Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 76 (2015) 74 – 79

Procedia Computer Science

2015 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS 2015)

Neural Network based Guide Robot Navigation: An Evolutionary Approach

G. Capi*, Member, IEEE, S. Kaneko, B. Hua

G. Capi and B. Hua are with the University of Toyama, Gofuku 3910, Japan
S. Kaneko is with the Electrical Engineering Department, National Institute of Technology, Toyama College, 13 Hongo-machi, Toyama, Japan

Abstract

Abstract— Intelligent robot navigation in urban environments is still a challenge. In this paper we test if it is possible to train neural networks to control the robot to reach the target location in urban dynamic environments. The robot has to rely on GPS and compass sensor to navigate from the starting point to the goal location in an environment with moving obstacles. We compare the performance of three neural architectures in different environments settings. The results show that neural controller with separated hidden neurons has a fast response to sensory input. The performance of evolved neural controllers is also tested in real robot navigation. In addition to the neural network based navigation, the robot has also to switch between other navigation algorithms to reach the target location in the university campus.

Peer-review under responsibility of organizing committee of the 2015 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS 2015)

Keywords:

1. INTRODUCTION

There are many challenges for robot navigating in urban environments because these environments are highly unstructured and have different characteristics. Therefore, the robot has to rely on different sensors and switch between different navigation algorithms for a safe reach of the target location. Lidoris et al. [6] presents a robot

* G. Capi and Hua Bin are with the University of Toyama, Gofuku 3910, Japan (phone: 81-76-445-6745; fax: 81-76-445-6745; e-mail: capi@ eng.u-toyama.ac.jp).
E-mail capi@ eng.u-toyama.ac.jp).

Peer-review under responsibility of organizing committee of the 2015 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS 2015)
doi:10.1016/j.procs.2015.12.279

navigation method in unknown urban environments relying only on the information extracted through the interaction with passers-by and its local perception capabilities. Thrapp et al. [7] presents a robust localization for the robot in outdoor environments using GPS and extended Kalman filter. The localization error was reduced up to 0.4m. In our previous work, we presented several navigation algorithms for the guide robot ([8]). The robot utilizes the Laser Range Finder (LRF), Camera, GPS and compass sensors to navigate in environments with different characteristics and reach the target location. The robot was able to select the appropriate navigation algorithm based on the environment conditions. In addition, we evolved neural networks for robot navigation in open squares environments. The neural controllers were evolved in simple environments without fixed or moving obstacles.

Vision based robot navigation in urban environments has been widely used. Several approaches use the teach and replay paradigm ([9], [10]). In the teaching stage, the robot moves manually through a desired route, and then in the replay stage the robot moves autonomously replicating the teaching route. However, if the robot deviates from the target route due to an unpredicted obstacle, it is difficult to get the robot back on the target route. Pedestrian lane navigation using visual sensor has been also widely investigated with very promising results. The vanishing point method recognizes the road in the image. Most of the approaches use consensus direction or local textures or image edges to determine the most probable vanishing point ([11], [12], [13]). Siagian et al [14] presented a vanishing point detection algorithm that uses long and robust contour segments. Most of visual robot navigation in urban environments focus on pedestrian lane detection and following. However, in open square environments, the robot cannot rely on the visual information to follow a specific route to the target location. The robot has to rely on other sensors in order to estimate the heading and the moving direction.

In this paper, we compare the performance of different neural controllers for robot navigation in dynamic environments. The robot has to rely only on LRF, GPS, compass and camera sensors. The controllers are neural networks with different structures. The neural controllers are evolved in static and dynamic environments with moving and stationary obstacles. The robot controlled by the evolved neural networks show different behaviors. The best evolved neural controllers in simulated environments are also tested in the real hardware of the Guide Robot showing a good performance. In addition, we improve the navigation algorithm in narrow pedestrians. The robot adjusts its speed based on the walking speed of the user.

The paper is organized as follows. In Section II, the task and environment are presented. The neural architectures and the evolutionary algorithm are discussed in Section III. Simulation and experimental results are presented in Section IV before concluding in Section V.

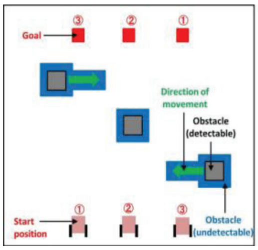
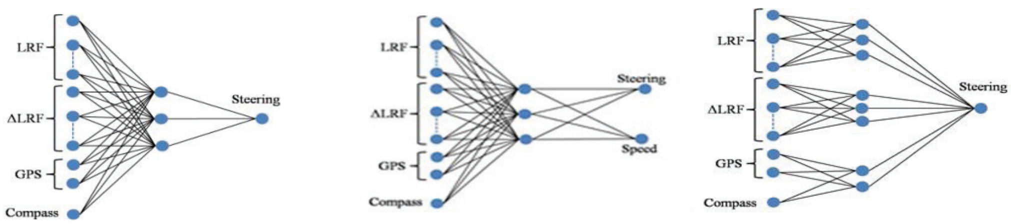


Figure 1. Neural Architectures.



