International Journal of Fuzzy Logic and Intelligent Systems, vol. 2, no. 1, March. 2002 pp. 1-8

A Study on Development of Visual Navigation System based on Neural Network Learning

Suk-Young Shin, Jang-Hee Lee, Yang-Jun You and Hoon Kang

School of Electrical and Electronic Engineering, Chung-Ang University
221, Huksuk-Dong, Dongjak-Gu, Seoul 156-756, Korea

Abstract

It has been integrated into several navigation systems. This paper shows that system recognizes difficult indoor roads without any specific marks such as painted guide line or tape. In this method the robot navigates with visual sensors, which uses visual information to navigate itself along the road. The Neural Network System was used to learn driving pattern and decide where to move. In this paper, I will present a vision-based process for AMR(Autonomous Mobile Robot) that is able to navigate on the indoor road with simple computation. We used a single USB-type web camera to construct smaller and cheaper navigation system instead of expensive CCD camera.

Key words : Vision, Navigation, Autonomous Mobile Robot, Neural Networks

I. Introduction

It has been developed that AMR (Autonomous Mobile Robot) navigates on the road with non-visual sensors as like ultrasonic sensor. It has been also researched in many methods that the algorithm avoids obstacles and constructs road map itself with distance data to the wall or obstacles using ultrasonic sensor, but there are difficulties to recognize objects with only distance data and then it needs visual sensor to acquire visual data which human can recognize comfortable with [6]. Human don't use distance data to avoid or recognize obstacles, but visual information. Therefore it is needed to research on navigation system using visual information.

However, it is necessary to control the direction and internal camera parameters of the video sensor actively in order to have them attend to the portion of the world that is to be studied. It also has the weakness of noise from circumstance, so it is required that pre-process to overcome these weaknesses and there are still remained a lot of problems now.

There are two different methods for the robot to navigate. In a controlled environment, we can define a few known landmarks before system design, and the navigation system can employ landmark detectors. Such navigation systems typically employ a model-based design method. However these methods have difficulties dealing with learning in complex, changing environments. Model-free design methods present a potentially powerful alternative. Model-free methods lack any predefined model of the driving environment during system design. In other words, a human programmer does not specify any environment model, either explicit or implicit; the system itself must automatically derive the model [1].

In this paper, I will present a vision-based process for AMR(Autonomous Mobile Robot) that is able to navigate on the indoor road with simple computation. We used a single USB-type web camera to construct smaller and cheaper navigation system instead of expensive CCD camera which was usually applied to navigation system.

We apply a preprocessor for this system to reduce noise and computation which is the one of the most significant problem in the image processing field. To make real-time system, we also have to diminish computations as much as possible. The block diagram of navigation system is shown in Figure 1.

II. Processing

The recognition of image is necessary for robot to navigate, but we have to convert image to appropriate data. It is required to reduce noise, extract edges, and detect straight lines in pre-processing.

If we were using color image captured from camera, it should compute lots of numerical formula for each RGB information. It would take a long time as well as become complicate. Therefore, the color image has been changed to gray scale image. The following shows how to change image to gray scale.

$$\text{Gray Scale Intensity} = \frac{299R + 587G + 114B}{1000}$$

2.1 Edge Extraction

We need to extract edges to find road boundaries from input image which was converted to gray scale. There are a

This paper was supported in part by Ministry of Commerce, Industry and Energy's developing the IWM(Intelligent Wearable Module) project in 2001(N09-A08-4301-04)

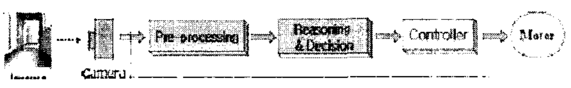
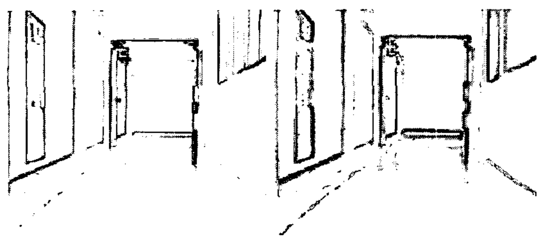


Fig 1. The block diagram of navigation syste

(a) Original Image. (b) Laplacian edge extraction.



