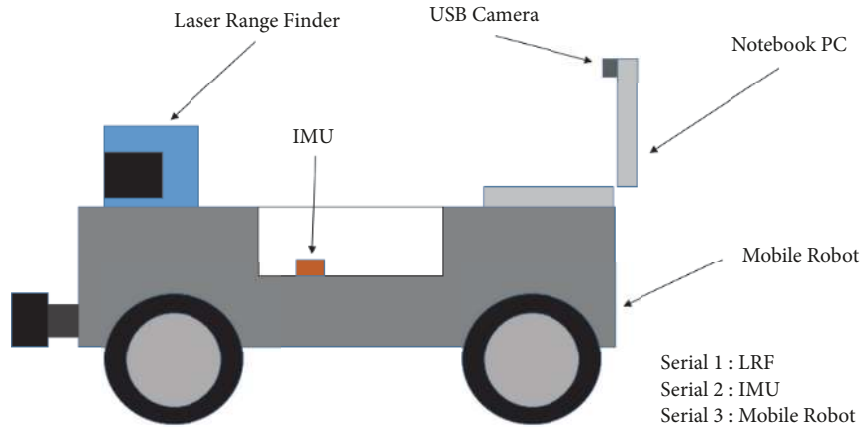


FIGURE 14: System configuration.

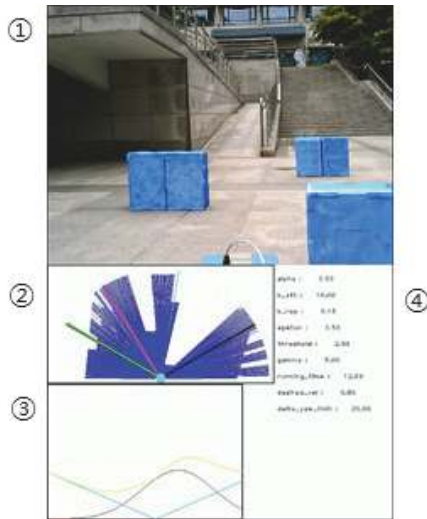


FIGURE 15: Program screen recorded by the PC on the robot.

The vehicle used was an Unmanned Solution™ ERP42v1 without a braking system. Its physical dimensions were $126 \times 80 \times 30$ cm. The range sensor was a SICK™ LMS200 laser measurement system, which could sense the distances from obstacles within $\pm 90^\circ$ range at 0.5° resolution. We used Xsens™ MTI-100 IMU to calculate the yaw angle of the system using Madgwick’s algorithm [27]. In all cases, the yaw angle of the starting point was 0° . A Hardkernel™ USB-CAM 720P USB camera with the resolution set to 640×480 was used to record the experiments. The CPU of the notebook PC was an Intel i5 (2 core, 4 thread, 2.67 GHz) and 4 GB of memory. The OS used was Windows 7. To implement the algorithms and show the graphs, Microsoft Visual C++ and OpenCV library were used. All of the data were recorded in real time. Figure 15 shows a snapshot of a recorded video. As we implemented all the algorithms in one PC program, the user should choose one of them before running the program.

The picture shows ① the front view from the USB camera, ② graphical sensor data, ③ calculated data (ODG-PF in this



FIGURE 16: Outdoor experimental environment.

example), and ④ the parameter values used. Thus, recorded videos can be used to analyze the cause of failure later.

5.2. Outdoor Experimental Environment. The experiments were performed at the back of the Innovation Hall at Korea University. One obstacle consisted of two $56 \times 36.5 \times 30$ cm expanded polystyrene boxes. Basically, the positions of the obstacles and the vehicle were the same as those in the simulation experiments. The outdoor experimental environment is shown in Figure 16.

5.3. Experiment Results under a Static Environment. The trajectory of each experiment was calculated later from the saved data using

$$x(n+1) = x(n) + v \cdot \cos \theta \cdot \Delta t, \quad (14)$$

$$y(n+1) = y(n) + v \cdot \sin \theta \cdot \Delta t. \quad (15)$$

The value of k_{att} for AFPFM was initially set at 100 because we used this value in the indoor experiments. For conventional PFM, we initially used $k_{att} = 10$ for the indoor experiments.

