Robot Control in Dynamic Environments Using a Fuzzy Clustering Network

Kai-Tai Song and Yuon-Hau Chen

Department of Electrical and Control Engineering
National Chiao Tung University
1001 Ta Hsueh Road, Hsinchu 300, Taiwan, R.O.C.

Abstract. A novel fuzzy clustering network has been developed for reactive motion control of a mobile robot in dynamic environments. The concept of dangerous index is proposed to represent obstacle configuration of the immediate environment of the robot. This index takes into account both the relative velocities and distances to the obstacles. A self-made real-time image card is employed to infer the dangerous index, which is then input to the proposed fuzzy clustering network for real-time obstacle avoidance. Thus more responsive control of the robot can be obtained when it handles multiple static and moving obstacles. A fuzzy behavior integration agent has been developed to combine the useful landmark-tracking behavior with responsive obstacle avoidance behavior. The effectiveness of the proposed method is confirmed by practical experiments using our laboratory-made mobile robot.

1 Introduction

Mobile robots capable of indoor navigation can have many useful applications such as working in hospitals, or they can be designed to work as cleaning robots, home robots, etc. In practice, such humanoid robotic systems face two important problems when put to work in the real world, namely goal-seeking path following capacity and real-time obstacle avoidance behavior. Several approaches to the mobile robot navigation in dynamic environments have been reported [1-6]. Most of them have been carried out in off-line manner. They required that the environment model must be known beforehand. Generally, these methods do not work well in complex and dynamically changing environments where unknown obstacles may be encountered on the planned path. Accordingly, sensor-based reactive path planning and real-time intelligent obstacle avoidance are required for of a truly humanoid robot.

Several types of detection sensors have been used for mobile robots, for instance, ultrasonic sensors, CCD cameras, laser range finders, and proximity sensors. Borenstein and Koren[7] proposed the virtual force field method for a mobile robot equipped with ultrasonic sensors. Ishikawa [8] presented an indoor navigation method using fuzzy control and emphasized on constructing an efficient control knowledge base. Song and Sheen [9] proposed a fuzzy-neuro network to handle reactive motion planning of mobile robot navigation. Some simple trap situations were considered in this method, but all obstacles were assumed to be static. Several approaches to sensor-based motion planning of mobile robots in uncertain dynamic environments have been proposed. In [10] moving obstacles was recognized employing a vision system. Chang and Song [11] proposed to construct an obstacle predictor in order to react in advance to handle multiple moving obstacles. In this paper, real-time vision system is exploited to demonstrate the capability of dealing with unexpected and unpredictable moving obstacles by means of the novel concept of dangerous index. The basic idea is to associate the environment with a number of representative prototype patterns, which is expressed in terms of dangerous index values. In order to establish the most representative prototype patterns, a competitive neural network was adopted for automatic sample clustering. These selected patterns are utilized to construct a fuzzy neural network. In each control loop, proper steering commands are generated according to the current input pattern. Rule tables have been built up to represent typical prototype patterns of immediate environment, such as multiple static obstacles, multiple moving obstacles, fast moving obstacles, etc. Thanks to the merit of membership property of the proposed fuzzy clustering network, these prototype-patterns are good enough for navigation in an general indoor environment. The fuzzy neural network has sufficient generalization capacity to handle situations other than those in the rule table. In order to combine this reactive behavior with the landmark tracking controller [12], we propose a fuzzy integration agent. The integration of these two behaviors is based on a fuzzy- grading scheme. Thus our indoor navigation system has the following functions : (1) The robot follows a preset path based on natural landmark? lamps on the ceiling; (2) it has the capacity to interact with unexpected stationary and moving obstacles on the path; (3) it returns to the desired path after avoiding obstacles.

This paper is organized as follows. Section 2 describes the develop of dangerous index representation method. The obstacle avoidance control scheme based on a fuzzy clustering network is developed. Section 3 explores the integration agency, which combines the landmark-tracking behavior and obstacle avoidance behavior. The effectiveness of the proposed design is demonstrated by practical experiments presented in Section 4. Section 5 concludes the paper.

2 Fuzzy Clustering Network for Real Time Obstacle Avoidance

The basic requirement for obstacle avoidance is to predict the possibility of colliding with any moving or stationary obstacle that appears on the planned path. Then the robot can possibly find a way to go around with them. It is well known that the distance to an obstacle is a useful information for designing the obstacle avoidance behavior of a mobile robot. However, when objects in the immediate environment are moving with certain speed toward the mobile robot, only considering the distance is not sufficient to guarantee safety. Consequently, the speed of moving obstacles should be estimated and taken into account for reactive motion control. We intuitively think that the detection of moving obstacles can be described by two important physical quantities, namely the relative velocities to obstacles and distance between the obstacles and the robot. The larger the relative velocity or the shorter the distance between the robot and obstacle is, the more dangerous obstacle will be. Consequently, the obstacle should be given a higher degree of dangerous index value. The way to define dangerous index to obstacles is described below.

On our mobile robot, a wide-angle CCD camera is employed for obstacle detection. As shown in Fig 1, the area covered by the vision sensor is divided into five groups, P1 ~ P5. Each group covers about 16° region. The relative velocity between obstacle and mobile robot is calculated for each region. Both stationary and moving obstacles are considered. Figure 2 depicts that the posture variation of the obstacle relative to the mobile robot is expressed as Δr ,($\Delta r = r_2 - r_1$) in the sample interval Δt . We can obtain the instantaneous velocity of the obstacles relative to mobile robot,$\Delta r / \Delta t$. This velocity can be decomposed into two components: V_T (perpendicular to the mobile robot's heading), and V_P (parallel to the mobile robot's heading). V_P is chosen as the relative velocity between an obstacle and the robot.

