[4] G. J. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic, Theory and Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1995.

[5] T. Kohonen, *Self-Organization and Associative Memory*. Berlin, Germany: Springer-Verlag, 1988.

[6] C. T. Lin and C. S. G. Lee, "Neural-network-based fuzzy logic control and decision system," *IEEE Trans. Comput.*, vol. 40, pp. 1320–1336, Dec. 1991.

[7] R. L. Mantaras, *Approximate Reasoning Models*. New York: Ellis Horwood, 1990.

[8] D. Nauck and R. Kruse, "A neuro-fuzzy method to learn fuzzy classification rules from data," *Fuzzy Sets Syst.*, vol. 89, no. 3, pp. 277–288, 1997.

[9] C. Quek and K. K. Ang, "A pseudo outer-product based fuzzy neural network using the compositional rule of inference and singleton fuzzifier," *IEEE Trans. Syst., Man, Cybern.*, submitted for publication.

[10] C. Quek and K .B. Tan, "Pseudo-outer product based fuzzy neural network fingerprint verification system," *Neural Networks*, submitted for publication.

[11] C. Quek and R. W. Zhou, "POPFNN-AARS(Non-S): A nonsingleton input pseudo outer product based fuzzy neural network using the approximate analogical reasoning schema," unpublished.

[12] H. Takagi and I. Hayashi, "Artificial-neural-network-driven fuzzy reasoning," *Int. J. Approx. Reas.*, 1989.

[13] I. B. Turksen and Z. Zhong, "An approximate analogical reasoning schema based on similarity measures and interval-valued fuzzy sets," *Fuzzy Sets Syst.*, vol. 34, pp. 323–346, 1990.

[14] H. Nakanishi, I. B. Turksen, and M. Sugeno, "A review and comparison of six reasoning methods," *Fuzzy Sets Syst.*, vol. 57, pp. 257–294, 1993.

[15] P. J. Werbos, "Backpropagation: Past and future," in *Proc. 2nd Int. Conf. Neural Networks*, 1988.

[16] ——, "Neurocontrol and fuzzy logic: Connections and designs," *Int. J. Approx. Reas.*, vol. 6, pp. 185–219, 1992.

[17] R. R. Yager, "Modeling and formulating fuzzy knowledge bases using neural network," *Neural Networks*, vol. 7, no. 8, pp. 1273–1283, 1994.

[18] L. A. Zadeh, "Outline of a new approach to the analysis of complex systems and decision processes," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-3, no. 1, pp. 28–44, 1973.

[19] ——, "The concept of a linguistic variable and its applications to approximate reasoning I, II, III," *Inf. Sci.*, vol. 8, pp. 199–249; pp. 301–357, 1975; and vol. 9, pp. 43–80, 1975.

[20] R. W. Zhou and C. Quek, "POPFNN: A pseudo outer-product based fuzzy neural network," *Neural Networks*, vol. 9, no. 9, pp. 1569–1581, 1996.

[21] ——, "A pseudo outer-product based fuzzy neural network and its rule-identification algorithm," in *Proc. ICNN'96*, June 1996, vol. 2, pp. 1156–1161.

[22] ——, "An automatic fuzzy neural network driven signature verification system," in *Proc. ICNN'96*, June 1996, vol. 2, pp. 1034–1039.

[23] R. Zwick, E. Carlstein, and D. V. Budescu, "Measures of similarity between fuzzy concepts: A comparative analysis," *Int. J. Approx. Reas.*, vol. 1, pp. 221–242, 1987.

Reactive Navigation in Dynamic Environment Using a Multisensor Predictor

Kai-Tai Song and Charles C. Chang

***Abstract*— A reactive navigation system for an autonomous mobile robot in unstructured dynamic environments is presented. The motion of moving obstacles is estimated for robot motion planning and obstacle avoidance. A multisensor-based obstacle predictor is utilized to obtain obstacle-motion information. Sensory data from a CCD camera and multiple ultrasonic rangefinders are combined to predict obstacle positions at next sampling instant. A neural network, which is trained off-line, provides the desired prediction on-line in real time. The predicted obstacle configuration is employed by the proposed virtual-force-based navigation method to prevent collision with moving obstacles. Simulation results are presented to verify the effectiveness of the proposed navigation system in an environment with multiple mobile robots or moving objects. This system was implemented and tested on an experimental mobile robot at our laboratory. Navigation results in real environment are presented and analyzed.**

I. Introduction

Although many approaches to motion planning for mobile robots in uncertain static environment have been proposed in the literature [16], [21], they do not effectively resolve navigation problems involving multiple moving obstacles in real-world applications. Providing information of all obstacle trajectories in advance, if at all possible, would allow us to plan a safe trajectory (i.e., a path and robot arrival times at every intermediate position) for the robot before it moves. One approach involves planning an initial path, then constructing trajectories or velocities by considering the trajectories of obstacles [8], [12], [19]. Another approach considers time as an additional dimension and plans for space–time [6], [9], [22]. Recently, Fiorini and Shiller [7] employed velocity information to determine potential collisions and computed the trajectories of a robot moving in a time-varying environment. A drawback to such approaches is that they are only applicable to situations where the *a priori* knowledge about obstacle motion is available. Due to the limitations in the sensor system, it is difficult to obtain the velocity information of multiple objects onboard the mobile robot in a dynamic environment.

Moreover, if unexpected events frequently occur, then complete trajectory replanning used repeatedly are inappropriate owing to real-time considerations. Under such circumstances, robots require reactive navigational schemes. The robot should constantly monitor its environment with onboard sensors and act appropriately when deemed necessary. Several investigators have presented reactive motion-planning schemes to cope with moving obstacles. Some methods of this type alter the robot's speed along the preplanned path to avoid unexpected moving obstacles [10], [14]. More flexible methods can modify both the robot's speed and direction of motion [15], [18], [25], [26]. While there is an increasing interest in solving dynamic motion-planning problems, most of the results are presented in computer simulations, relative fewer of them have been actually

Manuscript received December 20, 1997; revised November 8, 1998 and April 24, 1999. This work was supported by the National Science Council of the R.O.C. under Grant NSC-86-2213-E009-046. This paper was recommended by Associate Editor K. Pattipati.

K.-T. Song is with the Department of Electrical and Control Engineering, National Chiao Tung University, Hsinchu 300, Taiwan, R.O.C.

C. C. Chang is with the Opto-Electronics Laboratories, Industrial Technology Research Institute, Chutung 310, Taiwan, R.O.C.

Publisher Item Identifier S 1083-4419(99)06991-5.

implemented on a real mobile robot. We believe this is mainly due to the difficulties encountered in the sensing and perceptual systems.

To resolve the sensing problem, Nair and Aggarwal proposed a vision system that can detect unexpected moving obstacles [17]. The system was able to detect moving obstacles 100 ms/frame using an HP-735 workstation. But the use of such information in obstacle avoidance was not investigated. Chang and Song examined the feasibility of extracting simple implicit motion information from multiple ultrasonic rangefinders [4]. This information is equivalent to the next-time relative positions of obstacles' nearest points to the robot. Using such projected information, they proposed a virtual force approach for a mobile robot navigating among moving obstacles [5]. However, the frequent used ultrasonic sensors cannot provide the comprehensive obstacle-motion information required for the navigation method. Other types of sensors, e.g., CCD cameras, provide substantially more information. These two types of sensors can compensate each other [24]. Through proper arrangement, a combination of them provides improved perceptual capacity. In this paper, sensor data from a CCD camera and 16 ultrasonic sensors are fused into an uniform representation to make more accurate environmental prediction. A multilayer feedforward network, which is trained off-line, provides the desired prediction on-line in real time. The predicted obstacle configuration is employed by the proposed virtual-force-based reactive navigation algorithm to prevent collision with multiple moving obstacles.

In conventional potential field based methods [1], [13], the virtual force is from the current obstacle positions. These methods are mainly employed to deal with stationary obstacles. They simply regarded the direction of the resultant force as the moving direction of the robot. The traveling speed of the robot is not affected. This however, contradicts the natural reaction to an applied force. It will not be sufficient to handle fast moving obstacles. In this paper, the virtual forces come from the predicted obstacle positions. Consequently, it is a precautionary approach to a more reactive motion planning. Moveover, the resultant force causes a motion change that follows Newton's law of motion, both the speed and moving direction of the robot are simultaneously adjusted. This navigation scheme can therefore handle moving as well as stationary obstacles much more efficiently. However, it also needs more attention in designing forces to achieve the desired obstacle avoidance behavior in dynamic environments.

