3D Vision-Based Autonomous Navigation System Using ANN and Kinect Sensor

Daniel Sales, Diogo Correa, Fernando S. Osório, and Denis F. Wolf

University of São Paulo, Mobile Robotics Lab, Av. Trabalhador São-Carlense, 400, P.O. Box 668, São Carlos, Brazil
{dsales,diogosoc,fosorio,denis}@icmc.usp.br

Abstract. In this paper, we present an autonomous navigation system based on a finite state machine (FSM) learned by an artificial neural network (ANN) in an indoor navigation task. This system uses a kinect as the only sensor. In the first step, the ANN is trained to recognize the different specific environment configurations, identifying the different environment situations (states) based on the kinect detections. Then, a specific sequence of states and actions is generated for any route defined by the user, configuring a path in a topological like map. So, the robot becomes able to autonomously navigate through this environment, reaching the destination after going through a sequence of specific environment places, each place being identified by its local properties, as for example, straight path, path turning to left, path turning to right, bifurcations and path intersections. The experiments were performed with a Pioneer P3-AT robot equipped with a kinect sensor in order to validate and evaluate this approach. The proposed method demonstrated to be a promising approach to autonomous mobile robots navigation.

Keywords: Mobile Robotics, Autonomous Navigation, Kinect, Artificial Neural Networks, Finite State Machine.

1 Introduction

AI techniques implementation on Autonomous Mobile Robots and Intelligent Vehicles has an important role on international scientific community [9][13][14]. One of the most desirable features for a mobile robot is the autonomous navigation capability. There are many known relevant works on this research field, as for example the Darpa Challenges (2004 and 2005 editions on desert and 2007 in Urban environment)[7][8] and ELROB [15][16].

Autonomous mobile robots usually perform three main tasks: localization, mapping/planning and navigation [17]. Localization task is related to estimate robot´s position in a well-known environment using its sensorial data. Mapping consists on creating an environment representation model, based on robot´s localization and sensorial data. Navigation is the capability to process environment information and act, moving safely through this environment.

C. Jayne, S. Yue, and L. Iliadis (Eds.): EANN 2012, CCIS 311, pp. 305–314, 2012.

In order to an autonomously navigate into structured environments composed by streets or corridors, the robot must know its approximate position, the environment map and the path to be performed (source/destination). So, navigation in this environment consists on following a well-defined path, considering the available navigation area.

The main focus of this work is the implementation of a Topological Autonomous Navigation System able to recognize specific features in a path on indoor environments (composed by corridors and intersections). This navigation system is intended for indoor service robots in several different tasks, from the simplest ones as carrying objects until critical ones as patrolling. The implemented system for these applications must be easy to configure and use. It must be also robust allowing the robot to both navigate and detect possible abnormalities.

The proposed approach does not require a well-defined environment map, just a sketch representing the main elements, resulting in a simple path sight. Furthermore, this approach does not require an accurate robot´s pose estimation. The main goal is to make the robot navigate in an indoor structured environment deciding when and how to proceed straight, left or right, even when these three possibilities are detected simultaneously (intersections).

The topological approach uses an ANN [19] to classify sensor data and a FSM to represent the steps sequence for each chosen path. The ANN learns all possible states, and a FSM generator is responsible to convert any possible path into a sequence of states. This way, the system combines this deliberative topological behavior with a simple reactive control for a safe navigation.

The next topics of this paper are organized as follows: section 2 presents some realted works; section 3 presents the techniques and resources used for state detection; section 4 presents the experimental results; section 5 presents the conclusion and possible future works.

2 Related Works

Several approaches have been used for navigation, using many different sensors (for example laser, sonar, GPS, IMU, compass) singly or fused [9][17][18]. One of the most used approaches recently is the Vision-Based navigation [20]. This method uses video cameras as the main sensor. Cameras are very suitable for navigation and obstacle avoiding tasks due to its low weight and energy consumption [1]. Furthermore, one single image can provide many different types of information about the environment simultaneously. It is also possible to reduce costs by using cameras rather than other types of sensors [2]. The Vision-based approach implementation is already usual in navigation systems for structured or semi-structured environments [3][7][9][10][11]. These systems classify the image, with track segmentation for safe navigable area identification.

