

Evolution of Neural Controllers for Robot Navigation in Human Environments

Genci Capi and Hideki Toda

Department of Electrical and Electronic Systems Engineering,
Faculty of Engineering, University of Toyama Gofuku Campus,
3190 Gofuku, Toyama, 930-8555, Japan

Abstract: Problem statement: In this study, we presented a novel vision-based learning approach for autonomous robot navigation. **Approach:** In our method, we converted the captured image in a binary one, which after the partition is used as the input of the neural controller. **Results:** The neural control system, which maps the visual information to motor commands, is evolved online using real robots. **Conclusion/Recommendations:** We showed that evolved neural networks performed well in indoor human environments. Furthermore, we compared the performance of neural controllers with an algorithmic vision based control method.

Key words: Robot navigation, vision, neural networks, evolutionary algorithm

INTRODUCTION

The general problem of designing a machine for real time navigation and obstacle avoidance in an arbitrary environment is ongoing. This problem is often described as a problem in artificial intelligence, fuzzy logic, sensor fusion, intelligent control, ad infinitum. When dealing with numerous, noisy, conflicting, incomplete and uncertain information from multiple streams, designing robotic computer systems and algorithms for autonomous navigation and obstacle avoidance is non-trivial.

Robot navigation in human environments has been previously investigated from a number of different approaches. In most of these methods, the robots utilize the distance, ultrasonic, or laser sensors to navigate in the environment. However, the main drawback of the sonar or laser sensors lies in the fact that one sensor is required for one distance measurement, that is, in order to obtain a complete picture of the environment around the robot, a number of sensors must be used. Moreover, to achieve the accuracy in detection, they will have to be placed perpendicular to the target.

Recently, vision based robot navigation has attracted many researchers. Variations have included using sensory input from stereo vision, monocular vision and the combination of vision with other sensors. Methods also vary in how they deal with temporal information, from using individual frames exclusively (Horswill, 1993) to computing optical flow fields from

multiple frames. Domains include road and off-road travel (Hebert *et al.*, 1995; Rosenblatt and Thorpe, 1995; Turk *et al.*, 1988; Crisman, 1992; Zeng and Crisman, 1995; Dickmanns *et al.*, 1990) and indoor robotic navigation (Horswill, 1993; Coombs and Roberts, 1992; Santos-Victor and Sandini, 1996; Santos-Victor *et al.*, 1995).

Neural Networks (NNs) had also been used by some other researchers for solving the said problem. In this connection, researches (Yang and Meng, 2000; Mondada and Floreano, 1995; Capi, 2007; Yamada, 2005; Pal and Kar, 1996; Gu and Hu, 2002) are important to mention. However, the performance of an NN depends on its architecture and connecting synaptic weights, optimal selection of which is a tedious job. A variety of tools based on supervised and reinforcement learning algorithms had been used by a few investigators for this purpose. Back-propagation algorithm is the most popular method to optimize the NN, but it may have the local minimum problem. Simulated Annealing (SA) (Goffe *et al.*, 1994), Genetic Programming (GP) (Koza, 1992), Genetic Algorithms (GA) (Goldberg, 1989) have also been used by some researchers for the said purpose. It is to be noted that GAs along with NN has added a new dimension to the field of robotic research, namely evolutionary robotics (Pratihari, 2003). Here, a suitable NN architecture is evolved by using a GA through proper interactions with the environment. In (Hui and Pratihari, 2004) the researchers have provided a comprehensive review on

Corresponding Author: Genci Capi, Department of Electrical and Electronic Systems Engineering, Faculty of Engineering, University of Toyama Gofuku Campus, 3190 Gofuku, Toyama, 930-8555, Japan
Tel/Fax: 0081-(0)76-445-6745

various aspects of evolutionary robotics. After realizing the advantages of GA-NN approach, Hui and Pratihari (2004) studied its performance for solving the navigation problems of a car-like robot through computer simulations.

Unlike previous works, in the experiments reported here, we consider evolution of neural controllers for robot navigation in unstructured environments. In our method, we convert the captured image in a binary one, which after the partition is used as the input of the neural controller.

