Velocity Grid Map Approach and Its Application to Collision-Free Navigation

Tomoshi Yamashita, Jae Hoon Lee, *Member, IAENG* and Shingo Okamoto

***Abstract*— This research proposes a new approach to recognize dynamic circumstance near a mobile robot using a laser scanner. It calculates the velocity of moving objects, using the time history of laser scan data, on a grid map. In addition, it implements collision-free navigation for an indoor mobile robot. For this function, the optimal motion for the robot is computed according to a safety index based on the proposed grid map, including velocity and obstacle position information. An algorithm using fuzzy theory is employed to reflect the characteristics of human motion planning to the robot. Computer simulation and experimental works were carried out to check the feasibility of the proposed algorithm for collision-free navigation of the indoor mobile robot.**

***Index Terms*— Fuzzy theory, Grid map, Collision avoidance, Laser scanner**

I. Introduction

UNLIKE typical robots used for industrial applications, recent robot technologies are advancing for service tasks in the human daily environment of the near future. Collision-free navigation is considered as a particularly essential capability for a robot that coexists with a human. Recently, there have been many studies carried out on this topic.

Representative examples of these follow. Fox et al. proposed a dynamic window approach for collision avoidance [1]. A navigation algorithm, called Nearness Diagram (ND), which enable a robot to move successfully in troublesome scenarios, was proposed by Minguez [2]. Huang et al. proposed a navigation method based on a potential field for a robot using an omni-directional camera [3]. LEE proposed a motion planning system using fuzzy theory that determined the priorities of thirteen possible heading directions using two ultrasonic modules [4]. Yang investigated “Fuzzy–Braitenberg Navigation Strategy” using the concept of Braitenberg vehicles [5]. A collision-free navigation for multi-agent systems was also investigated by Ono et al [6]. Jaradat et al. proposed a hybrid navigation system by combining the theories of fuzzy and potential field [7].

The underlying assumption of the above researches is that the robot has the capability to recognize its surrounding area and acquire sufficient information about the motion of nearby obstacles using its embedded sensors. However, it is difficult to realize a dynamic environment in an actual experiment. Therefore, methods to recognize obstacles near the robot have also been investigated actively. For example, Bis et al. proposed a method using velocity occupancy space that allows a robot to avoid moving obstacles [8]. Tamura et al. investigated an algorithm to recognize obstacles and classify them as three types, stationary, movable, and moving, using an obstacle map [9].

Generally, human sensory systems cannot provide, with high accuracy, the exact position or velocity information related to circumstance conditions, including static and moving obstacles. However, humans can move safely in a dynamic environment using skillful motion intelligence and different sensory systems such as stereo eyes. This study focuses on the application of the human collision-avoidance method to the robot. It assumes that a human decides his moving direction intuitively based on the conditions related to safety around him and the direction to his goal. To realize this objective, we feel that fuzzy theory is the appropriate algorithm to process the ambiguous information and plan a safe motion for the robot navigation.

The objective of this research, therefore, is to apply human intelligence to a collision-avoidance algorithm for the mobile robot. To achieve this, a method to calculate the velocity and motion change of obstacles using a Velocity Grid Map is proposed. In addition, this is applied to a collision-avoidance system based on fuzzy theory. Based on the proposed method, the mobile robot can recognize circumstance conditions and move to the goal position autonomously. The feasibility of the proposed method is demonstrated through computer simulations and experiments of collision-free navigation in an indoor, narrow, dynamic environment, with two people.

II. Mobile Robot System

A. Configuration of the Mobile Robot

The mobile robot used in this research is shown in Fig. 1. Its motion is generated by two independent active wheels driven by DC motors. A Laser Range Finder (LRF, model: UTM-30LX, made by Hokuyo Co., Japan) is installed on its upper plate as the sensor to obtain ‘distance to obstacle‘s information; its height is about 250 mm. The size of this mobile robot is Width: 306 mm, Depth: 229 mm, Height: 295 mm. The mass is approximately 3.0 kg.

The simplified configuration of the system is shown in Fig. 2. Both wheels of the robot are controlled by the embedded controller according to a velocity motion command given from the laptop computer on the platform. The robot position is computed in the controller using odometry data and

Manuscript received December 23, 2013; revised January 30, 2014. This work was supported by JKA and its promotion funds from Auto Race*.