(c) Sobel edge extraction. (d) Canny edge extraction.

Fig. 2. Results of some edge extraction methods.

lot of methods to extract edges such as Laplacian, Prewitt, Sobel, Canny, and etc. We had empirically found that the best result could be get from Sobel filter for this research.

Laplacian transform is a general method to extract edges, shown in figure 2 (b). This method has properties of detecting edges faster than others, using 2nd order differential operator, and extracting with only one mask in all direction. Sobel filter has the strength that it can extract edges in row and column component separately. However, it is slow when compared to Laplacian and sensitive to brightness as much as it detects noise as edges, shown in figure 2(c).

It gives effect of noise reduction by applying Gaussian filter that blurs image. After Gaussian filter it extracts line with the Sobel filter, which is Canny filter. It has a characteristic to extract only explicit lines and strong to noise, shown in figure 2(d). Canny is slow, since it takes mask convolution twice. The results of each edge extractions are described on figure 2 respectively.

In addition, too many vanishing lines are detected on the same line with a little different slopes, because Canny makes thick lines. It causes increasing of the error between computed and practical vanishing point. Even though we could take a great advantage of noise reduction from Canny extraction, we decided to use Sobel filter in consideration of computation time and detecting vanishing lines.

2.2. Threshold

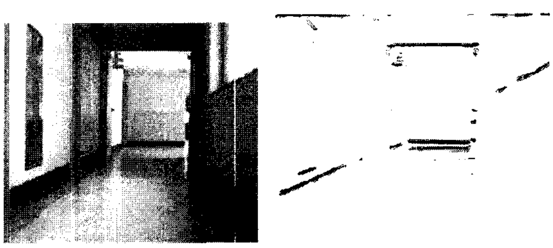
There are three different thresholds which are for row and column component of Sobel filter and line length of Hough transformation. The threshold values have considerable influence upon the result of edge extraction. The effects of threshold value are shown well in figure 3.

Because finding appropriate threshold by it self is involved in another field, we made the threshold can be set by human in initial state.

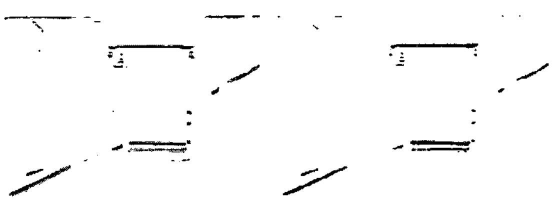
When it changes input image to gray scale and binary image, it is also get pretty different result according to binary conversion threshold. However general method, middle value, was used in this system.

2.3 Noise

We could take some advantages when we use the row component mask of Sobel filter. Especially, it has significant effect of noise reduction. One of the most serious problems was sunlight through window and reflection of fluorescent lights on the floor. It made a specific shape which could be recognized as a boundary of road or obstacles, soit made difficult to get correct information.



(a) Original Image (b) Threshold = 50



(c) Threshold = 100 (d) Threshold = 100

Fig. 3. The influence of threshold value upon the image. extracted lines are decreased as threshold increase.

There are another big merit from applying only row component that we can reduce those noises greatly and acquire suitable images to recognize road boundaries as well. The figure 3 was applied only row component Sobel filter.

III. Road Characteristic

3.1 Continuous Characteristic

The road boundaries are always continuous except intersection. As matter of fact intersection boundaries are not disconnected. They are just changing to the opposite directions, one to the left and the other to the right. We could say that road boundary is strictly continuous.

The road the image doesn't tend to change suddenly. We assumed that the road boundary is linear in near section. Even though the road image has curvature lines, the near section of the road boundaries are located on the left and the right end sides of input image, can be regarded as linear line. So we could find out vanishing point which was defined from extending both road boundary lines in the near section of the road.