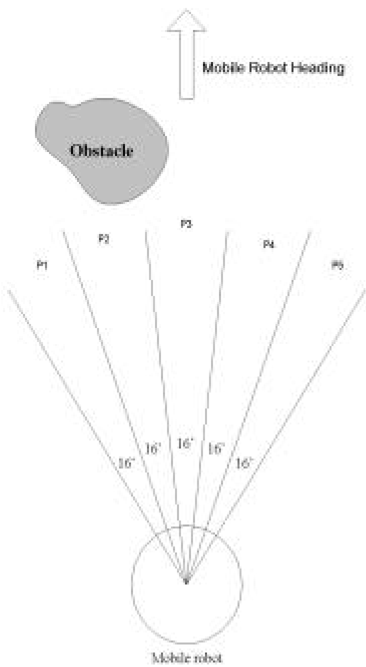

Fig.1. Obstacle detection using a wide-angle CCD camera

Exploiting the relative velocity, we define the dangerous index of the obstacle relative to the mobile robot. The dangerous index of a detected obstacle is given by:

$$DI = C_o \times \frac{V_P}{r} \quad (1)$$

where V_P is the relative velocity between the obstacle and the mobile robot , r is the distance between the obstacle and the mobile robot at the last sample instance and C_o is an adjusting coefficient in order to have dangerous indexes in the range [0,100].

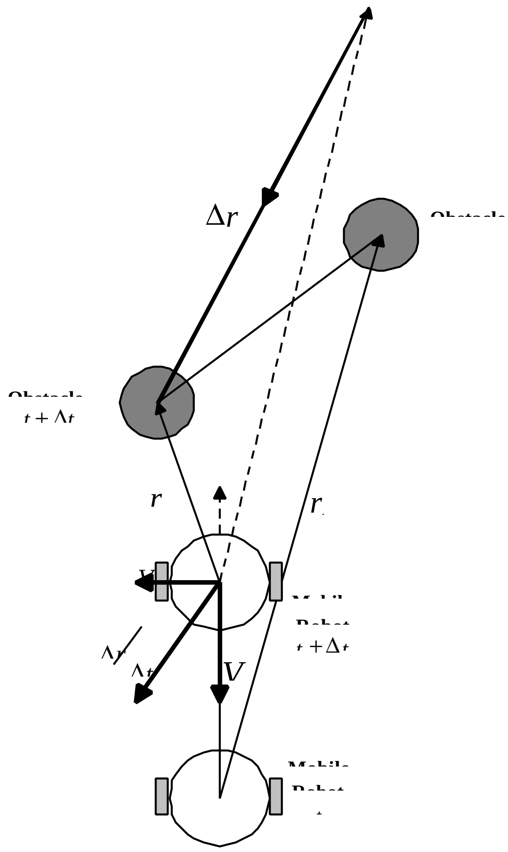

Fig.2. Relative velocity between an obstacle and the mobile robot

Now we consider the method to represent environment configurations for a mobile robot. This is practically a computer aided design procedure. By assigning various obstacle deployment in a simulation program, we recorded the dangerous indexes of detected obstacles in each group P1~P5 for each sample instant. In the simulation program, we considered two types of navigation environments: stationary and dynamic. All obstacles in stationary environment were assumed to be static. It was more complicated for dynamic environments where the obstacles can move. We considered the following configurations: single or multiple moving obstacles with low or high speeds. All the moving obstacles were modeled to move directly toward the robot. We then established a sample space representing almost every possible configuration that the mobile robot faced during an indoor navigation. The sample space is a 5-D space. Figure 3 illustrates one example of the 2-D sample space. Only P1 and P2 values are presented in this case, the other nine sets can be found in [13]. Altogether 3595 samples were generated. In order to find the most representative patterns in the established sample space, a competitive clustering network[14] was utilized to classify the prototype patterns.

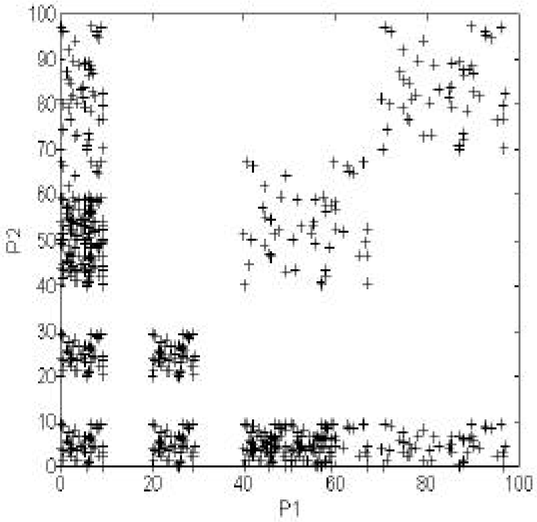

Fig. 3. The sample space of visual region of P1 and P2

The competitive learning algorithm employs a learn-only-if-it-wins principle to update a neuron value in each iteration. Figure 4 shows the competition learning results after 20000 epochs (it converges after 4000 epochs). Again only the weights of P1 and P2 visual region are presented. As expected, we obtained 120 clusters. However, since there were still apparent similarities among the clusters, we further condensed them into 61 clusters according to our common-sense judgement.