Although these works present good results, the scope for a conventional camera is restricted, and many implementations demand camera data fusion with laser sensors (Sick Lidars, IBEO, Velodyne), radars or even special vision systems such omnidirectional cameras [7][8][9][18]. This fusion becomes necessary specially when depth information is needed. This kind of information is not originally provided by a conventional camera.

Is worth noting that fusion-based approaches are usually expensive, so the use of Kinect sensor can lead to lower cost solutions. Kinect is a low-cost 3D sensor developed by Microsoft for XBOX 360 videogame which allows the player to use its own body as the controller in games. It is composed by a RGB camera associated to an infrared transmitter and receiver allowing depth estimation of the environment elements. Since its sensorial advantages were found out, many independent researches were being held in order to explore this device features in applications from health to robotics [25]. The main advantage of this sensor is the capability of depth maps construction, allowing a very accurate distance estimation for the obstacles detected in front of the robot.

In order to implement an autonomous navigation system, purely reactive approaches are not suitable. Immediate reaction to sensor captured data is not enough to ensure a correct robot control in a more complex path. A more robust system must be developed, providing sequence and context information that are absent in purely reactive models.

In Robotics, FSM-based approaches [5] are widely used. FSMs are useful because the system can be easily described as a sequence of states (context changes). Inputs (sensors data) are considered to allow state changes and to define the adequate reaction for each state (motor action). The proposed system in this work is based on this idea, so the path is described as a FSM and the current state observed through captured sensor data processing.

The use of a Machine Learning technique such ANN is a very interesting way of process sensorial data [6]. ANNs are robust to noise and imprecision on input data. They are also able to detect states and transitions between these states and very efficient to generalize knowledge. This way, this method is very useful for features detection e state recognizing.

The association of ANNs to FSMs has been researched since the 90s [21][22][23][24], when the ANN models were developed and improved, occuping an important place on AI and Machine Learning Researches. Recently, some works were developed using the association of these techniques to robotics problems, from car parking [18] until robots and vehicles navigation in indoor and outdoor environments [4]. These applications have two main problems: the high cost of its main sensor (laser) and the low amount of environment information (bidimensional detection only, with a depth information of a planar cut).

Thus, the proposed solution with kinect sensor shows extremely lower costs in addition to a more complete and accurate environment information, since tridimensional detection is performed simultaneously to conventional image capture.

3 System Overview

The proposed system is composed by three main steps. The first one is the ANN training in order to recognize all possible states (features) using previously collected environment data. The second one is the FSM Generation for any chosen path.

The third step is the autonomous navigation combining this deliberative control (topological path planning) with a reative control to keep the robot aligned. Figure 1 shows the system setup and navigation overview.

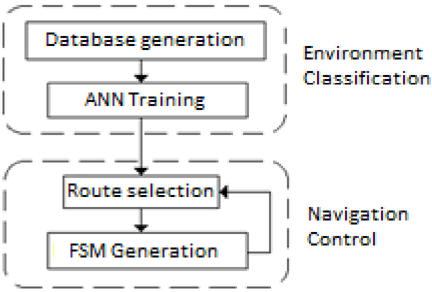

Fig. 1. System overview

3.1 ANN Training

This step consists on training the ANN to efficiently recognize all possible situations in a specific environment. ANN inputs are obtained from sensorial data, and the output is the detected class (state). For the environment used in this work, eight classes were defined (they are better described on next section). The dataset is created saving a "log" of collected input data moving the robot through the environment. The adopted learning algorithm is supervised, so a specialist must "manually" classify this data before ANN training, forming the input/output pairs. Several topologies were tested, and the best results were obtained from a feed-forward MLP, with 632 neurons on input layer, 316 neurons on hidden layer and 8 neurons on output layer. Figure 2 shows a simple graphic representation of this MLP.

Kinect captures are stored on a 640x480 matrix (one element per pixel), and each value is the distance between the sensor and the object represented in that pixel. For navigation purposes, there is no need to use all this 307200 points (this could also make ANN training impracticable due to the high ammount of input neurons). This way, an interest zone on Kinect´s capture is selected, in which the information is enough for state detection.