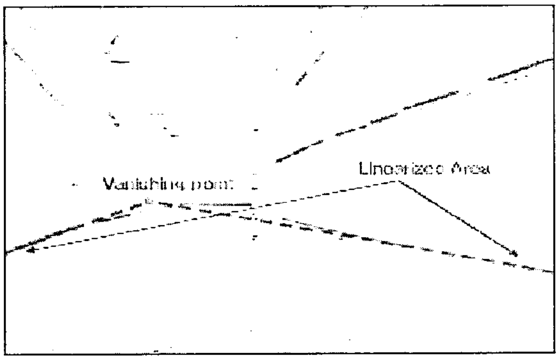


Fig. 4. Road boundary is linearized around end of each side

3.2 Vanishing Point

It is well known that perspective projection maps each set of parallel lines from 3-D space into the set of half lines in the projective plane with a common endpoint called a vanishing point. The vanishing point is related with looking for main direction of road or corridor in 3-D space. All extended lines finally converge to vanishing point in 3-D space structure.

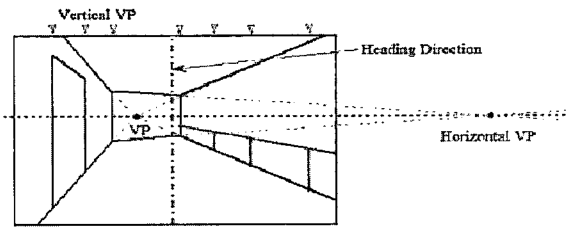


Fig. 5. Location of vanishing points which are corresponding to longitudinal, transversal, vertical direction in the hallway image.

All extracted lines can be partitioned into three different subsets so that lines from each subset correspond to a different subset of parallel edges in the scene and intersect the associated vanishing point. Those are longitudinal, vertical, and horizontal vanishing point. The locations of the three vanishing points in these scenes are mutually dependent, described in figure 5 [2].

The vertical vanishing point as well as longitudinal vanishing point was used to the robot in this research. The Longitudinal vanishing point should give great effective variable to decide where the robot has to move. Of course the robot can navigates only with longitudinal vanishing point and some other research used only longitudinal vanishing point to steer. But, we additionally used vertical vanishing points complementary as another control input. If robot was oriented to the right than vanishing point, the more vertical vanishing points would normally be found on the right side from vanishing point. If paintings or doors were on the left side wall and nothings were on the other side, even though the robot oriented to the left, more vertical vanishing points would be found on the left side contrary. Since it is not always correct, this variable should be used minor effective variable.

3.3 Heading Direction

The orientation line always locates on the middle of horizon in the input image as shown in figure 5. When the vanishing point corresponds to heading point, the robot will move to the goal point. So it must try to reduce the deviation (Δy) between two points. If vanishing point was detected on the left side from heading direction, it should make a turn to the left. Otherwise, it should move to the right.

IV. Road Recognition

4.1 Hough Transformation

Hough transformation was used to recognize road boundaries and slope of them from image in this paper .Hough transformation is known as a method to extract the straight line in the image. Hough transformation can detect the slope of line with high accuracy. Furthermore, Hough transformation can detect the straight lines such as dotted lines [3].

The input for the Hough transform is an image that has been preprocessed. Let us consider binary image on the x-y plane and u-v plane that is a space of sinusoidal function shown in Figure 6.

If a pixel exists on the x-y plane, Hough transform is mapping x-y plane into u-v plane expressed in equation (1) on the u-v plane.

$$Vc = x\cos(u_c) + y\sin(u_c) \tag{2}$$

where, x and y are coordinates of the pixel on the x-y plane.

The property that the Hough algorithm relies on is that each of the curves have a common point of intersection, namely. Conversely, the sinusoidal curve passes through the point in the u-v plane if lies on the line in the x-y plane [4].

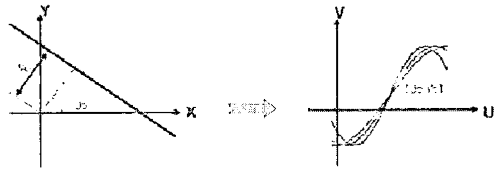


Fig. 6. Hough Transformation. Transform from X-Y plane to U-V plane. (a) Relation of rectangular to polar representation of a line. (b) Sinusoidal in Hough space associated with points on a line.