The idea of the proposed real-time obstacle avoidance is to associate different prototype patterns with various environment configurations. This is an extension of the idea of reactive navigation presented in [9]. We tried to map proper steering commands of the mobile robot according to these clustered prototype patterns. The structure of this obstacle avoidance method is shown in Fig. 5. It consists of four elements: target direction, rule table, fuzzy Kohonen clustering network (FKCN) and the competitive clustering network described in previous paragraph. The fuzzy-neural network controller combines a fuzzy Kohonen clustering network with a fuzzy rule table. In the upper part, the fuzzy rule table provides the reference velocity commands associated with prototype patterns. The lower part is adopted from an FKCN structure, which is responsible for calculating the similarities between the input environment pattern and the prototype patterns.

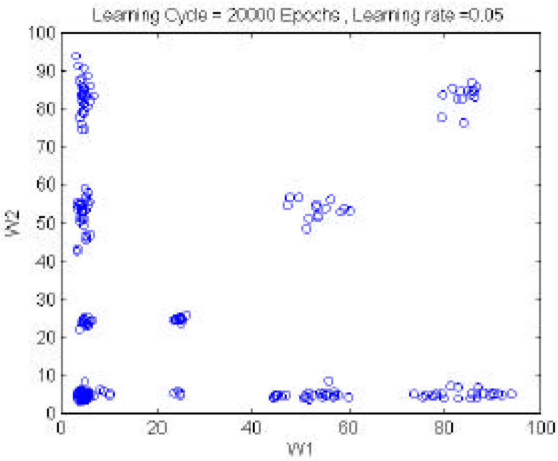

Fig. 4. The results of competition learning after 20000 epochs (only the weights for P1 and P2 visual regions are depicted)

The fuzzy Kohonen clustering network was adopted from the idea of unsupervised pattern recognition proposed by Huntsberger and Ajjimarangsee[15]. As shown in Fig. 5, it consists of three layers. The first layer is the input layer, to which the to-be-recognized environmental configuration is presented. The connection weights between the first and second layers represent the pre-assigned prototype patterns. The second layer is the distance layer. This layer is responsible for comparing an input pattern with the prototype patterns. The output d_{ij} of node j in the distance layer is calculated for the input pattern X_i presented to match the prototype pattern W_j:

$$d_{ij} = \|X_i - W_j\|^2 = (X_i - W_j)^T (X_i - W_j) \qquad (2)$$

where $j = 0 \sim C\text{-}1$, and C denotes the number of prototype patterns. The third layer is the membership layer. This layer is provided to map the distance d_{ij} to a membership value u_{ij}. The determination of the membership can be summarized by the following equations:

$$m_{ij} = \begin{cases} 1 & if \quad d_{ij} = 0 \\ 0 & if \quad d_{ik} = 0, (k \neq j, k \geq 0, j \leq c-1) \end{cases}$$

otherwise (3)

$$m_{ij} = \left(\sum_{l=0}^{c-1} \left(\frac{d_{ij}}{d_{il}} \right) \right)^{-1}.$$

It is easy to show that the sum of u_{ij} for $j=0$ to C-1 is 1. Since each prototype pattern is associated with a rule, the membership value represents the degree of activation of a rule. It is worth noting that unlike the original FKCN, this method does not use an unsupervised learning algorithm to determine the prototype patterns. Instead, a competitive clustering network was utilized off-line to choose the most representative prototypes, which will be recognized as the weights of the FKCN and assigned to the network. The prototype patterns, which were obtained from competitive learning, represent the typical indoor navigation configuration. Accordingly, sixty-one rules are designed associated with each prototype pattern. The rules are constructed to prevent from collision with obstacles by means of determining appropriate wheel velocities of the mobile robot. The rule tables are shown in Table1~5. Several useful navigation modes have been further categorized in these rules: stationary obstacles, single low-speed moving obstacle, single high-speed moving obstacle, multiple low-speed moving obstacles and multiple high-speed moving obstacles. Consequently, these rule tables can be applied in a modular manner, depending on the characteristics of to-be-encountered environment. When all sixty-one rules are applied, the robot can handle complex environments.

The target direction was also taken into account and divided into five levels. The equation for target division is as follows:

$$t = \begin{cases} 1 & for \quad 0^{\circ} \le f < 60^{\circ} \\ 2 & for \quad 60^{\circ} \le f < 120^{\circ} \\ 3 & for \quad 120^{\circ} \le f < 180^{\circ} \\ 4 & for \quad 180^{\circ} \le f < 270^{\circ} \\ 5 & for \quad 270^{\circ} \le f < 360^{\circ} \end{cases} \tag{4}$$

where f is the direction of the target with respect to the current heading of the robot.

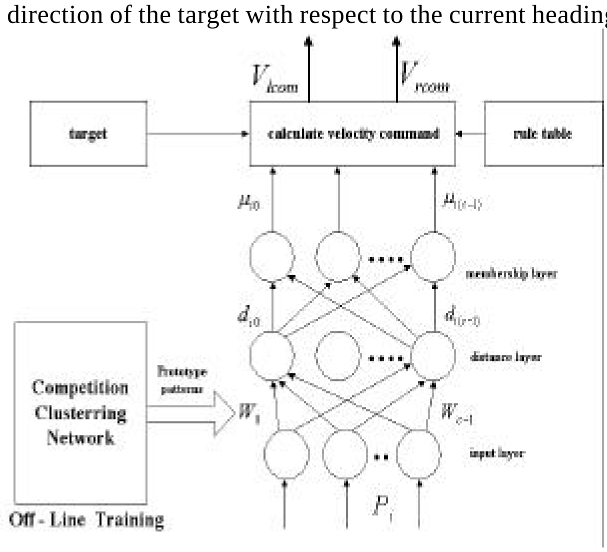

Fig.5. Fuzzy clustering network for obstacle avoidance

Table 1. Rule table for the situation of stationary obstacles

DANGEROUS INDEX PATTERN						T=1		T=2		T=3		T=4		T=5	
	P1	**P2**	**P3**	**P4**	**P5**	VL	VR	VL	VR	VL	VR	VL	VR	VL	VR
1	6.01	4.24	5.17	3.75	8.20	10	30	15	25	30	30	25	15	30	10

