This paper appears in the Proceedings of the World Congress on Neural Networks, vol. 2, pp.602-607, July 17-21, 1995, Washington D.C.

FUZZY-NAV: A Vision-Based Robot Navigation Architecture using Fuzzy Inference for Uncertainty-Reasoning*

J. Pan, D. J. Pack, A. Kosaka, and A. C. Kak
Robot Vision Laboratory
1285 EE Building, Purdue University
West Lafayette, IN 47907-1285

Abstract

We report here a new but robust vision-based control architecture for indoor mobile-robot navigation. On one hand, this architecture takes advantage of the high-throughput of neural networks for the processing of camera images. And, on the other, it employs fuzzy logic to deal with the uncertainty in the inferences drawn from the vision data. In particular, we will present in this paper the sixteen fuzzy terms that have proved sufficient for the desired navigational behavior. The reader should also note that the architecture we present in this paper allows our robot to simultaneously navigate and avoid obstacles, both static and dynamic.

1 Introduction

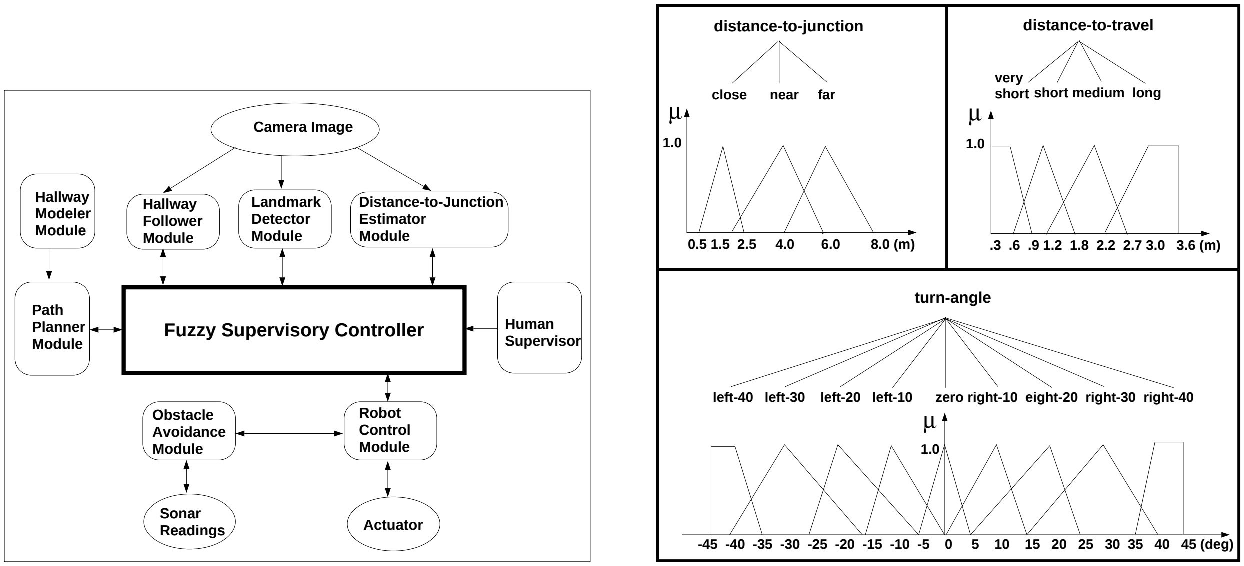
As the readers probably know, our laboratory has previously produced two radically different control architectures for vision-based navigation by indoor robots. The FINALE system, presented in [4], allowed our robot to navigate at roughly 8 meters/minute using model-based vision in which expectation maps, constructed from a geometrical model of the hallways, are compared with camera images to determine the location of the robot. The FINALE system is heavily geometrical, in the sense that it requires that a 3D model of the hallways be known. Our second architecture, the NEURO-NAV system presented in [9], was more human-friendly, in the sense that it used topological models of hallways. As discussed in [9], from the topological models of space the path planner in NEURO-NAV outputs a sequence of navigational commands like "*[straight to the second T-junction, turn right, straight to the third door on the left].*" In executing these commands, the robot invokes an ensemble of neural networks for steering control. For example, the input to one of the neural networks, the Hallway Follower, consists of the Hough space of the camera image; each output node of this network corresponds to a command to the robot to turn right or left by a certain number of degrees. With the same hardware as for FINALE, the NEURO-NAV system is able to generate a steering command every three seconds. Readers should refer to [1, 2, 5, 6, 11, 13, 14] for other approaches.