Figure 17 is the trajectories of the experiments. The time interval between markers in all the figures of trajectories is 0.8s. The velocity was set to constant but, in the beginning of

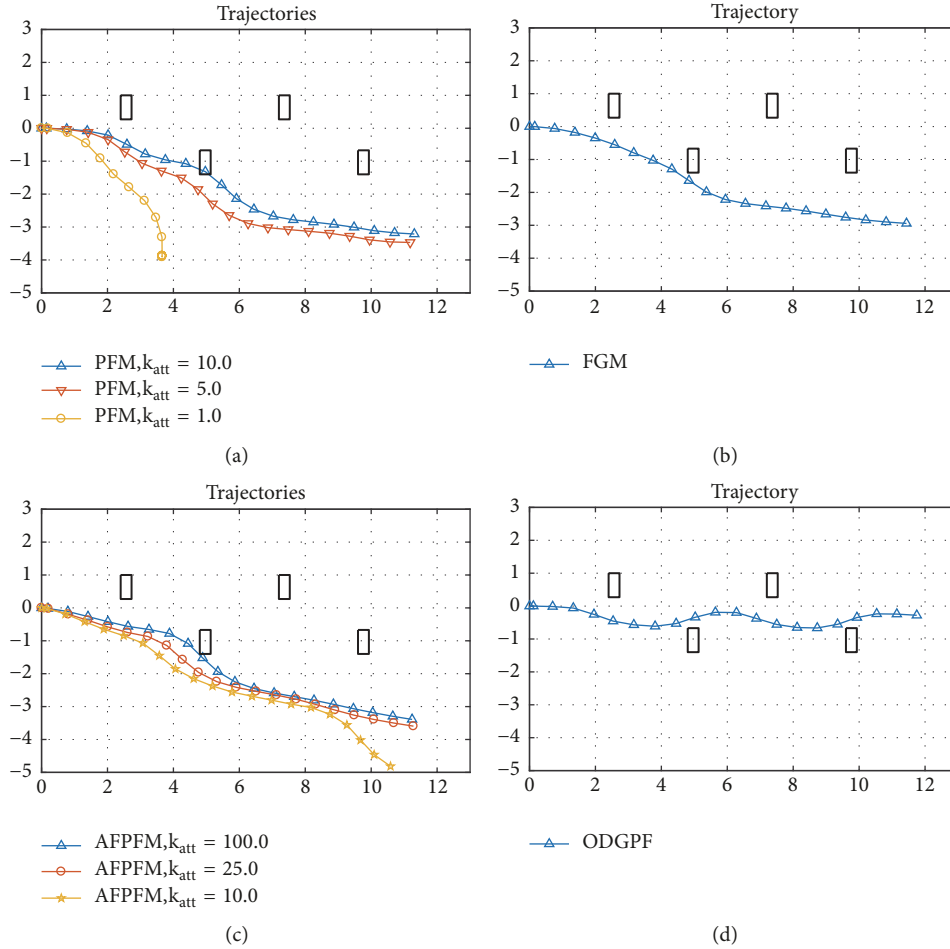


FIGURE 17: Trajectories of the vehicle with (a) PFM, (b) FGM, (c) AFPFM, and (d) ODG-PF.

each navigation, the robot was accelerated from zero speed. The sampling time of IMU is 0.01s and it was used as the reference clock.

For the PFM with $k_{att}=10$ (Figure 17(a)), and AFPFM with $k_{att}=100$ (Figure 17(c)) cases, the vehicle collided with the second obstacle.

As occurred in the simulations, Figure 17(a) shows that PFM did not perform well and collided with obstacles when k_{att} was large and strayed from the desired angle when k_{att} was small. When k_{att} was too small, it strayed from the goal angle, and it collided with the second obstacle when $k_{att}=10$.

The vehicle with FGM performed very well when it is needed to avoid obstacles in complex environments. There were no collisions in several experiments, although it did not attain an efficient trajectory.

Figure 17(c) shows the performance of the vehicle with AFPFM. When k_{att} was large ($k_{att}=100$), the vehicle collided with the second obstacle, which was not the same results as in the simulation, indicating that there might have been a problem in the vehicle's dynamic model. Moreover, there was a pedestrian at the end of the experiment with $k_{att}=10$.

ODG-PF performed the best in all cases. We carried out the same experiments several times and there were no

collisions. It always managed to avoid the obstacles, as shown in Figure 17(d).

The experimental results consistently show that ODG-PF efficiently and successfully avoided obstacles. Other algorithms were able to avoid obstacles but the deviations from the original y position were large.

5.4. Experiment Results under the Dynamic Environment. As simulated in Figure 13, we performed three scenarios for moving obstacle avoidance.