(a) NN1 (b) NN2 (c) NN3

Figure 2. Neural Architectures

2. TASK AND ENVIRONMENT

We consider open square environments where the robot has to reach a target location determined by its GPS data. In such environments, the robot can utilize only the GPS and compass sensors to navigate toward the target location. The LRF sensor is utilized to avoid hitting obstacles. In simulation, the robot navigates in a square environment of 30m x 30m (Fig. 1). The target location is considered as a square of 1m x 1m. The environment is dynamic with stationary and moving obstacles. The obstacles size are 1m x 1m. The moving obstacle speed is randomly selected in the range of 0.5m/s to 0.7m/s. The initial robot locations are IL1(10m,3m), IL2 (15m,3m), IL3 (20m,3m). The center coordinates of three target locations are TL1 (10m,27m), TL2 (10m,27m), TL3 (10m,27m).

3. EVOLUTION OF NEURAL CONTROLLERS

Three feedforward neural network architectures are shown in Fig. 2. The input of the neural controllers are 7 neurons encoding the LRF data (from 20 degree to 160 degree every 20 degrees), 7 neurons encoding the difference with one step before LRF data readings (Fig. 3), 2 neurons encoding the difference between the robot and target location GPS data, and 1 neuron encoding the data of compass sensor. The NN1, NN2, NN3 has 3, 8, and 3 neurons in the hidden layer, respectively. In the NN1 and NN3 all the input units are fully connected with the hidden units. The structure of NN2 is different. The hidden units are divided in three groups of 3, 3 and 2 units. Each group of hidden units is connected only with specific group of input units, as shown in Fig. 2(b). The output unit of the NN1 and NN2 controls the steering of the robot while the moving speed is set to be constant (0.5 m/s). The NN3 has 2 output units that control the steering and speed of the guide robot.

$$y_i = \frac{1}{1+e^{-x_i}} \tag{1}$$

where the incoming activation for node i is:

$$x_i = \sum_j w_{ji} y_j \tag{2}$$

and j ranges over nodes with weights into node i.

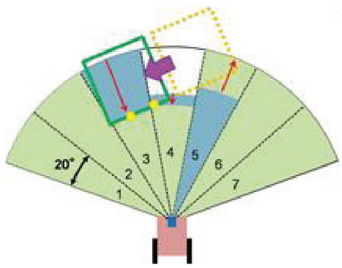
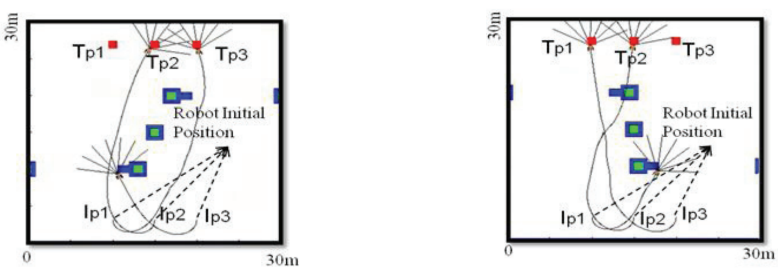


Figure 3. LRF data used as input of neural controllers.



(a) (b)

Figure 4. NN1 performance in dynamic environments.

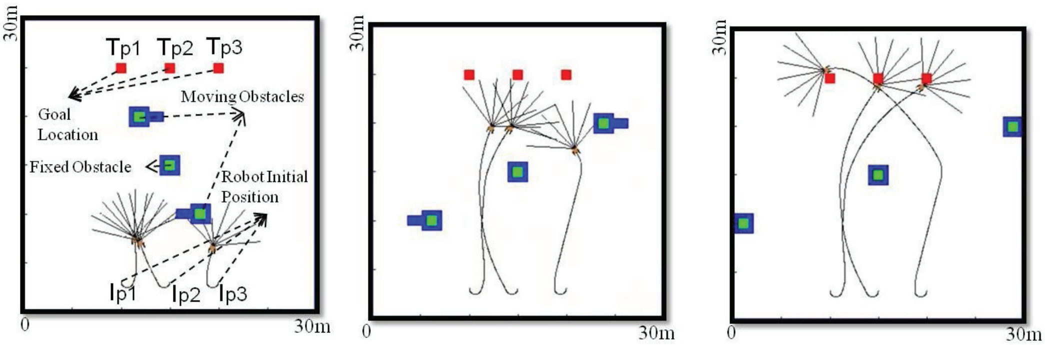


Figura 5. NN2 performance in dynamic environments.