Every author is with Mechanical Engineering Course, Graduate School of Science and Engineering, Ehime University, 3 Bunkyo-cho, Matsuyama, Ehime, 790-8577, Japan.
Tomoshi Yamashita (e-mail: y840026b@mails.cc.ehime-u.ac.jp).
Jae Hoon Lee (e-mail: jhlee@ehime-u.ac.jp).
Shingo Okamoto (e-mail: okamoto.shingo.mh@ehime-u.ac.jp).

transferred to the computer. The range data from the LRF is also transferred to the computer. Thus, this unit is utilized as the main computer to calculate the algorithms for recognition and motion planning.

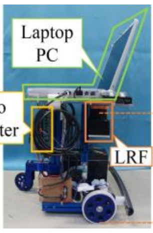


Fig. 1. Mobile robot.

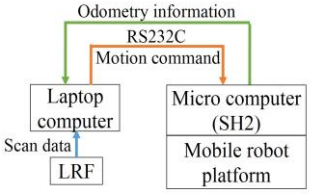


Fig. 2. Configuration of the mobile robot.

B. Kinematic Model of the Mobile Robot

The kinematic diagram of the mobile robot is shown in Fig.3, where O-XY denotes the fixed world coordinate system and o-xy denotes the moving coordinate system attached at the center of the robot. The position and orientation of the mobile robot are represented by (x_R,y_R,θ_R) with respect to the fixed coordinate system. The velocity kinematics of the differential drive mobile robot are given by (1), where v_R and ω_R denote the translational and angular velocity of the mobile robot with respect to the moving coordinate system, ω_l and ω_r are the angular velocities of the left and right wheels, R_w is the radius of both wheels, and T is the tread between the wheels. The position of the robot, odometry, is computed by integrating the velocities based on (1) with respect to the fixed coordinate system.

$$\begin{pmatrix} v_R \\ \omega_R \end{pmatrix} = \begin{pmatrix} R_W/2 & R_W/2 \\ R_W/T & -R_W/T \end{pmatrix} \begin{pmatrix} \omega_r \\ \omega_l \end{pmatrix} \qquad (1)$$

For the control of the mobile robot, the reference angular velocities of both wheels are calculated using (2) with the reference velocity for the robot. This is the inverse kinematic relationship of (1).

$$\begin{pmatrix} \omega_r^{ref} \\ \omega_l^{ref} \end{pmatrix} = \begin{pmatrix} 1/R_w & T/2R_w \\ 1/R_w & -T/R_w \end{pmatrix} \begin{pmatrix} v_R^{ref} \\ \omega_R^{ref} \end{pmatrix} \qquad (2)$$

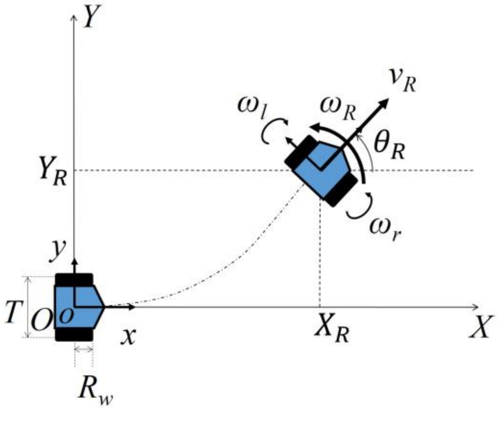


Fig. 3. Kinematic description of the mobile robot.

III. Velocity Calculation Algorithm using Velocity Grid Map

This section describes the method used to compute an obstacle's velocity that is used for recognizing nearby environmental conditions.

A. Outline of Velocity Grid Map

To recognize the motion of people around the robot, a method to track each individual human is employed. However, this requires high-level technologies and computing power in real applications. In addition, this is fundamentally different from human navigation methods that utilize, not exact individual tracking results, but somewhat ambiguous human feel to sense the motions of moving obstacles, for example, velocity. To realize the condition near the robot like as the human manner, a 'Velocity Grid Map' is proposed in this paper.

This algorithm utilizes the history of the range data around the robot. In order to acquire the velocities of the obstacles, the range data is calculated on the grid map as shown in Fig. 4. o-$x_g y_g$ denotes the grid map coordinate system. Unlike typical tracking methods, it does not track each individual object but computes the motion change of condition with the range data accumulated during some time interval. The size of each grid is set to 500 mm × 500 mm. In this research entire grid map is 6 horizontal × 20 vertical grids. Additionally, the grid map has four layers to retain four time intervals of range data where the oldest data is in the first layer, and the newest data is in the fourth layer, respectively. The newest layer is updated with the current scan data for each time step. The oldest layer is deleted and the other three layers are shifted accordingly.