2	5.10	5.09	24.9	4.53	4.35	10	30	15	25	15	25	25	15	30	10
3	5.79	4.29	25.3	24.4	4.24	5	35	10	30	15	25	10	30	10	30
4	5.55	23.5	24.8	7.04	4.49	30	10	30	10	30	10	30	10	35	5
5	4.68	5.01	3.82	23.2	25.1	10	30	15	25	20	25	20	25	20	25
6	24.9	24.6	4.31	4.47	4.78	30	30	30	30	30	30	30	15	30	10
7	25.1	25.1	24.9	5.64	5.53	30	10	30	10	30	10	30	10	31	10
8	4.26	5.65	23.9	23.7	25.1	10	30	10	30	10	30	10	30	10	30
9	23.7	24.2	24.2	24.7	25.2	-10	10	-10	10	-10	10	-10	10	-10	10

Table 2. Rule table for the situation of single low-speed moving obstacles

DANGEROUS INDEX PATTERN						T=1		T=2		T=3		T=4		T=5	
	P1	**P2**	**P3**	**P4**	**P5**	VL	VR	VL	VR	VL	VR	VL	VR	VL	VR
10	4.90	4.84	5.60	5.29	47.5	15	30	15	30	20	30	15	30	15	3
11	4.16	4.71	3.98	53.7	5.42	10	30	10	30	10	30	10	30	10	3
12	5.03	4.92	54.2	4.87	4.67	10	30	15	30	10	30	30	15	30	1
13	6.14	46.9	4.59	5.33	5.50	30	10	30	10	30	10	30	10	30	1
14	55.5	4.86	4.58	5.69	5.17	30	15	30	15	30	20	30	15	30	1

Table 3. Rule table for the situation of single high-speed moving obstacle

DANGEROUS INDEX PATTERN						T=1		T=2		T=3		T=4		T=5	
	P1	**P2**	**P3**	**P4**	**P5**	VL	VR	VL	VR	VL	VR	VL	VR	VL	VR
15	5.09	4.17	4.19	4.62	86.2	20	70	30	70	30	50	30	70	30	70
16	5.06	4.86	4.90	89.9	5.04	20	70	30	70	20	50	20	70	20	70
17	4.88	4.50	85.5	4.87	4.12	20	70	30	70	30	70	70	30	70	20
18	3.14	93.9	3.20	3.18	4.96	70	30	70	20	70	20	70	20	70	20
19	77.8	4.32	3.74	4.32	4.37	70	30	70	35	70	30	70	20	70	20

When in use, the CCD camera provides the environment information. The dangerous index values are calculated for each direction and the input pattern is obtained. For each input pattern P_i, the membership values are computed using the fuzzy clustering network. These membership values are used for the activation of associated rules with respect to the prototype patterns. In the output stage, the output velocity command of the network can be calculated by the weighted sum of the membership values and the assigned velocities in each associated rule.

Table 4. Rule table for the situation of multiple low speed moving obstacles

DANGEROUS INDEX PATTERN						T=1		T=2		T=3		T=4		T=5	
	P1	**P2**	**P3**	**P4**	**P5**	VL	VR	VL	VR	VL	VR	VL	VR	VL	VR
20	60.3	53.1	6.54	5.09	4.55	20	10	30	10	30	10	30	10	30	10
21	4.61	52.9	53.4	5.58	5.39	30	5	30	10	30	10	30	10	30	10
22	5.84	5.06	43.1	53.3	55.1	5	30	10	30	10	30	10	30	10	30
23	10.1	4.83	5.03	57.2	55.6	5	30	10	30	15	30	10	30	10	30

24	51.3	4.13	53.3	6.72	4.21	30	10	30	10	30	15	30	10	30	5
25	55.2	4.69	5.61	58.4	8.87	10	20	10	20	10	20	10	20	10	20
26	56.8	4.08	5.09	5.09	55.9	20	30	15	30	30	30	30	15	30	20
27	5.67	57.7	4.88	56.4	5.47	10	30	10	30	10	30	30	10	35	10
28	5.13	55.1	4.01	5.32	59.5	30	20	30	20	30	20	30	20	30	20
29	5.22	5.17	47.9	6.69	58.7	5	35	10	30	10	30	10	30	10	30
30	47.3	54.6	51.5	5.66	5.17	30	5	30	10	30	10	30	5	30	5
31	53.6	52.0	4.25	53.3	4.99	30	10	30	10	30	10	30	10	30	10
32	47.7	56.7	4.13	3.91	55.0	30	20	30	20	30	20	30	15	30	15
33	55.5	4.35	53.3	57.3	5.29	10	30	10	30	10	30	10	30	10	30
34	57.2	4.44	4.50	56.0	56.5	20	30	20	30	20	30	20	30	20	30
35	5.30	57.0	54.0	55.1	5.07	10	30	10	30	10	30	30	10	30	10
36	4.05	55.1	51.5	4.41	51.1	5	30	5	30	5	30	5	30	5	30
37	5.20	56.2	4.45	55.3	55.9	5	30	10	30	10	30	10	30	10	30
38	5.28	4.31	56.9	57.4	56.9	5	30	10	30	10	30	10	30	10	30
39	58.4	52.8	55.7	55.7	4.85	30	10	30	10	30	10	30	5	30	5
40	4.92	52.6	56.6	56.4	53.5	5	30	10	30	10	30	10	30	10	30