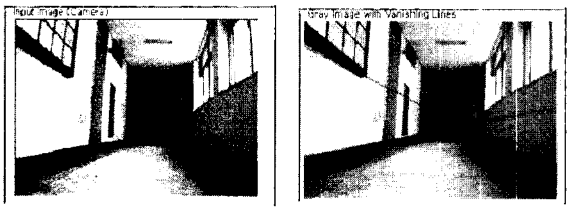
This process is repeated as much as the number of pixels.

When pixels more than one are exist on a line on the x-y plane, curves on the u-v plane correspond to those pixels cross each other on a point. To find straight lines make u-v plane be quantized, and then find accumulator values greater than specified threshold value.

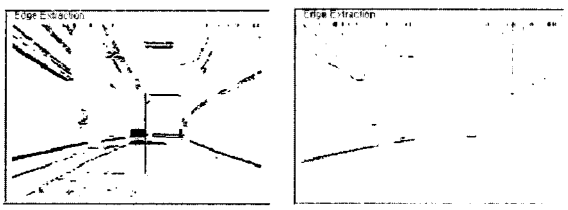
Therefore, the intercept of line and slope of a straight line can be detected by Hough transformation.

4.2 Finding Vanishing Points

We can figure out road boundary lines Hough transform. More than one line was found from each side boundary, so more than one vanishing point should be detected. Because of this reason it was needed to find where exact vanishing point would be located. Following methods could be considered, deciding segment which includes most vanishing points, using clustering methods, and finding centroid of vanishing points. I chose an approach to the detection of vanishing point with centroid method because it is easy and fast to calculate.



(a) Input image from camera. (b) Road boundaries are found from input image.



(c) row edges are extracted for finding longitudinal vanishing point. (d) column edges are extracted for finding vertical vanishing point.

Fig 7. The system finds the vanishing lines, longitudinal vanishing point, and vertical vanishing point

The column of vanishing point becomes the baseline to find out the number of vertical vanishing point on each side. It counts the number of vertical lines on the left and right side respectively. The vertical vanishing points could be also found with Hough transform, but it took many computations which made system slow. These values are not also considerable variable for navigation vertical vanishing points. It doesn't necessary to find exact values with these reasons. If extracted line was longer than specific length in certain column, it would regard its column as vertical vanishing point for making system faster.

Figure 7 describes the result of finding longitudinal and vertical vanishing points respectively. Figure 7(b) describes the longitudinal vanishing lines are detected by Hough transform. As explained, more than one vanishing lines are detected on same edges. The longer blue line means the center of detected vanishing points which are small blue lines in figure 7(c). Figure 7(d) shows that it found vertical vanishing points from extracted edges with column mask. The brown color lines were marked on the top of column where the vertical vanishing points were detected. The red line, it splits the image half, presents where the robot is heading to.

4.3 Navigation Strategy

The deviation between the vanishing point and the heading point was used to steer orientation as the most effective variable. The deviation and the number of vertical vanishing points on each side present the relative position and orientation of the robot on the road. The robot is needed to move its center, objective line, to vanishing point for making robot parallel to the road.

It didn't find vanishing point sometimes, when the robot usually faced to the wall too close. In case of no finding vanishing point, the robot moves backward a little with slow speed for its safety. Following figure 8 shows flowchart of navigation strategy and algorithm.

V. Neural Networks

5.1 Applying Back propagation Algorithm.

Back propagation algorithm was applied for this system to learn driving pattern. There are two ways to learn driving pattern, on-line and off-line method. The on-line learning is that the system learns driving pattern during human controls the robot manually. The other way, off-line methods, is that the system learns pattern with predefined pattern table in advance.

In this research the learning data, desired values, are made previously in table of normal navigation patterns and then train the robot with it. The neural network is consist of 3 input nodes, 8 hidden nodes, and 2 output nodes for this robot.

The deviation (Δy) and the number of vertical vanishing points were used as input data of the neural network. As you see Figure 9, the outputs of network will be steering value to the each DC-motor. These values decide where it should move and how fast it can move around. Following figure describes the neural network module used for the robot navigation system.