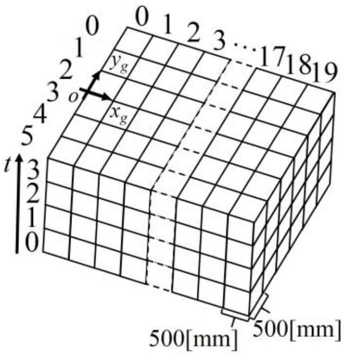


Fig. 4. Grid map for plotting 4 times scan data.

It could happen that the data around the borderline between the grids could cause an error in computing the velocity, owing to a sudden change; in particular, some of them could be suddenly swapped between two grids. Therefore, the divided data could have a negative effect on the velocity calculation. To address this situation, each layer of the grid map has two overlapped grids as shown in Fig. 5. The offset distance is set to the half-length (250 mm) of the grid in the X and Y-directions. As a result, the range data can be positioned in the center of the grid.

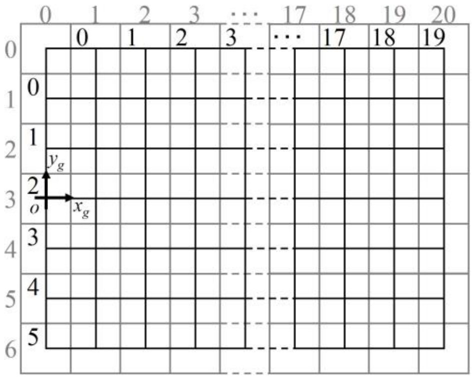


Fig. 5. A layer of the velocity grid map including two overlapped grid maps.

B. Preprocessing to Extract Obstacle Part

It is assumed that the robot moves in an area whose map is prepared in advance in the form of range data. In order to reduce computation time, the object part included in the range data is extracted by comparing it to a predefined map. The process is explained in detail as follows.

STEP 1: The map data closest to each range data is found. They are then associated with each other.

STEP 2: The distance between the two points is calculated.

STEP 3: Based on a threshold distance, the range data similar to the map data is considered to be from a static object. Conversely, the range data that is far from the map data is considered a moving object. The threshold is set to 300 mm in this research.

STEP 4: Steps 1 to 3 are repeated for all range data scanned in the time interval.

C. Calculation of Obstacle's Velocity on Grid Map

The obstacle's velocity is calculated using the positional relationship between the average coordinate value of the newest and the oldest object data in one grid. The calculation is carried out for each grid and repeated for all grids. Its detailed explanation follows.

STEP 1: The average coordinate value of the range data for each grid is calculated using (3) and (4) as shown in Fig. 6.

$$x_{ave} = (\sum_{i=1}^{N_L} x_{Li}) / N_L \,. \tag{3}$$

$$y_{ave} = (\sum_{i=1}^{N_L} y_{Li}) / N_L \,. \tag{4}$$

where x_{Li} and y_{Li} denote the coordinate value of the *i* th range data in the grid, and N_L is the number of range data included in the grid.

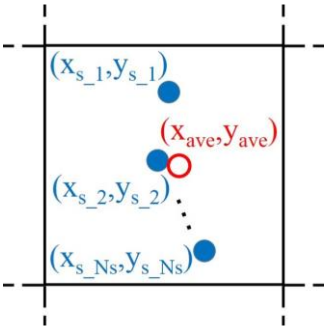


Fig. 6. Average coordinate value of scan data in a grid.

STEP 2: The distance between the newest average coordinate value and the oldest in a grid, d_g, is calculated as follows:

$$d_g = \sqrt{(x_{newest} - x_{oldest})^2 + (y_{newest} - y_{oldest})^2} \,. \tag{5}$$

where x_{newest} and y_{newest} are the coordinate values of the newest range data, x_{oldest} and y_{oldest} are the coordinate values of the oldest range data. A schematic diagram of distance d_g is shown in Fig. 7.

STEP 3: The velocity of an obstacle, v_o, is calculated for each grid as follows:

$$v_o = d_g / t_{scan} \,. \tag{6}$$

where t_{scan} is the interval time between the oldest and newest scans.

STEP 4: Steps 1 to 3 are applied to all grids.

