The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Mobile Robot Navigation with Reactive Free Space Estimation

Tae-Seok Lee, Gyu-Ho Eoh, Jimin Kim and Beom-Hee Lee, *Fellow, IEEE*

Abstract— **One of the fundamental requirements of an autonomous mobile robot is that it must not collide with obstacles. This paper addresses the problem of controlling an autonomous robot to avoid obstacles for reactive route following navigation. The four-wheeled mobile robot is equipped with three monocular cameras for route following and a range sensor for obstacle avoidance. The equipped robot moves route environments using a reactive navigation method. When an obstacle is detected by a range sensor, the proposed obstacle avoidance method estimates the free space in the route and generates the turning direction vector for heading the robot to the free space which is wider than other space. We executed experiments about the navigation algorithm of the robot with an obstacle including curved path environments in this paper. Through the experiment results in various cases, it is discovered that the proposed method has a better time performance for obstacle avoidance in comparison with other conventional technique.**

I. Introduction

A fundamental competency for mobile robot navigation is the ability to plan and execute collision free motion through unknown environments in real time. In order to plan collision free motion for a mobile robot, numerous obstacle avoidance algorithms have been proposed based on reactive robot control techniques. Several decades before, Borenstein and Koren developed the vector field histogram(VFH) [1]. After that many other researches have been conducted. Currently, in addition to the traditional problems, obstacle avoidance has been researched for autonomous cars [2], [3], unmanned aerial vehicles [4], unmanned submarine vehicles [5], [6], and other moving agents. In many of these researches, the studies for ground vehicles are especially active. Generally the ground vehicles are commonly moving along the lane on the road. In this reason, several studies about route following navigation have been researched [7], [8]. Thus, we address the obstacle avoidance problem in route environments.

Current researches on obstacle avoidance are using vision sensors. There are several problems when the vision sensors are used. In order to detect an obstacle from image, obstacle detection using monocular camera is achieved by comparing the current view image with the memorized one [9], otherwise stereo camera has to be equipped to acquire a disparity map [10]. There is another problem caused by illumination. Feature components of obstacles from image could change by illumination conditions. In order to avoid these problems, sensor integration method with range sensors and vision sensors is widely used [11], [12]. Range sensors are used for obstacle detection and vision sensors are used for acquisition of other environment information. This sensor integration makes that the robot system has stable and accurate performance.

Many reactive obstacle avoidance methods execute heuristic obstacle avoidance motion. When the robot senses an obstacle, conventional methods push the robot away from the obstacle with priori defined amount [12] or they increase the traversability cost value [13]. However, obstacle avoidance could execute by more algorithmic process in route space.

We suggest a novel method for real-time reactive obstacle avoidance in route following navigation. After estimating the exact position of the sensed obstacle on the track path using range sensor data, the robot calculates widths of the space between the obstacle and two boundary lines of the route, and then it generates immediately the turning direction vector for heading the robot to the space which is wider than the other side. Due to the complexity of this free space estimation process is not high, the proposed algorithm allows us to take advantage of available computing resources without losing the ability to respond reactively to unexpected obstacles.

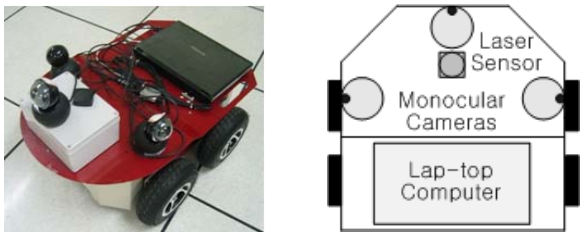
The paper is organized in five sections. In Section II, we introduce the visual route following navigation system what we use in this paper. In Section III, we describe the reactive free space estimation algorithm for obstacle avoidance. The experiments in various situations are presented in Section IV, and our conclusion is in Section V.