The range of input should be normalized, because each input nodes has too much large values if it didn't. Since it has been set to 320*240 pixel image, input of the deviation has range from -160 to 160 and the input of the number of vertical vanishing point can have from 0 to 320. So, we changed the range -160 ~ 160 for the input of the deviation and -3.2 ~ 3.2 for each input of the number of vertical vanishing point. The output was set to have only positive value from 0 ~ 1.0 which means the robot can move only forward In this neural network module. The output node1.0 which means the maximum speed of motor, otherwise' -1 ' rotates it backward, and 0 stops motor. The output value has minus value which means moving backward, only in case of the robot could not find vanishing point.

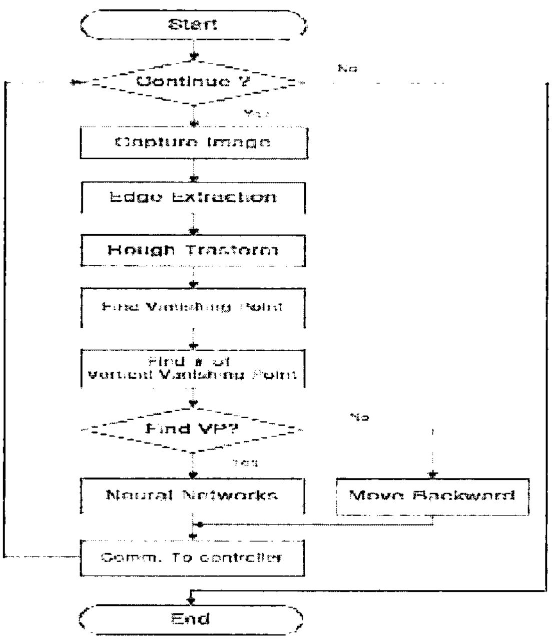


Fig. 8. The flowchart of navigation strategy for the mobile robot.

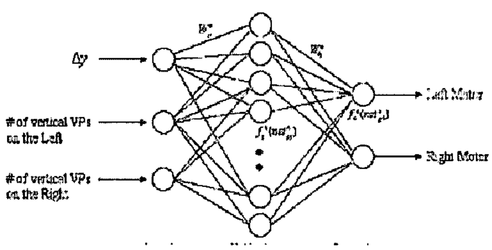


Fig. 9. Back propagation module for the navigation system.

After all, the neural network learns the relationship between the steering angle and the deviation of vanishing point and the number of vanishing points.

5.3 Simulation for each pattern model.

We made two driving patterns for simulation of learning algorithm, those were normal and log-sigmoidal driving pattern. The normal pattern was defined to make a turn faster as the deviation became great. It made a smooth and slow turn and maintained the speed gap between motors as much as possible in the opposite case. The normal driving pattern is shown in figure 10 (a).

The learning outputs are shown for the normal driving pattern in figure 10. The learning was improved as hidden nodes were increased till 64 nodes in complicate patterns as like normal driving pattern. It took a long time to converge the error below 0.001 with 4 or 8 hidden nodes.

If normal driving model was a kind of nonlinear pattern, the log-sigmoidal model would be a linear pattern that was made from formula. For applying this pattern formula should be changed a little like followings.

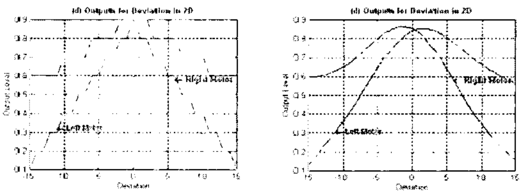
$$\text{Output for left}: \ O_{left}(x) = \frac{1}{1+e^{0.2(x+0.1s)}} \tag{3}$$

$$\text{Output for right}: \ O_{left}(x) = \frac{1}{1+e^{-0.2(x+0.1s)}} \tag{4}$$

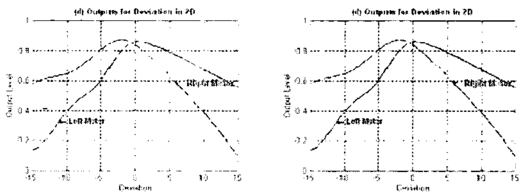
where s=(the number of left VVP - the number of right VVP).

The log-sigmoidal function should be expanded5 times by multiplying coefficient 0.2 to the input term the reason of its own characteristic. If we uses original log-sigmoid function, the robot will steer suddenly even in small deviation as like motor on-off method. To have minor influence from vertical vanishing points coefficient 0.1 was multiplied to s.