In Scenarios 1 and 2, the positions of two static obstacles were the same as for static obstacle avoidance and we performed experiments with all of the aforementioned algorithms. The experimental results of vehicles with PFM, AFPFM, and FGM show that their trajectories were not much different from those of the static obstacle cases because they were far away from the moving obstacle. As shown in Figures 17–19, the speed of mobile robot was constant except in the beginning of each experiment, where the mobile robot was accelerated from zero speed to the specified speed. We will upgrade ODG-PF to consider the acceleration of the mobile robot in the future work.

In Scenario 3, we analyzed all of the algorithms using different static obstacle positions and scenarios.

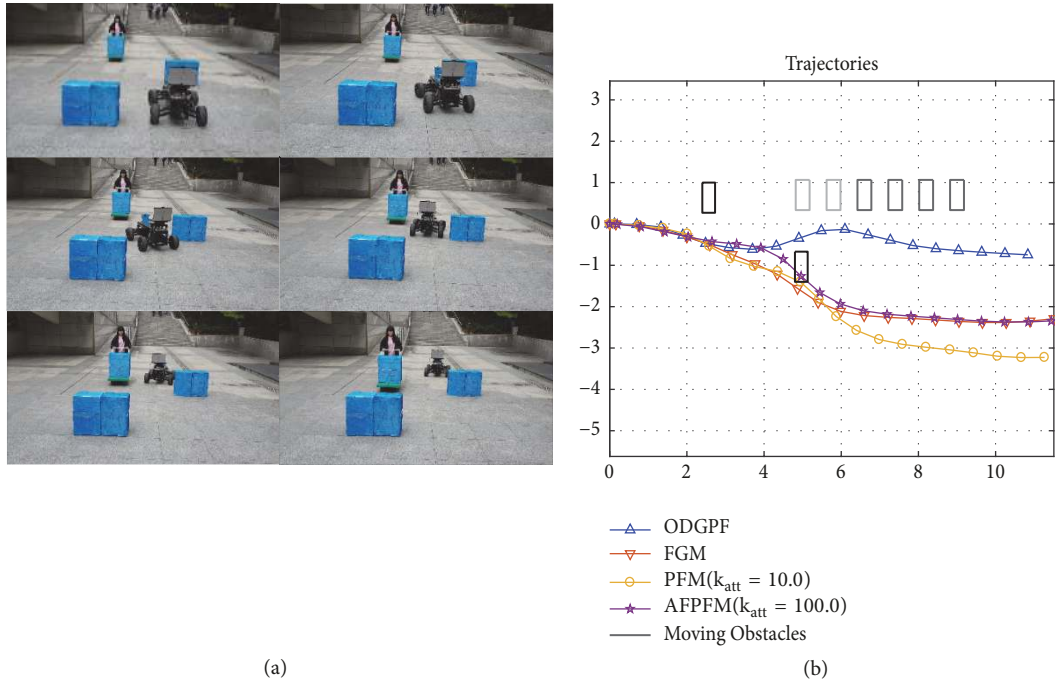


FIGURE 18: Experimental results of the vehicle with ODG-PF for Scenario 1.

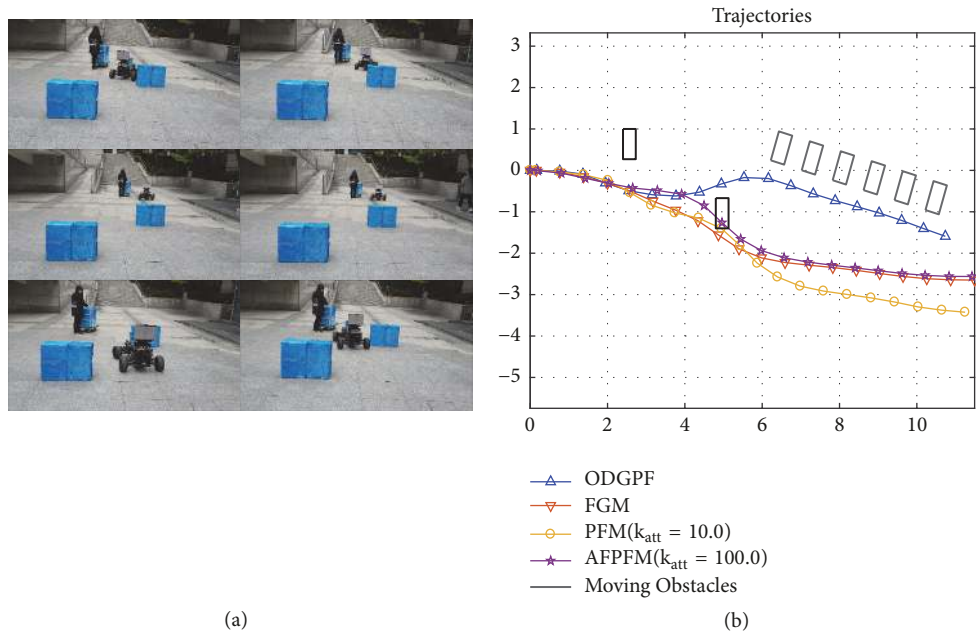


FIGURE 19: Experimental results of the vehicle with ODG-PF for Scenario 2.