II. Visual Route Following Navigation

Our route navigation system is based on the visual path following algorithm. Many visual route following navigation algorithms have been researched in outdoor environments, the reason is that outdoor navigation is a challenging work because the robot has to operate in various environment conditions. Detail and successful navigation results are presented in [7], however it does not consider obstacles in the environments. There is a slight mention about the obstacle avoidance problem in [8], but it also remains the deep

This work was supported in part by a Korea Science and Engineering Foundation (KOSEF) NRL Program grant funded by a Korean government (MEST) (No. R0A-2008-000-20004-0), in part by the Brain Korea 21 Project, and in part by the Industrial Foundation Technology Development Program of MKE/KEIT [Development of CIRT(Collective Intelligence Robot Technologies)].

Tae-Seok Lee, Gyu-Ho Eoh, Jimin Kim and Beom-Hee Lee are with the School of Electrical Engineering and Computer Sciences, Seoul National University, Seoul, Korea (e-mail: {felix84, geni0620, torin00, bhlee}@snu.ac.kr). B. H. Lee is an IEEE fellow and currently a professor in the School of Electrical Engineering and Computer Sciences, Seoul National University, Seoul, Korea.



(a) (b)

Fig. 1. The robot platform for visual route following navigation: (a) A picture of the sensor equipped robot, (b) Top view illustration of the robot.

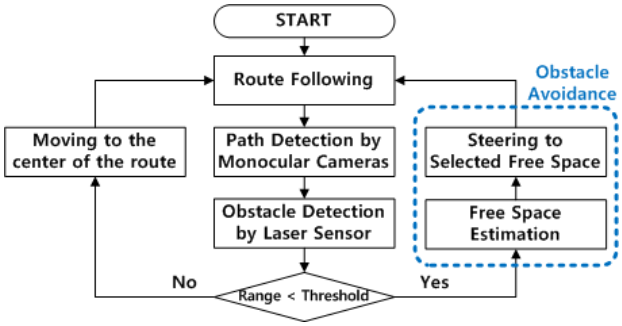


Fig. 2. Visual route following navigation structure.

implementation about obstacle avoidance as a future work.

In this paper, the visual route following navigation system is constructed to have a steady performance whether it is in indoor or outdoor. As shown in Fig. 1, we equipped a four-wheeled skid steering mobile robot with three web cameras and a laser range finder. The two monocular web cameras are installed at both sides of the robot, respectively, and the rest camera is installed at the front of the robot. We extract path lines of the route using the side cameras and we obtain information about the path in front of the robot using the forward camera. Using the laser range finder which is attached behind the forward camera, we detect unknown obstacles up to 4m. This framework is similar to [10] in that two side cameras work only for path line detection, however the rest of the details especially in obstacle detection part are different. The previous work [10] used a visual obstacle avoidance technique. The problem is that visual obstacle avoidance sometimes fails to detect obstacles. In this paper, the obstacle detection process is performed using the laser range finder. Because the laser range finder gives more accurate and stable performance than the visual information analysis for object detection, we used the laser range finder.

Fig. 2. represents the overall scheme of the visual route following and the obstacle avoidance method for this work. This navigation system follows a reactive approach because reactive methods are useful in unpredictable situations. After acquiring the measurement data from cameras and the range sensor, the robot adjusts its steering direction to the desired position depending on the velocity of the robot, the friction coefficient between the robot and the ground, and the distance from the robot to the obstacle. If there are no obstacles on the route, the robot tries to maintain its position

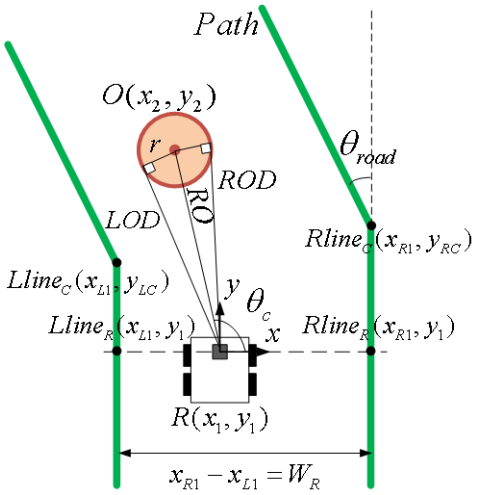


Fig. 3. System configuration of visual route following navigation.

to the center of the route. When an obstacle is detected by the laser range finder, the robot conducts obstacle avoidance motion. The proposed obstacle avoidance technique is mainly described in Section III.