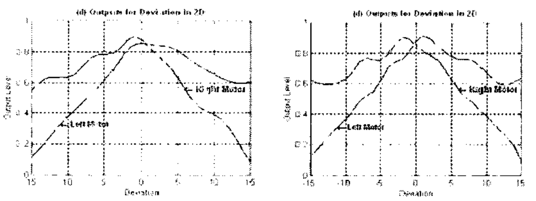
The difference between the number of vertical vanishing point gave the effect to shift the function on the x-axis. When s has positive number, the more vertical lines are detected on the left side, the outputs are shifted to the left (negative). it means that the left output has greater value and the right output has lower value. Therefore the robot can make a turn faster. Vertical vanishing point has the effectiveness of helping the robot's steering. The result of learning for the sigmoidal driving pattern is shown in figure 10.



(a) Desired output. (b) The # of hidden node = 4.



(c) The # of hidden node = 8. (d) The # of hidden node = 16.

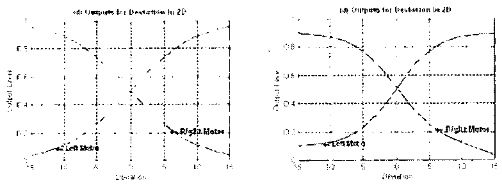


(b) The # of hidden node = 32. (e) The # of hidden node = 64.

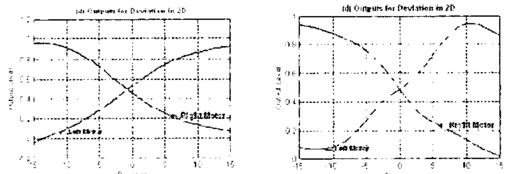
Fig 10. Relationship between outputs and deviation for normal driving pattern in mean squared error(MSE) = 0.0008

As the number of hidden node exceeds 8 nodes, the performance gets worse on the contrary. log-sigmoidal graph was disfigured from the 16 hidden nodes with same error. We can see that the more hidden nodes are not always having better performance, especially for the simple function.

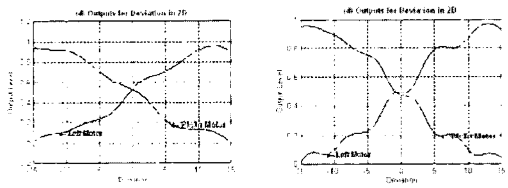
In consideration of necessary for the robot to navigate with perfect desired output, 8 hidden nodes and learning coefficient, were used for learning patterns. The results of learning process for the normal and log-sigmoidal patterns are shown in figure 12 and figure 13 respectively.



(a) Desired output. (b) The # of hidden node = 4.



(c) The # of hidden node = 8. (d) The # of hidden node = 16.



(b) The # of hidden node = 32. (b) The # of hidden node = 64.

Fig. 11. Relationship between outputs and deviation for log sigmoidal driving pattern in mean squared error (MSE) = 0.000

VI. System Architecture

Notebook computer with Pentium III-600Mhz was used as a main processor and we applied USB type PC-camera to construct cheaper and smaller mobile robot system in stead of expensive CCD camera. Controller is consist of 80C196KC

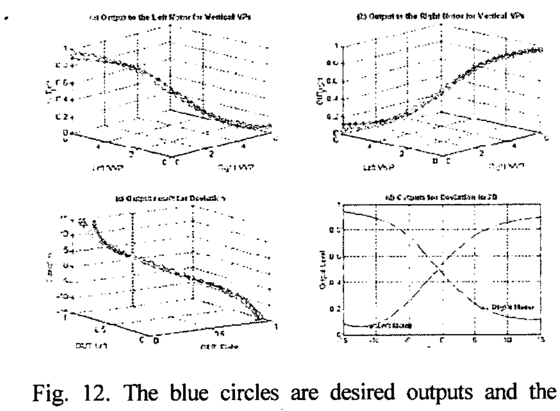


Fig. 12. The blue circles are desired outputs and the red stars are outputs from networks after learning for log- sigmoidal driving pattern.

