Mobile Robot Driving using Moving Window

Sung-Yug Choi*, Jin-Gu Kang**, Hwa-Ra Hur*** and Jin-Hwa Ju****

* Research Inst. of Computer Information & Communication, Pusan National University, Busan, Korea
(Tel : +82-51-510-2598; E-mail: sychoi3@korea.com)
**Department of Electrical Engineering, Keukdong College, U'mseong, Korea
(Tel : +82-43-879-3434; E-mail: jgukang@hanmail.net)
***Department of Electrical Engineering, Songho College, Hoengseong, Korea
(Tel : +82-33-340-1030; E-mail: hrhur@songho.ac.kr)
****Department of Electrical Engineering, Osan College, Osan, Korea
(Tel : +82-31-370-2685; E-mail:jinhju@osan.ac.kr)

Abstract: This paper introduces a method that can detect obstacles and corridor environments from the images captured by a CCD camera in an automobile or mobile robot is proposed. Processing the input dynamic images in real time requires high performance hardware as well as efficient software. In order to relieve these requirements for detecting the useful information from the images in real time, a "Moving Window" scheme is proposed. Therefore, detecting the useful information, it becomes possible to search the obstacles within the driving corridor of an automobile or mobile robot. The feasibility of the proposed algorithm is demonstrated through the simulated experiments of the corridor driving.

Keywords: Mobile robot, Corridor driving, Moving window, Obstacle detection

1. INTRODUCTION

This paper introduces a system in which obstacle detecting and obstacle avoidance are integrated. In this system, range data from vision sensors is continuously sampled and updated immediately while the robot is driving. Simultaneously, the obstacle avoidance algorithm uses the instantaneous center of corridor information to avoid newly detected obstacles.

The optimal size of the moving window can be determined as follows [1]: The size of the moving window should be big enough to include some part of a corridor information at any time, considering the maximum admissible speed of a mobile robot and the change of corridor conditions for maximum safety. Besides, the number of moving windows is minimized so that the area within the maximum safety distance could be searched for obstacles. One advantage of this integrated system is its ability to progressively adapt the strength of an obstacle avoidance reaction to the information of evidence for the existence of an obstacle.

This paper consists of six sections including the instruction, and proposes the optimal size of window in section Ⅱ, and discusses detection of corridor by moving windows and recognition of obstacles in section Ⅲ and Ⅳ respectively. And then it describes an experiment for detection of a useful information that is information of obstacles or corridor environments Ⅴ and deal with conclusion in section Ⅵ.

2. DECISION OF WINDOW SIZE

Input images have the size of 640× 480 pixel at every sampling time. It can cause the increase of unnecessary computation to apply preprocessing to total image in order to draw some necessary information. Therefore, the research on the decrease of computation time by designating a necessary part as a sub-block is being in progress [2-6]. This section explains edge detection schemes based on the image processing and the central moment that plays a key role in determining the position of windows. And then it decides the optimal size of windows. The proposed moving window technique is applied for the corridor navigation by a mobile robot, which is illustrated as Fig. 1.

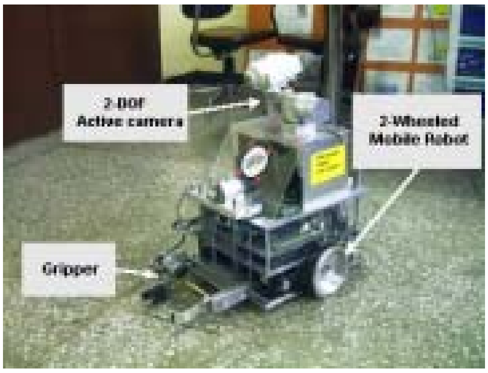

Fig 1. Mobile Robot.

A design frequently used for computer-controlled a robot consists of two drive wheels, each with its own controlled DC motor. One or two free-wheeling castors provide stability. Such a design was chosen for the robot of the nursing robot, as shown in Fig. 2. Two DC motors, with built-in reduction gears and incremental encoders, drive two wheels constituting the front axle of the robot. The resolution of the encoders is such that one pulse represents 2 [mm] of tangential travel of the drive wheel. The motors are coupled to the wheel shafts through a 2400 [pulse/rev].

In the rear, there is one free-wheeling castor. Although castors have been said to cause slipping during direction changes, it has been proven that this is may not always occur. A more complicated design that allows two DOF-motion in the plane and is based on two wheel-pair assemblies was presented in. However, this robot has been found to be very difficult to control and no satisfactory solution has been found yet.

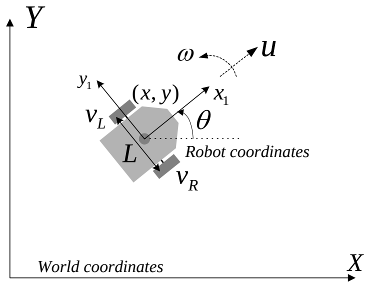

Fig. 2. Constructional diagram of the mobile robot.