Kinect sensor has a very interesting property: each line of the depth matrix can be compared to a standard laser scan. This way, it is possible to say that a single kinect data capture provides the same data amount than 480 consecutive laser scans.This is another great advantage of using Kinect, high amount of available data for each single scan. An interesting way to minimize the effects of noisy inputs is to rely on data redundancy. This can be done processing various lines of the depth matrix for state detection. As mentioned earlier, it is impracticable to use all the 480 lines of information as inputs for the ANN, so an "interest area" must be defined.

For this work, 80 lines of depth matrix were selected, the correspondent pixels are represented at Figure 3. The mean of each column is calculated, so the result is one line only with the mean information, similar to a conventional laser scan. This line is used as the ANN input.

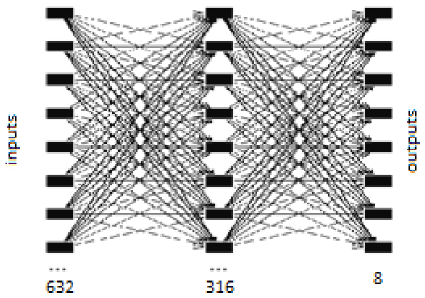

Fig. 2. ANN Topology

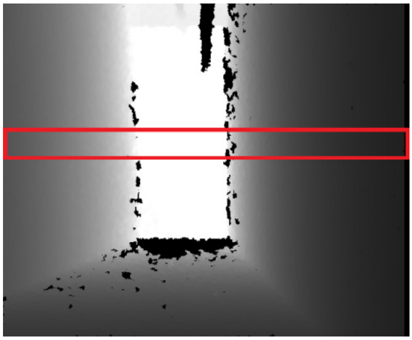

Fig. 3. Interest area on captured kinect´s frame

Each output neuron is associated to one of the 8 implemented states. The ANN is trained once, and must be able to work for any chosen path.

3.2 FSM Generation

Once the topological map for the environment is known, it is easy to establish a route between two points of this map, manually or with an algorithm.

Every route can be seen as a sequence of steps (states), so it is trivial to generate a FSM to represent this path. A single algorithm converts any possible path into a sequence of states and expected actions (also considering that one state can have more than one associated action). This way, after path selection the FSM is stored on system to be used by control unit.

3.3 Navigation Control

The hybrid control combines the deliberative control obtained from FSM-based topological navigation with a reactive control which must keep the robot into its expected route, avoiding collisions.

The Topological Navigation allows the robot to follow its planned path and also know its approximate position, but doesn´t controls the robot “into” every situation (state). When the robot is in a corridor (“straight state”), a reactive control is activated to keep the robot centered and aligned in this corridor. This is the main benefit of this hybrid control: take advantage of deliberative model for path control and simultaneously guarantee a safe navigation with reactive control.

For this implementation, it is assumed that robot´s initial position is always known, as the topological map also. The current position is estimated based on current state detection.

In this approach it isn´t necessary to estimate the robot´s exact position, it “knows” its approximate position based on current state and there is a reactive control responsible for keeping it safe.

Input data processing makes it possible to determine if the robot still at the same state (part of the path) or if a context change is needed. State transitions must occur only with the detected state is compatible with next expected state (this information is related to the stored FSM). Figure 4 shows the navigation control flowchart.

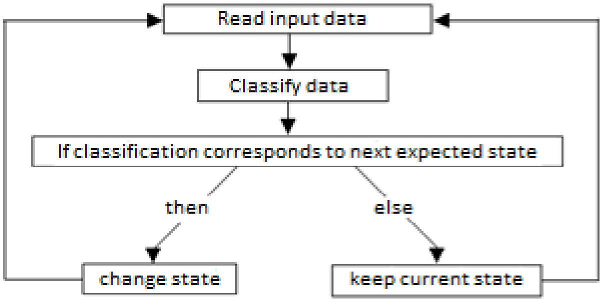

Fig. 4. Navigation control flowchart

4 Experiments and Results

Experiments were carried out in a real environment using a Pioneer P3-AT robot equipped with a dual-core computer for processing and Kinect as the only sensor.

