

A heuristic obstacle avoidance algorithm using vanishing point and obstacle angle

Yongkuk Kim¹ · SangJoo Kwon¹

Received: 27 February 2015 / Accepted: 20 April 2015 / Published online: 28 April 2015
© Springer-Verlag Berlin Heidelberg 2015

Abstract Although there exists a class of algorithms for coping with unknown obstacles in mobile robot navigation, most of them produce rather conservative paths because the varying density of obstacles is not directly considered in the real-time motion planning stage. In this paper, we develop a heuristic obstacle avoidance method in terms of the vanishing point and obstacle angle (VP–OA) to compromise through an adjustable weighting factor between the lane tracking and the obstacle avoidance performance depending on the frequency of emerging obstacles. The suggested algorithm has the advantage of generating smooth local paths close to a human's car driving. Comparison simulations and experiments with other popular algorithms validate the effectiveness of the proposed scheme.

Keywords Obstacle avoidance · Mobile robot navigation · Vanishing point · Lane following

1 Introduction

In the global path planning, the autonomous mobile robot determines an optimal path based on the environmental map with open spaces and structured obstacles. However, local path planning is also necessary to cope with unexpected obstacles using sensory information in real time. When a mobile robot deviates from the expected global path or experiences an abrupt change of environment, it has to undergo a time-consuming process to generate a modified path. To help with this problem, an obstacle avoidance algorithm must be

considered as a local path planning approach, which is indispensable for the fast and safe navigation of a mobile robot in uncertain environments.

The previous studies on the obstacle avoidance for a mobile robot can be largely classified into the directional approach and the velocity space approach. In the directional approaches, a best moving direction is determined every sampling time in terms of the certainty values on the existence of obstacles for a designated area around the robot. As some representative examples, the vector field histogram (VFH) method [1] divides the surroundings of the robot into small sectors and determines the next moving direction of the robot based on the obstacle density in the polar histogram. The VFH+ version [2] enables smooth motions to avoid obstacles by considering the possible radii of turns and the robot dimensions in the configuration space. In succession, the VFH* method [3,4] was suggested to solve the problem that some local obstacles lead the robot to dead ends by incorporating A* algorithm which performs a look-ahead verification. As another directional approach, the nearness diagram (ND) algorithm [5] is beneficial in crowded areas, where the surroundings are divided into five areas by considering the proximity to obstacles and available spaces, and the desirable robot motions are determined for each region.

As a different framework for obstacle avoidance in the velocity space, the dynamic window approach (DWA) [6] establishes a search space which consists of the admissible velocities for the robot under dynamic constraints, where the optimal velocities to avoid obstacles are found within the dynamic window. However, sometimes it may suffer from the local minimum problem that the robot cannot find a moving direction at dead ends, which was investigated in the global DWA [7] in conjunction with an independent navigation function. The convergence of the DWA algorithm has been improved by applying the model predictive con-

✉ SangJoo Kwon
sjkwon@kau.ac.kr

¹ School of Aerospace and Mechanical Engineering, Korea Aerospace University, Goyang 412-791, Korea

trol technique and using the control Lyapunov function [8]. Also, the DWA has been integrated with the D* search algorithm for handling densely populated moving obstacles [9] and extended to a 3D planning for arbitrarily shaped mobile robots [10]. Similarly, the curvature velocity method in [11] also seeks to find optimal velocities to avoid obstacles in the velocity space. Also, a few combined algorithms for the directional approach and the velocity space approaches were developed in [12, 13].

In exploring uncertain environments, mobile robots usually experience the change of obstacle environment. Hence, if we can take the varying density of obstacles into consideration in the motion planning stage, a more natural path such as mimicking human's car driving might be generated. For example, when there are no obstacles in the course of navigation, it is wise to choose a minimum distance or a minimum time trajectory. When the obstacles are rarely found, it is reasonable to put more emphasis on the target tracking performance rather than on the obstacle avoidance to keep the velocity decrease of the robot as small as possible. On the contrary, in congested areas with emerging obstacles, the main concern must be how to arrive at the final goal safely by choosing a conservative path to be certain to avoid collisions with obstacles. However, most of the currently available algorithms do not explicitly consider the real-time change of the obstacle population in generating the collision avoidance motion of mobile robots. Although some of them choose the movable and unmovable directions of the robot by controlling the threshold for the certainty value of obstacle existence [1–4] or the coefficients involved in the objective functions [6–9], those parameters are fixed regardless of the obstacle distribution in reality.

To address this issue, this paper investigates how to accomplish a dexterous driving of a mobile robot by reflecting the obstacle density in the collision avoidance problem. First, we define a vanishing point and obstacle angle (VP–OA) in Sect. 2 as a heuristic measure to compromise between the driving and obstacle avoidance performances. Then, the VP–OA-based obstacle avoidance scheme is proposed in Sect. 3, where the frequency of obstacles in a designated search area is considered through a weighting factor which is adjustable in real time. Through the comparison simulations with representative algorithms in Sect. 4 and the experimental results in Sect. 5, it is shown that the suggested obstacle avoidance algorithm is very efficient and powerful. Finally, the conclusion is drawn in Sect. 6.