Figure 18 show some images from the Scenario 1 experiment and the trajectory. The vehicle’s velocity was 0.8 m/s in both the simulations and the experiments.

Figure 19 show some images of the experiments and the trajectory for Scenario 2. We can see that they also agreed with the simulation results shown in Figure 13(b).

As in Scenarios 1 and 2, we performed experiments with the vehicle and all of the aforementioned algorithms

for moving obstacle Scenario 3. As shown in Figure 13(b), all of the algorithm except for FGM successfully avoided both the static obstacles and the moving obstacles in the simulations. This scenario is harsh because at the end of the experiments, the gap between a static obstacle and the moving obstacle was only slightly wider than the vehicle’s width. In the simulations, we noted that all the algorithms successfully passed through the gap when it was relatively

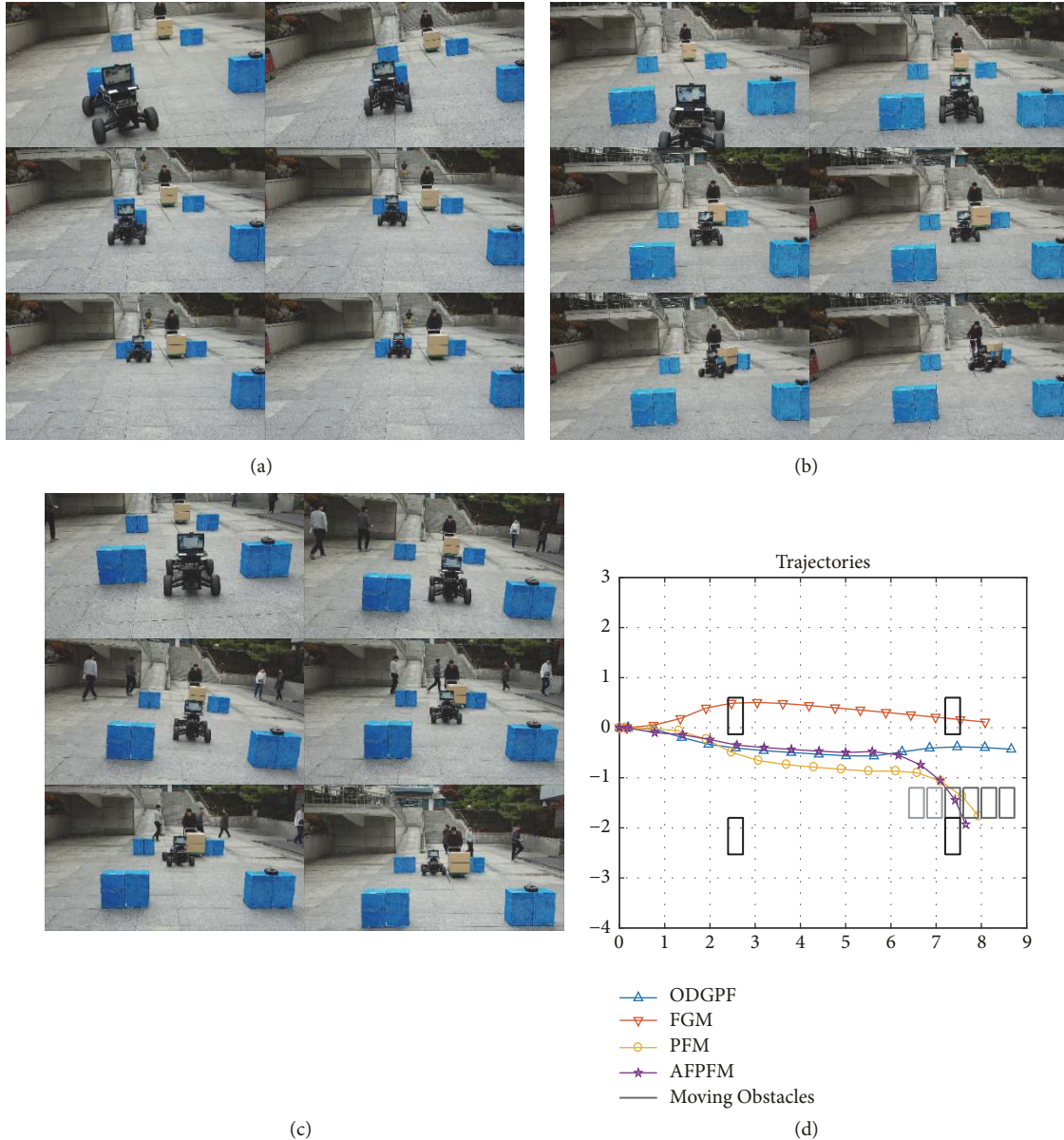


FIGURE 20: Experimental results for Scenario 3: (a) FGM, (b) AFPFM ($k_{att}=100$), (c) ODG-PF, and (d) trajectories.

wide. We shrunk the gap in the simulations to a degree where one of them started to move differently. In Figure 13(b), the vehicle with FGM went toward the maximum gap, which was to the left of the vehicle when it started to move. We also performed experiments with a wider gap with which all the algorithms worked well.

Figure 20 shows some images from Scenario 3 experiments and trajectories. In results different from the simulation, only ODG-PF successfully passed through the gap between the static obstacles and the moving obstacle. The reasons for success and failure are discussed later.

For PFM and AFPFM, the direction is decided by the arctangent value of the total field vector. As described in (1), the total field is the sum of the attractive field and the repulsive field. The repulsive field becomes larger when the

distance to the obstacles becomes smaller. At a certain point, the repulsive field starts to overwhelm the attractive field and the arctangent value directs the vehicle backwards. We analyzed the reason for failure, which is explained in the next section. The mobile robot was not able to change direction abruptly, which resulted in a collision.