Most of the above researchers tested their motion planning algorithms through computer simulations. However, more recently, the importance of conducting experiments using the real robot to test the performance of motion planner has been felt by various investigators (Thrun, 1996; Kim *et al.*, 2004; Bruce and Veloso, 2002; Akbarzadeh *et al.*, 2000). In the soft computing-based navigation schemes, training is given to off-line and the performance of optimal motion planner is tested on a real robot. However, simulation of visual sensors is really difficult. Due to the lighting conditions, the evolved neural controllers in simulation often have a poor performance when they are transferred in real robot. In order to overcome this, we evolved the neural controllers online using real robots. We utilized the Tate Rob, a robotic platform developed in our laboratory.

MATERIALS AND METHODS

TateRob platform: In the experiments presented in this study, we use the TateRob Fig. 1. The key performance specifications of TateRob are:

- Total mass 20.5 kg
- Length 0.5×0.5×0.4 m
- Maximum speed 3 m sec⁻¹



Fig. 1: TateRob platform

Due to the desired tasks and the environment in which the robot has been designed to operate, vision was chosen as the primary sensor for navigation. It is also expected that in the environments with sufficient features and color information TateRob can make accurate vision-based odometry estimation. Therefore, the vision system consists of a stereo camera. By changing the lens, the peripheral and foveal vision can be realized.

In order to operate in environments with insufficient lighting conditions, the TateRob is equipped with Laser range sensor. The SICK LMS 200 used in our experiments has a field-of-view of 180° and returns 181 distance readings (one per degree). The maximum error is +/-3 cm per 80 m.

The robot is actuated by two 24 V batteries. One of the batteries actuates the motors and the other one the main CPU and the sensors. A PC/104 stack running the Linux operating system provides the software interface to record and process all the sensor information in real time. The robot can communicate with the operator by wireless LAN in a maximum distance of 100 m.

The robot can operate in the following different modes:

- The robot can move in the environment controlled by a joystick:
 - Directly connected with the robot
 - Connected with operator PC and remotely controlling the robot
- The operator can control the robot remotely by sending commands like move forward, rotate right or left
- The robot can operate autonomously by processing the sensors data in the onboard computer

Neural controller: In order to calculate the sensory input of the neural controller, the captured color image is converted to a binary image, as shown in Fig. 2. The input of the visual processing module is the image frame captured from the robot's camera. The image first is converted to a grey scale and then to a binary image Fig. 3. In our implementation the size of captured image is 240×320 pixels.

We implemented a feed-forward neural controller Fig. 4 with 20, 4 and 2 units in the input, hidden and output layers, respectively. A set of visual neurons, arranged on a 2×10 grid, receive information about the grey level of the corresponding pixels in the image provided by the camera of the robot. Each input unit covers an area of 50×32 pixels in the image. In order to increase the image processing speed, only the half-bottom of the captured image is processed (100×320 pixels).