2 Vanishing point and obstacle angle

2.1 Lane tracking using vanishing point angle

The vanishing point (VP) is defined as a single point where the two straight lines in a 3D space along the boundaries of a

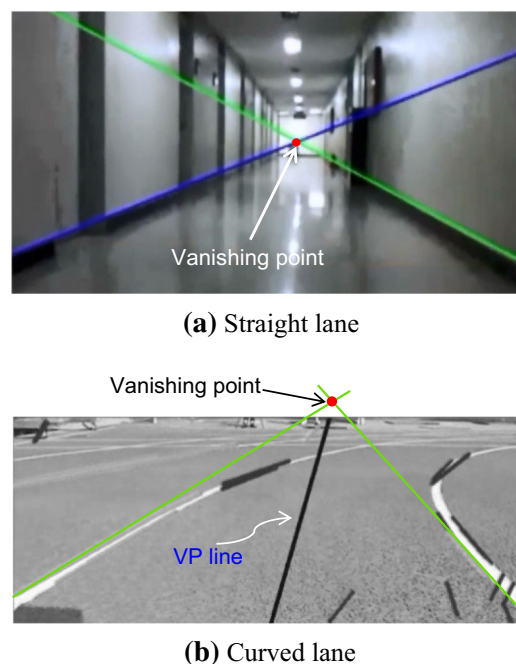


Fig. 1 Extraction of vanishing points using boundary and lane detection. **a** Straight lane, **b** curved lane

lane meet when they are projected onto a 2D image according to the sense of perspective. For example, the boundaries at both sides of the indoor passage in Fig. 1a are parallel in reality, but the extracted and extended lines from the captured image gather at one point. In a curved lane like Fig. 1b, the vanishing point can be readily determined as far as the two tangential lines at both sides are given through the image processing. The vanishing point technique was popularly utilized in many applications concerned with the mobile robot navigation including camera calibration [14], vision-based attitude estimation [15], obstacle detection [16], and lane tracking problems [17]. Tracking the vanishing points enables the mobile robot to explore an unknown environment along the boundaries of indoor passages or outdoor roads.

As schematically described in Fig. 2, the current pose of the robot with respect to a specific lane can be represented by the VP line and VP angle. If the robot approaches one side of a lane, the central axis of the image captured from the robot camera is coincident with the moving direction of the robot. The virtual VP line connects the robot to the vanishing point, where the angle between the central axis and the VP line is defined as the VP angle. As far as the moving direction of the robot is not parallel to the local lanes and is not located at the exact center between the left and right boundaries of the lanes, the VP angle always happens with nonzero values. If the wheel velocities of the robot are controlled to make the VP angle converge to zero, the robot is naturally driven to the central part of the passage by the steering action. Actually, it is the same as reducing the VP

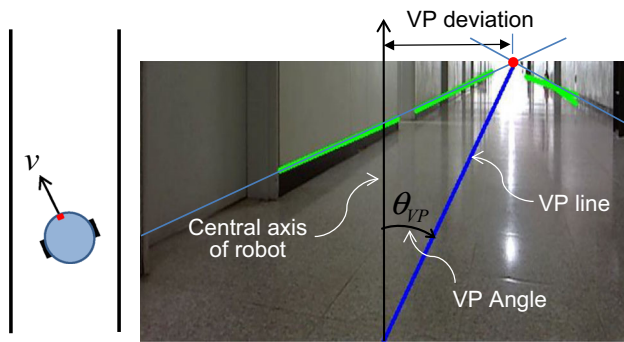


Fig. 2 Definition of the vanishing point angle (VP angle)

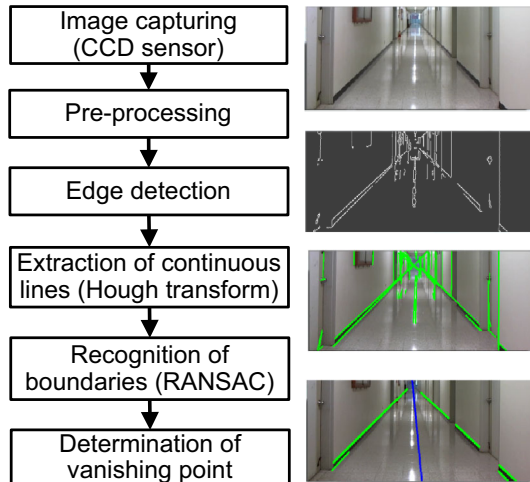


Fig. 3 Image processing procedure to find the vanishing points

deviation denoted in Fig. 2 to zero as investigated in [18]. The image processing procedure to find the vanishing points is denoted in Fig. 3, where the virtual lines at both boundaries are updated every sampling time and the RANSAC algorithm [19] can be applied to recognize the final boundaries from a lot of candidates of continuous lines.