The mobile robot navigates unknown route under the following configuration as shown in Fig. 3. This system is based on several assumptions.

- The origin of the coordinates is the center of the robot and the y axis of the coordinates is parallel with path lines of the route, because this moving local coordinates is easy to apply to reactive control. According to this characteristic, the robot position $R(x_1,y_1)$ is always (0,0). The cross points between the coordinates and the path lines, $LLine_R$ and $RLine_R$, also have 0 as y value.
- The width of the route, W_R, is constant. It guarantees that (1) is always satisfied:

$$W_R = x_{R1} - x_{L1} \tag{1}$$

  where x_{R1} and x_{L1} are x position of $LLine_R$ and $RLine_R$. If W_R varies on the route, we have to calculate exact value of W_R using pixel information of image from the forward camera. However, we only consider the situation that the variation of W_R is very small.
- We can obtain the angle of the curved route, θ_{road}, and the starting position of curve, $LLine_C$ and $RLine_C$ using the forward camera. θ_{road} has positive value when the route curved to the right side, and negative value in the opposite case.
- The shape of the obstacle is circle and the radius of the obstacle, r, is known.

III. Reactive Free Space Estimation Algorithm

When the laser range finder detects an obstacle in this configuration, the obstacle avoidance method is executed. As stated in Section I, the proposed method estimates the free space of the both sides of the obstacle within two boundary lines. Then it chooses the wider space as a next waypoint

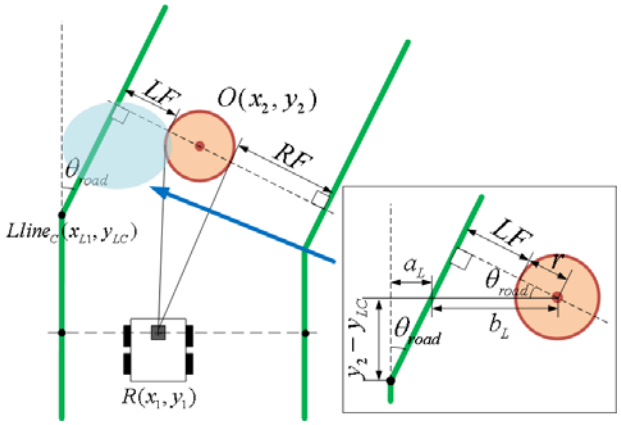


Fig. 4. Illustration of estimating the free space of both sides of the obstacle.

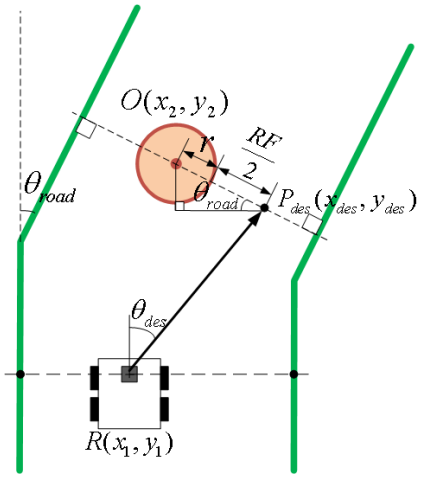


Fig. 5. Illustration of generating a turning direction vector for heading the robot to the center position of wide space.

because the chosen space is regarded as safer than the other.

First, we can acquire the distance from the left end of the obstacle to the robot, *LOD*, and from the right end of the obstacle, *ROD*, also. Then we can calculate *RO*, the distance from the center of the obstacle to the robot as shown in (2) and Fig. 3.

$$RO = \sqrt{LOD^2 + r^2} \tag{2}$$

x and y components of the obstacle's position, $O(x_2,y_2)$ are obtained using following equations.

$$x_2 = x_1 + RO \cdot \sin(\frac{\pi}{2} - \theta_c) \tag{3}$$

$$y_2 = y_1 + RO \cdot \cos(\frac{\pi}{2} - \theta_c) \tag{4}$$

When $O(x_2,y_2)$ is found, the widths of the free space between the obstacle and the boundary line of the route are calculated. Using the triangle similarity shown in Fig. 4, we can easily get the width of the left side of the obstacle (Left Free space: *LF*) with $LLine_C$, θ_{road}, $O(x_2,y_2)$, and the radius of the obstacle, r, as (5) and (6).