The work reported in this paper represents an advance over NEURO-NAV, an advance that will allow our robot to ultimately go beyond just navigation and also engage in more complex tasks such as simultaneously finding an object while it is navigating. The robot might, for example, look for a fire-extinguisher whose location is only approximately known in the hallway. This enhanced level of intelligence is made possible by using fuzzy logic to couple the output of the various neural networks to a supervisory controller which decides what commands to issue to the actuators. More specifically, while in the NEURO-NAV system each steering command produced by one of the neural network was considered as a categorical command, in our present work it is now treated as a command with a certain ambiguity associated with it – a much more natural thing to do. Similarly, the NEURO-NAV system treated each distance estimate produced by one of the neural networks categorically. In our new system, an ambiguity interval is associated with each such estimate. Some references[7, 8, 10, 15, 16] are helpful to understand how fuzzy inference takes place in real-time applications.

In what follows, in Section 2 we will first describe the overall architecture of our new system that we have named FUZZY-NAV. Section 3 will then delve into the Fuzzy Supervisory Controller, the heart of FUZZY-NAV. We will display some of the rules and talk about the linguistic variables and the associated fuzzy terms and their membership functions. Finally, in Section 4 we will discuss the working of the entire system and show some results.

*This work was supported by the Office of Naval Research under Grant ONR N00014-93-1-0142.

2 Architecture of FUZZY-NAV



(a) (b)

Figure 1: *(a) Architecture of FUZZY-NAV. (b) Linguistic variables and their associated fuzzy terms used in the Fuzzy Supervisory Controller.*

As depicted in Fig.1(a), the heart of the FUZZY-NAV system consists of the Fuzzy Supervisory Controller that is connected to six peripheral modules drawn from the earlier NEURO-NAV system. So that the reader can understand the workings of the supervisory controller, we need to describe these peripheral modules. That we will do now, albeit briefly, for the convenience of the reader.

As discussed in detail in [9], the Hallway Follower consists of two competing neural networks that help the robot stay in the middle of the hallway when there are no obstacles present. Different regions of the Hough transform of the camera image are fed into the input nodes of these neural networks. The two neural networks together have a total of ten output nodes corresponding to the following steering commands: *left-40 left-30 left-20 left-10 go-straight right-10 right-20 right-30 right-40* and *no-decision*

The Distance-to-Junction Estimation Module gives the robot a sense of how far it is from a hallway junction. As shown in [9], certain regions of the Hough space representation of the camera image correspond to those straight lines in a hallway that are horizontal but perpendicular to the hallway in which the robot is traveling. By feeding these regions of the Hough space into an appropriately trained neural network, an estimate of the distance to the end of the hallway can be constructed. The output nodes of this network are *far, near, close* and *no-decision.*

The Obstacle Avoidance Module in Fig.1(a) consists of a semi-ring of ultrasonic sensors. Through these sensors, the module can estimate both the direction of and the distance to the obstacle. When the distance to the obstacle falls below a certain preset threshold, the Collision Avoidance Module interrupts the CPU and takes over the Robot Control Module. As long as an obstacle can be detected by any of the sonar sensors, the Robot Control Module remains completely under the control of the Obstacle Avoidance Module. When no obstacles are detected, control reverts back to the Fuzzy Supervisory Controller. The Landmark Detector Module contains neural networks, each specialized for a specific recognition task. One of the main neural networks is for recognizing door frames of closed doors from a certain range of perspectives. This module, obviously of critical importance if our robot is to succeed in simultaneous navigation and object finding, is still undergoing development and refinement in our laboratory.

It is clear then that the Fuzzy Supervisory Controller uses the information received from the Hallway Follower, the Distance-to-Junction Estimator, and the Landmark Detector neural networks to decide how to control the robot. This control must of course be in accordance with the sequence of top-level commands received initially from the Path Planner. As discussed in [9], the Path Planner uses a topological model of the hallways and the information supplied by the human about the desired destination location to spit out a sequence of navigational commands. We are now ready to discuss in greater detail the working of the Fuzzy Supervisory Controller.