CPU,LM629 DC-motor controllers, and LMD18200 motor drivers. The architecture of system is shown in figure 14.

Input image size was set to 320*240 in consider to computation time. The image processing and learning algorithm are computed in notebook, and then it sends the outputs of neural networks to the 80C196KC through the serial cable. 80C196 writes desired movement values on the LM629s. LM629 makes PWM pulse and direction value according to received value from 80C196 and interfaces to a motor via an incremental encoder for feedback control.

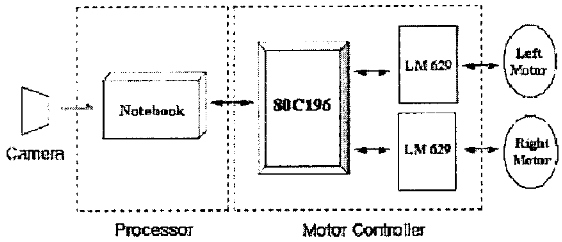


Fig. 13. The block diagram of the robot control system.

VII. Result

The whole navigation system is shown in figure 15. The left-top image is an input image from camera and the right-top image shows gray scale image with vanishing lines. The right-bottom image shows the extracted edges with row and column mask separately and longitudinal vanishing points. The obtained input values and the output values from networks are on the bottom box.

The result of processing on the slightly curved corridor is shown in Figure 16. It shows how vanishing points are detected and the outputs are changed the outputs according to the inputs, the deviation and the number of the vertical vanishing point.

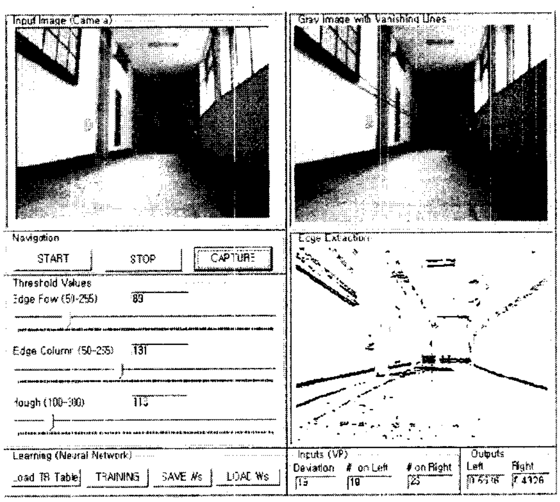
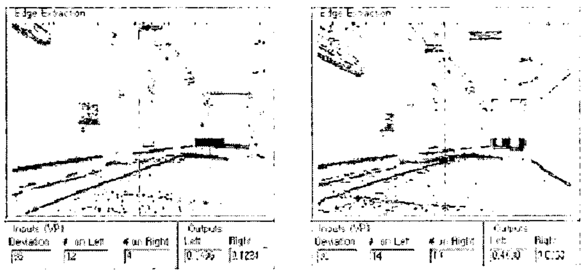
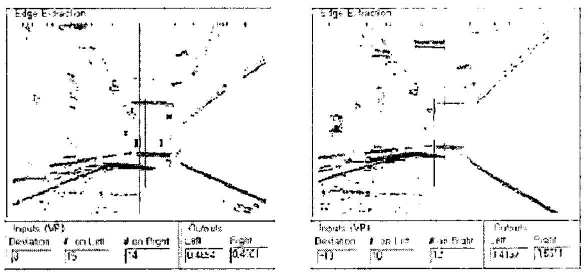


Fig. 14. The picture of running navigation system on the corridor.



(a) Step 1. Vanishing point is on the right side from orientation line.

(b) Step 2. Vanishing point moves a little left as robot moves to the right.



(c) Step 3. Vanishing point is getting close to orientation line.

(d) Step 4. Vanishing point passed the orientation line, so it will move to the left.

Fig. 15. It shows the results in sequence as it moves on the road.

Ⅷ. Conclusion

The proposed method in this paper showed a good performance not only on the straight but also on the a little curved road. This approach is meaningful that it does not need complex geometrical computation and can be implemented easily. This research shows that the neural network can learn relationship between vanishing point and steering angle. The vanishing point gave beneficial effect to decide steering values.