$$a_L + b_L = x_2 - x_{L1} = (y_2 - y_{LC}) \tan\theta_{road} + \frac{LF + r}{\cos\theta_{road}} \tag{5}$$

$$LF = (x_2 - x_{L1} - (y_2 - y_{LC}) \tan\theta_{road}) \cos\theta_{road} - r \tag{6}$$

Even the route gets bent, the width W_R is fixed, thus using (7) the width of the right side (Right Free space: *RF*) is obtained as (8).

$$LF + RF + 2r = W_R \tag{7}$$

$$RF = W_R - (x_2 - x_{L1} - (y_2 - y_{LC}) \tan\theta_{road}) \cos\theta_{road} - r \tag{8}$$

The proposed technique compares the two calculated widths, *LF* and *RF*, and then chooses the wider one. The center point of the chosen width becomes the next goal point of the robot. The position that the robot desires to go, $P_{des}(x_{des},y_{des})$, is obtained by adding the parts of cosine and sine components of the distance between $O(x_2,y_2)$ and P_{des} to $O(x_2,y_2)$. We know the distance between $O(x_2,y_2)$ and P_{des} as the half of the chosen width plus r. x and y components of the P_{des} are represented in (9) and (10). The angle that the robot wants to turn, θ_{des}, is also obtained as shown in Fig. 5. and (11). These figure and equations are defined when the right side of the obstacle is wider than the left. If the left side is wider than the right, we just apply (12) in equations from (9) to (11).

$$\begin{aligned} x_{des} &= x_2 + (\frac{RF}{2} + r)\cos\theta_{road} \\ &= x_1 + RO \cdot \sin(\frac{\pi}{2} - \theta_c) + (\frac{RF}{2} + r)\cos\theta_{road} \end{aligned} \tag{9}$$

$$\begin{aligned} y_{des} &= y_2 - (\frac{RF}{2} + r)\sin\theta_{road} \\ &= y_1 + RO \cdot \cos(\frac{\pi}{2} - \theta_c) - (\frac{RF}{2} + r)\sin\theta_{road} \end{aligned} \tag{10}$$

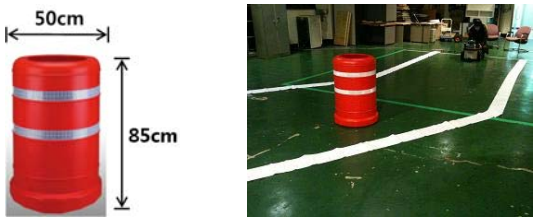
$$\theta_{des} = \frac{\pi}{2} - \arctan\left(\frac{RO \cdot \cos(\frac{\pi}{2} - \theta_c) - (\frac{RF}{2} + r)\sin\theta_{road}}{RO \cdot \sin(\frac{\pi}{2} - \theta_c) + (\frac{RF}{2} + r)\cos\theta_{road}}\right) \tag{11}$$

$$(\frac{RF}{2} + r) \Leftrightarrow -(\frac{LF}{2} + r) \tag{12}$$

If the robot has an orientation angle θ_{robot}, then the turning angle what robot actually steer θ_{steer} is as follows:

$$\theta_{steer} = \theta_{des} - \theta_{robot} \tag{13}$$

If θ_{road} is $\pm\pi/2$, then *LF* and *RF* become invalid values. Generally, in this case θ_{des} tends to small value and the robot goes straight along the route. When the robot passes the curved point then one of the side cameras cannot extract the boundary line of the route. It indicates that the route is



(a) (b)

Fig. 6. Experimental environments: (a) The obstacle, (b) The route of the experiment.