As a result, the algorithm calculates the obstacles' velocity for each grid.

To resolve negative effect of error data, the velocity of a grid is computed using its own range data and those of the four grids near it. Then, the smallest calculated value is used as the representative velocity of the object in the grid. As shown in Fig. 8, there are four overlapped grids of Grid-map2 near a grid on Grid-map1.

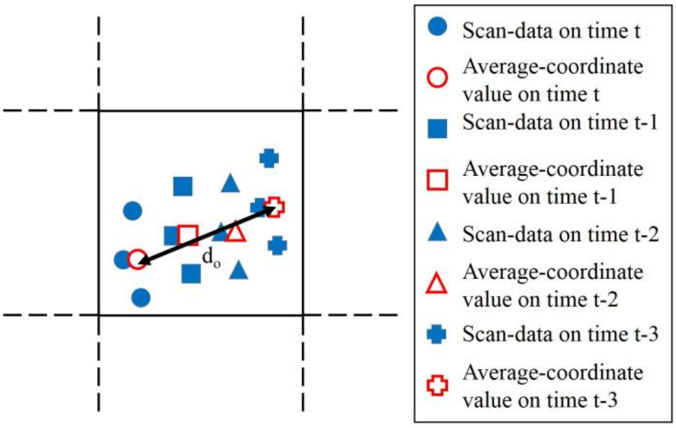

d_o
● Scan-data on time t
○ Average-coordinate value on time t
■ Scan-data on time t-1
□ Average-coordinate value on time t-1
▲ Scan-data on time t-2
△ Average-coordinate value on time t-2
✚ Scan-data on time t-3
✚ Average-coordinate value on time t-3

Fig. 7. Distance between the newest and oldest average coordinate values.

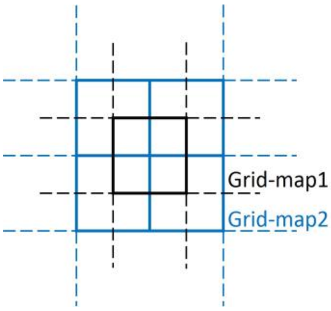


Fig. 8. 5 grids for calculating a velocity of a center grid.

D. Calculation of the Direction of Obstacle's Motion

The method to calculate the direction of an obstacle's motion is explained below.

STEP 1: The velocities of the range data in the grid and the four overlapped grids are computed. The minimum is selected as the representative value. In the case that there is no range data, the value becomes zero.

STEP 2: The direction of the obstacle's motion is calculated for the range data in the grid as follows:

$$v_{o\_rob} = v_o \times \cos\theta_{rob} \,. \tag{7}$$

The resultant value, v_{o_rob}, denotes the velocity of the obstacle with respect to the direction from the robot to the obstacle. The sign indicates the direction of the obstacle's motion, namely, whether the obstacle is moving nearer to or away from the robot. The magnitude represents the speed of the motion. This is depicted in Fig. 9.

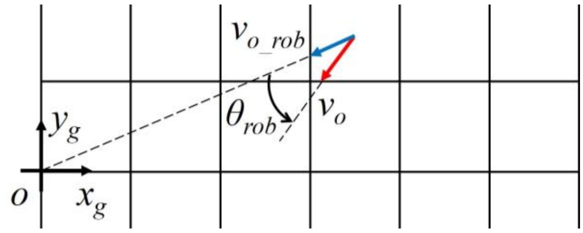


Fig. 9. A direction of the obstacle's motion on the grid.

STEP 3: Steps 1 and 2 are adapted to all grids.

The above computation is applied to all grids that have range data.

IV. Collision Avoidance Algorithm

The collision avoidance method proposed in this research is explained below.

A. Outline of Collision Avoidance System

In this research, it is assumed that the robot moves toward the goal position in a narrow corridor where pedestrians exist. Fig. 10 shows the schematic diagram where the mobile robot and two moving obstacles exist. The detailed searching area of the LRF is shown in Fig. 11, where *O-XY* denotes the fixed coordinate system and *o-xy* denotes the moving coordinate system attached at the center of the robot. *O-XY* and $o\text{-}x_g y_g$ are parallel coordinates system. The position and orientation of the robot are represented by (x,y,θ) on the fixed coordinate system. The resulting safe direction computed by the proposed algorithm, α, and the direction toward the goal position, β, are represented with respect to the moving coordinate system.