A moving obstacle may cross the predefined path of the mobile robot instead of coming toward it. To avoid a crossing obstacle, the robot has to check the tangent component V_T (see Fig. 2). A threshold value (60 cm/sec.) is defined. If $|V_T| <$ threshold value, the obstacle will be recognized as moving towards the mobile robot, otherwise it will be recognized as a crossing obstacle. The robot will reduce its speed until the crossing obstacle passes.

3 Integration with Landmark Tracking

The mobile robot follows a predefined path utilizing the lights on the ceiling. We defined two parameters:?and X_d to represent the vehicle's position and heading relative to the lights(see Fig. 6). The parameter ? is defined as the angle between Y axis of the image plane and the line formed by two mass centers of consecutive lamps.

Table 5. Rule table for multiple high speed moving obstacles.

DANGEROUS INDEX PATTERN						T=1		T=2		T=3		T=4		T=5	
	P1	**P2**	**P3**	**P4**	**P5**	VL	VR	VL	VR	VL	VR	VL	VR	VL	VR
41	86.9	85.6	4.21	4.82	4.53	100	20	80	20	120	30	120	20	100	20
42	4.91	78.5	85.6	5.25	5.17	20	100	20	100	110	30	120	20	100	20
43	4.86	4.78	83.6	79.6	4.51	20	100	30	110	20	110	120	20	120	10
44	4.81	4.39	4.61	87.5	90.5	20	100	30	110	20	120	20	100	20	110
45	82.8	6.66	82.9	7.63	5.35	120	20	110	20	120	20	120	20	120	20
46	86.9	4.92	5.28	88.6	4.86	60	70	60	70	60	70	60	70	60	70
47	90.4	4.88	5.33	5.75	84.8	60	60	60	60	60	60	60	60	60	60

48	5.09	84.8	4.58	80.1	6.06	20 100	20 100	100 20	120 20	20 120
49	4.84	84.6	4.10	4.25	81.4	20 100	70 60	70 60	70 60	70 60
50	3.44	4.23	78.9	4.36	88.1	20 120	20 100	20 110	20 120	20 120
51	81.8	85.3	79.8	4.05	5.53	120 20	120 20	120 20	120 20	120 20
52	86.3	84.5	33.9	85.6	5.96	120 20	120 20	120 20	120 20	120 20
53	83.3	84.4	4.89	4.96	83.8	70 60	70 60	70 60	70 60	70 60
54	87.2	6.78	85.9	85.9	5.02	60 70	60 70	60 70	60 70	60 70
55	88.3	5.22	5.05	85.8	84.6	60 70	60 70	60 70	60 70	60 70
56	4.45	85.9	80.2	82.1	4.35	20 120	20 100	20 120	20 120	20 120
57	3.91	77.6	86.0	4.39	83.9	20 120	20 110	20 120	20 120	20 110
58	4.64	82.1	4.24	78.9	79.8	20 110	20 120	20 120	20 120	20 110
59	4.87	4.46	81.5	85.2	89.9	10 120	10 120	20 120	10 110	20 120
60	86.5	83.1	87.3	87.2	4.93	120 20	110 20	120 20	120 20	120 20
61	5.47	80.3	88.1	81.4	81.7	20 120	20 120	20 120	20 120	20 120

If the vehicle is on the left of the desired path, the angle will be less than 90 degrees. If the vehicle is on the right of the desired path, then the angle will be greater than 90 degrees. The parameter X_d is defined as the distance between center of the line formed by mass center of two consecutive lamp images and the image center. Thus, a negative X_d represents left-turn of the heading of the robot. These two parameters were employed to develop a lamp tracking controller[12]. The switching from the tracking mode to the obstacle avoidance mode and vice versa is important for smooth navigation performance. A fuzzy logic integration agent was developed for the switching between two modes. The fuzzy integration agent takes the sensory data and determines which mode the robot should be controlled. Figure 7 illustrates the complete navigation design. The inputs to this integration agent are the processed image data X_d and S. The former can be obtained from the CCD which covers the lamps on the ceiling and means the heading deviation from the lamps. It tells us whether the CCD keeps covering the lamps. The second input S can be obtained from the other wide-angle CCD camera, which detects the obstacles in the environment. We can calculate the dangerous indexes of obstacles in each visual region, P1~P5. We choose the average value of the dangerous indexes in P2, P3 and P4, i.e. S = (P2+P3+P4)/3. It tells us whether an obstacle will appear in front of the vehicle. The

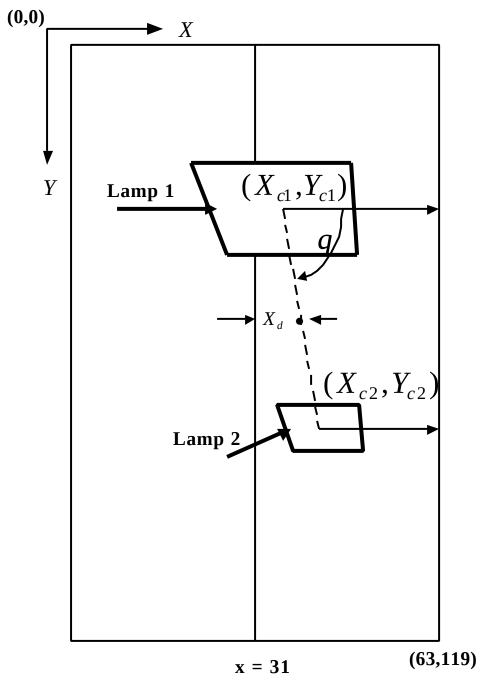

Fig. 6. Illustration of image data for lamp following

outputs of the integration agent are the grades of obstacle avoidance behavior and lamp-following behavior, respectively. The higher grade takes the control.