The indoor environment used in the tests was represented with 8 states, illustrated at Figure 5. The created states are: “straight path”, “left turn”, “right turn”, “left and right turns”, “left turn and straight”, “right turn and straight”, “blocked” and “intersection”.

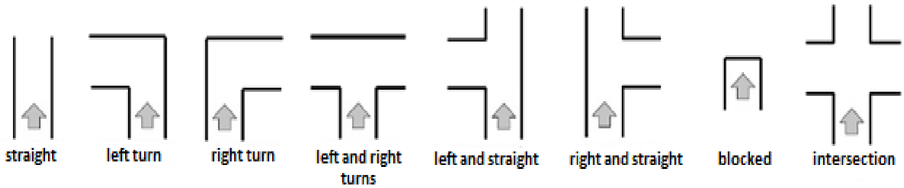

Fig. 5. Possible situaions on environment

The implemented ANN was designed and trained with Stuttgart Neural Network Simulator (SNNS), and then converted to C language with SNNS2C tool to be integrated with the robot´s control unit.

The ANN training database was collected after controlling the robot manually through the environment in several angles and positions, and then sliced into the 8 classes. At this step, about 180 examples were collected for each class, resulting in a database with 1421 input/output pairs, used for supervised learning.

The training algorithm used was Resilient Propagation (R-Prop). This algorithm is achieving great results for feed-forward networks in many applications due to its good training time and convergence. Training parameters were set up as follows: $\delta 0 = 0.1$, $\delta max = 50$, $\alpha = 5.0$ and number of epochs = 1000.

Five different topologies were tested, with different number of hidden layer neurons and number of hidden layers. The tests were held with 16, 80 and 316 neurons on hidden layer. These amounts were considered after empirical tests. The input layer is composed by 632 neurons (vector with the mean of the 80 lines of depth matrix), and output layer is composed by 8 neurons, 1 neuron for each possible class.

A great variation on training times was observed. With 1000 epochs, the training time for 632-16-8 net was 20 minutes, 2 hours for 632-80-8 net, and 8 hours for 632-316-8 approximately.

ANN validation was done with stratified 5-fold cross-validation method. This way, 5 train and test sets were generated, with 80-20 proportion on data (80% used for training and 20% for test, with same proportion of elements from the 8 class on the datasets). The networks with best results were 632-316-8 and 632-80-80-8, with 92,2% and 92% accuracy respectively, as can be observed at Table 1. The main difference between these networks is the training time: 8 hours for the first one versus 2 hours for the second one.

Table 1. ANN Accuracy after 1000 training cicles

ANN	Partition 1		Partition 2		Partition 3		Partition 4		Partition 5		Average	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
632-16-8	89	**88**	95	**87**	92	**81**	92	**86**	91	**85**	91,8	**85,4**
632-80-8	97	**89**	96	**91**	97	**90**	97	**92**	96	**89**	96,6	**90,2**
632-316-8	99	**92**	99	**93**	98	**92**	98	**91**	98	**93**	98,4	**92,2**
632-16-16-8	90	**80**	93	**85**	94	**85**	90	**86**	94	**86**	92,2	**84,4**
632-80-80-8	98	**92**	99	**91**	98	**91**	99	**92**	99	**94**	98,6	**92**

The confusion matrix for partition 2 in 632-316-8 net is shown next, on Figure 6. It is possible to note that error per class is close to zero, which means that very few classification errors occur.

133	1	5	0	0	0	0	0
0	156	0	0	0	0	0	0
4	0	158	0	0	0	0	0
0	0	0	97	0	0	0	0
0	0	0	0	170	0	1	0
0	0	0	0	0	171	0	0
0	0	0	0	0	0	96	0
0	0	0	0	0	0	0	136

Total: 1128 examples, 11 errors
Accuracy: 99%

a) train

33	0	2	0	0	0	0	0
0	37	2	0	0	0	0	0
1	2	37	1	0	0	0	0
0	0	1	24	0	0	0	0
0	0	0	0	39	0	3	1
0	0	0	0	0	41	0	2
0	0	1	0	0	0	24	0
0	0	1	0	1	1	0	31

Total: 285 examples, 19 errors
Accuracy: 93%

b) test

Fig. 6. Train confusion matrix (a) test confusion matrix (b)

Despite the excellent results considering the accuracy mean, a 100% safe navigation is guaranteed only if no errors occur in state detection. As a 100% accuracy is not achieved with the ANN, something must be done to guarantee that no unexpected state changes occur due to a wrong classification. This way, a “filter” was implemented, removing possible oscillations resulting from classification errors. This means that a state transition will occur only after some consecutive detections of the expected state, indicating confidence on transition detection.