Nevertheless, the vehicle avoided both the static obstacles and the moving obstacle closely but successfully with ODG-PF. The Gaussian repulsive field of the ODG-PF described obstacles efficiently and did not fail to detect them.

Figure 20(d) shows the trajectories of the vehicle using different obstacle avoidance algorithms. Besides the ODG-PF case, the experimental results were not the same as the simulation results, and so we think that the vehicle model used in the simulation might have had some errors in it.

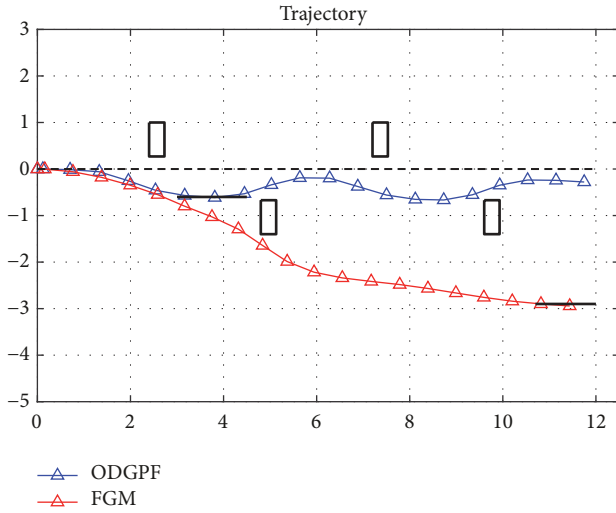


FIGURE 21: Maximum deviation from the initial direction line.

The experimental trajectory of FGM was different from that of the simulation, which we think is due to the vehicle model. As in the simulation result shown in Figure 13(b), FGM tried to follow the maximum gap.

5.5. Evaluation and Discussion. To evaluate the results of the different algorithms, we should consider whether the vehicle with each algorithm collided with any obstacles or not. If the vehicle with a particular algorithm did not collide with any obstacles, we should evaluate the algorithm with some criteria. When considering the efficiency (depicted in Figure 21), the maximum deviation from the initial direction line can be a good criterion.

Table 1 contains the collision results and the maximum deviation from the initial line for each algorithm. Avoiding obstacles is more important than efficiency because this is a matter of safety. Efficiency matters only if there are no collisions with obstacles.

As is evident from the results in Table 1, ODG-PF avoided obstacles with outstanding efficiency and AFPFM was more efficient than conventional PFM.

The experiments with moving obstacles show that only ODG-PF successfully avoided the moving obstacles in all the 3 scenarios. The experimental results of Scenario 3 show that, in a tough situation, only ODG-PF successfully avoided both static and moving obstacles.