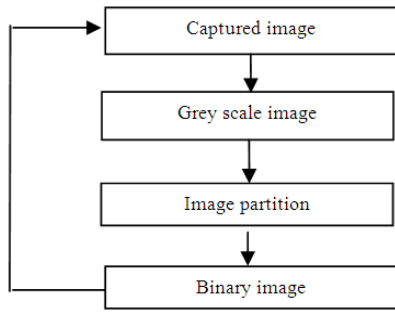


Fig. 2: Image processing algorithm

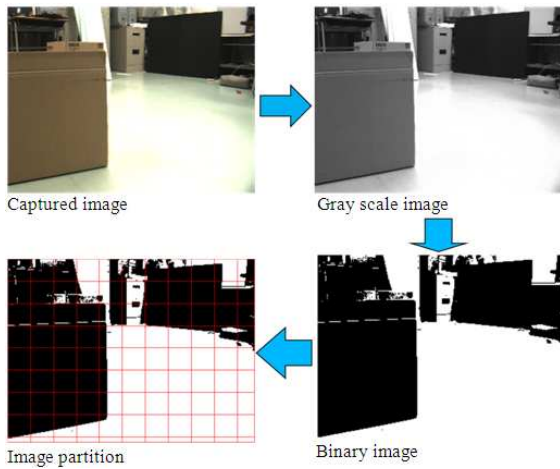


Fig. 3: Sample of image processing

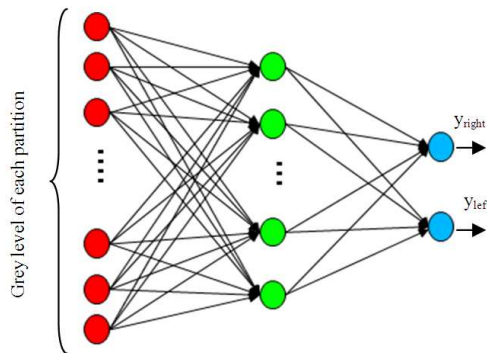


Fig. 4: Neural controller

The activation of input units, scaled between 0 and 1, is given by the average grey level of all pixels within the partition. The hidden and output units use sigmoid activation function:

$$y_i = \frac{1}{1 + e^{-x_i}} \quad (1)$$

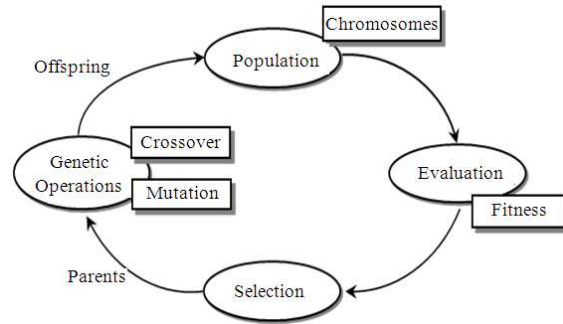


Fig. 5: GA cycle

where the incoming activation for node i is:

$$x_i = \sum_j w_{ji} y_j \quad (2)$$

and j ranges over nodes with weights into node i .

The output units directly control the right and left wheel angular velocities where 0 corresponds to no motion and 1 corresponds to full-speed forward rotation. The left and right wheel angular velocities, ω_{right} and ω_{left} , are calculated as:

$$\begin{aligned} \omega_{right} &= \omega_{max} * y_{right} \\ \omega_{left} &= \omega_{max} * y_{left} \end{aligned} \quad (3)$$

Where:

ω_{max} = The maximum angular velocity
 y_{right} and y_{left} = The neuron outputs

The maximum forward velocity is considered to be 0.5 m sec^{-1} .

Evolutionary algorithm: GA is a search algorithm based on the mechanics of natural selection and population genetics. The search mechanism is based on the interaction between individuals and the natural environment. GA comprises a set of individuals (the population) and a set of biologically inspired operators (the genetic operators). The individuals have genes, which are the potential solutions for the problem. The genetic operators are crossover and mutation. GA generates a sequence of populations by using genetic operators among individuals. Only the most suited individuals in a population can survive and generate offspring, thus transmitting their biological heredity to the new generation.

GA operates through a simple cycle of four stages, as shown in Fig. 5. Each cycle produces a new generation of possible solutions for a given problem. At

the first stage, an initial population of potential solutions is created as a starting point for the search. In the next stage, the performance (fitness) of each individual is evaluated with respect to the constraints imposed by the problem. Based on each individual's fitness, a selection mechanism chooses "parents" for the crossover and mutation operators.

The crossover operator takes two chromosomes and swaps part of their genetic information to produce new chromosomes. The mutation operator introduces new genetic structures in the population by randomly modifying some of the genes, helping the search algorithm to escape from local minima's traps.

The offspring's produced by the genetic manipulation process are the next populations to be evaluated. GA can replace either a whole population or its less fitted members only. The creation-evaluation-selection-manipulation cycle repeats until a satisfactory solution to the problem is found or some other termination criteria are met.

For any evolutionary computation technique, a chromosome representation is needed to describe each individual in the population. In our implementation, the genome of every individual of the population encodes the weight connections of the neural controller as a real number. The genome length is 88 and the connection weights range from -5 to 5. Each individual of the population controls the TateRob during a lifetime of 50 sec (50 m sec×1000 time steps). At the end, the fitness function is calculated. The fitness function selects robots for their ability to move among obstacles as long as possible for the duration of the life of the individual. Therefore, the fitness is the distance traveled by the robot. Every-time that the robot hits an obstacle (detected by the laser sensor), the robot moves backward and the remained lifetime is reduced by 100 steps.