In mobile robots it is desirable to place the two drive wheels as far apart as possible, for the following reasons:

1. The stability of the robot is improved.
2. The effect of the encoder resolution on the orientation error of the robot is decreased. In the worst case, the orientation error $\Delta\theta$ is given approximately by follow.

$$\Delta\theta = \frac{\Delta x}{a} \tag{1}$$

As seen from (1), the orientation error $\Delta\theta$ is reduced by increasing the distance between the drive wheels.

3. During straight-line motion, mechanical disturbances might cause the motors to run temporarily at different angular speeds, resulting in a temporarily curved path. It can be shown by trigonometry that the radius of the curved path is directly proportional to the wheel separation distance a.

4. Differences in the two wheel diameters will also cause a curved path with a radius proportional to the distance a.

On the other hand, an exaggerated base width will adversely affect the mobility within a room. In the present design the distance between the two drive wheels is 300 mm.

It is required to minimize the size of windows for image processing in real time, since, the information of corridor must be found from windows. The size of a window will be determined according to the condition that at least some part of a corridor must be included inside one sub-block.

Any point in real world corresponds to a point in the image obtained through a CCD camera as shown in Fig. 3.

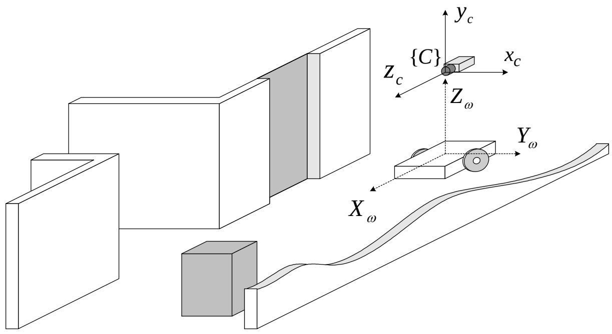

Fig. 3. Coordinate systems between a camera and a point in the real world.

3. DEFINITION OF MOVING WINDOW

3.1 Corridor detection

This section deals with the definition of moving windows and explains how to detect information of corridor using the information from the driving robot. After labeling as a preprocessing, from the area that is marked as the edge region inside a moving window, a central moment is obtained. Note that the position of corridor information in the input image with the sampling time 10 [ms], does not change drastically, compared to the size of a moving window. Based on the position of central moment, a check point is determined to be used as the center of a moving window.

The proposed moving window technique is applied for the corridor navigation by a mobile robot, which is illustrated as Fig. 4.

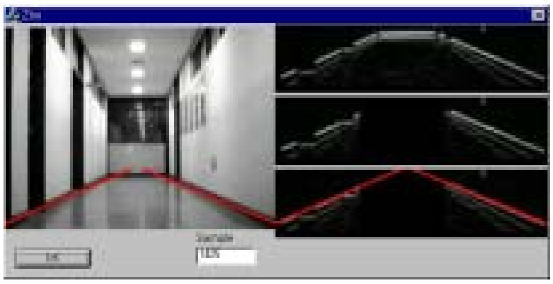

Fig. 4. Corridor information.

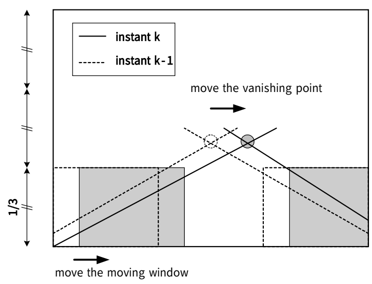

Fig. 5. Driving situations of a mobile robot in the corridor.

3.2. Expectation of moving window position

This subsection describes the decision of searching area at the instant ($k+1$). The searching area is called as "Moving Window" in this paper, which can be represented as a check point, x_k. In other words, the current position of a moving window (k), x_k, can be estimated based upon the information on the check point of the previous one, and the error between the estimated, x_k^-, and the measured, x_k^+, will be compensated via Kalman filter [7-8]. Note that once those are determined, the moving windows can move along the X-axis. Therefore x_k is enough to represent the location of a moving window.

The state x_k^- is estimated as,

$$x_k^- = F x_{k-1}^+ + \omega_{k-1} \tag{2}$$

where, x^+_{k-1} means a measurement value in instance $(k-1)$ and ω_{k-1} represents the Gaussian noise from the system model. And a system matrix, F, is defined as,

$$F = \begin{pmatrix} 1 & \Delta_t \\ 0 & 1 \end{pmatrix} \tag{3}$$

where Δ_t represents the measurement time of the check point.