Even though there were some noises as like sunlight and shades during the navigation, satisfied results were obtained. The vanishing lines of noise would also converge to the vanishing point, it didn't give any effect in this method. It was proved that this system was stronger to the noise than others.

However, this system still has processing speed problem. Though we made great effort to reduce computation and improve speed, it is not fast enough yet.

Reference

[1] Juyang Weng, Shaoyun Chen, "Visual Learning with Navigation as an Example", *IEEE Intelligent Systems,* Volume : 15 Issue :5 , Sept-Oct. 2000 Page(s): 63-71

[2] Sinisa Segvic and Slobodan Ribaric, "Determining the Absolute Orientation in a Corridor Using Projective Geometry and Active Vision", *IEEE Transactions on Industrial Electronics*, Volume : 48 , No :5 , JUNE 2001 Page(s): 696-710

[3] Nishira H., Kojima A., Murakoshi H., and Ishijima S., "The multi-layer neural network applied to a car detection system", *IEEE International Workshop* on 1992, Robot and Human, Proceedings, pp. 88-92, 1992

[4] Rafael C. Gonzalez and Richard E. Wood, "Handbook of Computer Vision Algorithms in Image Algebra -2nd edition", CRC Press LLC, 2001.

[5] Martin T. Hagan, Howard B. Demuth, and Mark Beale, "Neural Network Design", PWS Publishing Co., 1996

[6] Gregory Dudek and Michael Jenkin, "Computational Principles of Mobile Robotics", Cambridge University Press, 2000

[7] Rafael C. Gonzalez and Richard E. Woods, "Digital Image Processing", Addison-Wesley Publishing Company, 1993

[8] Ohya A., Kosaka A., and Kak A., "Vision-Based Navigation by a Mobile Robot with Obstacle Avoidance Using Single-Camera Vision and Ultrasonic Sensing", Robotics and Automation, *IEEE Transaction on*, vol. 14 Issue 6, Dec. 1998

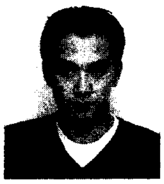
[9] Jill D. Crisman and Charles E. Thorpe, "SCARF: A Color Vision System that Tracks Roads and Intersections", Robotics and Automation, *IEEE Transactions on*, vol. 9 Issue. 1, pp. 49-58, Feb. 1993

Suk-Young Shin
received the B.S. degree in control and instrumentation engineering and M.S. degree in electrical and electronic engineering from Chung-Ang University, Seoul, Korea, in 1998 and 2002, respectively. His research interests are Image Processing, Neural Networks and Vision Navigation Systems.

Phone : +82-2-816-8234
Fax : +82-2-816-1856
E-mail : icecom@sirius.cie.cau.ac.kr

Jang-Hee Lee
received the B.S. degree in electrical and electronic engineering from Chung-Ang University, Seoul, Korea, in 2001. He is currently pursuing the M.S. degree. His research interests are Neural Networks, Vision Navigation Systems, Robotics and Intelligent Control.

Phone : +82-2-816-8234
Fax : +82-2-816-1856
E-mail : whizkid@sirius.cie.cau.ac.kr



Yang-Jun You
received the B.S. degree in electrical and electronic engineering from Chung-Ang University, Seoul, Korea, in 2002. He is currently pursuing the M.S. degree. His research interests are Image Processing, Neural Networks, Vision Navigation Systems and Intelligent Systems.

Phone : +82-2-816-8234
Fax : +82-2-816-1856
E-mail : rookie2001@sirius.cie.cau.ac.kr

Hoon Kang
received the B.S. and M.S. degrees in electronic engineering from Seoul National University, Seoul, Korea, in 1982 and 1984, respectively. He received the Ph.D. degree in electrical engineering in Georgia Institute of Technology in 1989. Since 1992, he has been a faculty member of School of Electrical and Electronic Engineering at Chung-Ang University. His research interests are Fuzzy Systems, Artificial Life, Cellular Automata, Neural Networks, Pattern Recognition, Evolutionary Computation, Vision System, Robotics and Intelligent Control.

Phone : +82-2-820-5320
Fax : +82-2-816-1856
E-mail : hkang@cau.ac.kr