The rest of the paper is organized as follows. Section II describes the obstacle-configuration predictor that fuses data from a CCD camera and multiple ultrasonic sensors. A novel navigation scheme based on virtual force concept is presented in Section III. Section IV illustrates a navigation design for a mobile robot among multiple moving as well as stationary obstacles. Section V presents the simulation results which indicate intelligent human-like avoidance behavior in dynamic environments. In Section VI, the experimental mobile robot is described and practical implementation results of the proposed method are presented. Conclusions and areas for future research are presented in Section VII.

II. Obstacle Prediction Using Multiple Sensors

In this work, a multisensor system is developed to inform the robot what area will be occupied by obstacles. This is an implicit form of considering the motion of obstacles. The obstacle-configuration prediction is possible if one succeed in fusing multisensor data to obtain the correlation among them. One possibility is to predict future sensor readings. The measurement from any given sensor must have a relationship to historical measurements from sensors nearby if the motions of the robot and obstacles follow certain fixed patterns when sensor data are taken. If one can find such relationship, the sensor's future measurement can be predicted. Knowing the future measurements is equivalent to knowing what area will be occupied by obstacles. It is helpful for navigation among moving obstacles because the robot can respond to anticipated changes in the environment.

Such an environment predictor has been implemented employing 16 Polaroid ultrasonic ranging sensors [4]. Ultrasonic transducers are simple and very effective to obtain the distance to an obstacle. However, they have two major drawbacks. First, the specular reflection may prevent the robot from seeing obstacles, subsequently causing unexpected collisions. Second, the wide beams of ultrasonic sensors cause poor angular resolution, making it impossible for the robot to pass through closely cluttered environments. Therefore, another type of sensor(s) for compensating ultrasonic sensors is highly desired. Herein, we examine the feasibility of using a CCD camera to enhance the onboard perceptual capacity of the mobile robot. The basic idea is to combine the camera data with ultrasonic sensor data to obtain more reliable environmental information.

A. CCD Camera Sensing

In our system, a CCD camera is exploited to provide perceptual information to the robot. Although the image from a CCD camera contains much useful environmental information, extracting and interpreting this information is a complex task. Utilizing two cameras to measure distances requires image matching and a higher computational cost. Moreover, it is considered that the robot must react responsively to the moving obstacles. The real-time performance is important. Consequently, a single camera was used to achieve this task. We assumed that all obstacles stand on the ground and each pixel is mapped to a ground distance (ground-plan constraint [11]). Finding the lowest edge is equivalent to finding the nearest obstacle.

1) Calibration: The vision system is initially calibrated for distance measurement. The CCD camera should be installed at a fixed position with a fixed orientation to cover the most important areas. Apparently, the camera image positions have a defined, continuous relationship with their actual ground positions. In this work, the polar coordinate system is adopted to represent ground positions owing to the consideration of fusion with ultrasonic sensor data. Therefore, the image-ground relationship can be represented as

$$\rho_g = f_1(x_i,\, y_i) \tag{1}$$

$$\theta_g = f_2(x_i,\, y_i) \tag{2}$$

where

- ρ_g distance from the ultrasonic sensor ring to the measured point;
- θ_g angle between the robot's heading and the measured point;
- x_i image position in the x-axis;
- y_i image position in the y-axis.

Each of these two equations contains a camera distortion part and many trigonometric and inverse trigonometric functions used in geometrical transformations. It will take much calculation time for these trigonometric functions. Moreover, because there is an uncertain part due to the camera distortion, to derive a precise equation for other parts becomes less meaningful. Consequently, a direct calibration measurement was carried out to find the relationship between image positions and ground positions. We express (1) and (2) as polynomials so that the regression model can be adopted. First, taking Taylor expansion for (1) at $x_i = 0$, we obtain

$$\rho_g = g_0(y_i) + c_1 x_i g_1(y_i) + c_2 x_i^2 g_2(y_i) + c_3 x_i^3 g_3(y_i) + \text{higher order terms}. \tag{3}$$

TABLE I
COEFFICIENTS OF THE REGRESSION MODEL FOR ρ_g

a_1	a_2	a_3	a_4	a_5
9.376e-9	-6.139e-9	4.456e-10	-5.983e-8	-7.188e-6
a_6	a_7	a_8	a_9	a_{10}
1.756e-5	9.111e-7	-1.417e-3	-5.304e-4	2.243

TABLE II
COEFFICIENTS OF THE REGRESSION MODEL FOR θ_g

b_1	b_2	b_3	b_4	b_5
1.908e-8	-3.750e-5	-7.615e-6	1.245e-6	2.468e-3
b_6	b_7	b_8	b_9	b_{10}
2.151e-2	3.408e-3	-1.216	-5.006	5.434e2

Then, taking Taylor expansion for $g_0(y_i)$, $g_1(y_i)$, $g_2(y_i)$, etc., in (3) at $y_i = 0$, we have

$$\begin{aligned}\rho_g =& (d_{00} + d_{01}y_i + d_{02}y_i^2 + d_{03}y_i^3 + \cdots)\\ &+ c_1x_i(d_{10} + d_{11}y_i + d_{12}y_i^2 + d_{13}y_i^3 + \cdots)\\ &+ c_2x_i^2(d_{20} + d_{21}y_i + d_{22}y_i^2 + d_{23}y_i^3 + \cdots)\\ &+ c_3x_i^3(d_{30} + d_{31}y_i + d_{32}y_i^2 + d_{33}y_i^3 + \cdots)\\ &+ \text{higher order terms.}\end{aligned} \tag{4}$$

Expanding the above equation and choosing a finite number of terms to approximate it, we obtain a polynomial expression for ρ_g. Then, we can use the regression method to find the coefficients. For this purpose, we record 110 sets of data that relate ground positions to the corresponding pixel positions in an image frame. According to our test results, taking the 10 most dominant terms, in which the orders of $x_iy_i \leq 3$ can achieve reasonable precision. Therefore, the regression model can be expressed as

$$\begin{aligned}\rho_g =& a_1 \cdot x_i^3 + a_2 \cdot y_i^3 + a_3 \cdot x_i^2y_i + a_4 \cdot x_iy_i^2 + a_5 \cdot x_i^2\\ &+ a_6 \cdot y_i^2 + a_7 \cdot x_iy_i + a_8 \cdot x_i + a_9 \cdot y_i + a_{10}\end{aligned} \tag{5}$$

where a_i's are the coefficients. Similarly, the regression model for θ_g can be expressed as

$$\begin{aligned}\theta_g =& b_1 \cdot x_i^3 + b_2 \cdot y_i^3 + b_3 \cdot x_i^2y_i + b_4 \cdot x_iy_i^2 + b_5 \cdot x_i^2\\ &+ b_6 \cdot y_i^2 + b_7 \cdot x_iy_i + b_8 \cdot x_i + b_9 \cdot y_i + b_{10}\end{aligned} \tag{6}$$

where b_i's are the coefficients. The calibrated coefficients are presented in Tables I and II, respectively.

This method achieves the accuracy of 5.23 cm average distance error (maximum error $<$ 10 cm) and 1.32° average angular error (maximum error $<$ 2.75°).

2) Real-Time Vision System: Our mobile robot is equipped with a real-time image processing board developed in this laboratory [23]. We take advantage of the onboard frame-rate edge detection capacity (60 fields/s) to identify obstacles presented in an image. However, two problems should be resolved to find obstacles. First, owing to the noise in source images, what kind of edges can be regarded as obstacles must be determined. Second, the floor of our laboratory building is covered by 30 × 30 cm tiles; edges may be extracted at the gaps between tiles when lights are reflected from them. These false edges (that are not from obstacles) must be eliminated. To resolve the noise problem, we regard a group of edges as an obstacle if it is constructed from at least three connected edges in terms of 8-connectivity. Our results demonstrate that the false edge problem is attributed not to the color difference between tiles and gaps but, instead, their reflectivity; the false edges usually occur when the tiles are very bright due to light reflection. Under such circumstance, although the gaps are darker than tiles, they also have relative high gray levels. Therefore, brightening the complete image can effectively eliminate this type of false edge. The algorithm is summarized in the following equation:

$$p_{i,\text{new}} = \begin{cases} p_i + q & \text{if } p_i < q \\ 255 & \text{if } p_i \geq q \end{cases} \tag{7}$$

where

$p_{i,\text{new}}$ new gray level of the ith pixel;
p_i original gray level of the ith pixel;
q integer between 0 and 255.