The weight connections of the neural controllers are encoded in the genome of the genetic algorithm. The genome lengths of the NN1, NN2, and NN3 are 54, 56 and 57, respectively. The size of the hidden layer for each network were chosen specifically such that all network architectures approximately have the same number of weights, thereby justifying a fair comparison across all models. The GA searches for the weight connections in the range of -5 to 5.

The fitness function is defined such as to minimize the distance between the robot and the goal location in the shortest possible time, as follows:

$$f = \sum_1^3 \sqrt{(x_G - x_R)^2 + (y_G - y_R)^2} + N_{steps} \quad (3)$$

where xT and xR are the target and robot x coordinates, yT and yR are the target and robot y coordinates and Nsteps is the number of steps required to reach the goal location. Each neural network controls the robot for 3 different initial and goal locations as IL1 to TL3, IL2 to TL2, IL3 to TL1.

Table 1. MPGA functions and parameters

Function name	Parameter
Number of subpopulations	4
Initial nr. of individuals/ subpop	100, 50, 30, 20
Crossover probability	0.8
Mutation rate / subpopulation	0.1, 0.03, 0.01, 0.003
Isolation time	20 generations
Migration rate	10%
Termination	50 generation

A real value parallel GA is employed for the evolution of neural controllers. In the parallel GA the population is divided in subpopulations that compete and cooperate with each other. In [15], it is shown that parallel GA outperformed single population GA in terms of quality of the solution. The MPGA functions and parameters are presented in Table I. Each subpopulation uses different mutation rates (Table I). The crossover operator (discrete crossover) and crossover probability are considered the same for all subpopulations.

4. RESULTS

In this section we present the robot performance in simulated environment. The fitness reached by the neural controllers during the course of evolution can serve as a useful first comparison among three neural controllers. However, the evolved neural controllers performance is also tested in environments with different settings. In the first environment where there is no static or moving obstacle, all the 3 neural controllers managed to reach the target locations. Fig. 4 shows the performance of NN1 in the environments with moving and static obstacles. The robot starting its motion from the Ip3 and moving toward Tp1 location hits the obstacle that moves from right to left. It is nearly the same situation with the environment in Fig. 4(b) where the robot hits the moving obstacle. In both environments, the robot hits the obstacle that moves near to the robot initial locations. Therefore, the robot was unable to response fast to the changes in the LRF sensor data.

The performance of NN2 and NN3 in dynamic environments are shown in Fig. 5 and Fig. 6, respectively. The NN2 response to moving obstacles is very fast.

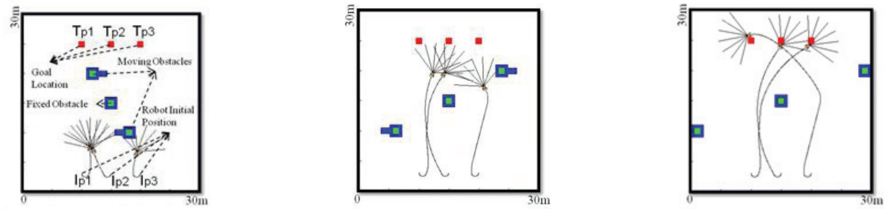


Figure 6. NN3 performance in dynamic environments.

Especially when the robot moving from Ip3 to Tp1 get near to the obstacle, the robot utilizes the LRF data to make a very fast right turn. A fast turn can also be seen by the robot starting from Ip1 when the robot rotates left and moves in front of the moving obstacle.

NN3 also managed to avoid obstacles and reach the target location (Fig. 6). However, the robot navigation strategy was different. Initially, the robot starting its motion from Ip3 and moving toward Tp1 location moves slowly in order to hit the obstacle moving from right to left. The robot follows a long route to reach the target. This is because the robot obstacle avoidance strategy is by keeping a enough distance to them.

The evolved neural controllers are also tested in the real hardware of the guide robot (Fig. 7). Before reaching the open space the robot has to navigate in environments with different characteristics. For a detailed explanation of the navigation algorithms refer to [8]. However, we improved the robot navigation algorithm. The main improvement is that the robot changes its speed based on the walking speed of the user.

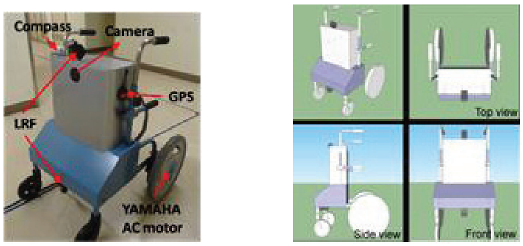


Figure 7. Guide robot.