If the covariance matrix of ω_{k-1} is denoted as Q_{k-1}, the covariance matrix of the estimation can be represented as $P^-_k = P^+_{k-1} + Q_{k-1}$. Besides, in the measurement step, the measurement vector y_k can be denoted as,

$$y^-_k = Hx_{k-1} + v_{k-1} \tag{4}$$

Note that v_{k-1} is uncorrelated with ω_{k-1} in the estimation step and its covariance matrix is denoted as R_{k-1} in the measurement step. Since the image itself does not show the change of distance per image frame, the measurement matrix can be represented as

$$H = [1 \quad 0] \tag{5}$$

The recursive Kalman filter to compensate the state vector by adding measurement deviation to the estimated can be described as follow:

$$x^+_k = x^-_k + G_k(y_k - Hx^-_k) \tag{6}$$

where Kalman filter gain G_k, which adjusts the state vector by assigning the appropriate weight between the measured and the estimated, can be selected as [9],

$$G_k = P^-_k H^T \left[HP^-_k H^T + R_k \right]^{-1} \tag{7}$$

Consequently, the estimated state for the state vector compensated in (7) can be represented as.

$$x^-_{k+1} = Fx^+_k + \omega_k \tag{8}$$

The whole procedure - the information of corridor constituents are detected through Prewitt Mask, a check point is determined through operations, moving windows are determined to detect corridor and obstacles, and based on these, the new position of a sub-block (that is a moving window) is determined using Kalman filter.

4. OBSTACLE DETECTION

When the position of a right corridor line and a left corridor line is denoted as $F_L(y)$ and $F_R(y)$ respectively, the center of corridor can be calculated as

$$F_c(y) = F_R(y) - F_L(y) \tag{9}$$

The intensity of a center of corridor is $f_c(x, y)$ within 0 to 256 representing gray level. An obstacle ahead of the robot (the obstacle here is actually another robot) has the characteristics of horizontal constituents such as a rear bumper and a trunk. So if an edge gradient is detected in the horizontal direction, the expected obstacle can be characterized. Using $f_{cB}(x, y)$ that is the binary horizontal edge, the estimated area of the obstacle, that is $f_{Ob}(x, y)$, can be extracted.

The estimated area of the obstacle starts from the bottom line of the obstacle(Ob1) in Fig. 5, and the right bottom(Right1) and the left bottom(Left1) are determined by taking 60[%] of the corridor width, which is the average of the obstacle width. In order to obstacle, the right top(Right2) and the left top(Left2) are determined by taking the half of the bottom width. Therefore, the scope for the obstacle recognition can be limited to the inside of a driving corridor.

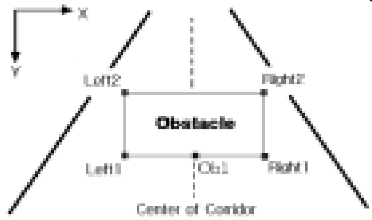

Fig. 5. Expected area for obstacle.

5. OBSTACLE RECOGNITION

A neural network has frequently been used in the previous studies for the obstacle detection from an input image [9]. In this study, a neural network is introduced as a pattern classifier to make a distinction between the obstacle and the other. The structure is a forward multi-layer neural network as shown in Fig. 6 and the input layer of the neural network consists of 35 neurons, the hidden layer 16 neurons, and the output layer 6 neurons. The sigmoid activation function is used for each node.

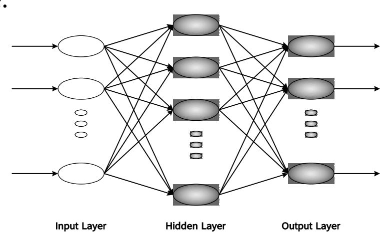

Fig. 6. Structure of a neural network.

The input reflects the attributes of the pixels in a subset, and it is normalized to have the value between 0 and 1 since the sigmoid function is made up of exponential functions.

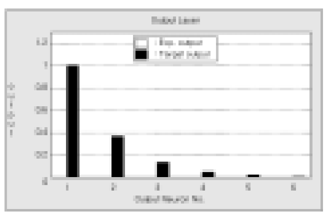

Fig. 7. Target value for the obstacle in the neural network.

In Fig. 7, the horizontal axis represents the number of the output layer neuron when the output of the neural network consists of six neurons, and the vertical axis represents the value of the output layer. In case the input image is an obstacle, the target output value – the black bar-- of the output layer neuron is the highest in No. 1 neuron followed by No. 2, No. 3, etc. That is, the output value is set to be decreased exponentially from the No. 1 output value.

When the input image isn't a obstacle, the target output value is the highest in No. 6 neuron followed by No. 5, No. 4, etc. The output is made decreased exponentially from the No. 6 output value. The reason for using the increasing and decreasing exponential functions, instead of putting two output layers for each of the obstacle and other condition, is to give some flexibility for the case that the unknown images are coming into the network.