2.2 Obstacle detection area

For computational efficiency, it is necessary to restrict the area to search for obstacles considering the current speed of the robot. In Fig. 4, we define the sectors beginning at the distance sensor as the obstacle detection area (ODA), where the radius is varied in real time as a function of the current speed:

$$R = \left(\frac{R_{\max} - R_{\min}}{v_{\max} - v_{\min}} \right) (v - v_{\max}) + R_{\max} \quad (1)$$

with the pre-specified maximum and minimum radii (R_{\max} , R_{\min}) and the velocities (v_{\max} , v_{\min}). In the where the ODA area has a constant value of A , the central angle of ODA can be determined as

$$A = \theta_{\text{ODA}} R^2 / 2 \rightarrow \theta_{\text{ODA}} = 2A / R^2 \quad (2)$$

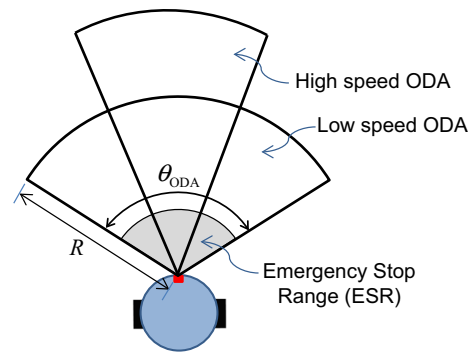


Fig. 4 Fan-shaped obstacle detection area (ODA)

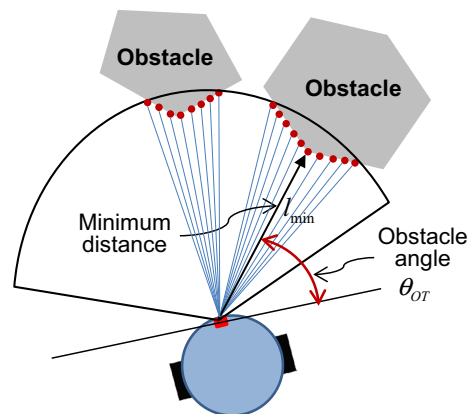


Fig. 5 Definition of obstacle angle (OT angle)

Consequently, the ODA at low speeds has a large central angle but a small radius to detect a wide range of surroundings. At high speeds, however, it has a small central angle but a large radius to detect distant objects in advance. If the ODA area is too large, it will increase the computation time accordingly and it may invoke excessive steering motions for any possible collisions. On the contrary, if the ODA is too small, it is hard to cope with abruptly emerging obstacles. It is also necessary to consider the robot's dimensions in establishing the ODA. For example, the width of the ODA must be at least larger than the distance between the wheels. In fact, a fundamental approach for considering the dimensional effect of the robot in the collision avoidance problems is to extend the dimension of obstacles for any pose of the robot in the configuration space as remarked in [3,9].

2.3 Collision avoidance using obstacle angle

As shown in Fig. 5, when the obstacles are detected by the distance sensor such as the laser range finder or ultrasonic sensor, the outer contour points of the obstacles in a specific region of ODA can be saved as many points as the sensor resolution. Here, we define the obstacle angle (OA) as the angle between the horizontal line of the robot and the mini-

minimum distance vector to the contour points. As long as there are no obstacles inside the ODA, the obstacle angle always remains zero. When some objects are detected in the ODA, the minimum distance is compared with the emergency stop range (ESR) denoted in Fig. 4 to judge whether to conduct collision avoidance motion or not. If the minimum distance is smaller than a specified radius of ESR, the mobile robot must make an emergency stop for the safety. Otherwise, it is directed to avoid obstacles by regulating the obstacle angle to zero, which is detailed in the following section.

3 VP–OA-based obstacle avoidance

3.1 Collision avoidance motion planning

As denoted in Fig. 6, the VP angle has a positive value in the clockwise direction with respect to the vertical axis of the robot and negative in the counterclockwise direction. Similarly, the direction of the obstacle angle is defined with respect to the horizontal axis. At a specific sampling time, if the minimum distance to obstacles exists on the right side from the vertical axis, the robot will steer to the left to follow a collision-free path until the obstacle angle reduces to zero. In the same way, if the minimum distance occurs at the left obstacle, the robot will move to the right.

In the lane tracking navigation, the VP tracking performance of a mobile robot is dependent upon how quickly the VP angle recovers to zero. In Fig. 6, if the VP line is toward the right side, the robot turns to the right and if it heads for the left side, the robot turns to the left. As the VP angle converges to zero, the robot transfers to the central part of the passage. To make the mobile robot keep lane tracking while avoiding obstacles, the VP angle and the obstacle angle must converge to zero simultaneously. However, in the course of traveling,

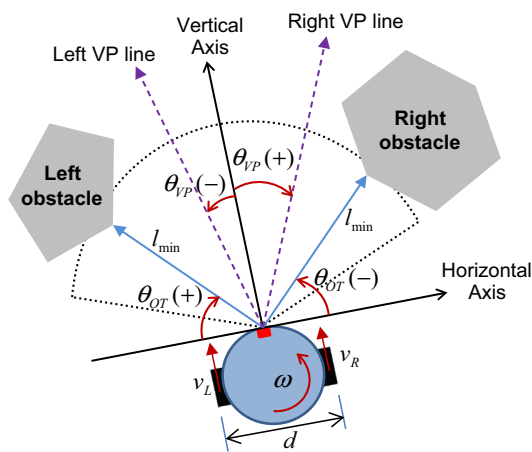


Fig. 6 Positive and negative directions of VP angle and obstacle angle

the convergence of the two angles will conflict frequently and they need to be compromised by following a certain rule.