Failure analyses show why other methods could not avoid the obstacles and there were different reasons for each of these. In PFM and AFPFM, the total field is a vector sum of the repulsive field and the attractive field. The vector direction (recommended angle of the vehicle by PFM or AFPFM) is calculated using the arctangent of the vector. As shown in Figure 22, if the attractive field is not large enough to push the repulsive field, the direction of the vector sum will be at the back of the vehicle. The problem is that using a large attractive field is not the solution because the repulsive field is inversely proportional to the distances to obstacles.

TABLE 1: Experimental results of static obstacle avoidance.

No.	Algorithm	Parameter	Collision	Max. Deviation
1	PFM	$k_{att}=10$	○	3.2 m
2	PFM	$k_{att}=5$	-	3.7 m
3	PFM	$k_{att}=1$	-	4.6 m
4	FGM	$\alpha=0.5$	-	3.4 m
5	AFPFM	$k_{att}=100$	○	3.2 m
6	AFPFM	$k_{att}=25$	-	3.0 m
7	AFPFM	$k_{att}=10$	-	4.3 m
8	ODG-PF	$\gamma=5.0$	-	0.6 m

Furthermore, vehicles have some physical constraints such as turning radii, maximum wheel angles, and body sizes. We cannot abruptly adjust the vehicle's direction to go backwards. Because turning rate is limited, turns trying to avoid obstacle can often lead to positions that are more dangerous.

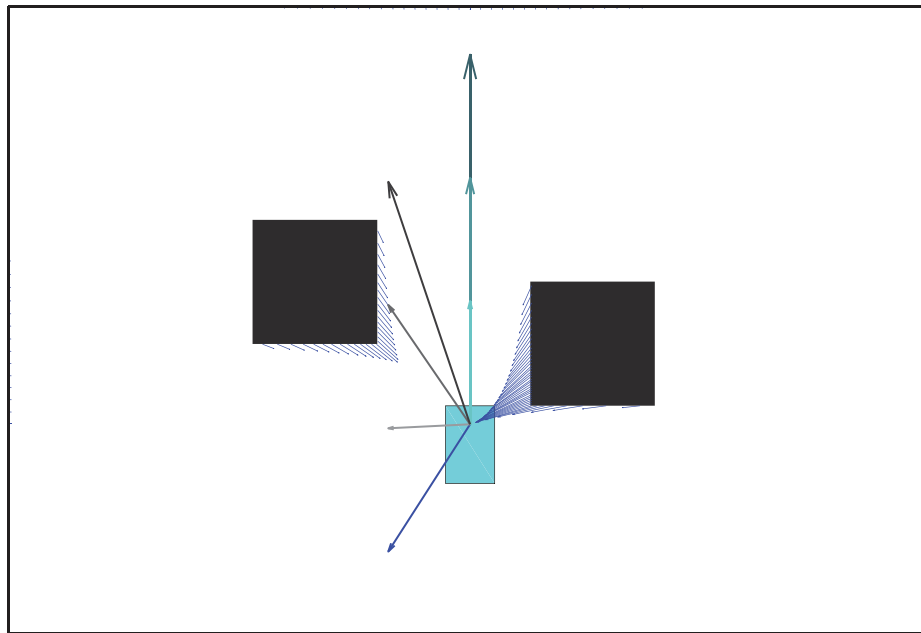
As depicted in Figure 23, the guide angle in FGM is calculated as in (4). The drawback of this calculation is that the direction of guide angle can be toward an obstacle when it is between the maximum gap angle and the goal angle because the guide angle is the weighted average of the two angles.

6. Conclusions

In this paper, we introduce a novel method for obstacle avoidance, ODG-PF, which avoids obstacles very efficiently and safely. Although AFPFM is an improvement that avoids the local minima problem, its attractive field coefficient should be adjusted when the environment is changed. As with conventional PFM, when the vehicle is in a narrow environment, the repulsive field becomes strong; thus k_{att} value needs to be large. Otherwise, the robot will be stuck at a local minimum or go toward the goal very inefficiently.