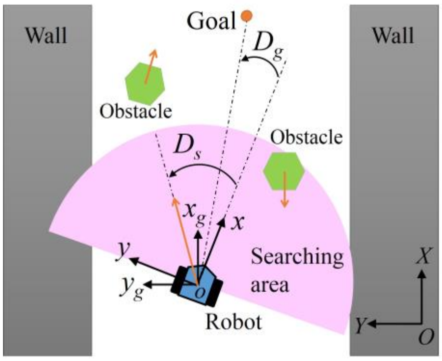


Fig. 10. Schematic diagram of an indoor corridor environment with a robot and two obstacles.

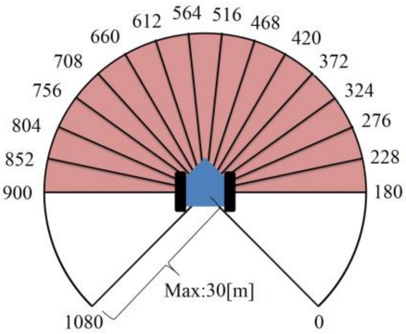


Fig. 11. Searching area of the mobile robot.

B. Recognizing the Environmental Condition around the Robot with Velocity Grid Map

The proposed algorithm recognizes the condition of circumstance by computing a safety level for all direction in the search area. In our proposal, the safety level for each direction is computed using fuzzy theory. The position and velocity information of the objects from the grid map are utilized. Safety level takes a continuous value between 0 and 1, where 0 indicates the most dangerous state and 1 is the safest. The shortest distance in each direction is considered as the distance information and the largest velocity in each direction is the velocity information. Both are translated to a safety level with the membership functions of the hypothesis shown in Figs. 12 and 13. The membership function of the safety level is shown in Fig. 14. The fuzzy rule to calculate the resultant safety level is shown in Table I.

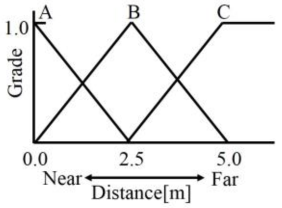


Fig. 12. Membership function of distance.

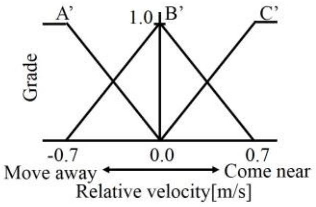


Fig. 13. Membership function of relative velocity.

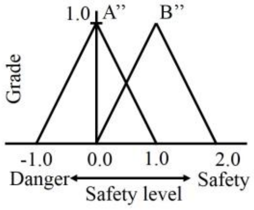


Fig. 14. Membership function of safety level.

TABLE I
Fuzzy rule to compute safety level with distance and relative velocity.

Distance / Relative vel.	A: Near	B: Medium	C: Far
A': Move away	B": Safety	B": Safety	B": Safety
B': Medium	A": Danger	B": Safety	B": Safety
C': Come near	A": Danger	A": Danger	A": Danger

C. Motion Planning for the Mobile Robot

The motion-planning portion decides the translational velocity, v, and angular velocity, ω, required to reach the goal position and simultaneously avoid collision with obstacles. Both are decided using the following equations:

$$v = v_{\max} \times S \,. \tag{8}$$

$$\omega = D_m \times C_w \,. \tag{9}$$

The maximum velocity, v_{max}, is set to 1.0 m/s in this study. S is the sum of safety level values of all directions. Thus, the translational velocity of the robot changes in the range from 0.0 m/s to 1.0 m/s. D_m is the moving direction described below. C_w is the constant number for the angular velocity.

The moving direction of the robot is determined using the different cases that are classified according to the safety level as follows:

Case 1: When the safety level on the angle of the goal direction is 1, the moving direction is the same as the direction to the goal position D_g. This means there is no obstacle.

$$D_m = D_g \,. \tag{10}$$

Case 2: If not Case 1, when the safety level of the current moving direction is less than 1 and larger than 0.5, the moving direction is the forward direction of the robot, D_{fr}.

$$D_m = D_{fr} \,. \tag{11}$$

Case3: If not Case 1 and 2, when the moving direction is computed with the following equation.

$$D_m = D_s + (S_g /(S_g + S_s)) \times (D_g - D_s) \,. \tag{12}$$

where the safest direction, D_s, denotes the direction whose safety level is larger than the other search directions. S_g and S_s

are the safety level of the direction to the goal and the safest direction, respectively.