The basic principle of grading is described as follows. If the robot detects that there is an obstacle very near to it on the preset path, then the grade of obstacle avoidance behavior should be higher than that of lamp tracking behavior. If the robot doesn't detect any obstacle on the preset path and CCD covers the lamp, the grade of obstacle avoidance behavior should be lower than that of lamp tracking. If the vehicle detects that there is an obstacle on the preset path and CCD loses the lamp, the grade of obstacle avoidance behavior should be higher than that of lamp tracking. Table 6 shows the grading rules for the obstacle avoidance behavior.

Before switching to the obstacle avoidance behavior, a virtual target is set for the obstacle avoidance behavior. Such a virtual target is generated automatically during the program execution. It is used to guide the mobile robot in our design of obstacle avoidance behavior. As illustrated in Fig. 8, the virtual target must be on the desired path so that the robot can return to the preset path after avoiding the obstacle. The robot will travel toward the virtual target in the obstacle avoidance behavior and this helps the CCD to cover lamps on the ceiling as long as it is possible. Consequently, the virtual target is not fixed; it moves along the desired path in accordance with the

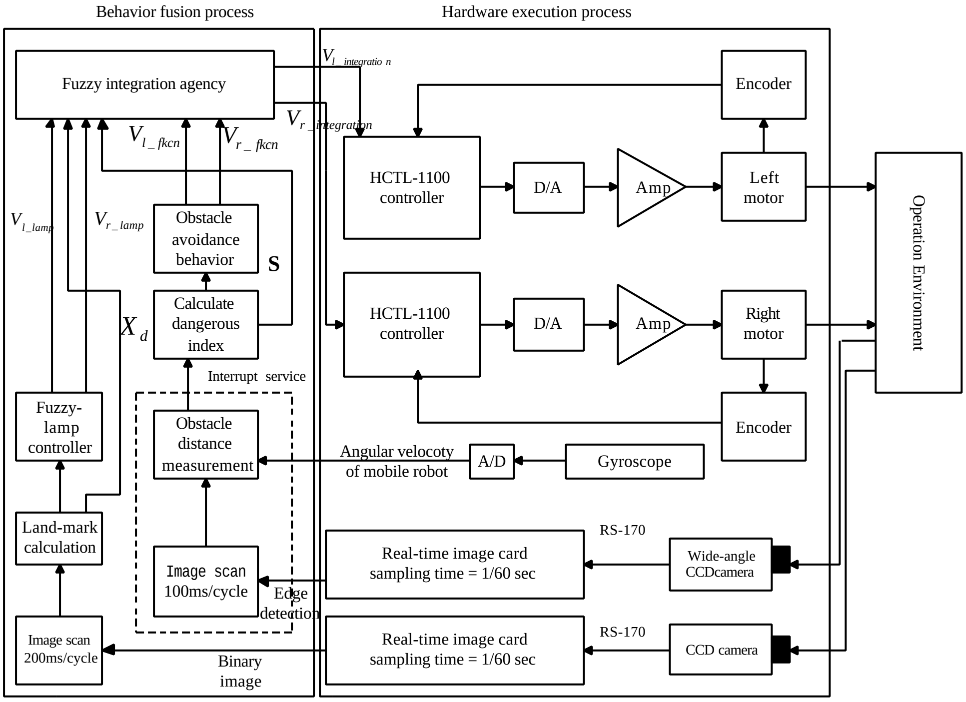

Fig.7. System block diagram of the navigation system

Table 6. Grading rules for obstacle avoidance behavior

Grade 1		X_d						
		NB	NM	NL	ZR	PL	PM	PB
s	Very Near	*High*	*High*	*High*	*High*	*High*	*High*	*High*
	Near	*High*	*High*	*High*	*High*	*High*	*High*	*High*
	Middle	*High*	*Medium*	*Low*	*Low*	*Low*	*Medium*	*High*
	Far	*High*	*Medium*	*Low*	*Low*	*Low*	*Medium*	*High*
	Very Far	*High*	*Medium*	*Low*	*Low*	*Low*	*Medium*	*High*

relation between the virtual target and the current location of the robot. The basic principle for moving the virtual target is to let it run ahead of the mobile robot.