First, we define the steering angle of the robot as the sum of the VP angle and the weighted obstacle angle:

$$\phi = \theta_{VP} + k_{OT}\theta_{OT} \quad (3)$$

where the weighting factor is to penalize the lane tracking performance to guarantee a safe collision avoidance. Then, in the case of a differential type mobile robot, the velocity commands of the left and right wheels can be determined by

$$v_L = V_{ref} + (K_P\phi + K_D\dot{\phi}), \quad v_R = V_{ref} - (K_P\phi + K_D\dot{\phi}) \quad (4)$$

to regulate the steering angle to zero as the robot moves forward, where V_{ref} is the reference velocity and (K_P, K_D) are the proportional and derivative control gains. The forward velocity and the steering rate of the robot are constrained by the relationships:

$$v = \frac{v_L + v_R}{2}, \quad \omega = \frac{v_R - v_L}{d} \quad (5)$$

with d the distance between the wheels. Hence, the steering direction of the robot will be alternated along with the sign change of the steering angle.

When the VP–OA-based navigation algorithm is applied to a mobile robot, it is expected to trace the collision-free path described in Fig. 7. At each traveling stage, the VP and obstacle angles are determined depending on the pose and velocity of the robot and the distance to the obstacles. First, in the case where no obstacles exist in the ODA, only the VP angle contributes to the required steering angle in (3). In the second stage where the robot detects an obstacle in the ODA, the obstacle angle jumps to a certain value and gradually decreases until the obstacle disappears from the ODA. Finally, as the VP angle decreases to zero, the robot returns to the central part of the way and naturally recovers straight driving along the lanes. For the obstacles and boundaries in Fig. 7, they can be thought of as enlarged ones in the configuration space (or C-space) [3, 9], where the robot's dimensions are reflected into the individual obstacles and the robot is transformed into a single point in the space.

3.2 Generation of humanlike driving motion

Considering the dynamic characteristics and the limit of actuation power, the steering rate and the forward velocity of the mobile robot are required to be limited to certain maximum values. To achieve smooth turns and prevent turnovers, it is also necessary to adjust the velocity reference in the control loop depending on the current steering rate. For example, we have

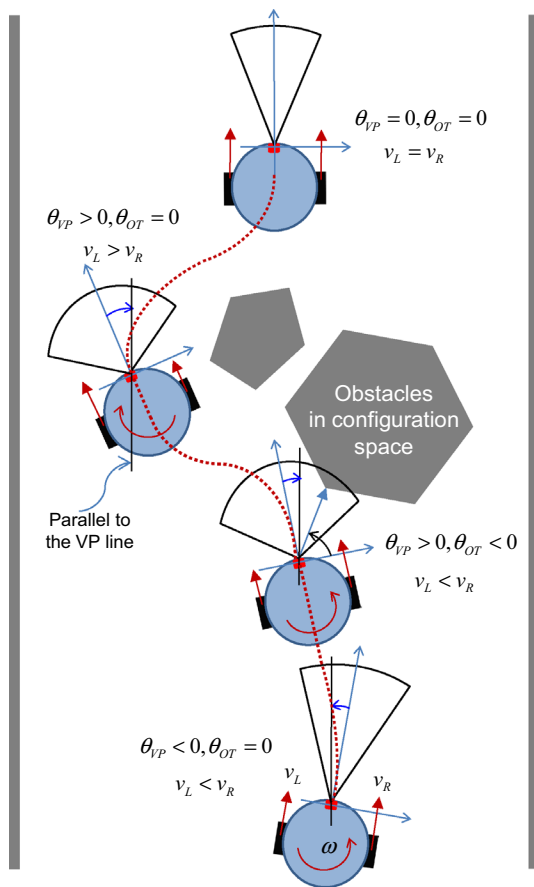


Fig. 7 Collision-free local path generation by the VP–OA-based obstacle avoidance algorithm

$$V_{\text{ref}} = \begin{cases} (V_{\text{max}} - V_{\text{min}})(1 - \frac{|\omega|}{\omega_{\text{max}}}) + V_{\text{min}} & (|\omega| < \omega_{\text{max}}) \\ V_{\text{min}} & (|\omega| \geq \omega_{\text{max}}) \end{cases} \quad (6)$$

where $(V_{\text{max}}, \omega_{\text{max}})$ are the pre-specified maximum values for the forward velocity and steering rate of the robot. When the steering angle gets large due to a sudden appearance of an obstacle or a lane change, it results in a large steering rate in the robot motion. Then, the reference velocity will be lessened as much. Only when the robot is running parallel to the straight lane with no steering angle does the reference velocity maintain the same value as the pre-specified maximum value.

This approach is similar to human car driving which slows down the speed when entering curves and speeds up when exiting them. As we instinctively reduce the car speed in heavily congested areas with many obstacles, it is necessary for the mobile robot to slow down when it meets the same situation for safe collision avoidance. It can be achieved by allowing the weighting factor for the obstacle angle in (3) to be adjustable in real time. For instance, we have

$$k_{OT} = \begin{cases} \left(\frac{N}{N_{\text{max}}}\right) \times \left(\frac{R}{l_{\text{min}}}\right) & \text{if obstacles exist in ODA} \\ 0 & \text{if no obstacles in ODA} \end{cases} \quad (7)$$

where R is the radius of the ODA in Fig. 4 at the current state, N_{max} is the maximum number of points along the arc of the ODA which is detectable according to the distance sensor resolution, and N the number of points actually detected along the obstacle contours in ODA. As the obstacles that occupy the ODA gets larger and the minimum distance to the obstacles gets smaller, the steering angle gets larger due to the increased weighting factor. Then, a higher steering rate will be assigned to the robot according to (4). Successively, the reference velocity for the robot gets decreased by the rule in (6).