Case4: For Case 3, if the safety level of the resultant moving direction is less than 0.5, the resultant moving direction is set to the safest direction.

$$D_m = D_s . \tag{13}$$

Finally, the translational and rotational velocity based on the moving direction and the safety level is given as the motion command for the robot.

D. *Computer Simulation of Collision Avoidance*

(a) Method of the Computer Simulation

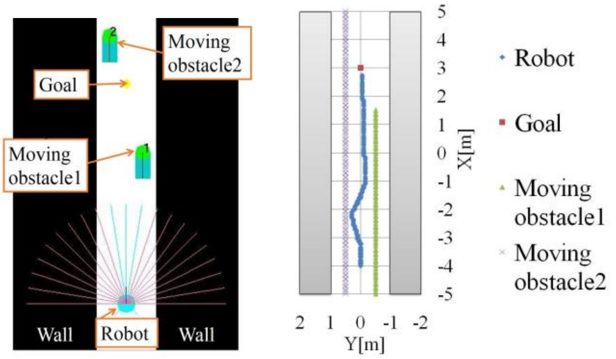
A computer simulation was carried out to verify the feasibility of the proposed motion planning algorithm. The model of the corridor (width: 1.8 m) was prepared as a virtual environment in the computer. The pedestrian was modeled as a circle whose diameter was set to 450 mm. In addition, the initial position, the maximum velocity of the mobile robot, the number of obstacles, initial positions and velocity of obstacles, and goal position were set based on a real situation. A simulation of the situation with two moving obstacles was carried out. The parameters used in the simulation are shown in Table Ⅱ.

TABLE Ⅱ
PARAMETERS USED IN SIMULATION

Parameter	Robot	Obstacle1	Obstacle2	Goal
Position(X) [m]	-0.2	0.55	-0.55	0.0
Position(Y) [m]	-4.0	6.0	2.0	3.0
Velocity [m/s]	0.7	0.8	0.8	

(b) Result of the Computer Simulation

The Calculated results using the proposed algorithm are shown in Fig.15. It was confirmed that the mobile robot using the proposed algorithm could reach the goal position without any collision with obstacles. Therefore, its feasibility of the proposed algorithm for collision-free navigation could be checked in this virtual environment.



(a) Virtual environment. (b) Movement trajectories.

Fig.15. Calculated result of the computer simulation with the proposed method.

V. EXPERIMENTAL WORKS

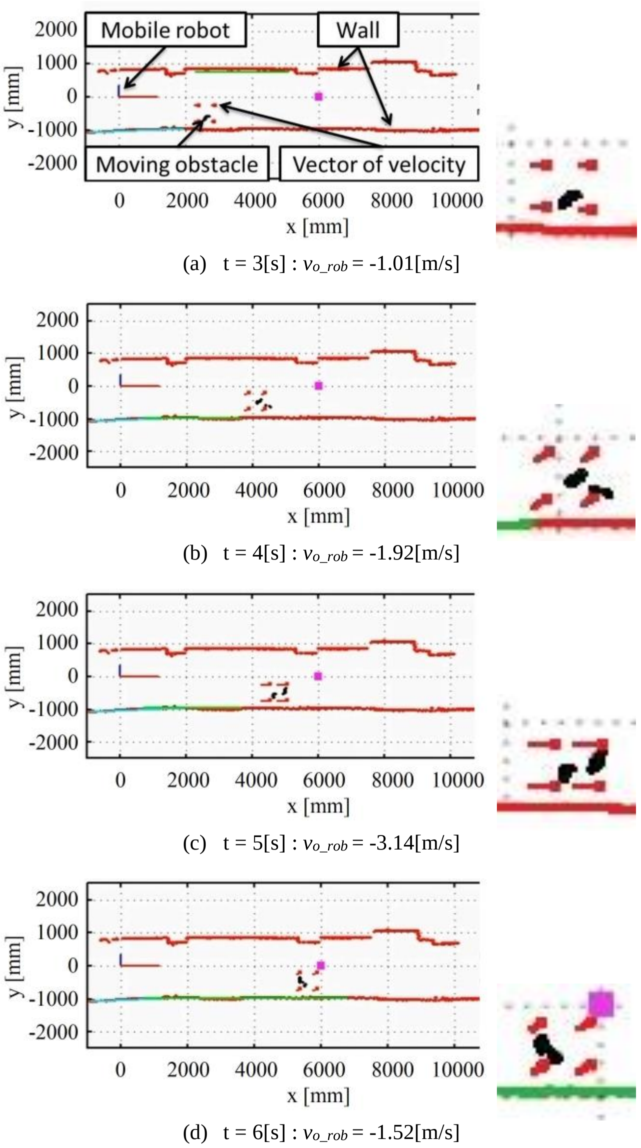
A. *Experiment to Calculate Obstacle's Velocity with Velocity Grid Map*