3 Fuzzy Supervisory Controller

This controller is in reality a real-time expert system that takes in the outputs of all the neural networks and decides what commands to issue to the Robot Control Module that drives the actuators. The Fuzzy Supervisory Controller uses three linguistic variables, *distance-to-junction, turn-angle, distance-to-travel*. Associated with these linguistic variables are a total of sixteen fuzzy terms, as shown in Fig.1(b). The semantics of the linguistic variables *distance-to-junction* and *turn-angle* should be obvious to the reader. The linguistic variable *distance-to-travel* stands for the current best estimate of how far the robot should plan on traveling straight barring any encounters with obstacles. While the value of the linguistic variable *distance-to-junction* is derived from the vision data by one of the neural networks, the linguistic variable *distance-to-travel* is given a value by the firing of one or more rules in the Supervisory Controller. As with all systems using fuzzy logic, the membership functions for the terms displayed in Fig.1(b) were arrived at empirically. In order to appreciate the features of our system, consider, for example, the following rules that are in the supervisory controller:

```
(rule-name rule20
IF (turn-angle right-20)
     (distance-to-junction far)
     (sonar-reading no-obstacle)
THEN
     (turn-angle right-20)
     (distance-to-travel long))


(rule-name rule31
IF (turn-angle right-30)
     (distance-to-junction far)
     (sonar-reading no-obstacle)
THEN
     (turn-angle right-30)
     (distance-to-travel medium))
```

It should now be obvious to the reader that the rule antecedents shown above are entirely in terms of the linguistic variables and the fuzzy terms. For example, in the first rule the first condition element, (turn-angle right-20), the linguistic variable is *turn-angle*, with *right-20* as one of the terms associated with this linguistic variable. We now describe the following key aspects of the workings of the Supervisory Controller that make possible the fuzzy inference.

Aspect 1: Unlike the more traditional rule-based systems, it should be possible for a rule to fire even when the data and the rule-antecedent are not identical patterns. Assume that the following fact has been asserted on the basis of the reports from the sensors:

(turn-angle *right-20*)

and assume that a rule has the following condition element in its antecedent:

(turn-angle *right-30*)

As is clear from the membership functions shown in Fig.1(b), there is an overlap between the membership functions for *right-20* and *right-30*. Therefore, the data fact (turn-angle *right-20*) should match the rule-antecedent condition element (turn-angle *right-30*), although of course not perfectly. In keeping with the fuzzy inference schemes described by [7], such a match would succeed, the degree of match represented by a number calculated by the following sup-min operation:

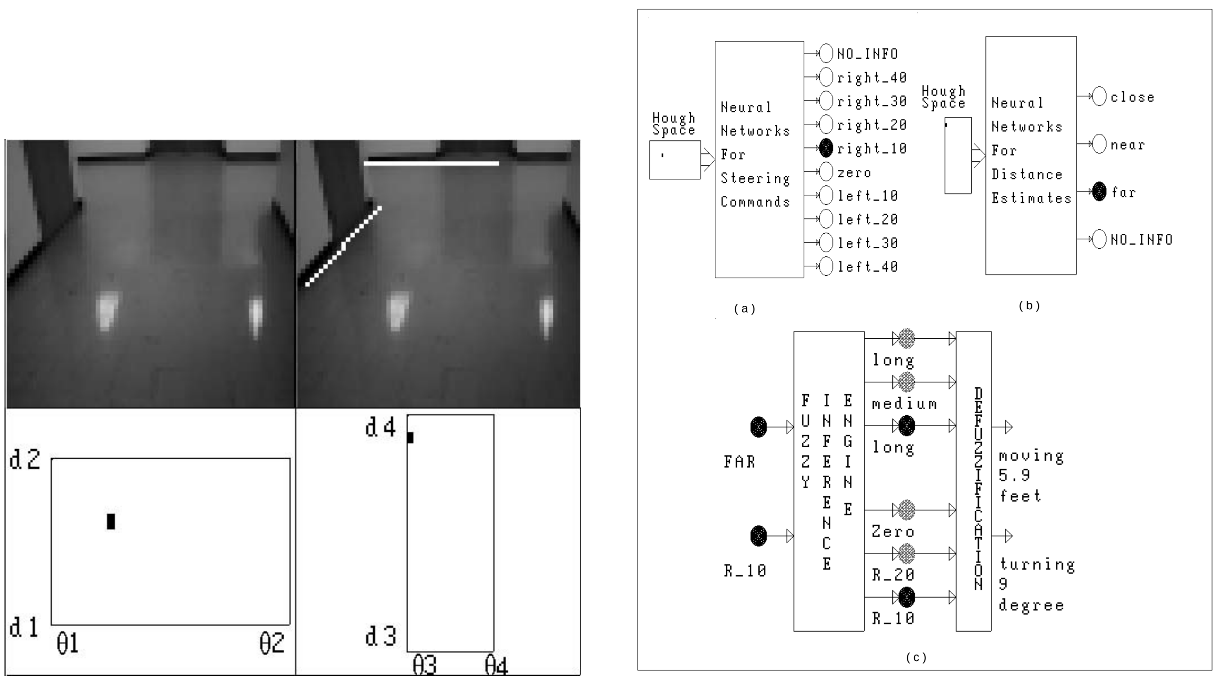
$$\sup\min\,(\mu_{right-20}, \mu_{right-30})$$

where $\mu_{right-20}$ and $\mu_{right-30}$ are the membership functions associated with the fuzzy terms *right-20* and *right-30*, respectively.