Notably, this algorithm is realized on the image processing board in hardware, hence it does not require extra processing time.

Fig. 1 presents a sample test result of the real-time image system using the proposed image-processing algorithm. Fig. 1(a) depicts the original image taken by the camera. Fig. 1(b) displays the detected edges (dark points), which are extracted from a brightened image ($q = 85$). The white lines in the figure are the nearest edges found using the proposed method. We note that the obstacles are detected and their distances to the mobile robot are obtained. The numbers indicated below the image, from left to right, are the estimated distance in directions from $-60°$ to 60°. Each covers 7.5°.

B. Sensor Data Fusion

The purpose of sensor data fusion in this study is mainly to obtain an uniform representation of multiple ultrasonic sensors and the CCD camera. Herein, we process the image data such that they have the same representation as ultrasonic sensor's, that is, a distance to the nearest obstacle in each observed direction. The following conditions are important for this design:

1) the camera may observe something that ultrasonic sensors cannot detect, and vice versa;
2) the camera-measured distances have a better directional resolution than ultrasonic sensors;
3) the camera-measured distance may be uncertain due to unclear edges.

A wide-angle CCD camera is installed on the mobile robot. The present design adopts three camera-measured distances within a single ultrasonic sensor beam angle. The fusion results are made in each camera-measured direction. Consequently, they have the same directional resolution as the camera-measured distances. Fig. 2 illustrates the sensor arrangement and sensing directions. We see from the figure that 21 camera-measured distances are taken over the angular span covered by seven ultrasonic sensors. The fusion principle adopts the shortest distance that is reliable (using prediction as a criterion) as the fused result.

C. Multisensor Neural Network Predictor

A multilayer feedforward network was adopted to accomplish obstacle configuration prediction. Fig. 3 depicts the neural network structure. In each direction, the distance data from neighboring directions and their previous readings are employed to predict the next-time distance reading. In the figure, $s_{fi}(t)$ represents the fused sensor data for the measurement direction i at sample instant t and $s_{ui}(t)$ represents the reading from ultrasonic sensor i at sample instant t. The inputs to the predictor are the historical measurements

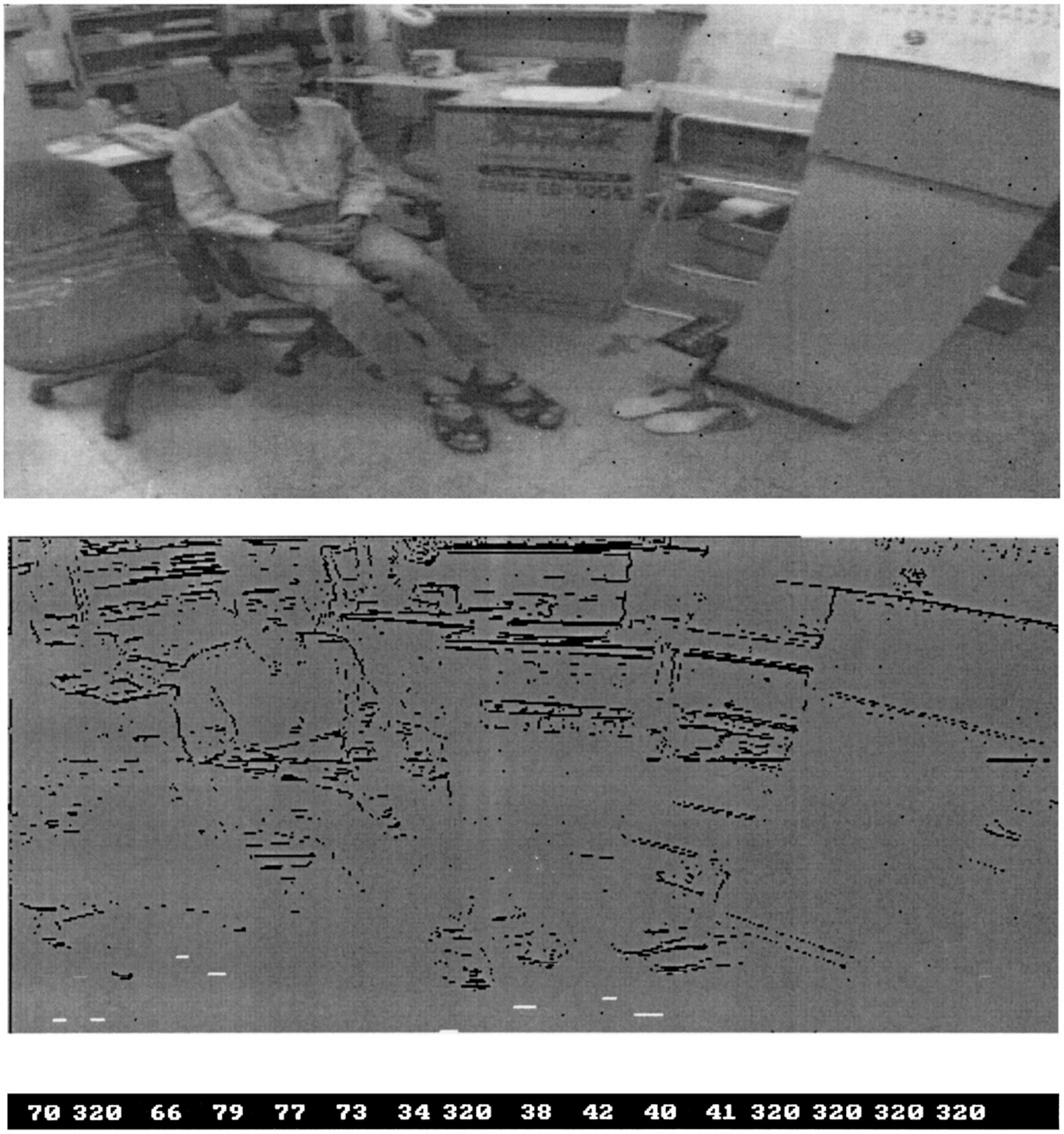

Fig. 1. Experimental results of identifying obstacles and their distances.

of sensors near the sensor whose reading in the next time instant is being predicted. Before the measurement data enter the neural net module, they are normalized to a value between 0 and 1. The larger the measured distance, the closer the value to 1. The outputs of the neural network are the normalized sensor data prediction, which is also a value between 0 and 1. To convert them into the predicted sensor data, the denormalization module multiplies them by the maximum sensor range. The number of neighboring sensors considered in the predictor depends on the velocities of the robot and obstacles. It also depends on the sensor arrangement of the robot. Considering moderate velocities of the robot and moving obstacles, the measurements within the central 135° are sampled to predict the obstacle motion. Within the camera covered area, the fused data are used; outside that area, ultrasonic sensor data alone are used. Therefore, the fused data in direction 1–21 plus the data from ultrasonic sensors 1 and 9 are taken as the inputs to the neural network. Since the previous sensor data as well as the present data are used in the obstacle configuration prediction, altogether there are 46 (23 × 2) inputs to the neural network. The fact that the camera has full-distance measurements within only about the central 45° accounts for why the predictions of fused data are made at these areas (other areas still use ultrasonic sensors alone for prediction). Therefore, the distances in the central nine camera-measured directions (nos. 7–15) are predicted by this neural network. The hidden layers of the neural network have forty-five and thirty processing elements, respectively. The structure of the neural net is thus 46-45-30-9.