In the first generation 60 neural controllers with randomly selected weight connections are generated. In the second generation the population size is reduced to 20, by selecting the best individuals of the first generation based on the fitness value. The evolution terminated after 9 generations.

RESULTS AND DISCUSSION

In order to evaluate the performance of evolved neural controllers, we developed an algorithmic navigation method where the right and left wheel angular velocities are calculated based on the average grey level of pixels in the left and right visual field. The angular velocity of right wheel is considered inverse

proportional with the average gray level of bottom left half of visual field and vice-versa Fig. 6.

Figure 7a and b show the performance of the algorithmic method for two different maximum velocities, 0.5 and 0.3 m sec⁻¹, respectively. At the beginning, the robot avoids collision with an obstacle and moves fast in the environment Fig. 7a. Then, obstacles become visible in the right half of the visual field. However, due to the restriction in the bottom part of the visual field, the obstacle in the front of the robot (middle upper part of the image) is not considered. Therefore, the small number of black pixels in the bottom right corner does not have a great effect on reducing the angular speed of the left wheel, making impossible for the robot to avoid hitting the obstacle.

When the robot was controlled using the algorithmic method, but the maximum velocity is reduced to 0.3 m sec⁻¹, the robot was able to navigate through the obstacles, as shown in Fig. 7b. Due to the low speed motion, the right and left angular velocities are updated in a shorter moving distance. Therefore, the robot trajectory is different compared with the previous experiment Fig. 8.

Figure 7c shows the performance of robot controlled by the best evolved neural controller. The maximum velocity is 0.5 m sec⁻¹. The robot avoids obstacles and outperformed the algorithmic method by moving smoothly among obstacles.