Aspect 2: Especially on account of the matchings made possible by fuzzy terms with overlapping membership functions, the same data can fire multiple rules. When multiple rules make assertions about the same linguistic variable, it becomes necessary to aggregate the membership functions of all the fuzzy terms asserted by the rule consequents. Consider, for example, the two rules, rule20 and rule31 defined above. Given the facts (turn-angle *right-20*), (distance-to-junction *near*) and (sonar-reading *no-obstacle*) asserted in the working memory, the rules named by *rule20* and *rule31* will be fired by the Fuzzy Supervisory Controller. As a result, for the linguistic variable *turn-angle*, both the fuzzy terms *right-20* and *right-30* will be asserted as commanded steering angles. Similarly, for the linguistic variable *distance-to-travel*, the fuzzy terms *long* and *medium* will be asserted as the distance the robot should expect to travel straight barring any obstacles. **The gist of Aspect 2 is that when the Supervisory Controller asserts multiple fuzzy terms for the same linguistic variable, their membership functions must be combined after they are weighted in accordance with the dictates of fuzzy logic [7]. And, then the composite membership functions must be defuzzified to yield a single numerical value for, say, the turn angle for the robot.** As far as final defuzzification is concerned, we use the "center of area" method .

We have implemented a fuzzy expert system that can achieve the two aspects above. Briefly speaking, we have some special data structures for representing the fuzzy terms and their membership functions so that Aspect 2 can be achieved. As for Aspect 1, we modify the so-called Rete networks[3] so that a fact may enable a rule despite the different symbols appearing in the fact and the rule antecedent. For lack of space, we have left out many crucial implementation details here that could help the reader understand better how the key aspects of Fuzzy Supervisory Controller are achieved. The interested readers are referred to a very detailed paper [12].