The relative-error-backpropagation (REBP) [4] was adopted as the training algorithm for the artificial neural network (ANN). This algorithm is to minimize the error function E defined by

$$E \triangleq \frac{1}{2}\sum_{j}\left[(T_{N,j} - A_{N,j})/T_{N,j}\right]^2 \tag{8}$$

where $A_{N,j}$ and $T_{N,j}$ are the actual and desired outputs of the jth neuron of the output layer. One can obtain the minimal relative error by minimizing E. Consequently, the absolute prediction error for distant obstacles can be allowed a larger value, while that of close

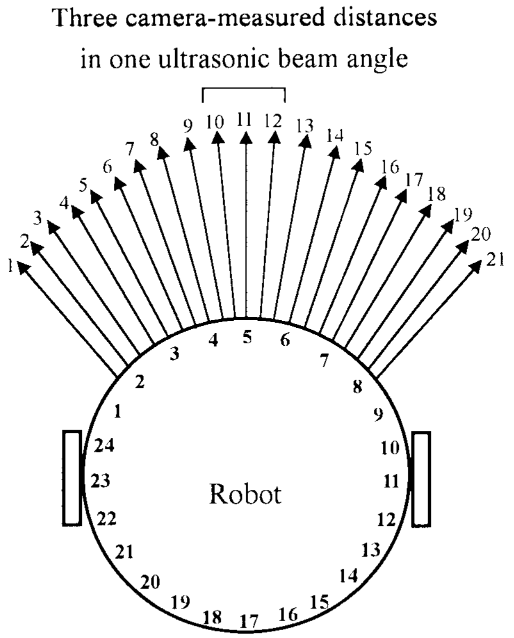

Fig. 2. Sensor arrangement for environmental configuration prediction; the numbers inside the circle indicate the ultrasonic sensor positions; the numbers at the arrows indicate the directions in which camera measures distances.

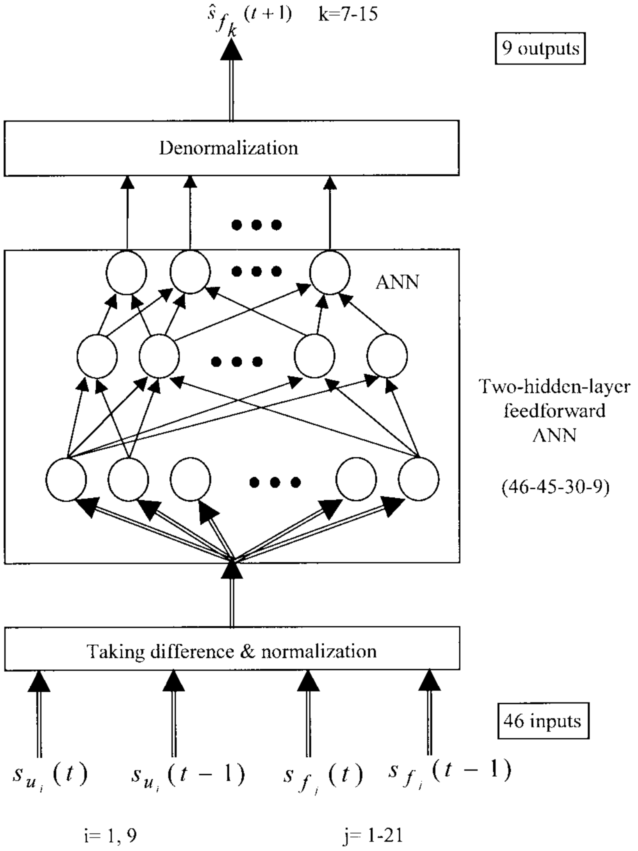

Fig. 3. ANN structure exploiting fused sensory data.

obstacles is kept small. This feature is suited for navigation applications. The neural network was trained off-line using data generated by sensor models. The entire training data for the neural network consist of thirty thousand sets of sensor readings. Approximately 29 h CPU time was spent on a Sun Classic workstation to train the neural network to convergence.

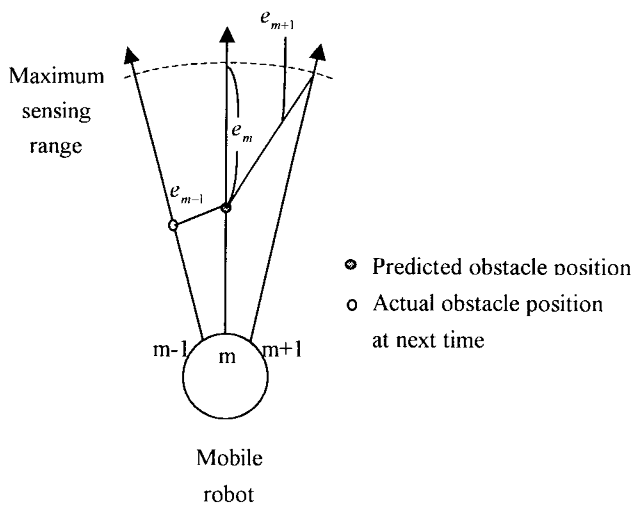

Fig. 4. Explanation of the APPE.

TABLE III
APPE Comparison of Using Fused Data and Using Ultrasonic Data (Obstacle Speed is 140–200 cm/s)

Sensor Data Used	APPE(cm) Using Current Data	APPE(cm) Using REBP
Ultrasonic Alone	34.5	24.8
Fused Data	35.6	15.9

It is clear that using fused data from the camera and ultrasonic sensors has a better directional resolution than using only ultrasonic sensors. However, with improved directional resolution, there is a higher possibility to lead to miss-calculation, in which the predicted obstacle is actually detected in the neighboring measurement direction. A situation in practice does little influence on the navigation performance. Next, a fair and unbiased comparison is made to confirm the prediction accuracy. Herein, we define an index termed apparent position prediction error (APPE). Assume that a measured (predicted) distance is regarded as a point obstacle in the measured (predicted) direction at that distance. The APPE is the distance between the predicted and the matched (at next sample instant) obstacles. The matched obstacle is the closest obstacle measured in the predicted or its neighboring directions. From Fig. 4

$$\text{APPE}_m = \min_{m-k<i<m+k} e_i \tag{9}$$

where

APPE_m APPE of the prediction in the m direction;

e_i distance from the predicted obstacle position in the m direction to the actual measured obstacle position in the ith direction;

k number of neighboring directions to be considered on each side of direction m.

The example shown in Fig. 4 reveals that although the obstacle will be measured in the $(m-1)$ direction, it is predicted in the m direction. The fact that e_{m-1} is smaller than e_m and e_{m+1} accounts for why it is regarded as the APPE_m. Notably, k is one for using ultrasonic sensors and four for using fused data; so they cover the same angle span. Table III presents the APPE comparison for predicting fast obstacles (140–200 cm/s). In this table, the APPE using current data indicates no prediction applied to the sensors. This finding indicates that the

APPE using the fused sensor data is markedly decreased from 24.8 to 15.9 cm. The fused data have less false data and a better prediction, thereby making their prediction results more useful for mobile robot navigation.

III. Reactive Navigation Among Multiple Moving Obstacles

We first consider an application task space in which both moving and stationary obstacles may present. Among multiple moving obstacles, the robot's safety is difficult to guarantee because

1) it may not detect an obstacle until the last second;
2) obstacles may behave carelessly or in a hostile way;
3) a moving obstacle may somehow simply lose control at the instant when it approaches the robot.

Owing to the above reasons, the following navigation principles are considered to prevent from collision with multiple moving obstacles.

1) The robot should move toward a place which is predicted to be safe.
2) The robot should control its speed according to the predicted positions of moving obstacles so that it can handle unexpected situations.
3) In addition, the robot should have no speed component toward an obstacle if it is already near the obstacle. This feature also guarantees its safety among static obstacles.

A reactive navigation method is developed herein for guiding a mobile robot among stationary and moving obstacles. The proposed method assumes that the goal and future positions of obstacles exert virtual forces on the robot. The subsequent virtual force changes the robot's motion so that it is more appropriate for the predicted environment. The subsequent force produces an acceleration, which is thereby transferred to the robot's velocity change for the next time interval. Note that in this system the future relative positions of obstacles are used for virtual force calculation.

A. Selection of Virtual Force

A rational choice of virtual force is critical in designing the navigation system for the robot among moving and stationary obstacles. With respect to such a choice, the following considerations must be made.

1) To cause the robot to move toward its destination, the goal has an attractive force associated with it. This is like a string which ties the robot and pulls it toward the goal. When there is an obstacle blocking the way to the goal, the string has to go around the obstacle. Consequently, a temporary virtual goal is set besides the obstacle.
2) A person decreases his speed of motion when he encounters obstacles to avert dangers caused by accidental changes in the obstacle's motion. If a sudden danger actually occurs, humans then quickly move away from the obstacle as if being pushed outward from it. Therefore, two zones around an obstacle are required: an outer deceleration zone that decreases the robot speed for relative safety, and an inner push-away zone that causes the robot to move away from the obstacle when it is too close to an obstacle.
3) In the deceleration zone, viscous force is applied because it decreases the robot's rate of movement in proportion to its speed. If the range for a viscous force is sufficiently large, the robot can achieve a stable speed owing to the balance between this force and the goal force. Therefore, this force can be included to control the robot's speed. Notably, even the robot is at a place covered by deceleration zones of many obstacles; it is sufficient to count only the one with the largest viscous force. The viscous force, however, cannot force the robot to stop because an attractive force is still exerted on it.