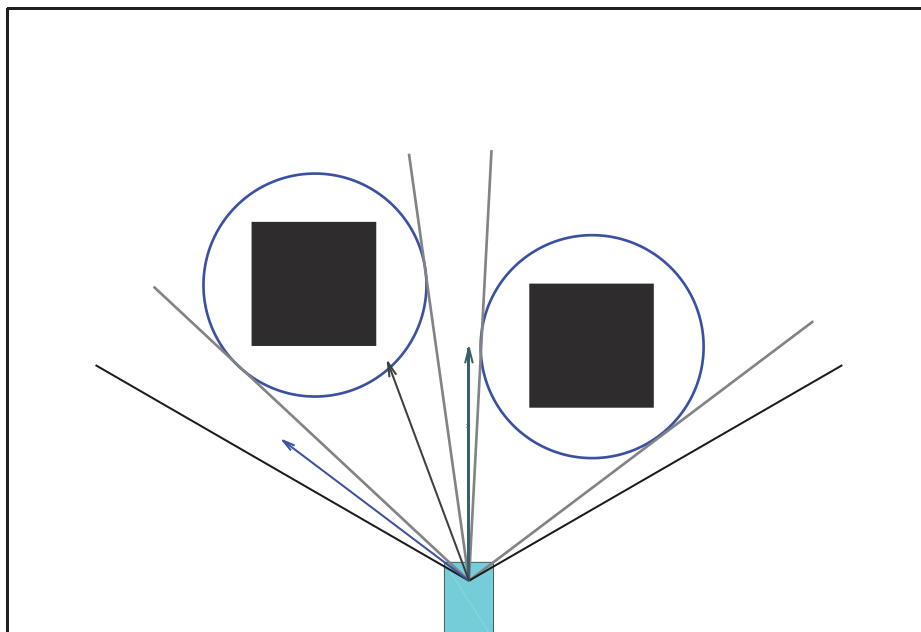
Simulations and experiments showed that vehicle movements with ODG-PF were very stable. It seems that this stable movement is because of two reasons. One reason is that it detects and defines obstacles first rather than directly calculating the repulsive field from the distance data, and the other is that it finds the angle with minimum values from the total field function. Thus, ODG-PF does not have the local minima problem that causes a vehicle to become stuck at certain points.

One more merit of ODG-PF is that it avoids both static and moving obstacles without any adjustments. In the performed moving obstacle scenarios, the vehicle using it did not collide with any of the static or moving obstacles and instead avoided them using the same method. This is also quite a good point in that it does not need to carry out any time-consuming activities such as image processing or computer vision processing. ODG-PF can be easily implanted in a real-time system because the overhead when calculating it is relatively light.



- Attractive Field
- Repulsive Field
- Total Field

FIGURE 22: Total field of PFM and AFPFM with varying k_{att} .



- θ_{gap}
- θ_{goal}
- θ

FIGURE 23: Guide angle calculation in FGM.

Data Availability

The readers can access the data through the homepage: <http://cml.korea.ac.kr>.

Conflicts of Interest

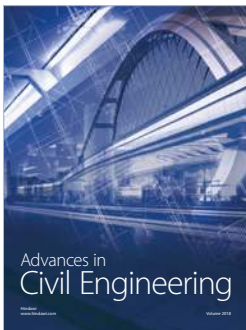
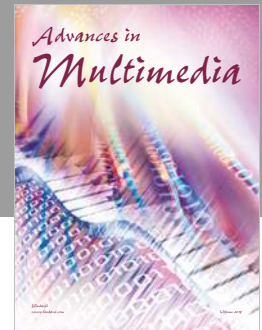
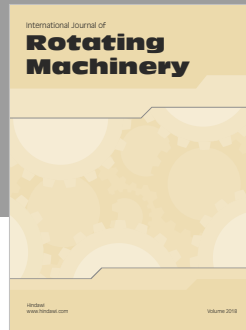
The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2016R1D1A1B01016071 and NRF-2016R1D1A1B 03936281) and also in part by the Brain Korea 21 Plus Project in 2018.