After testing and validating the ANN, it was implemented on the real robot, recognizing features for a specific indoor environment. Figure 7 shows a successful classification of the “left turn and straight” state. The frame on the left is a graphic representation of the depth matrix, and the frame on the right is a “real” frame captured from kinect.

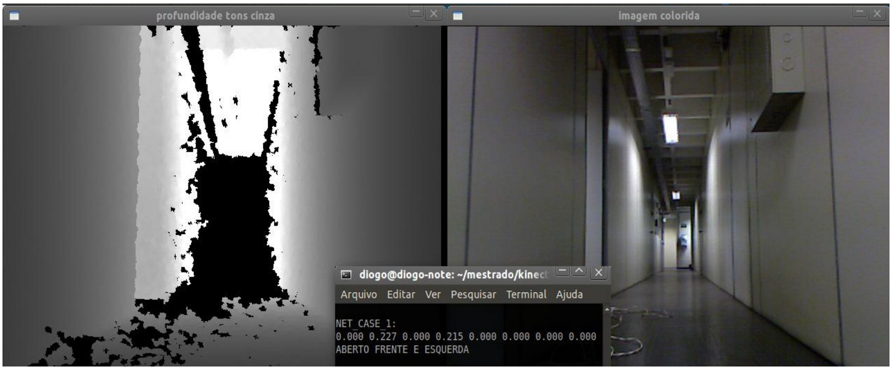

Fig. 7. State “left turn and straight” classification

5 Conclusion

Excellent results were achieved, with high accuracy level for the ANN individually, and 100% accuracy on navigation task after filter implementation on all experiments carried out. This shows that the association of ANN and FSM is a very suitable approach for autonomous robotic navigation.

This system is very flexible, as it can be re-trained to recognize new situations, settings and features, and also use and combine other sensorial systems.

Kinect was presented as a very suitable sensor for features detection on indoor environmets, allowing the development of robots with low-cost 3D vision-based navigation systems.

The main challenge for future works is to apply this same methodology for autonomous outdoor navigation, using other sensorial systems (also fusing sensors), as kinect is not designed for outdoor environments.

Acknowledgments. The authors acknowledge FAPESP and CNPq for their support to INCT-SEC (National Institute of Science and Technology - Critical Embedded Systems - Brazil), processes 573963/2008-9 and 08/57870-9 and financial support to authors (master´s grant).