With a large weighting factor, it bothers the lane following performance of the robot because it decreases the forward velocity and sometimes generates excessive steering motions. Roughly speaking, when $0 < k_{OT} < 1$ with rare obstacles, the VP angle dominates the steering angle and the robot is more faithful to the lane following behavior rather than obstacle avoidance. However, when $k_{OT} > 1$ in populated obstacles, the obstacle angle influences the steering angle more than the VP angle and the main priority in the robot motion is to escape from that region safely without any collision. Actually, it is a great advantage of the VP–OA-based approach that a mobile robot is able to cope with the variation of the obstacle density through the auto-tuned weighting parameter.

4 Performance simulation

4.1 VFH, DWA, and VP–OA approach

Among the existing obstacle avoidance methods for mobile robots, two representative ones are the class of VFH methods to find a moving direction in the Cartesian space (or configuration space) and the class of DWAs formulated in the velocity space. In the VFH methods [1–4], the obstacle information in the two-dimensional histogram is transformed into a one-dimensional polar histogram, which divides the entire 360 degrees around the robot into small sectors and stores the polar obstacle density (POD) based on the certainty value. As described in Fig. 8, the sectors are clustered into the valleys of movable (bright) and unmovable groups (dark) by applying a threshold to the POD. Among the valleys beyond a certain width considering robot dimension, the nearest path to the goal is selected and the corresponding steering angle is determined. The performance of the VFH methods greatly depends on the number of sectors and the pre-determined threshold value [20].

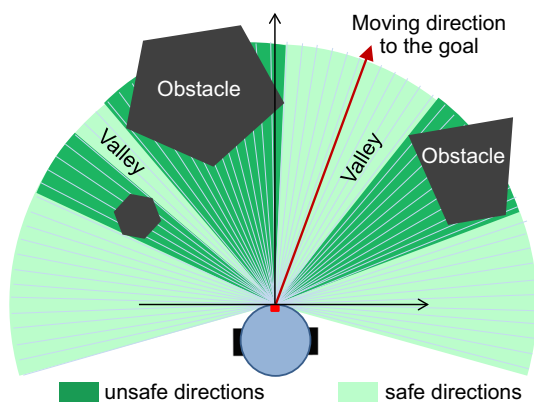


Fig. 8 Schematic of the VFH class algorithms

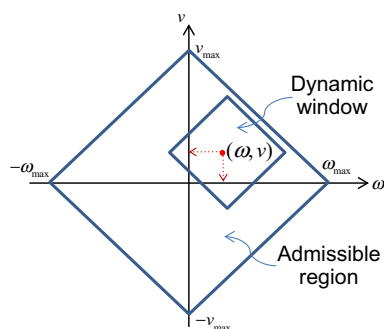


Fig. 9 Schematic of the dynamic window

In the class of DWA algorithms [6–10], a dynamic window shown in Fig. 9 is created every sampling time as a certain region in the velocity space which can be entered into at the next sampling time by considering the current velocity and steering rate of the robot and the torque limit of wheel actuators. Then, among the velocity sets in the dynamic window, it determines a pair of forward velocity and steering rate (v, ω) which maximizes an objective function in the form of

$$G(v, \omega) = \alpha \cdot (1 - \text{heading}(v, \omega)) + \beta \cdot \text{distance}(v, \omega) + \gamma \cdot \text{velocity}(v, \omega) \quad (8)$$

where the sub-functions representing the steering angle, distance to obstacles, and forward velocity, respectively, are normalized ones. The DWAs are advantageous for implementing fast and smooth navigation of a mobile robot because of its small computing power requirement.

As the threshold value to separate the movable areas gets smaller in the VFH methods, the resulting local paths will become more conservative for safe collision avoidance. Similarly, the coefficients of the objective functions in the DWA methods determine the configuration of the steering motion. Since the thresholds in the VFH methods and the weighting coefficients in the DWAs usually have fixed values, it is difficult to cover various environments with varying obstacle

populations. On the contrary, the VP–OA approach proposed in this paper enables the mobile robot to conduct humanlike driving motions by coordinating the lane following performance and the obstacle avoidance performance through the flexible weighting factor.

4.2 Comparative simulation

The developed VP–OA algorithm is compared with the VFH and DWA methods through numerical simulations. In Fig. 10, the cylindrical obstacles with 1 m diameter are randomly distributed along a straight route 20 m long and 8 m wide. The reference velocity of the robot for the three algorithms commonly varies between the maximum 0.5 m/s and the minimum 0.1 m/s.