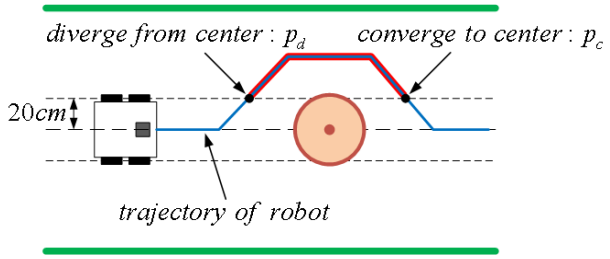


Fig. 7. Diverged trajectory segment of the robot (red line) during obstacle avoidance motion.

suddenly curved to that direction, thus we steer the robot to the curved direction. When the denominator of (11) is close to zero, also is close to zero. Since it is similar to the above situation, the robot follows the same procedure.

When the obstacle avoidance motion is executed, the robot acts in two ways. The first way is that the robot stops first, and then it acts turning and going. Another one is that the robot continuously turns when it keeps moving. This paper uses the latter method because the robot's velocity is maintained. Since, the robot continuously turns with θ_{steer} maintaining its velocity v, we assign the command which consists of θ_{steer} and v. The robot has a rotational velocity using assigned θ_{steer}:

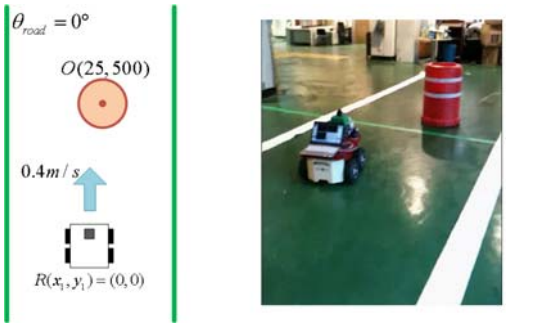
$$w_r = k \cdot \theta_{steer} \tag{14}$$

where w_r is the robot's rotational velocity and k is the experimentally determined system gain that depends on the system's processing speed and the robot dynamics.

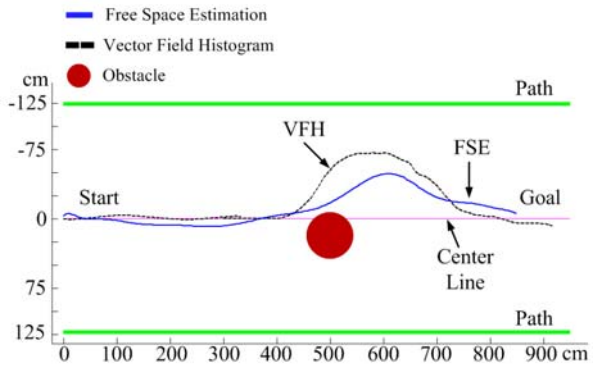
IV. Experimental Results

Obstacle avoidance tests were conducted using a four-wheeled skid steering mobile robot, a Korean-made vehicle, with three monocular cameras and a laser range finder as shown in Fig. 1 (a). Three 40° field of view Logitech QuickCam Sphere AF web cameras and a URG-04-LX-UG01 laser scanner by Hokuyo were mounted to the robot. We implemented the algorithm using MS Visual Studio program. Running time was measured using single-threaded execution on a 2.1GHz Core 2 Duo. Fig. 6 (b) represents the real experimental environments.

The robot was moving with a constant velocity of 0.4m/s. Here, we used the obstacle as in Fig. 6 (a) and its radius r is 25cm. We set the width of the route W_R as 2.5m. The robot started at the center of the route and the obstacle is located 5m ahead of the robot.The obstacle within 4m of the robot can be



(a)



(b)

Fig. 8. Experiment A: (a) Route of the experiment, (b) Results of vector field histogram and free space estimation algorithm.

detected by the laser scanner. If there is no obstacle on the route, the robot passes the center of the route. When the robot is conducting the obstacle avoidance motion it diverges from the center of the route. After the obstacle avoidance motion, the robot comes back to the center again. If the robot passes more than 20cm away from the center line of the route, we considered that the robot has left the center of the route as shown in Fig. 7. We can measure the navigation time between the diverging point and the converging point. In this paper, we call this navigation time of the diverged trajectory segment as the diverging time and the diverging time was set as a performance index and measured in each experiment. We compared the diverging time of the proposed algorithm and the vector field histogram algorithm in three kinds of the route as shown in Fig. 8 (a), Fig. 9 (a), and Fig. 11 (a).