The following is the output of the neural network for mobile robot corridor driving.

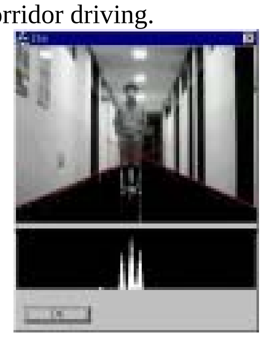

Fig. 8. Obstacle detection.

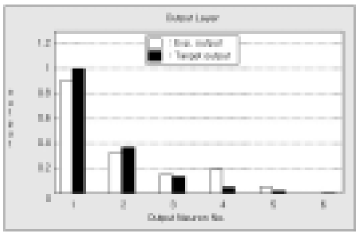

Fig. 9. Neural network output for an obstacle.

As shown in Fig. 8, a man is recognized as an obstacle by the decreasing outputs, the white bars in Fig. 9.

6. EXPERIMENTS AND RESULTS

This paper presents is algorithm for selecting sub-blocks and optimizing the sizes is proposed. This new moving window technique shows excellent performance in detecting information of corridor and. It was applied corridor driving of a mobile robot. The size of a moving window is big enough to include some part of corridor information at any time, considering the maximum admissible speed of a robot and the change of conditions for maximum safety. Besides, the number of moving windows is minimized so that the area within the maximum safety distance could be searched for obstacles. The optimized moving windows at the current instance are moved to the estimated positions for the next instant by the curve fitting and Kalman filtering techniques.

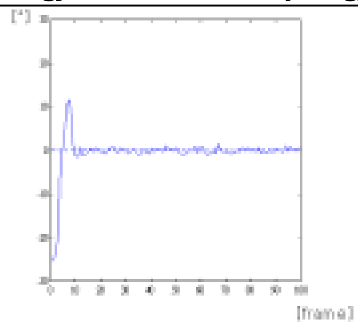

Fig. 9. Driving error ration of mobile robot.

Moreover, when the proceeding direction is determined by fitting the detected information on corridor, the unexpected loss of information caused by external environment is resolved by extrapolating the information of the previous instants. Therefore, this method is useful for the case that the input image disappears temporarily because of CCD's delay in the shadow below a light. By locating a center of corridor between the two detected information of corridor to search obstacles, the front areas can be considered as obstacles. It is also applicable to AGV (Autonomous Guided Vehicle), which moves slowly driving in corridor.

REFERENCES

[1] Reinhold Behringer and Nikolaus Muller, "Autonomous Road Vehicle Guidance from Autobahnen to Narrow Curves," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 5, pp. 810-815, Oct. 1998.

[2] Christopher E. Smith, Charles A. Richards, Scott A. Brandt and Nikolaos P. Papanikolopoulos, "Visual Tracking for Intelligent Vehicle-Highway Systems," *IEEE Transactions on Vehicular Technology 45(4)*, pp. 744-759, 1996.

[3] Makoto Hirano, "Development of Vehicle-Following Distance Warning System for Trucks and Buses," *IEEE-IEE Vehicle Navigation & Information Systems conference*, Ottawa-VNIS'93, 1993.

[4] D. Koller, K. Daniilidis and H. -H. Nagel, "Model-Based Object Tracking in Monocular Image Sequences of Road Traffic Scenes," *International Journal of Computer Vi*sion, 10:3, pp. 257-281, 1993.

[5] S. Y. Choi and J. M. Lee, "Lane Recognition and Obstacle Detection Using Moving Windows," *Journal of The Institute of Electronics Engineers of Korea*, vol. 36-S, no. 1, pp. 93-103, Jan. 1999 (in Korean).

[6] S. Y. Choi, T. S. Jin and J. M. Lee, "Optimal Moving Windows for Real-Time Road Image Processing," *Journal of Robotic Systems*, vol. 20, issue 2, pp. 65-77, Feb. 2003.

[7] W. Kwon, J. W. Lee, D. M. Shin, K. S. Rho, D. Y. Kim, and S. H. Lee, "Experiments on Decision Masking Strategies for a Lane Departure Warning System," *Proc. of IEEE Conf. on Robotics and Automation*, pp. 2596-2601 Detroit, May 1999.

[8] C. H. Lee, H. T. Kim, S. W. Seol, K. G. Nam, J. M. Lee, "Multiple Vehicle Tracking Algorithm Using Kalman Filters," *Journal of The Institute of Electronics Engineers of Korea*, vol. 36-S, no. 3, pp. 359-365, Mar. 1999.

[9] Charles E. Thorpe, Takeo Kanade, *Vision and Navigation*, Kluwer Academic Publishers, pp. 9-82, 1990.