4 Experimental Results

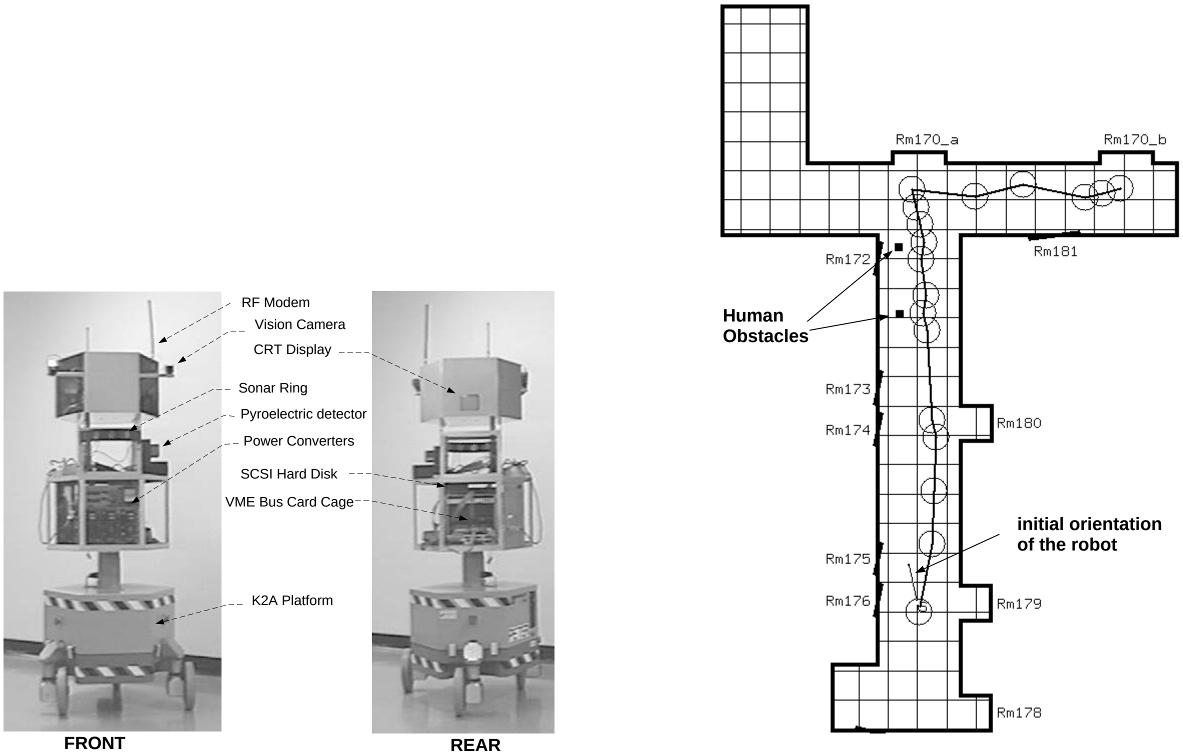


(a) (b)

Figure 2: *(a)Shown here are the downsampled image on the left upper corner, the extracted edges on the left lower corner, the Hough space for generating the steer command on the right upper corner and the Hough space of the horizontal lines for generating the distance estimate on the right lower corner. (b) Shown here are the outputs of the neural networks given the Hough spaces in Fig.2(a) and the outputs of fuzzy inference engine.*

Our robot contains a Cybermotion K2A platform and a VME-based computing hardware. The VME card cage contains a MC68040 microprocessor based MVME167 single-card computer, an Imaging Technology frame grabber, a sonar transducer board and a board for multi-channel digitization of analog signals. A semi-ring of five ultrasonic sonars provides the range information for obstacle avoidance. The MVME167 board runs the VxWorks real-time operating system.

Fig. 2(a) shows a downsampled camera image in the upper left corner; the extracted edges are shown in the right upper image. The edge image is then mapped into Hough space. As described in detail in [9], different regions of the Hough space are used as input for the different neural network in the system. The portion (d1,d2)x($\theta1, \theta2$) is used as input region for the Hallway Follower module, whereas the portion bounded by (d3,d4)x($\theta3, \theta4$) is used as input for the Distance-to-Junction Estimator Module. The precise values for d1, d2, d3 , d4, $\theta1$, $\theta2$, $\theta3$ and $\theta4$ are not important to the discussion here as they depend on



(a) (b)

Figure 3: *(a)Shown here are pictures of the front and the rear of our mobile robot. (b) One typical run of the robot with human obstacles.*

such factors as the pan and tilt angles of the camera. Shown at the lower left in Fig.2(a) is that part of the Hough space derived from the edge image of upper right which is used as input to Hallway Follower Module. And, shown at the lower right is that part which is used as input to the Distance-to-Junction Estimator Module. Shown in Fig.3(a) are pictures of the front and the back of the mobile robot. The reader will notice in the back of the robot a CRT display where composite images like those shown in Fig.2(b) pop up as the robot is navigating down the hallway. As the vision data is processed by each of the three vision-based modules shown in Fig.1(a), one of the three iconic figures displayed in Fig.2(b) also pops up. For example, after the Hallway Follower has processed its part of the Hough representation of the camera image shown in Fig. 2(a), the iconic figure at upper left in Fig.2(b) pops up on the CRT display. In the example shown, the output node *right_10* lights up, meaning that the Hallway Follower is recommending that the robot turn to the right by 10 degrees in order to stay in the middle of the hallway. Subsequently, the iconic figure shown at upper right of Fig.2(b) shows up and the output node *far* lights up, implying that according to the Distance-to-Junction Estimator the distance to the junction is far. All of this information is fed into the Fuzzy Supervisor Controller. When this happens, the iconic figure at the bottom of Fig.2(b) pops up on the screen. What is shown in that figure is that the Fuzzy Supervisory Controller takes in the recommendations *far* for distance to junction and *right-10* for the turn angle and issues forth its decisions regarding what the robot actions should be. As the iconic diagram shows, in this case a total of three rules were fired that made assertions about how far the robot should continue to travel in a straight line, two of these asserted that the value of the linguistic variable *distance-to-travel* should be *long* while one rule asserted that it should be *medium*. The same three rules also asserted that the the value of the linguistic variable *turn-angle* should be *zero*, *right-20* and *right-10*. The defuzzification of these assertions made by the Fuzzy Supervisory Controller results in the values of 5.9 feet for the latest estimate of how far the robot should travel before expecting to see the junction and that the turn angle at this time should be 9 degrees in order to bring the robot to approximately the middle of the hallway.