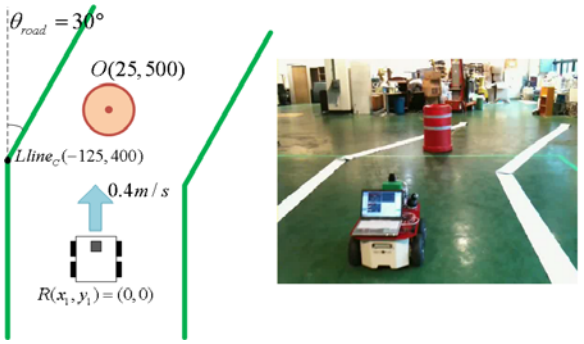
A. Straight route

In first test we placed the robot in straight route which has

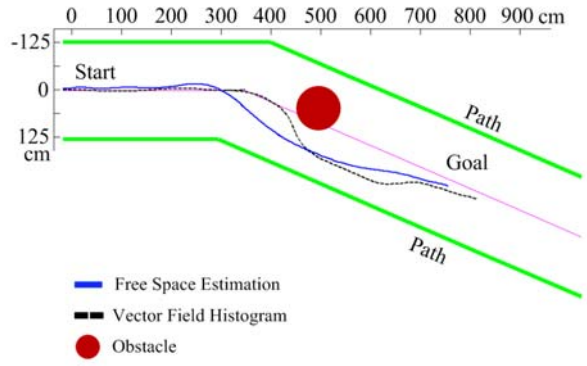
TABLE I
Width of the Free Space

θ_{road} (°)	Real Width of the Free Space (cm)		Estimated Width of the Free Space (cm)		Error of the Width (cm)
	Left	Right	Left	Right	
0	125	75	133.9	66.1	8.9
30	54.9	145.1	45.1	154.9	9.8
45	10.4	189.6	12.3	187.7	1.9
Average of the Error (cm)					6.87

The width of the road W_R=250cm.
The radius of the obstacle r=50cm.



(a)



(b)

Fig. 9. Experiment B: (a) 30° curved route of the experiment, (b) Results of vector field histogram and free space estimation algorithm.

θ_{road} as 0° as shown in Fig. 8 (a). The obstacle's position is (25, 500) from the robot. TABLE I summarizes the real width of the space next to the obstacle and the average of the estimated width by the proposed algorithm 10 times. The error of the width estimation is 8.9cm. The error implies the proposed algorithm was successfully started. The blue solid line in Fig. 8 (b) represents the trajectory of the robot with the free space estimation algorithm. When enough space exists, the obstacle avoidance movement was successfully worked.

The black dashed line in Fig. 8 (b) shows the trajectory of the robot with the vector field histogram. The vector field histogram also succeeded, but the diverging time was distinguished. As shown in TABLE II, the free space estimation algorithm derived the shorter diverging time than the vector field histogram. Furthermore, when the robot uses the free space estimation algorithm, it had less turning motion than the other as represented in Fig. 8 (b). The sudden turning motion causes the slip of the wheels, and it makes the odometry error of the robot. Therefore, we can say that the free space estimation method may lead the less odometry error than the vector field histogram algorithm.

B. 30° curved route

Experiment B was conducted on curved route of 30° as shown in Fig. 9 (a). The starting point of the robot and the obstacle's position are same as Experiment A. The route is curved at 4m forward of the robot's starting position. The obstacle is located 1m behind the curved point. If the route is straight, the left side of the obstacle is wide, so the robot turns

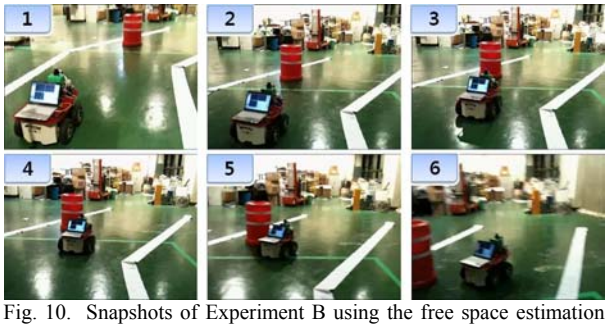


Fig. 10. Snapshots of Experiment B using the free space estimation algorithm.

to the left direction. However, in this situation, the right side of the obstacle is wider than left.