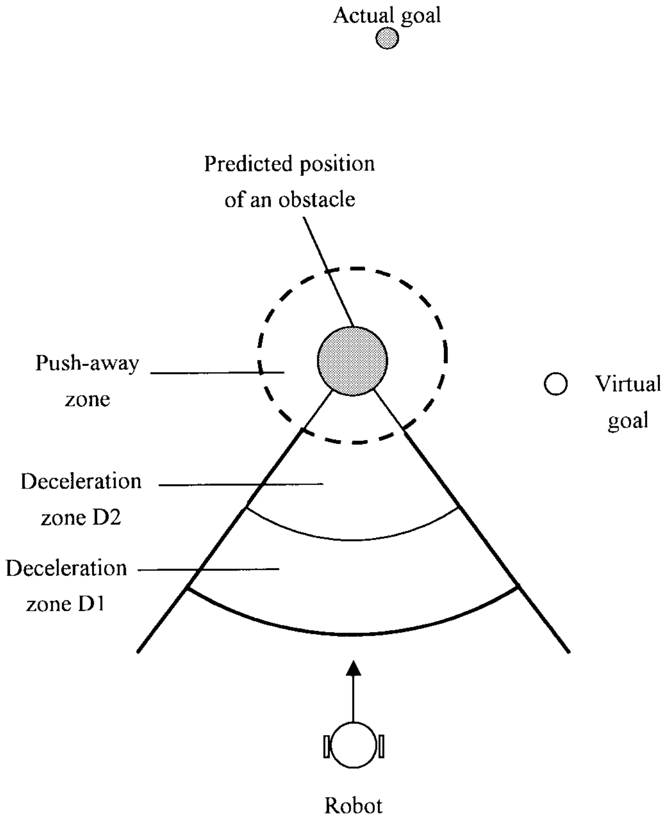

Fig. 5. Virtual goal and virtual obstacle forces.

4) In the push-away zone, a larger force can hopefully be exerted on the robot as it approaches an obstacle, ultimately causing the force to be sufficiently large to drive the robot away from the obstacle. For such a purpose, a virtual spring installed between the robot and obstacle is appropriate because the spring force exerted on the robot is directed away from the object. The exerted force increases as the robot penetrates deeper into the push-away zone. With an adequate amount of spring constant, this force guarantees that the robot never collides with a static obstacle.

Fig. 5 illustrates the concept of virtual goal and force zones. Two deceleration zones, $D1$ and $D2$ will be described in Section IV. The strengths and ranges of each force zone are the parameters to be determined. They must be carefully checked to achieve the desired behavior. The navigation system using these virtual forces fulfills the principles introduced previously.

B. Kinematic Constraints

The virtual force approach assumes a velocity controlled robot and provides motion speed commands $v[k]$ and steering angles $\beta[k]$. For a mobile robot with two independent drive wheels, a transformation to the speeds of the two drive wheels is required. The formula is as follows:

$$\begin{cases} v_r[k] = \left(2v[k] + \dfrac{W\beta[k]}{T_s}\right)\Big/2 \\ v_l[k] = \left(2v[k] - \dfrac{W\beta[k]}{T_s}\right)\Big/2 \end{cases} \tag{10}$$

where

v_r	speed of right wheel;
v_l	speed of left wheel;
W	distance between the two drive wheels;
T_s	sampling period.

Although kinematic constraints are considered during the parameter design, the reaction speed may still exceed the bounds. Therefore, the commands must be checked and modified before passed on to the wheel motors. Wheel speed constraints are absolute limits. If the speed of one wheel reaches the bound, the other wheel's speed is also adjusted accordingly to keep the robot on the desired path. When necessary, speed difference between the two drive wheels is adjusted such that the robot moves quickly on a gently curving path, rather than slowly on a sharply curving one.

These constraints obviously affect navigational performance. Herein, we employ a heuristic approach to resolve such constrained motion planning problems because efficiency is considered more important in this case. Although there is still small navigation difference due to kinematic and dynamic constraint, our reactive navigation scheme can resolve it by continually adjusting the robot's motion according to updated sensory information. On the other hand, previous investigations have demonstrated that planning robot motion in a dynamic environment with complete information is computationally difficult, even with only a bounded-velocity module [3], [20]. Importantly, deriving exact solutions would be even more difficult if there are other kinematic constraints and only incomplete information is available.

IV. Navigation Design

The kinematic constraints on our experimental robot, which has two independent drive wheels, include the maximum wheel speeds ($v_{\max}$), the maximum speed change ($\delta v_{\max}$) during a sampling period (T_s), and the maximum speed difference between the two drive wheels ($\Delta v_{d,\max}$). Herein, we adopt those constraints as references when designing the virtual force system.

As mentioned earlier, the goal force together with the viscous force in the deceleration zone can be used to control the robot speed when dealing with a moving obstacle. The closer the robot to an obstacle implies the larger viscosity that the virtual force system should have at that place. As Fig. 5 indicates, we divide the obstacle force zone into three areas. The two deceleration zones exist in front of the robot because the robot is assumed to move only forward using a single camera. The most distant area is deceleration zone $D1$, in which we expect the robot to be able to move at the desired speed at any place in this area. The middle area is the deceleration zone $D2$. This area is already near the obstacle; the robot in this area is expected to slow down. The closest inner area is the push-away zone, in which a spring force in addition to the viscous force is exerted on the robot to keep it away from the obstacle.

In the following, we describe the method of determining force related parameters for implementing the navigation system. The parameters include the ranges of three force zones, the strength of the goal force, the viscosity coefficients, and the spring constant. Notably, the range of each zone is expressed in terms of the *predicted distances* between the robot and obstacles. Expected navigation behaviors are transformed into equations to find the parameters. However, there are more variables than the equations. Therefore, numerical procedures are utilized to find the solutions.

A. Goal Force

To determine the virtual force strength, the goal force and the viscosity should be considered simultaneously so that the robot speed in $D1$ can be controlled. We expect that even when the robot moves at its maximum speed $v_{\max}$ toward an obstacle, it can immediately decrease to the desired speed $v_{D1,0}$ when coming to the inner bound of $D1$. This can be expressed as

$$v_{\max} - \frac{v_{\max} \times \mu_{D1,0}}{m} \times T_s + \frac{F_g}{m} \times T_s = v_{D1,0} \tag{11}$$

where

$\mu_{D1,0}$	viscosity at the most inner place in $D1$;
m	robot's mass;
F_g	rated goal force when there is an obstacle in the goal's direction.

In this equation, the second term on the left side means the speed change in a sample interval due to the viscous force. The third term on the left side means the speed change in a sample interval due to the goal force. On the other hand, because the robot is also expected to maintain this speed, then

$$v_{D1,0} \times \mu_{D1,0} = F_g \tag{12}$$

which means at this speed the viscous force can cancel out the goal force. Here $v_{D1,0}$ and the range of $D1$ can be determined by the designer, so these two equations can be used to determine F_g and $\mu_{D1,0}$.

When an obstacle is detected between the robot and its goal, a virtual goal is selected by the following principles.

1) The virtual goal should be at a distance from any obstacle by a safe margin.
2) The virtual goal should be close to the real goal.
3) If there are two virtual-goal candidates nearly as close to the real goal as each other, then select the one that has more free space around.
4) If no appropriate virtual goal can be set at the direction that differs from the goal direction by less than 90°, then use the original goal. In this case, the robot slows down, and replanning is invoked.

When obstacles are at directions other than the goal's direction, the goal force can be larger to make the robot approach more quickly. However, the components of this force toward obstacles at different directions cannot exceed F_g so that in the following we can find force parameters by considering only one obstacle (explained later). Therefore, the effective goal force f_g is

$$f_g = F_g / \cos(\theta) \tag{13}$$

where θ is the angle between the robot's heading and the direction of the obstacle that is closest to the heading.