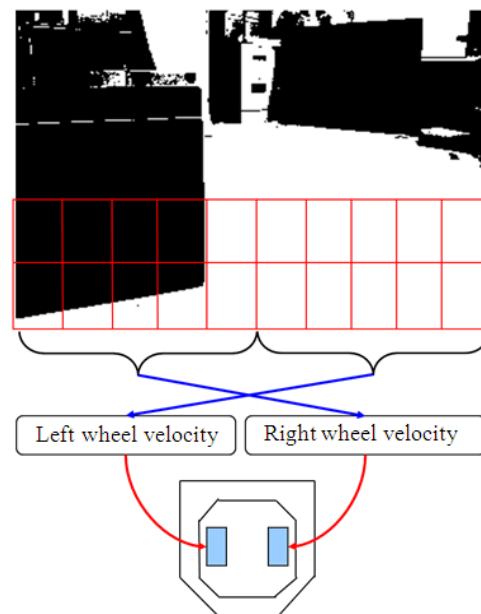


Fig. 6: Algorithmic method for robot navigation

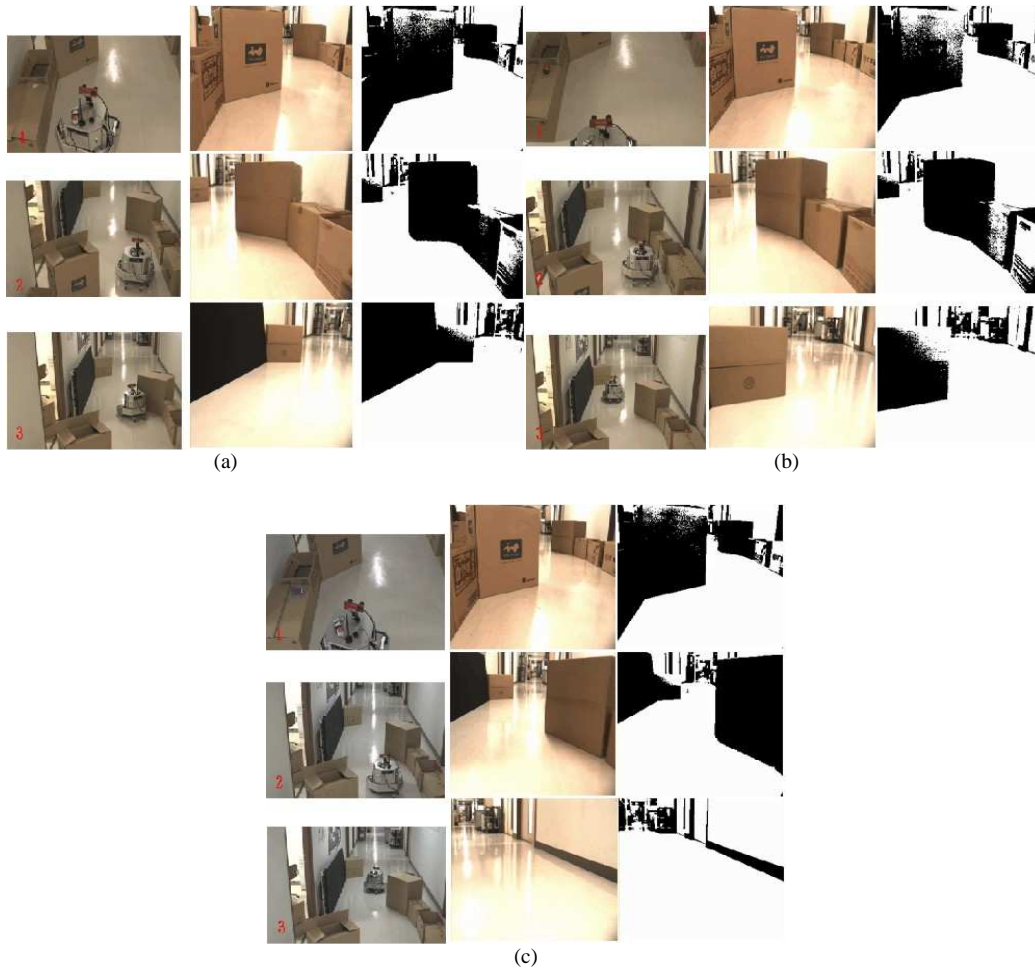


Fig. 7: Video capture of robot motion. (a) Algorithmic method with max. speed 0.5 m sec^{-1} ; (b) Algorithmic method with max. speed 0.3 m sec^{-1} ; (c) Neural Controller with max. speed 0.5 m sec^{-1}

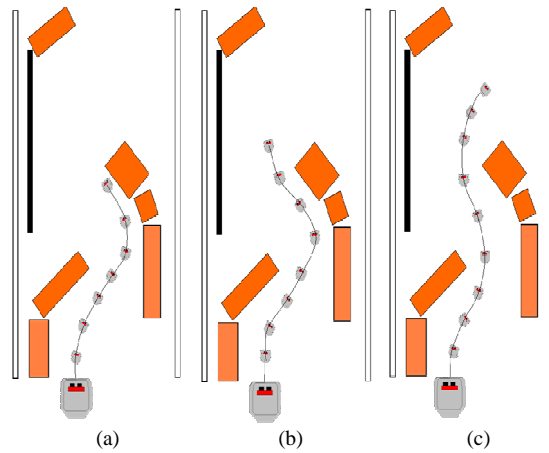


Fig. 8: TateRob trajectory. (a) Algorithmic method with max. speed 0.5 m sec^{-1} ; (b) Algorithmic method with max. speed 0.3 m sec^{-1} ; (c) Neural Controller with max. speed 0.5 m sec^{-1}