(a) Experimental Method

The experiment to calculate an obstacle's velocity with the proposed method was performed in a real corridor environment with width about 1.8 m. One human as a moving obstacle moved away from the mobile robot is in rest at the origin position of the corridor.

(b) Experimental Result

The computed velocity information of each grid is displayed as an arrow on the velocity grid map in Fig. 16. The figures on the right of Fig. 16 show their enlargement near the moving object. The vector size denotes the magnitude of the obstacle's velocity. The resultant velocity information does not show exactly the same value as the real motion, but it is sufficient to recognize the trend of the moving obstacle.



(a) t = 3[s] : v_{o_rob} = -1.01[m/s]

(b) t = 4[s] : v_{o_rob} = -1.92[m/s]

(c) t = 5[s] : v_{o_rob} = -3.14[m/s]

(d) t = 6[s] : v_{o_rob} = -1.52[m/s]

Fig.16. Calculated velocity of the obstacle.

B. *Experiment of Collision Avoidance*

(a) Experimental Method

The collision-avoidance experiment of the proposed algorithm was performed in a corridor (width: 1.8 m) of a laboratory as shown in Fig. 17. Two humans approached from near the goal to near the robot. The robot moved toward the goal position. The initial position of the robot was set near the origin point of the world coordinate. The goal position was set to (X, Y) = (6.0 m, 0.0 m). The robot's maximum velocity was set as 1.0 m/s.

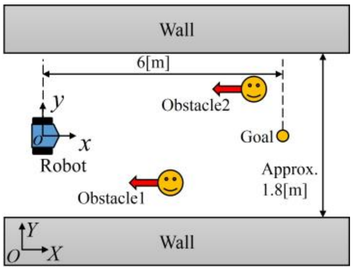


Fig. 17. Schematic diagram of the scenario for collision avoidance experiment.

(b) *Experimental Result*

The experimental results are as follows. It was confirmed that the robot could reach near the goal position successfully with no collisions with obstacles or walls. The motion trajectory of the robot is displayed in Fig. 18. Fig. 19 shows the motion trajectories of both the robot and the humans depicted on the map. The robot started near the origin position and moved linearly toward the goal position. Then, after moving about 2 m from its initial position, it changed direction to avoid a collision with the first human. After moving about 4 m from its initial position, it changed direction again to avoid a collision with the second human. The change of motion command for the mobile robot is displayed in Fig. 20. When the robot avoided collision with the first person, its velocity was decreased (t=3 to7 s). After avoiding the first person, its velocity increased. When the robot avoided the collision with the second person, its velocity was again decreased (t=8 to 11 s). Photos of the experimental result are shown in Fig.21.

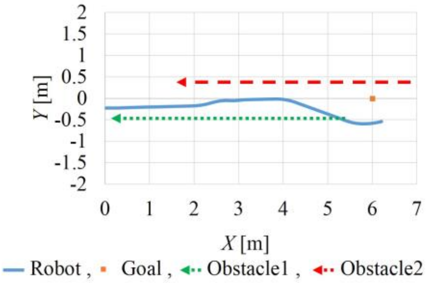


Fig. 18. Motion trajectory of the mobile robot.

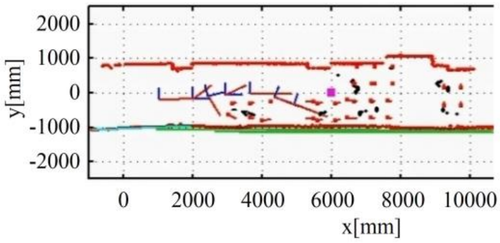


Fig. 19. Motion trajectory of the mobile robot and the human on the map.