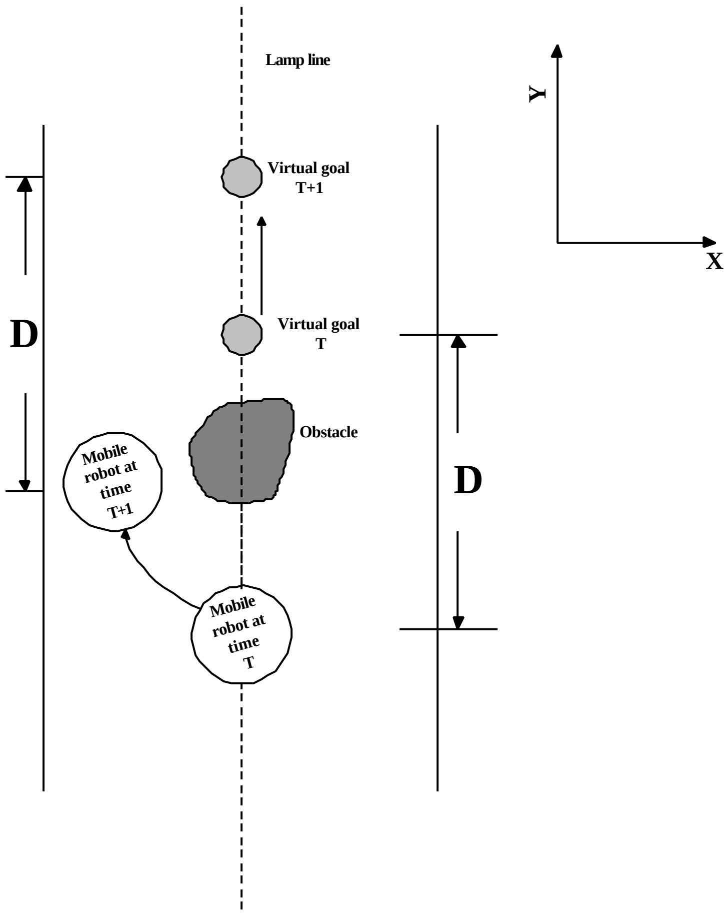

Fig.8. Setup of the virtual targets

4 Experimental Results

Our experimental mobile robot is of circular shape with a diameter of about 60cm. It has two independent drive wheels and two free casters for balance. Fig. 9 shows the robot in the experiment. A Gyrostar ENV-05DB-52 gyroscope was mounted on the vehicle. The absolute angular information from gyroscope was fused with the measurement from shaft-encoders by means of Kalman filtering to estimate the position and orientation of the robot. Two CCD cameras were provided on this mobile robot. One was installed at a height of 165cm with a 16mm lens and was used for lamp-detection and tracking control. The CCD camera tilts up and covers two lamps in one picture at a time in order to calculate the information of the landmark. Another CCD camera was installed at a height of 123cm with a 3.7mm wide-angle lens. This CCD camera tilts downward and covers a visual region of 12 ~ 400cm ahead with angular span of 80° to detect unexpected obstacles. The self-developed image cards (Fig. 10) were employed for real-time image processing of these cameras. The control tasks are distributed among two HCTL-1100 motion controllers and an IPC (Intel Pentium-133) put on board the robot.

Fig.9. Experimental mobile robot

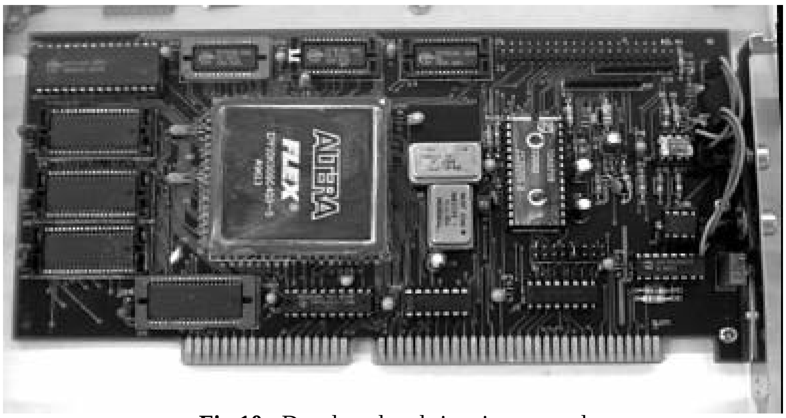

Fig.10. Developed real-time image card

Navigation experiments were carried out to verify the effectiveness of the proposed reactive control system. Figure 11 illustrates an experimental result recorded in our laboratory building. The robot could deal with moving and static obstacles in an unstructured environment. In the figure the robot's positions were depicted for the time base of each second. In the experiment, the obstacles were a chair and two walking persons. One walking person was moving toward the vehicle and the other was crossing the predefined path as a lateral moving obstacle. In the beginning, the robot followed the predefined path employing the lamp-tracking controller. Before encountering the first moving obstacle Mov1, the robot approached the preset path as expected. At the 8th second, it detected a walking person and switched to obstacle avoidance behavior. Just after it passed this person, it switched back to lamp-tracking behavior and followed the preset path. The robot turned to another long corridor when the passed lamps reached a preset value to initiate a turning operation. Since the robot would not be exactly under the lamps after making the turn, it tried to approach the