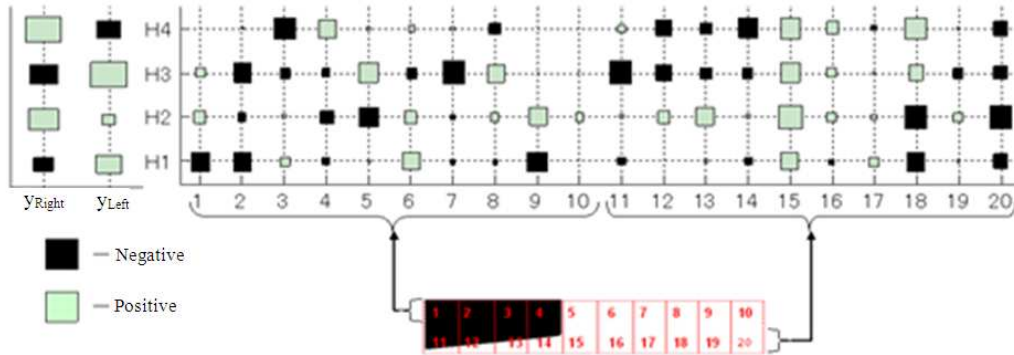


Fig. 9: Hilton diagram of weight connections

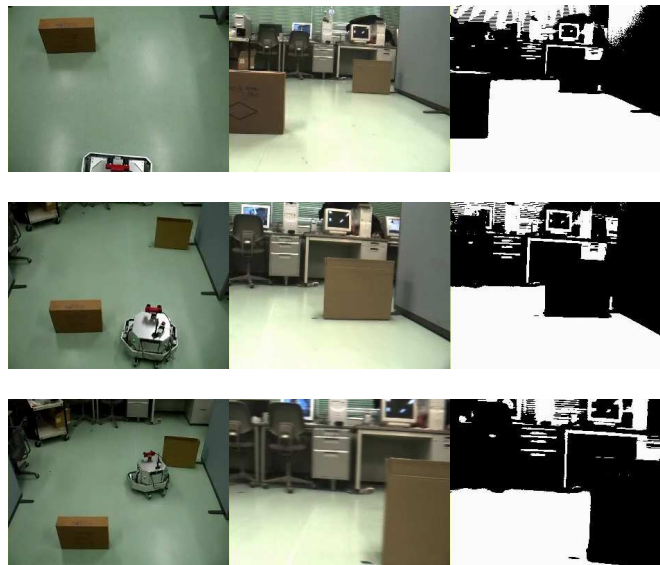


Fig. 10: Performance of neural controllers in different environment

The Hinton diagram of the weight connections Fig. 9 shows that input neurons positioned in the center of the visual field have positive connections with the hidden neurons. In addition, most of hidden neurons also have positive connections with output neurons that control the right and left wheel angular velocities. Therefore, the robot moves fast in the forward direction when there is no obstacle in the front. When an obstacle enters in the visual field in the left or right side, the respective wheel' angular velocity is reduced due to the negative connection weights and the robot avoids hitting the obstacles.

The performance of the evolved neural controllers was also tested in environments different from those in which they had been evolved Fig. 10. Figure 10 shows that evolved neural controllers generalize well in new environments. However, in certain specific

environments, the evolved neural controllers failed to perform well. For example, in some specific lighting conditions, the TateRob hit the obstacles during the navigation.

CONCLUSION

This study presented an evolutionary based method for robot navigation in human environments. Neural controllers were evolved online using the TateRob and the validity of the proposed method has been proved experimentally. The results show that the evolved neural network outperforms an algorithmic method for robot navigation. In the future, it will be interesting to investigate how the size of the visual field and the number of partitions influence the neural controller performance.