B. Deceleration Zone $D1$

At $D1$'s outer bound, the robot is expected to travel with its maximum speed. We prefer that the robot can decrease its speed from here on. At this place, the viscous force can just cancel the goal attraction, and the viscosity, $\mu_{D1,1}$, can be determined by

$$v_{\max} \times \mu_{D1,1} = F_g. \tag{14}$$

At other places in $D1$, the viscosity is linearly interpolated between $\mu_{D1,0}$ and $\mu_{D1,1}$. As long as the range of $D1$ is sufficiently large, the robot speed can be controlled approximately to

$$v_d = \frac{F_g}{\mu_p} \tag{15}$$

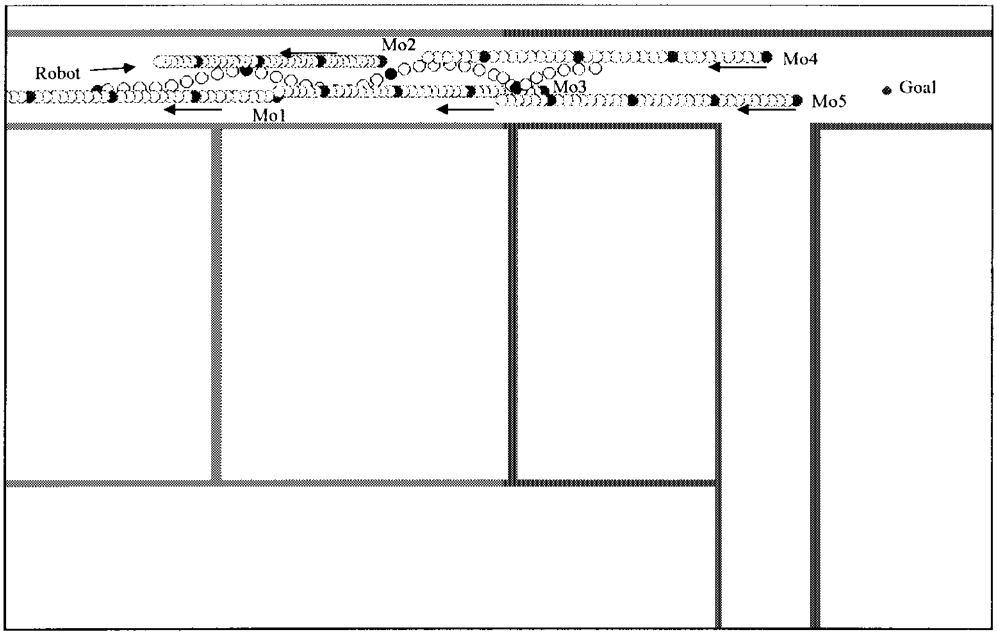

Fig. 6. Simulation result of navigation among multiple moving obstacles. All obstacles started to move at sample instant 0. The speeds of Mo1, Mo2, Mo3, Mo4, and Mo5 are, respectively, 80, 60, 72, 92, and 80 (cm/s).

where v_d is the desired speed, and μ_p is the viscosity at the robot's location p. However, it is preferred that the robot moves fast and thus $D1$ should be as small as possible. Because the goal force, $\mu_{D1,0}$ and $\mu_{D1,1}$ are already known, given an initial speed, the robot's speed and position can be calculated for every sample instant. To find the smallest range of $D1$, we assume the robot is at its highest speed before entering $D1$. Then a numerical routine is utilized to iteratively check different ranges for $D1$ to see if the robot's speed is controlled approximately to (15). Notably, here (and later on) we consider the case of only one obstacle, and assume that the goal is at the same direction of the obstacle. In practical situations, multiple obstacles may present and the robot may turn toward another obstacle when traveling to the goal. Since the projection of goal force in any other obstacle's direction is not larger than the above assumption for one obstacle case, the robot's speed is still suitable for the other obstacles. Therefore, this design is valid for the case of multiple obstacles.

C. Deceleration Zone $D2$

The viscosity at the outer bound of $D2$ is actually the same as $\mu_{D1,0}$. Because the possible highest speed just outside $D2$ can be calculated and the desired speed before entering push-away zone is predetermined, the robot's speed and position at every sample instant can be calculated. A numerical routine is employed to iteratively check different viscosity at inner bound of $D2$, $\mu_{D2,0}$, and to find the smallest $\mu_{D2,0}$ that can achieve the desired speed before entering push-away zone.

D. Push-Away Zone

The robot in push-away zone must be closely controlled so that there is no speed component toward the obstacle before the clearance between the robot and obstacle is smaller than the minimum detectable range of the sensor. The range of this zone depends on the robot's mobility and can be set by the designer. Because all other parameters are known, the spring constant can be determined using a numerical procedure. Here we require a spring constant that makes the robot slow down to a stop just before it passes this zone.

V. Simulations

Computer simulations have been performed to verify the navigational performance of a mobile robot among multiple moving obstacles using the proposed method. In the simulation, the robot was cylindrical in shape with a diameter of 60 cm. It was equipped with a camera and a ring of 24 ultrasonic rangefinders, each having a beam opening angle of 22.5° and effective range of 30–600 cm. The sampling time was 500 ms.

Fig. 6 shows the capability to avoid multiple moving obstacles that move straightforward without paying any attention to the robot. The robot smoothly avoids the first three moving obstacles as expected. At the fifth sampling instant, the robot predicts Mo1 approaching and sets a virtual goal in its front-left. At the tenth sampling instant, the robot predicts Mo2 approaching and sets a virtual goal in its front-right. At the fifteenth sampling instant, the robot adjusts its heading to the center. It does so because it attempts to move toward the goal. At the seventeenth sampling instant, Mo3 is predicted to approach the robot; therefore, the robot turns left to avoid it. At the twentieth sampling instant, the robot predicts Mo4 approaching and turns right. At the twenty-seventh sampling instant, the robot decelerates (which is less apparent elsewhere) since, under this circumstance, the robot is in a more or less cluttered environment. Notably, Mo4 is already close in the robot's left side, Mo5 is predicted to approach from the front, and a wall is in its front-right. The robot cannot choose an absolutely safe place to locate the virtual goal. Only a relatively safe virtual goal is set; therefore, the robot must slow down its speed. At the twenty-ninth sampling instant, Mo4 is no longer a threat and, thus, the robot turns left and then accelerates to avoid Mo5 and moves toward the real goal.

Fig. 7 reveals that the mobile robot can travel smoothly among multiple moving and static obstacles. At the tenth instant, the robot sets the virtual goal in its front-left to avoid Ob1. Its speed also decreases because of the wall. At the twenty-seventh instant, it turns right to avoid Ob3. At the forty-fourth instant, the robot turns left to avoid Mo1. The appearance of Ob2 narrows the passage and, consequently, the robot attempts to move safely in between the two obstacles. After the robot turns to the vertical corridor, two obstacles

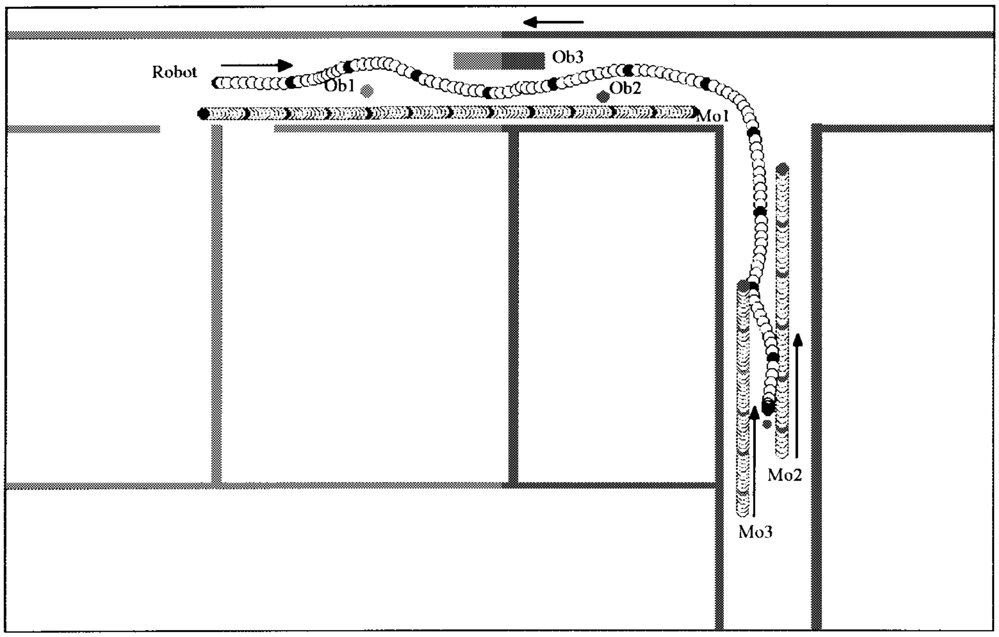

Fig. 7. Environment and the recorded trajectories. Mo1 started to move at the beginning with a speed of 80 cm/s; Mo2 started to move at the forty-fifth instant with a speed of 100 cm/s; and Mo3 started to move at the forty-fifth instant with a speed of 80 cm/s.

approach from the front. At the ninety-fourth instant, the robot begins to turn right slightly to keep a safe distance from Mo2. Then, at the 101st instant, it turns left quickly to avoid Mo3 coming head-on. Finally, it reaches the goal safely.