As represented in Fig. 9 (b), the free space estimation algorithm and the vector field histogram succeeded. However, similar as Experiment A, the vector field histogram had sharper trajectory than the proposed algorithm. We got 1.7 second difference of the diverging time as shown in TABLE II. The vector field histogram method begins the obstacle avoidance motion when the obstacle is appeared in active window region and the obstacle is located between the robot and the goal. When the size of the active window is large, the robot starts the avoidance motion early. The large active window brings smoother trajectory than the result with current setting. However it takes very high computation thus it is not suited to the fast moving robot, and also in curved route. If the robot determines its avoidance direction too early, the robot may choose the direction to the narrower space. Then the robot collides with the obstacle and crosses the line of the route. Since the robot does not know the route information in advance, it sets the goal in forward direction. When the robot turns along the route, it updates its direction of the goal. Thus the vector field histogram was effective after passing the curved point $LLine_C$ or $RLine_C$. In this experiment, we adjusted the size of the active window so that the robot chooses its avoidance direction after it passes the curved point. In the free space estimation algorithm case, the robot predicts the width of the free space and sets the goal to the center of the free space. Therefore, the robot turned earlier and had the smoother trajectory than the vector field histogram case. The robot was apart from the center of the route earlier than the vector field histogram case but it came back earlier too, therefore the diverging time was shorter than the vector field histogram. Fig. 10. shows the experiment when the robot used the free space estimation algorithm.

TABLE II
DIVERGING TIME OF THE EXPERIMENTS

θ_{road} (°)	Vector Field Histogram (s)	Free Space Estimation Algorithm (s)	Difference of Time (s)
0	6.8	5.0	1.8
30	9.6	7.9	1.7
45	9.1	7.2	1.9
Average of the Time Difference (s)			1.8

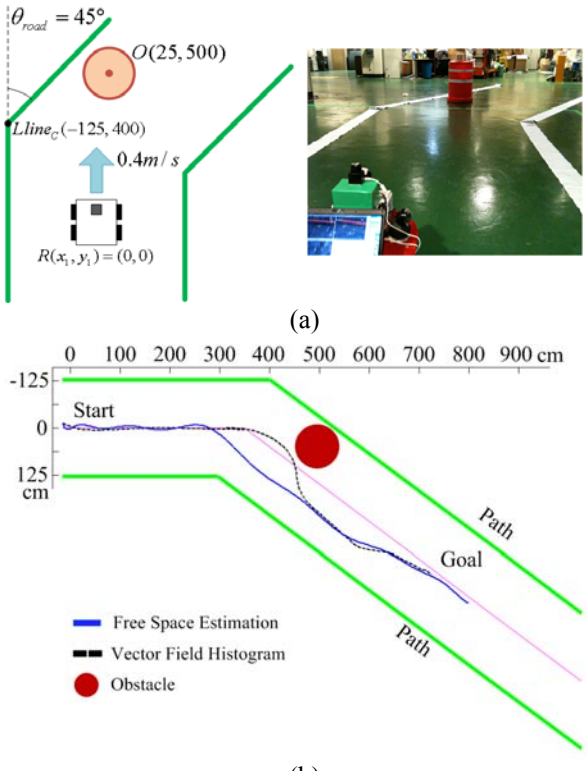


Fig. 11. Experiment C: (a) 45° curved route of the experiment, (b) Results of vector field histogram and free space estimation algorithm.

C. 45° curved route

In Experiment C, θ_{road} was set 45° as shown in Fig. 11 (a). The starting position of the robot and the position of the obstacle are same as Experiment A. There is only one difference in θ_{road} between Experiment B and Experiment C. Because θ_{road} is large, the left room of the obstacle has very narrow width. As shown in TABLE I, the free space estimation algorithm exactly estimated the width of both side of the obstacle.