In applying the VFH algorithm, the certainty grid around the robot has 33 by 33 cells where the magnitude of each cell is 10 by 10 cm, the POD sectors angle is one degree, and the other parameters follow the values as were given in [1], which include the smoothing function parameter $l = 25$,

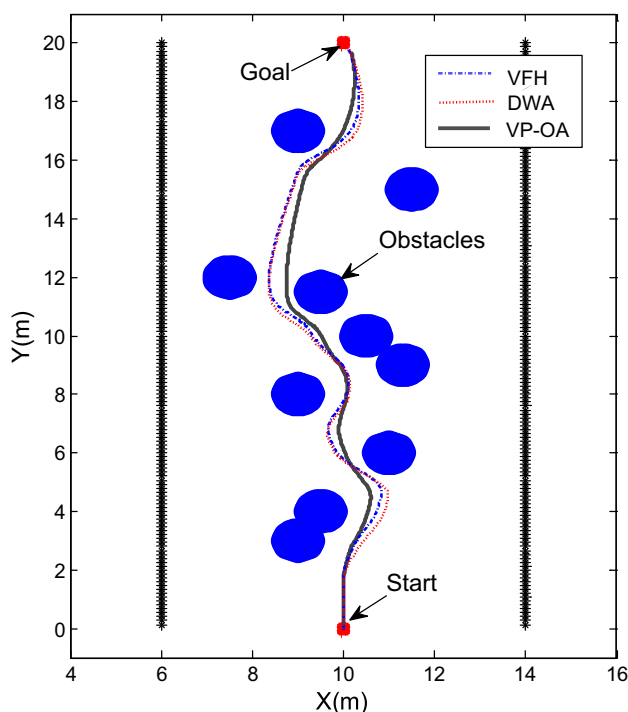


Fig. 10 Comparison of obstacle avoidance trajectories (simulation)

Table 1 Performance comparison (simulation)

	Elapse time (s)	Traveling distance (m)	Minimum distance (m)
VP–OA	48.1	20.55	0.49
VFH	52.7	21.46	0.53
DWA	53.5	21.69	0.52

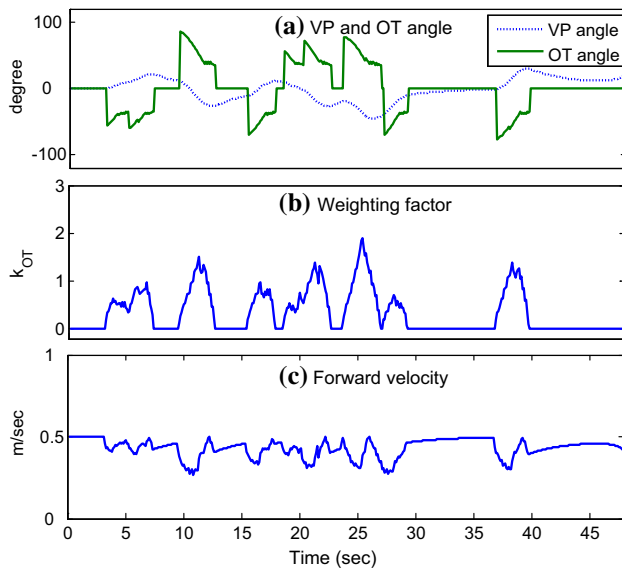


Fig. 11 Variation of the VP angle, obstacle angle, and weighting factor (simulation)