References

1. Zingg, S., Scaramuzza, D., Weiss, S., Siegwart, R.: MAV Navigation through Indoor Corridors Using Optical Flow. In: IEEE International Conference on Robotics and Automation (ICRA 2010), Anchorage, Alaska (May 2010)
2. Scaramuzza, D., Siegwart, R.: Appearance Guided Monocular Omnidirectional Visual Odometry for Outdoor Ground Vehicles. IEEE Transactions on Robotics 24(5) (October 2008)
3. Shinzato, P.Y., Wolf, D.F.: Features Image Analysis for Road Following Algorithm Using Neural Networks (September 2010)
4. Sales, D., Osório, F., Wolf, D.: Topological Autonomous Navigation for Mobile Robots in Indoor Environments using ANN and FSM. In: Proceedings of the I Brazilian Conference on Critical Embedded Systems (CBSEC), São Carlos, Brazil (2011)
5. Hopcroft, J.E., Ullman, J.D.: Introduction to Automata Theory, Languages and Computation. Addison - Wesley (1979)
6. Marino, A., Parker, L., Antonelli, G., Caccavale, F.: Behavioral Control for Multi-Robot Perimeter Patrol: A Finite State Automata approach. In: ICRA (2009)
7. Thrun, S., et al.: Stanley: The Robot that Won the DARPA Grand Challenge. Journal of Field Robotics 23(9), 661–692 (2006)
8. Urmson, C., et al.: Autonomous driving in urban environments: Boss and the Urban Challenge. Journal of Field Robotics 25(8), 425–466 (August 2008); Special Issue on the 2007 DARPA Urban Challenge, Part I
9. Buehler, M., Iagnemma, K., Singh, S. (eds.): The 2005 DARPA Grand Challenge: The Great Robot Race (Springer Tracts in Advanced Robotics), 1st edn. Springer (October 2007)
10. Nefian, A.V., Bradski, G.R.: Detection of Drivable Corridors for Off-Road Autonomous Navigation. In: ICIP 2006: Proceedings of the IEEE International Conference on Image Processing, pp. 3025–3028 (2006)
11. Álvarez, J.M., López, A.M., Baldrich, R.: Illuminant Invariant Model-Based Road Segmentation. In: IEEE Intelligent Vehicles Symposium, Eindhoven, Netherlands (June 2008), `http://www.cvc.uab.es/adas/index.php?section=publications`
12. Bishop, C.M.: Neural Networks for Pattern Recognition. Oxford University Press, Oxford (1995)
13. Bishop, R.: Intelligent vehicle applications worldwide. IEEE Intelligent Systems and their Applications 15(1), 78–81 (2000)
14. Bishop, R.: A Survey of Intelligent Vehicle Applications Worldwide. In: Proceedings of the IEEE Intelligent Vehicles Symposium 2000, pp. 25–30 (2000)
15. Kuhnert, K.-D.: Software architecture of the Autonomous Mobile Outdoor Robot AMOR. In: Proceedings of the IEEE Intelligent Vehicles Symposium 2008, pp. 889–894 (2008)
16. Schilling, K.: Assistance Systems for the Control of Rovers. In: SICE Annual Conference, Tokyo (October 2008)
17. Wolf, D.F., Osório, F.S., Simões, E., Trindade Jr., O.: Robótica Inteligente: Da Simulação às Aplicações no Mundo Real [Tutorial]. In: de Leon F. de Carvalho, A.P., Kowaltowski, T. (org.) JAI: Jornada de Atualização em Informática da SBC, vol. 1, pp. 279–330. SBC - Editora da PUC. RJ, Rio de Janeiro (2009)
18. Goebl, M., Althoff, M., Buss, M., Farber, G., Hecker, F., Heissing, B., Kraus, S., Nagel, R., Leon, F.P., Rattei, F., Russ, M., Schweitzer, M., Thuy, M., Wang, C., Wuensche, H.J.: Design and capabilities of the Munich Cognitive Automobile. In: IEEE Intelligent Vehicles Symposium 2008, pp. 1101–1107 (2008)

19. Haykin, S.: Neural Networks: A Comprehensive Foundation. Prentice Hall PTR, Upper Saddle River (1998)
20. Sales, D., Shinzato, P., Pessin, G., Wolf, D., Osório, F.: Vision-based Autonomous Navigation System using ANN and FSM Control. In: Proceedings of the IEEE Latin American Robotics Symposium (LARS), São Bernardo do Campo, Brazil (2010)
21. Giles, C.L., Horne, B.G., Lin, T.: Learning a class of large finite state machines with a recurrent neural network. Neural Networks 8(9), 1359–1365 (1995)
22. Omlin, C.W., Giles, C.L.: Constructing Deterministic Finite-State Automata in Recurrent Neural Networks. Journal of the ACM 43(6), 937–972 (1996)
23. Frasconi, P., Gori, M., Maggini, M., Soda, G.: Representation of finite state automata in Recurrent Radial Basis Function networks. Machine Learning 23(1), 5–32 (1996)
24. Cleeremans, A., Servan-Schreiber, D., Mcclelland, J.L.: Finite State Automata and Simple Recurrent Networks. Neural Computation 1(3), 372–381 (1989)
25. KinectHacks Supporting the Kinect Hacking news and community, `http://www.kinecthacks.com/` (accessed in: February 10, 2012)