Fig. 8 presents a simulation result of multiple robots, which are identical in locomotion and sensing capacity. We assume that their ultrasonic sensors do not interfere with each other. Each robot wants to move to their individual goals. At the fourteenth sampling instant, both robots 3 and 4 predict that robot 1 will block their way and, therefore, set virtual goals on their right hand side because of their goal positions. As we see in Fig. 8, robot 3 begins to turn left at the fifteenth sampling instant despite the fact that robot 1 still blocks its way. This action confirms the merit of predicting environment configuration. Robot 1 does not avoid robot 3 because robot 3 does not block its way. Robot 3 cannot approach its goal quickly since, at that time, robot 2 also blocks its way. For robot 4, after it turns right to avoid robots 1 and 3, it can travel smoothly toward its goal. The average speeds of the four robots are 125, 132, 121, and 127 cm/s, respectively. This simulation demonstrates that the robot using prediction and the proposed reactive navigation scheme can satisfactorily move among multiple mobile robots.

In practice, when the mobile robot is near a movable obstacle, it must react even more responsively to take care of a sudden motion change of the obstacle. Therefore, we recommend to decrease the control sampling time when the robot enters the deceleration zone $D2$. The neural network predictor can be skipped in this mode because the advantage of prediction is much less when using a very short sampling time, and thus the computation load is decreased, which helps to achieve a quicker reaction.

VI. Experimental Results

Fig. 9 illustrates the experimental mobile robot developed in our laboratory. Its diameter is 60 cm and it is about 105 cm tall. It has two independent drive wheels and two casters for balance. Its maximum speed is 40 cm/s. A ring of 16 Polaroid ultrasonic sensors was installed around it at alternating heights of 30 and 75 cm with equal angle spans. The effective range of each sensor is 43–320 cm. These sensors are fired sequentially in four groups. Herein, we adopt the error-eliminating-rapid-ultrasonic-firing (EERUF) method proposed by Borenstein and Koren [2], a complete set of 16 sensor measurements takes about 140 ms. The robot has a CCD camera installed at a height of 123 cm. A 3.7 mm wide-angle lens is used. The lens views downward and can cover distances from 12 cm to over 400 cm within 120° span. The image data are preprocessed by a real-time image processing board developed in our laboratory [23]. Using the method described in Section II, it takes 50–80 ms to obtain 16 distance measurements covering 120°. Next, dead-reckoning position estimation is performed by fusing shaft-encoders and a gyroscope. The sampling time for a navigation command is 560 ms normally and 100 ms for a dangerous mode. The control and computational tasks

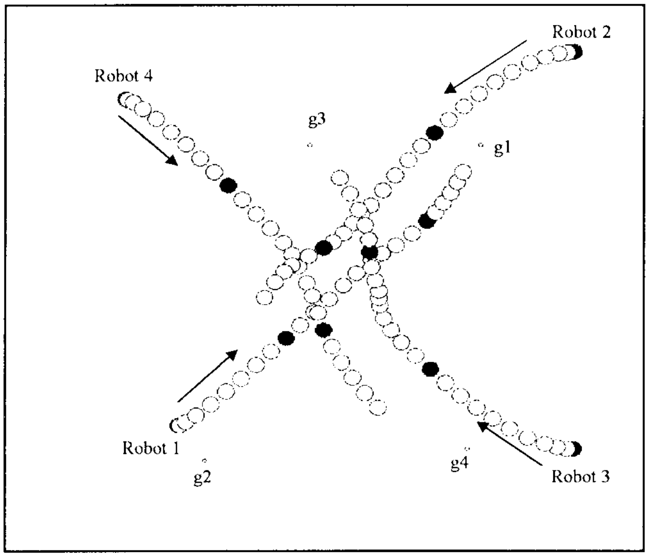

Fig. 8. Simulation result of multiple-robot navigation.

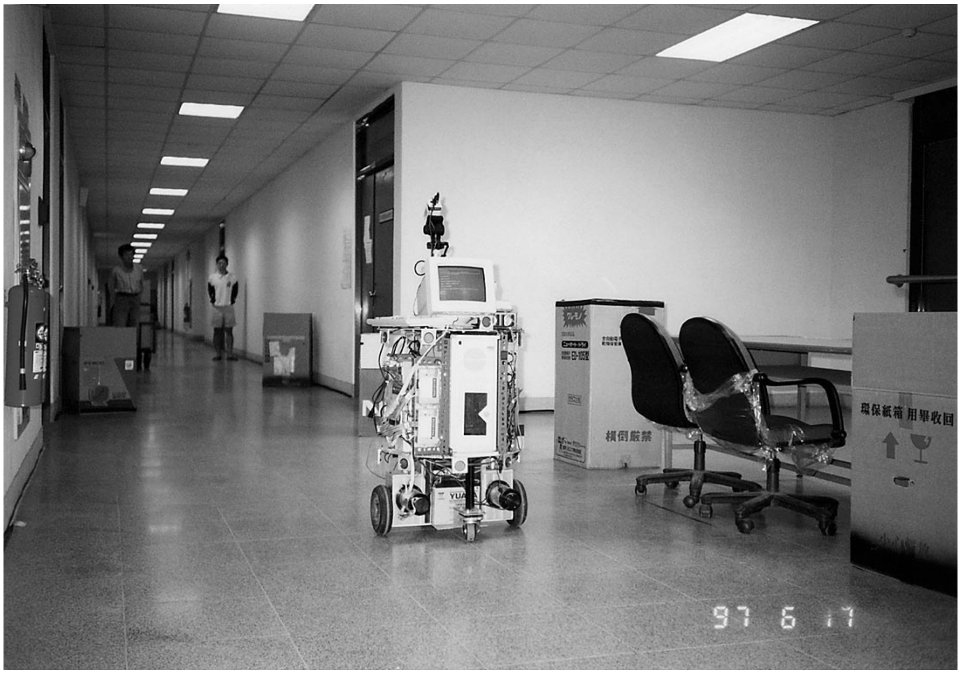

Fig. 9. Experimental mobile robot in this study.

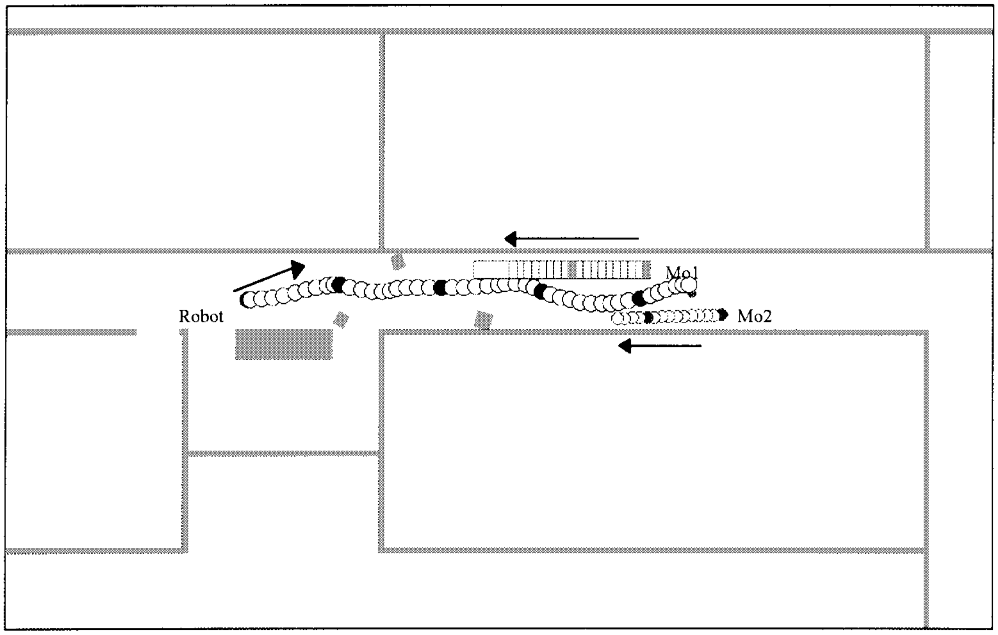

Fig. 10. Experiment confirming the capability of avoiding multiple moving as well as static obstacles. Mo1 started to move at the twentieth instant with a speed of 25 cm/s; and Mo2 started to move at the thirtieth instant with a speed of 25 cm/s.

are distributed among two HCTL-1100's (for servo motor control), an Intel 8751 (for ultrasonic sensor measurements), and an Intel Pentium-133 (for other tasks).