Shown in Fig.3(b) is one typical run of the robot controlled by FUZZY-NAV. The robot must work around the obstacle as it tries to reach its destination. During this maneuvering around the obstacle, the robot is completely under the control of the Obstacle Avoidance Module. After the robot has cleared the obstacle, it resumes its normal vision-based navigation to the destination point. The Obstacle Avoidance Module is also invoked if the robot gets too close to any of the walls.

5 Conclusion

We have discussed how a supervisory controller using fuzzy-inference can be used to control the navigational behavior of a mobile robot using vision for sensory feedback. In order to deal with the ambiguities that are associated inherently with any interpretation from sensory information of any kind, one of course has the option of using one of many different calculi of uncertainty. However, of all such calculi – Bayesian, Dempster-Shafer, fuzzy logic, etc. – we believe that fuzzy logic has the advantage of being the most convenient to use. With Bayesian methods, there is always the problem of what to do with conditional dependencies and independencies; and with the use of Dempster-Shafer based methods, it is non-trivial to translate sensory outputs into probability masses. By contrast, with fuzzy logic it is relatively easy to set up an intuitive connection with the membership functions that characterize the ambiguities associated with the different parameters.

References

[1] N. Ayache and O. D. Faugeras, "Maintaining representations of the environment of a mobile robot," *IEEE Transactions on Robotics and Automation*, Vol. 5, No. 6, 1989, pp. 804-819.

[2] C. Fennema, A. Hansen, E. Riseman, J. R. Beveridge and R. Kumar, "Model-directed mobile robot navigation," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 20, No. 6, November/December 1990, pp.1352-1369.

[3] C. L. Forgy, "Rete: A fast algorithm for the many pattern/many object pattern match problem," *AI*, Vol. 19, pp. 17-37, 1982.

[4] A. Kosaka and A. C. Kak, "Fast vision-guided mobile robot navigation using model-based reasoning and prediction of uncertainties," *Computer Vision, Graphics, and Image Processing – Image Understanding*, Vol. 5, pp.271-329, Nov. 1992.

[5] K. Konolige, K. Meyers and A. Saffiotti, "FLAKEY, an autonomous mobile robot," technical document, Stanford Research Institute International, July 20, 1992.

[6] D. J. Kriegman, E. Triendl, and T. O. Binford, " Stereo vision and navigation in buildings for mobile robots," *IEEE Transactions on Robotics and Automation*, Vol. 5, No. 6, 1989, pp.792-803.

[7] C. C. Lee, "Fuzzy logic in control systems: Fuzzy logic controller – Part 2," *IEEE Transaction on Systems, Man and Cybernetics* , Vol. 20, pp. 419-435. 1990.

[8] E. H. Mamdani, "Application of fuzzy logic to approximate reasoning using linguistic synthesis", IEEE Transaction on Computers, Vol. 26, No. 12, pp. 1182-1191, 1977.

[9] M. Meng and A. C. Kak, "NEURO-NAV: A neural network based architecture for vision-guided mobile robot navigation using non-metrical models of the environment," in *Proc. 1993 IEEE Int. Conf. on Robotics and Automation*, Vol. 2, pp.750-757.

[10] M. Mizumoto and H. Zimmermann, "Comparison of fuzzy reasoning methods," Fuzzy Sets and Systems, Vol. 8, pp. 253-283, 1982.

[11] H. P. Moravec, "The Stanford Cart and the CMU Rover", *Proc. IEEE* 71(7), July 1983, pp.872-884.

[12] J. Pan, "Design of a Large-Scale Expert System using Fuzzy Logic for Uncertainty-Reasoning and its application to Vision-Based Mobile Robot Navigation," Preliminary Report, Department of Electrical Engineering, Purdue University, 1994.

[13] F. G. Pin, H. Watanabe, J. Symon and R. S. Pattay, "Autonomous navigation of a mobile robot using custom-designed qualitative reasoning VLSI chips and boards," *1992 IEEE Int'l Conf. on Robotics and Automation*, pp. 123-128.

[14] M. Sugeno, T. Murofushi, T. Mori, T. Tatematasu, J. Tanaka , "Fuzzy algorithmic control of a model car by oral instructions," *Fuzzy Sets and Systems*, No. 32, pp. 207-219, 1989.

[15] R. R. Yager and L. A. Zadeh, "Fuzzy sets, neural networks and soft computing", Van Nostrand Reinhold, 1994.

[16] L. A. Zadeh, "The role of fuzzy logic in the management of uncertainty in expert system," *Fuzzy Sets and Systems 11*, North-Holland, pp. 199-227,1983.