Fig. 11 (b) shows the trajectories of the robot using free space estimation and the vector field histogram. In free space estimation case, the robot passed just center of the right side free space, however in the vector field histogram case, the robot had a very dangerous moment of collision with the obstacle. The trajectory was fluctuated, so the robot had a sharp shaped trajectory and long diverging time. The navigation time during the diverging of the robot from the center using the vector field histogram was 1.9 second longer than the proposed algorithm. Actually there were a lot of failed experiments with the vector field histogram for Experiment C. The free space estimation algorithm showed the distinguished performance in this experiment.

V. Conclusion

In this paper, reactive visual route following navigation is described and the reactive based obstacle avoidance method is developed. The developed algorithm estimates the free space next to the obstacle using cameras and a range sensor, and selects the wider space between both sides. We derived the free space estimation algorithm in Section III. We demonstrated the performance of the reactive free space estimation algorithm through the experiments presented in Section IV. The performance was compared with the vector field histogram. The experimental results confirmed the validity of the proposed algorithm.

The experiments were limited to the static obstacle in this paper. As a future work, it would be instructive to test the proposed algorithm with the moving obstacle. By applying the multiple obstacles detection technique, the implementation with the multiple obstacles will be also executed. During the experiments catching a curved point of the route, $LLine_C$ and $RLine_C$, by a monocular web camera was very difficult because of its narrow field of view. Next time, we will use a monocular camera with wider field of view or a stereo camera for experiments.

References

[1] J. Borenstein and Y. Koren, "The vector field histogram-fast obstacle avoidance for mobilerobots," *IEEE Transactions on Robotics and Automation*, 7 (3): pp. 278–288, 1991.

[2] S. Petti and T. Fraichard, "Safe navigation of a car-like robot within a dynamic environment," *Proceedings of the European Conference on Mobile Robots*, Sep 2005.

[3] W.H. Huang, B.R. Fajen, J.R. Fink, and W.H. Warren, "Visual navigation and obstacle avoidance using a steering potential function," *Robotics and Autonomous Systems*, 54 (4), pp. 288-299, 2006.

[4] J. Saumders and R. Beard, "Reactive vision based obstacle avoidance with camera field of view constraints," *AIAA Guidance, Navigation and Control Conference and Exhibit*, 2008.

[5] T.C. Smith, R. Evans, L.P. Tychonievich, and J. Mantegna, "AUV control using geometric constraint-based reasoning," *IEEE Symposium on Autonomous Underwater vehicle Technology*, pp. 150-155, 1990.

[6] K.Y. Kim, J.W. Park, and M.J. Tahk, "UAV collision avoidance using probabilistic method in 3-d," *Proceedings of the International Conference on Control, Automation and Systems*, pp. 826-829, Oct. 2007.

[7] A.M. Zhang and L. Kleeman, "Robust appearance based visual route following for navigation in large-scale outdoor environments," *The International Journal of Robotics Research*, pp. 331-356, 2009.

[8] A. Diosi, A. Remazeilles, S. Segvic, and F. Chaumette, "Outdoor visual path following experiments," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4265-4270, 2007.

[9] Y. Matsumoto, M. Inaba, and H. Inoue, "Visual navigation using view-sequenced route representation," *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 83-88, 1996.

[10] H.C. Lee, T.S. Lee, S.H. Lee, G.H. Eoh, and B.H. Lee, "Visual path following and obstacle avoidance using multiple cameras for outdoor environments," *Proceedings of the International Conference on Ubiquitous Robots and Ambient Intelligence*, pp. 709-711, Oct, 2009.

[11] L. Matthies, T. Litwin, K. Owens, A. Rankin, K. Murphy, D. Coombs, J. Gilsinn, T. Hong, S. Legowik, M. Nashman, and B. Yoshimi, "Performance evaluation of UGV obstacle detection with CCD/FLIR stereo vision and LADAR," *Proceedings of the IEEE ISIC/CIRA/ISAS Joint Conference*, pp. 658-670, 1998.

[12] A. Tsalatsanis, K. Valavanis, and A. Yalcin, "Vision based target tracking and collision avoidance for mobile robots," *Journal of Intelligent Robot Systems*, 48, pp. 285-304, 2007.

[13] K. Macek, R. Philippsen, and R. Siegwart, "Path following for autonomous vehicle navigation based on kinodynamic control," *Journal of Computing and Information Technology*, - CIT 17, 1, pp. 17-26, 2009.