The experiments were conducted in our laboratory building. A portion of the floor (33 m × 21 m) was used in the experiments. The experimental results presented in Fig. 10 reveals that the robot can effectively handle multiple moving and stationary obstacles in an unstructured environment. There were a table, two chairs, seven cartons, a passing person, and another person pulling a cart in this experiment. The robot smoothly passed through three cartons in front of it. Then, it encountered the walking person with the cart. At the twentieth instant, it predicted that they were approaching, so the virtual goal was set to the right to cause the robot to avoid the obstacle. Immediately after it passed the obstacles, another approaching person was predicted near; therefore, the virtual goal moved to the left. At the forty-first instant, the actual goal became active in guiding the robot.

VII. Conclusion

We have presented in this paper a precautionary navigation system for an intelligent mobile robot in dynamic environments. Sensory data from a CCD camera and 16 ultrasonic rangefinders are successfully fused into an uniform representation for distance measurement to the robot's immediate surroundings. On the issue of reactive control, we proposed the neural network based obstacle-configuration predictor exploiting multiple sensors. This neural network is trained off-line using the relative-error-backpropagation algorithm. It is then employed to predict on-line in real-time the obstacle positions in next sample instant. We have presented a novel reactive navigation scheme to enable the robot make use of the predicted environmental information. The main advantage of this method is that the robot's speed as well as its moving direction can be determined simultaneously. It is more efficient in dealing with multiple moving obstacles than traditional potential field approaches. The proposed methods have been implemented and tested on a real mobile robot constructed in the laboratory. Simulation and experimental results confirm that the navigation system can cope with unstructured and dynamic environments. To the best of our knowledge, there are no reported experimental results of mobile robots that have better reactive performance in a dynamic unknown environment than the result presented in this paper. There are many directions for future investigation of autonomous navigation of mobile robots. For practical realization of reactive control based on sensory information, the reliability of the perceptual sensors affects the navigation performance. Further investigation will focus primarily on improving on-board perceptual performance. One direction is to use two or more cameras for improving the scene understanding and cover more surroundings. Adding knowledge to the system can also enhance the perception capability. Some false sensory data could be eliminated using the learned knowledge. The reactive control presented in this paper can only handle the local situations. Another future work is the integration with a higher level world model and task planner. More knowledge about the global environment can improve robot motion planning, thereby resulting in better navigation.

References

[1] J. Borenstein and Y. Koren, "Real-time obstacle avoidance for fast mobile robots," *IEEE Trans. Syst., Man, Cybern.*, vol. 19, no. 5, pp. 1179–1187, 1989.

[2] ——, "Noise rejection for ultrasonic sensors in mobile robot applications," in *Proc. IEEE Int. Conf. Robot. Automat.*, Nice, France, 1992, pp. 1727–1732.

[3] J. Canny and J. Reif, "New lower bound techniques for robot motion planning problems," in *Proc. 28th IEEE Symp. Foundations Comput. Sci.*, Los Angeles, CA, 1987, pp. 49–60.

[4] C. C. Chang and K. T. Song, "Environment prediction for a mobile robot in a dynamic environment," *IEEE Trans. Robot. Automat.*, vol. 13, pp. 862–872, Dec. 1997.

[5] ——, "Dynamic motion planning based on real-time obstacle prediction," in *Proc. of 1996 IEEE Int. Conf. Robot. Automat.*, Minneapolis, MN, 1996, pp. 2402–2407.

[6] M. Erdmann and T. Lozano-Perez, "On multiple moving objects," *Algorithmica*, vol. 2, no. 4, pp. 477–522, 1987.

[7] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *Int. J. Robot. Res.*, vol. 17, no. 7, pp. 760–772, 1998.

[8] T. Fraichard and C. Laugier, "Path-velocity decomposition revisited and applied to dynamic trajectory planning," in *Proc. IEEE Int. Conf. Robot. Automat.*, Atlanta, GA, 1993, pp. 40–45.

[9] K. Fujimura and H. Samet, "A hierarchical strategy for path planning among moving obstacles," *IEEE Trans. Robot. Automat.*, vol. 5, pp. 61–69, Feb. 1989.

[10] N. C. Griswold and J. Eem, "Control for mobile robot in the presence of moving objects," *IEEE Trans. Robot. Automat.*, vol. 6, pp. 263–268, Apr. 1990.

[11] I. Horswill, "Polly: A vision-based artificial agent," in *Proc. AAAI'93*, Menlo Park, CA, 1993, pp. 824–829.

[12] K. Kant and S. W. Zucker, "Toward efficient trajectory planning: The path-velocity decomposition," *Int. J. Robot. Res.*, vol. 5, no. 3, pp. 72–89, 1986.

[13] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Int. J. Robot. Res.*, vol. 5, no. 1, pp. 90–98, 1986.

[14] K. J. Kyriakopoulos and G. N. Saridis, "An integrated collision prediction and avoidance scheme for mobile robots in nonstationary environments," in *Proc. IEEE Int. Conf. Robot. Automat.*, Nice, France, 1992, pp. 194–199.

[15] J. F. G. de Lamadrid and N. L. Gini, "Path tracking through uncharted moving obstacles," *IEEE Trans. Syst., Man, Cybern.*, vol. 20, no. 6, pp. 1408–1422, 1990.

[16] J. Latombe, *Robot Motion Planning.* Boston, MA: Kluwer, 1991.

[17] D. Nair and J. K. Aggarwal, "Moving obstacle detection from a navigating robot," *IEEE Trans. Robot. Automat.*, vol. 14, pp. 404–416, June 1998.

[18] Y. S. Nam, B. H. Lee, and M. S. Kim, "View-time based moving obstacle avoidance using stochastic prediction of obstacle motion," in *Proc. IEEE Int. Conf. Robot. Automat.*, Minneapolis, MN, 1996, pp. 1081–1086.

[19] T. Pan and R. C. Luo, "Motion planning for mobile robots in a dynamic environment with moving obstacles," in *Proc. IEEE Int. Conf. Robot. Automat.*, Cincinnati, OH, 1990, pp. 578–583.

[20] J. Reif and M. Sharir, "Motion planning in the presence of moving obstacles," in *Proc. 26th IEEE Symp. Foundations Comput. Sci.*, Portland, OR, 1985, pp. 144–154.

[21] J. T. Schwartz and M. Sharir, "A survey of motion planning and related geometric algorithms," *AI J.*, vol. 37, pp. 157–169, 1988.

[22] C. L. Shih, T. Lee, and W. A. Gruver, "A unified approach for robot motion planning with moving polyhedral obstacles," *IEEE Trans. Syst., Man, Cybern.*, vol. 20, no. 4, pp. 903–915, 1990.

[23] K. T. Song and F. C. Sheu, "Visual guidance control based on velocity mapping," in *Proc. 3rd IFAC Symp. Intelligent Autonomous Vehicles*, Madrid, Spain, 1998, pp. 193–198.

[24] K. T. Song and W. H. Tang, "Environment perception for a mobile robot using double ultrasonic sensors and a CCD camera," *IEEE Trans. Ind. Electron.*, vol. 43, pp. 372–379, May/June 1996.

[25] T. Tsubouchi, S. Kuramochi, and S. Arimoto, "Iterated forecast and planning algorithm to steer and drive a mobile robot in the presence of multiple moving objects," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, Pittsburgh, PA, 1995, pp. 33–38.

[26] Q. Zhu, "Hidden-Markov model for dynamic obstacle avoidance of mobile robot navigation," *IEEE Trans. Robot. Automat.*, vol. 7, pp. 390–397, June 1991.