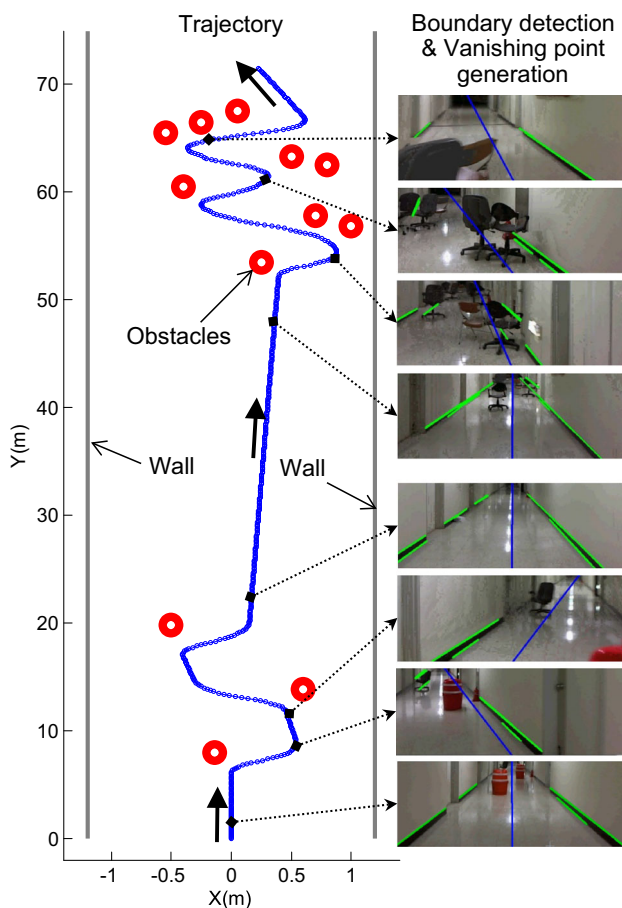


Fig. 12 Indoor navigation of a mobile robot applying the VP-OA algorithm

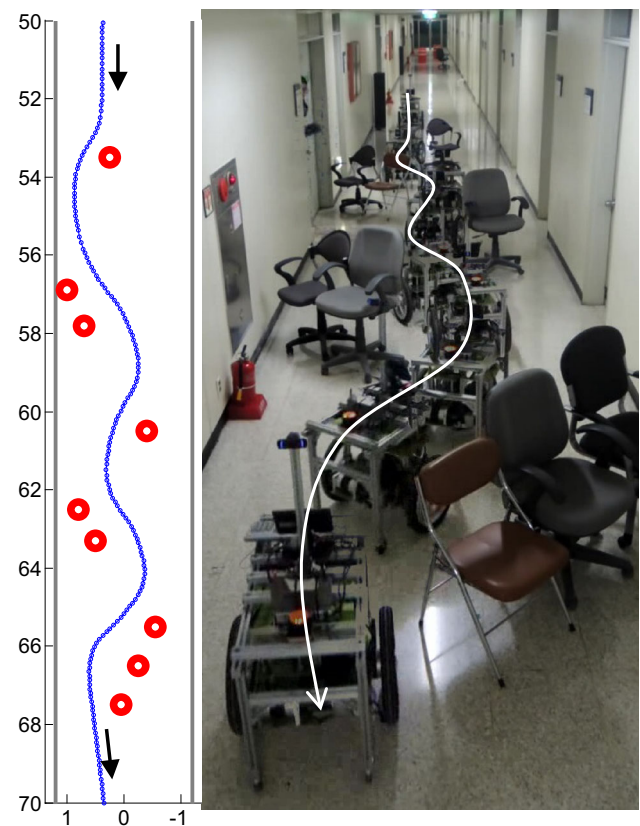


Fig. 13 Obstacle avoidance motion with the VP-OA algorithm

the steering control parameter $s_{\max} = 90$, and the allowable angular velocity $\alpha_{\max} = 90 \text{ deg/s}$. For the DWA algorithm, the weighting parameters in (8) were determined as $\alpha = 0.2$, $\beta = 2$, $\gamma = 0.2$ by following the suggestion in [6].

As a result, the obstacle avoidance paths for the three algorithms show similar patterns. However, it can be confirmed that the VP-OA approach produces smoother local paths due to the flexible weighting factor and it considerably reduces the total elapse time, traveling distance, and the minimum distance to obstacles, as is shown in Table 1. As indicated in Fig. 11, the VP angle happens and converges to zero as the robot is tracking the vanishing point, and the obstacle angle jumps to peak values when the obstacles are detected in ODA and soon disappears as soon as the robot escapes from them. As well, the weighting factor varies depending on the obstacle density in local areas and it becomes zero when there are no obstacles in ODA, which makes the distinctive difference of the VP-OA algorithm from the other ones.

5 Experiment

To validate the VP-OA obstacle avoidance algorithm, it has been applied to a differential type mobile robot with two

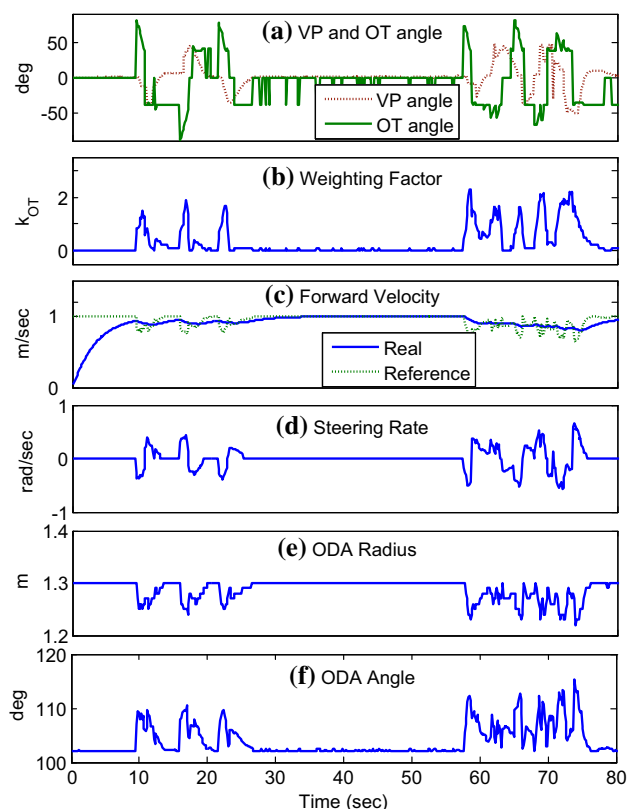


Fig. 14 Variation of the parameters and state variables

active wheels, which is equipped with a webcam for lane following navigation and a laser range finder for obstacle detection. The mobile robot navigation was performed in an indoor passage about 70 m long and 2.5 m wide, where the obstacles were distributed as shown in Figs. 12 and 13. For the initialization of the algorithm, the maximum velocity of the mobile robot was fixed as 1 m/s and the minimum velocity as 0.1 m/s and also the initial radius of ODA as 1.3 m.

The trajectories in Figs. 12, 13, 14 demonstrate the agile motion of the mobile robot to avoid obstacles while tracking the lane, specifically in the last section of high obstacle density as shown in Fig. 13. In Fig. 12, the vanishing points are successfully generated according to the image processing procedure suggested in Fig. 3. In Fig. 14, it is clear that the weighting factor has higher values as the obstacles become more congested, where in the middle section with no obstacles, the VP angle stays near zero, but the obstacle angle shows a little chattering because of the effect of the wall. As the steering rate increases due to the high VP and obstacle angles, the forward velocity is accordingly lessened by following the rule of (6) to achieve a humanlike driving motion, which makes the magnitudes of the ODA radius and the ODA angle alternated according to (1) and (2). Reducing the ODA radius but increasing the angle is similar to decreasing the visual point but increasing the field of view to turn to the left

or to the right when an obstacle appears in front of a human driver.