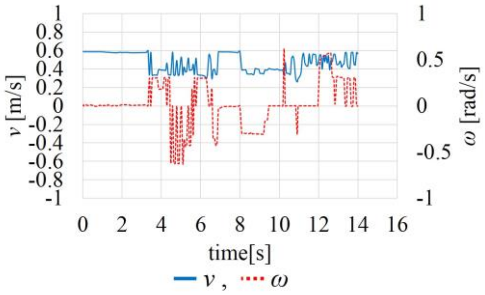
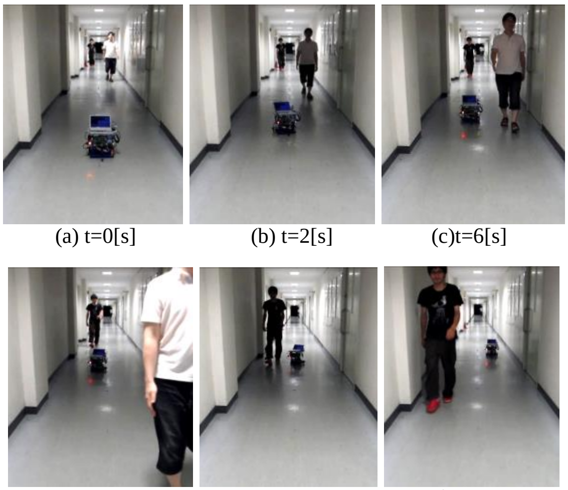


Fig. 20. Change of velocity command for the mobile robot.



(a) t=0[s] (b) t=2[s] (c)t=6[s]

(d) t=8[s] (e) =10[s] (f) t=12[s]

Fig.21. Experimental result to avoid collision with two walking humans.

VI. Conclusions

In this study, a collision-avoidance algorithm that reflects human motion sense was developed for an indoor mobile robot using LRF. The conclusions are as follows.

(1) A unified method to sense the obstacles' velocities using a grid map was proposed.

(2) A collision-free navigation algorithm based on fuzzy theory with the velocity grid map was proposed.

(3) Computer simulation was carried out to verify the feasibility of the algorithm as a collision-avoidance system for a mobile robot.

(4) Through experiments, it was confirmed that the algorithm could be utilized as a navigation method for an indoor mobile robot. In addition, it could generate a safe motion plan for a robot with no exact velocity measurement but an ambiguous trend of object motion changes.

References

[1] Dieter Fox, Wolfram Burgard, and Sebastian Thrun, "The Dynamic Window Approach to Collision Avoidance," IEEE Robotics & Automation Magazine, vol.4.1, p23-33,1997

[2] Javier Minguez and Luis Montano, "Nearness Diagram (ND) Navigation: Collision Avoidance in Troublesome Scenarios," IEEE Transactions on Robotics and Automation, vol.20.1, p45-59, 2004.

[3] Wesley H. Huang, Brett R. Fajen, Jonathan R. Fink, and William H. Warren, "Visual navigation and obstacle avoidance using a steering potential function," Robotics and Autonomous Systems vol.54.4, p288-299, 2006.

[4] Tsong-Li Lee and Chia-Ju Wu, "Fuzzy Motion Planning of Mobile Robots in Unknown Environments," Journal of Intelligent and Robotic Systems vol.37.2, p177-191, 2003.

[5] X. Yang, R. V. Patel, and M. Moallem, "A Fuzzy–Braitenberg Navigation Strategy for Differential Drive Mobile Robots," Journal of Intelligent and Robotic Systems vol.47.2, p101-124, 2006.

[6] Y. Ono, H. Uchiyama, and W. Potter, "A mobile robot for corridor navigation: a multi-agent approach," ACM-SE 42 Proceedings of the 42nd annual Southeast regional conference, p379-384, 2004.

[7] Mohammad Abdel Kareem Jaradat, Mohammad H. Garibeh, and Eyad A. Feilat, "Autonomous mobile robot dynamic motion planning using hybrid fuzzy potential field," Journal of Soft Computing, vol.16.1, p153-164, 2012.

[8] Rachael Bis, Huei Peng, and Galip Ulsoy, "Velocity occupancy space: Robot navigation and moving obstacle avoidance with sensor uncertainty," ASME 2009 Dynamic Systems and Control Conference, pp. 363-370, 2009.

[9] Yusuke Tamura, Yu Murai, Hiroki Murakami, and Hajime Asama, "Identification of types of obstacles for mobile robots," Journal of Intelligent Service Robotics, vol.4.2, p99-105, 2011.

ISBN: 978-988-19252-5-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)