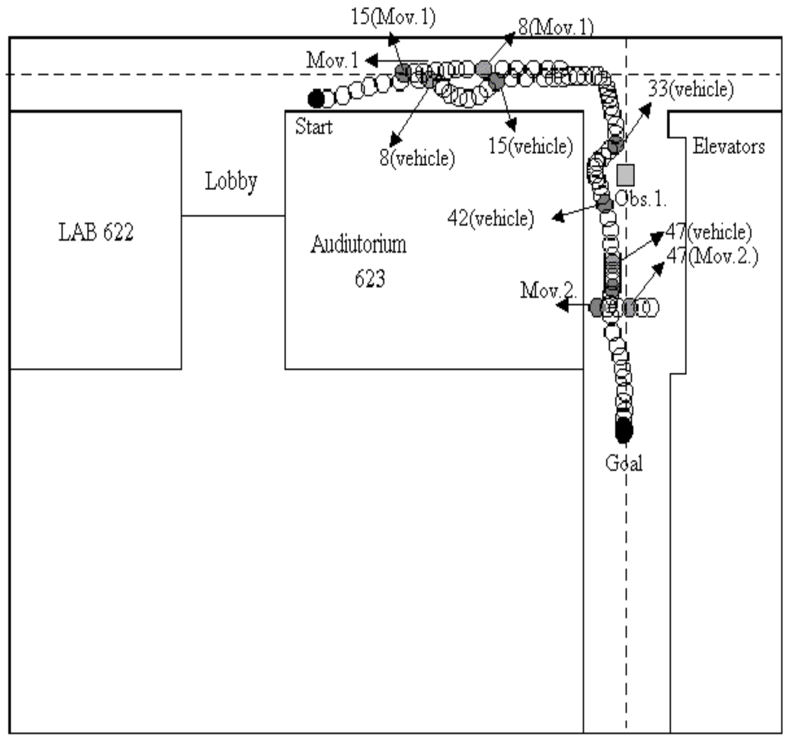

Fig 11. Experimental results of navigation in a corridor

predefined path by means of the lamp-tracking controller. At the 33rd second, the robot detected a chair and switched to obstacle avoidance behavior again. After the robot passed through the chair, it switched back to lamp-tracking behavior and approached the preset path. A walking person, Mov2, appeared in front of the robot at the 47th second. Since the tangent relative velocity V_T was greater than the default threshold value (60cm/sec), the moving obstacle would be recognized as a crossing obstacle. The robot employed a crossing-obstacle-avoidance operation and reduced its velocity until the obstacle passed completely. After Mov2 passed, the robot switched to lamp-tracking behavior and finally achieved its goal position.

5 Conclusions

In this paper, a novel robot control system has been developed using a fuzzy clustering network. We successfully merged pattern recognition technique into robot control applications. Because this method features a direct mapping from sensory data to robot action, reactive behavior is much easier to realize. This design enables the robot to handle both moving and static obstacles. Navigation experiments show that the vision-based dangerous index patterns are sufficient to represent the environment configuration. In the future, the extension of this scheme to include intelligent pattern recognition techniques will be further investigated.

6 Acknowledgement

This work was supported by the National Science Council of the Republic of China under grant NSC-87-2213-E009-140.

References

1. Fujimura, K. and Samet, H.: A Hierarchical Strategy for Path Planning Among Moving Obstacles, IEEE Trans. on Robotics and Automation, Vol. 5, No. 1 (1989) 61-69
2. Sharma, R.: Safe Motion Planning for a Robot in a Dynamic Environment, in Proc. of IEEE Int. Conf. on Robotics and Automation, Sacramento, CA (1991) 438-443
3. Beom, H. R. and Cho, H. S.: Path Planning for Mobile Robot Using Skeleton of Free Space, in Proc. IFAC Symp. Information Control Problems for Manufacturing, Toronto, Canada, May (1992) 509-513
4. Fiorini, P. and Shiller, Z.: Motion Planning in Dynamic Environments Using Velocity Obstacles, The International Journal of Robotics Research, Vol. 17, No. 7 (1998) 760-772
5. Weng, J. and Chen, S.:Vision-Guided Navigation using SHOSLIF, Neural Networks, Vol 11 (1998)1511-1529
6. Khatib, O.: Real-time Obstacle Avoidance for Manipulator and Mobile Robots, Int. J. of Robotics Research, Vol. 5, No. 1 (1986) 90-98
7. Borenstein, J and Koren, Y.: Real-time Obstacle Avoidance for Fast Mobile Robot, IEEE Trans. Syst. Man Cyber., Vol. 19, No. 5 (1989)1179 – 1187
8. Ishikawa, S.: A Method of Indoor Mobile Robot Navigation by Fuzzy Control," in Proc. IEEE Int. Conf. Intelligent Robots and Systems, Osaka, Japan, Nov. (1991) 1013-1018
9. Song, K. T. and Sheen, L. H.: A Neuro-fuzzy Control Design for Obstacle Avoidance of A Mobile Robot, Proc. FUZZ-IEEE/IFES'95, Yokohama, (1995) 71-76

10. Nair, D.and Aggarwal, J. K.: Moving Obstacle Detection From a Navigation Robot, IEEE Trans. on Robotics and Automat. , Vol. 14, No. 3 (1998) 404-416
11. Chang, C. C. and Song, K. T.: Environment Prediction for a Mobile Robot in a Dynamic Environment, IEEE Trans. on Robotics and Automat., Vol.13 , No.6 (1997) 862-872
12. Song, K. T. and Lin, Y. S.:Landmark-based Guidance of an Autonomous Vehicle Using Fuzzy Logic Velocity Control and Real-time Computer Vision, Journal of the Chinese Fuzzy Systems Association, Vol. 5, No. 1(1999) 43-51
13. Chen, Y. H.: Indoor Navigation System of a Mobile Robot, Master Thesis, NCTU, Taiwan (1998)
14. Looney, C. G.: Pattern Recognition Using Neural Network ? Theory and Algorithms for Engineers and Scientists, Oxford University Press, Inc. (1997)
15. Huntsberger, T. and Ajjimarangsee, P. :Parallel Self-organizing Feature Maps for Unsupervised Pattern Recognition, Int'l. J. General Systems, Vol. 16, No.4 (1990) 357-372