REFERENCES

- Akbarzadeh, K., E. Kumbla, E. Tunstel and M. Jamshidi, 2000. Soft computing for autonomous robotic systems. *Comput. Elect. Eng.*, 26: 5-32. DOI: 10.1016/S0045-7906(99)00027-0
- Bruce, J. and M. Veloso, 2002. Real-time randomized path planning for robot navigation. *Proceedings of the International Conference on Intelligent Robots and Systems*, Oct. 1-5, ACM Press, Lausanne, Switzerland, pp: 2383-2388. DOI: 10.1109/IRDS.2002.1041624
- Capi, G., 2007. Multiobjective evolution of neural controllers and task complexity. *IEEE Trans. Robot.*, 23: 1225-1234.
- Coombs, D. and K. Roberts, 1992. Bee-bot: Using peripheral optical flow to avoid obstacles. *Intell. Robots Comput. Vis.*, 1825: 714-721. DOI: 10.1117/12.131575
- Crisman, J.D., 1992. Color Region Tracking For Vehicle Guidance. *Active Vision*. MIT Press, Cambridge MA., ISBN: 0262023512, pp: 107-220.
- Dickmanns, E.D., B. Mysliwetz and T. Christians, 1990. An integrated spatio-temporal approach to automatic visual guidance of autonomous vehicles. *IEEE Trans. Syst. Man Cybernet.*, 20: 1273-1284. DOI: 10.1109/21.61200
- Goffe, W.L., G.D. Ferrier and J. Rogers, 1994. Global optimization of statistical functions with simulated annealing. *J. Econ.*, 60: 65-99. DOI: 10.1016/0304-4076(94)90038-8
- Goldberg, D.E., 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, MA. ISBN: 0201157675.
- Gu, C. and H. Hu, 2002. Neural predictive control for a car-like mobile robot. *Robot. Autonom. Syst.*, 39: 73-86. DOI: 10.1016/S0921-8890(02)00172-0
- Hebert, M., D. Pomerleau, A. Stentz and C. Thorpe, 1995. Computer vision for navigation: The CMU UGV project. *Proceedings of the Workshop on Vision for Robots*, Aug. 6-8, IEEE Computer Society Press, USA., pp: 97-102.
- Horswill, I.D., 1993. Polly: A vision based artificial agent. *Proceeding of the 11th International Conference on Artificial Intelligence (AI'93)*, AAAI, pp: 824-829. <http://www.aaai.org/Papers/AAAI/1993/AAAI93-123.pdf>
- Hui, N.B. and D.K. Pratihari, 2004. Neural network-based approaches vs. potential field approach for solving navigation problems of a car-like robot. *Mach. Intell. Robot Control*, 6: 29-60.
- Kim, J.H., D.H. Kim, Y.J. Kim and K.T. Seow, 2004. *Soccer Robotics*. Springer, Amsterdam, ISBN: 3642060064.
- Koza, J., 1992. *Genetic Programming*. The MIT Press, ISBN: 0262111705.
- Mondada, F. and D. Floreano, 1995. Evolution of neural control structures: some experiments on mobile robots. *Robot. Autonom. Syst.*, 16: 183-195. DOI: 10.1016/0921-8890(96)81008-6
- Pal, P.K. and A. Kar, 1996. Sonar-based mobile robot navigation through supervised learning on a neural net. *Autonom. Robots*, 3: 355-374. DOI: 10.1007/BF00240650
- Pratihari, D.K., 2003. Evolutionary robotics-a review. *Sadhana*, 28: 999-1003. DOI: 10.1007/BF02703810
- Rosenblatt, J. and C. Thorpe, 1995. Combining multiple goals in a behavior-based architecture. *Proceeding of the IEEE International Conference on Intelligent Robots and Systems*, Aug. 5-9, ACM Press, Pittsburgh, PA., pp: 136-141. DOI: 10.1109/IROS.1995.525787.
- Santos-Victor, J. and G. Sandini, 1996. Uncalibrated obstacle detection using normal flow. *Mach. Vis. Appl.*, 9: 130-137. DOI: 10.1007/BF01216818
- Santos-Victor, J., G. Sandini, F. Curotto and S. Garibaldi, 1995. Divergent stereo in autonomous navigation: From bees to robots. *Int. J. Comput. Vis.*, 14: 159-177. DOI: 10.1007/BF01418981
- Thrun, S., 1996. An approach to learning mobile robot navigation. *Robot. Autonom. Syst.*, 15: 301-309. DOI: 10.1016/0921-8890(95)00022-8
- Turk, M.A., D.G. Morgenthaler, K.D. Gremban and M. Marra, 1988. VITS-a vision system for autonomous land vehicle navigation. *IEEE Trans. Patt. Anal. Mach. Intell.*, 10: 342-361. DOI: 10.1109/34.3899
- Yamada, S., 2005. Evolutionary behavior learning for action based environment modeling by a mobile robot. *Applied Soft Comput.*, 5: 245-257. DOI: 10.1016/j.asoc.2004.07.004
- Yang, S.X. and M. Meng, 2000. An efficient neural network approach to dynamic robot motion planning. *Neural Network*, 13: 143-148. DOI: 10.1016/S0893-6080(99)00103-3
- Zeng, N. and J.D. Crisman, 1995. Categorical color projection for robot road following. *Proceeding of the IEEE International Conference on Robotics and Automation*, May 21-27, ACM Press, Nagoya, Japan, pp: 1080-1085.