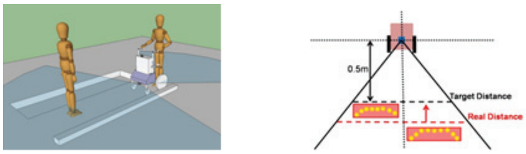


Figure 8. Speed adaptation based on LRF data.

The user's right and left leg are recognized by the LRF placed in the bottom part of the robot (Fig. 8). The real distance is calculated as follows:

$$Rd = {(D_r - D_l)}/{2} \qquad (4)$$

where Dr and Dl are the distance to the right and left leg respectively. The target distance between the user and the robot is set to 0.5m. A PD controller is implemented to control the robot to keep the target distance with the user.

Figure 9. Guide robot navigation in urban environment.

The video capture of the experiment is shown in Fig. 9. The robot stops moving when the user stops walking. The robot utilizes the face recognition algorithm and LRF data to stop in front of a walking person in a narrow pedestrian lane. In open square, the robot controlled by the evolved neural controller navigates to the target location avoiding hitting walking person.

5. CONCLUSION

In this paper, we compared the performance of different neural controllers for robot navigation in urban environments. The results showed that the neural network with hidden units connected to specific input units outperformed the fully connected input-hidden networks. Due to the specific connection between the LRF hidden and steering unit, the robot responded fast to the in front moving obstacles. The robot was able to navigate toward the target location avoiding hitting moving and static obstacles. The neural controllers were also tested in the real hardware of the Guide robot showing a good performance.

REFERENCES

[1] S. Thrun, M. Beetz, M. Bennewitz, W. Burgard, A.B. Creemers, F. Dellaert, D. Fox, D. Hahnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz. ``Probabilistic Algorithms and the Interactive Museum Tour-Guide Robot Minerva''. International Journal of Robotics Research, 19(11): 972-999, 2000.

[2] W. Burgard. A.B. Cremers. D. Fox, D. Hahnel. G. Lakemeyer. D. Schulz, W. Steiner. and S. Thrun, The interactive museum tour-guide robot. Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98).

[3] A. Yamazaki, K. Yamazaki, T. Ohyama, Y. Kobayashi, Y. Kuno, A techno-sociological solution for designing a museum guide robot: regarding choosing an appropriate visitor, Proceeding HRI '12 Proceedings of the seventh annual ACM/IEEE Int. Conf. on Human-Robot Interaction, pp. 309-316.

[4] S.X. Yang, and M. Meng, 2000. An efficient neural network approach to dynamic robot motion planning. Neural Network, 13: 143-148.

[5] G. Capi, "A Vision based Approach for Intelligent Robot Navigation", International Journal of Intelligent Systems Technologies and Applications (IJISTA), Vol. 9, Issue 2, July 2010.

[6] G. Lidoris, F. Rohrmuller, D. Wollherr, and M. Buss, "The Autonomous City Explorer Project Mobile robot navigation in highly populated urban environments," 2009 IEEE Int. Conf. on Robotics and Automation, pp. 1416-1422, 2009.

[7] R. Thrapp, C. Westbrook and D. Subramanian: "Robust Localization Algorithms for an Autonomous Campus Tour Guide, "In Proceedings of 2001 IEEE Int. Conf. on Robotics and Automation, pp.2065-2071, 2001.

[8] G. Capi, M. Kitani, K. Ueki, Guide robot intelligent navigation in urban environments, Advanced Robotics, Volume 28, Issue 15, 1043-1053, 2014.

[9] Z. Chen and S. Birhfield, "Quantitative vision-based mobile robot navigation," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, May 2006, pp. 2686 – 2692.

[10] C.-K. Chang, C. Siagian, and L. Itti, "Mobile robot vision navigation & localization using gist and saliency," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2010.

[11] H. Kong, J.-Y. Audibert, and J. Ponce, "General road detection from a single image," *IEEE Transactions on Image Processing*, vol. 19, no. 8, pp. 2211 – 2220, August 2010.

[12] O. Miksik, "Rapid vanishing point estimation for general road etection," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, May 2012.

[13] C.-K. Chang, C. Siagian, and L. Itti, "Mobile robot monocular vision navigation based on road region and boundary estimation," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2012, bb, pp. 1043–1050.

[14] C. Siagian, C. Chang, and L. Itti, "Mobile robot navigation system in outdoor pedestrian environment using vision-based road recognition," in Proc. IEEE International Conference on Robotics and Automation (ICRA), May 2013.

[15] G. Capi and K. Doya, "Evolution of recurrent neural controllers using an extended parallel genetic algorithm", Robotics and Autonomous Systems, Vol. 52/2-3, pp. 148-159.