References

- [1] H. Choset, *Sensor based motion planning: The hierarchical generalized Voronoi graph*, California Institute of Technology, Pasadena, USA, 1996.
- [2] E. Masehian and M. R. Amin-Naseri, "A voronoi diagram-visibility graph-potential field compound algorithm for robot path planning," *Journal of Robotic Systems*, vol. 21, no. 6, pp. 275–300, 2004.
- [3] P. Bhattacharya and M. L. Gavrilova, "Voronoi diagram in optimal path planning," in *Proceedings of the 4th International Symposium on Voronoi Diagrams in Science and Engineering (ISVD '07)*, pp. 38–47, July 2007.
- [4] S. Garrido, L. Moreno, M. Abderrahim, and F. Martin, "Path Planning for Mobile Robot Navigation using Voronoi Diagram and Fast Marching," in *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2376–2381, Beijing, China, October 2006.
- [5] T. Arney, "An efficient solution to autonomous path planning by approximate cell decomposition," in *Proceedings of the 2007 3rd International Conference on Information and Automation for Sustainability, ICIAFS*, pp. 88–93, Australia, December 2007.
- [6] R. V. Cowlagi and P. Tsiotras, "Beyond quadtrees: Cell decompositions for path planning using wavelet transforms," in *Proceedings of the 46th IEEE Conference on Decision and Control 2007, CDC*, pp. 1392–1397, USA, December 2007.
- [7] G. I. Delgado, R. O. Casallas, and G. E. Jacinto, "Path planning in static scenarios using image processing and cell decomposition," in *Proceedings of the 2014 IEEE International Autumn Meeting on Power, Electronics and Computing, ROPEC 2014*, Mexico, November 2014.
- [8] P. Melchior, B. Orsoni, O. Lavialle, A. Poty, and A. Oustaloup, "Consideration of obstacle danger level in path planning using A* and fast-marching optimisation: comparative study," *Signal Processing*, vol. 83, no. 11, pp. 2387–2396, 2003.
- [9] S. Garrido, L. Moreno, and D. Blanco, "Exploration of a cluttered environment using Voronoi Transform and Fast Marching," *Robotics and Autonomous Systems*, vol. 56, no. 12, pp. 1069–1081, 2008.
- [10] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '85)*, vol. 2, pp. 500–505, 1985.
- [11] A. Oustaloup, B. Orsoni, P. Melchior, and H. Linares, "Path planning by fractional differentiation," *Robotica*, vol. 21, no. 1, pp. 59–69, 2003.
- [12] V. J. Lumelsky and A. A. Stepanov, "Path-planning strategies for a point mobile automaton moving amidst unknown obstacles of arbitrary shape," *Algorithmica*, vol. 2, no. 1-4, pp. 403–430, 1987.
- [13] J. Borenstein and Y. Koren, "The vector field histogram—fast obstacle avoidance for mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pp. 278–288, 1991.
- [14] I. Ulrich and J. Borenstein, "VFH+: reliable obstacle avoidance for fast mobile robots," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '98)*, pp. 1572–1577, May 1998.
- [15] I. Ulrich and J. Borenstein, "VFH*: local obstacle avoidance with look-ahead verification," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '00)*, pp. 2505–2511, April 2000.
- [16] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics and Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.
- [17] S. J. Anderson, S. C. Peters, T. E. Pilutti, and K. Iagnemma, "An optimal-control-based framework for trajectory planning, threat assessment, and semi-autonomous control of passenger vehicles in hazard avoidance Scenarios," *International Journal of Vehicle Autonomous Systems*, vol. 8, no. 2–4, pp. 190–216, 2010.
- [18] J. Minguez and L. Montano, "Nearness diagram navigation (ND): A new real time collision avoidance approach," in *Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2094–2100, Japan, November 2000.
- [19] J. Minguez, J. Osuna, and L. Montano, "A "divide and conquer" strategy Based on situations to achieve reactive collision avoidance in troublesome scenarios," in *Proceedings of the Proceedings- 2004 IEEE International Conference on Robotics and Automation*, pp. 3855–3862, USA, May 2004.
- [20] J. W. Durham and F. Bullo, "Smooth nearness-diagram navigation," in *Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, pp. 690–695, France, September 2008.
- [21] M. Mujahed and H. Jaddu, "Smooth and safe Nearness-Diagram (SSND) Navigation for autonomous mobile robots," *Advanced Materials Research*, vol. 403-408, pp. 4718–4726, 2012.
- [22] V. Sezer and M. Gokasan, "A novel obstacle avoidance algorithm: "Follow the gap method"," *Robotics and Autonomous Systems*, vol. 60, no. 9, pp. 1123–1134, 2012.
- [23] J. W. Park, H. J. Kwak, Y. C. Kang, and D. W. Kim, "Advanced Fuzzy Potential Field Method for Mobile Robot Obstacle Avoidance," *Computational Intelligence and Neuroscience*, vol. 2016, Article ID 6047906, 13 pages, 2016.
- [24] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 15, no. 1, pp. 116–132, 1985.
- [25] Y. Koren and J. Borenstein, "Potential field methods and their inherent limitations for mobile robot navigation," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1398–1404, April 1991.
- [26] S. S. Ge and Y. J. Cui, "Dynamic motion planning for mobile robots using potential field method," in *Proceedings of the 8th IEEE Mediterranean Conference on Control and Automation*, 2000.
- [27] S. O. Madgwick, AHRS algorithms and calibration solutions to facilitate new applications using low-cost MEMS, Electronic Thesis, University of Bristol: Bristol, UK 2014.



Hindawi

Submit your manuscripts at
www.hindawi.com