In an open space without any clear boundaries, it may not be adequate for a mobile robot to perform a lane following navigation. However, if a waypoint or a final goal are given through the global path planning, they can be regarded as the vanishing points to follow in the VP–OA algorithm. Even though the VP angle is zero because no vanishing points are defined, there is no problem to generate local paths for obstacle avoidance according to the steering angle in (3).

6 Conclusion

In this paper, we have suggested an efficient obstacle avoidance algorithm heuristically developed based on the notion of vanishing point and obstacle angle. Specifically, it is useful to accomplish a humanlike driving motion of a mobile robot in the lane following navigation, where the obstacle avoidance and lane tracking performance can be naturally compromised through the weighting parameter depending on the varying obstacle density. The proposed VP–OA obstacle avoidance scheme can be readily combined with a global path planning to have an optimal path. Finally, a modified version of the VP–OA algorithm to escape from the local minimum problem at dead ends can be implemented by providing a prediction function for global search.

Acknowledgments This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF-2012R1A1B3003886).

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Bornstein J, Koren Y (1991) The vector field histogram-fast obstacle avoidance for mobile robots. *IEEE Trans Robot Autom* 7(3):278–288
2. Ulrich I, Borenstein J (1998) VFH+: reliable obstacle avoidance for fast mobile robots. In: *Proceedings of the 1998 IEEE international conference on robotics and automation (ICRA)*
3. Ulrich I, Borenstein J (2000) VFH: Local obstacle avoidance with look-ahead verification. In: *Proceedings of the 2000 IEEE international conference on robotics and automation (ICRA)*, pp 2505–2511
4. Shoval S, Ulrich I, Borenstein J (2003) Robotics-based obstacle-avoidance systems for the blind and visually impaired. *IEEE Robot Autom Mag* 10(1):9–20
5. Minguez J, Montano L (2004) Nearness diagram (ND): collision avoidance in troublesome scenarios. *IEEE Trans Robot Autom* 20(1):45–59
6. Fox D, Burgard W, Thrun S (1997) The dynamic window approach to collision avoidance. *IEEE robotics and automation magazine*, pp 23–33

7. Brock O, Khatib O (1999) High-speed navigation using the global dynamic window approach. In: Proceedings of the 1999 IEEE international conference on robotics and automation (ICRA), vol 1, pp 341–346
8. Ogren P, Leonard NE (2005) A convergent dynamic window approach to collision avoidance. *IEEE Trans Robot* 21(2):188–195
9. Seder M, Petrovi I (2007) Dynamic window based approach to mobile robot motion control in the presence of moving obstacles. In: Proceedings of the 2007 IEEE international conference on robotics and automation (ICRA), pp 1986–1991
10. Demeester E, Nuttin M, Vanhooydonck D, Van Brussel H (2005) Global dynamic window approach for holonomic and non-holonomic mobile robots with arbitrary cross-section. In: Proceedings of the 2005 IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 2357–2362
11. Simmons R (1997) The curvature-velocity method for local obstacle avoidance. In: Proceedings of the 1997 IEEE international conference on robotics and automation (ICRA), pp 23–33
12. Li G, Wu G, Wei W (2006) ND-DWA: a reactive method for collision avoidance in troublesome scenarios. In: Proceedings of the sixth world congress on intelligent control and automation (WCICA 2006), vol 2, pp 9307–9311
13. Chou CC, Lian FL, Wang CC (2011) Characterizing indoor environment for robot navigation using velocity space approach with region analysis and look-ahead verification. *IEEE Trans Instrum Meas* 60(2):442–451
14. Caprile B, Torre V (1990) Using vanishing point for camera calibration. *Int J Comput Vis* 4(2):127–139
15. Kessler C, Ascher C, Frietsch N, Weinmann M, Trommer GF (2010) Vision-based attitude estimation for indoor navigation using vanishing points and lines. In: Proceedings of 2010 IEEE/ION position location and navigation symposium (PLANS), pp 310–318
16. Nair D, Aggarwal JK (1998) Moving obstacle detection from navigating robot. *IEEE Trans Robot Autom* 14(3):404–416
17. Chang CK, Siagian C, Itti L (2012) Mobile robot monocular vision navigation based on road region and boundary estimation. In: Proceedings of the 2012 IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 1043–1050
18. Vassallo RF, Schneebeli HJ, Santos-Victor J (2000) Visual servoing and appearance for navigation. *Robot Auton Syst* 31(1):87–97
19. Bazin JC, Pollefeys M (2012) 3-line RANSAC for orthogonal vanishing point detection. In: Proceedings of the 2012 IEEE/RSJ international conference on intelligent robots and systems (IROS), pp 4282–4287
20. Yim WJ, Park JB (2014) Analysis of mobile robot navigation using vector field histogram according to the number of sectors, the robot speed and the width of the path. In: Proceedings of 2014 14th international conference on control, automation, and systems, pp 1037–1040