Vision-Based Autonomous Topological Navigation in Outdoor Environments

Daniel O. Sales, and Fernando S. Osório

Abstract— **In this paper, we present a proposal of an autonomous topological navigation system for structured outdoor environments. This system uses a camera as the main sensor, and an association of Finite State Machines (FSM) with Artificial Neural Networks (ANN) for path features detection. The environment is represented by a Topological map in which each state is related to a specific track shape. This way, the robot can become able to autonomously reach any destination, going through a sequence of these states, as for example straight path, right and left turns and intersections. The experiments were performed in a real structured outdoor environment in order to validate and evaluate this approach. The proposed system demonstrated to be a promising approach to autonomous vehicles navigation.**

I. Introduction

The development of robust autonomous intelligent systems for robotic applications is a very important research field. Several applications are related to robotics, from industry to military tasks.

The autonomous driving capability is one of the most desirable features for a mobile robot. Research related to this goal is being developed since the 1990´s, and groups such as NavLab have been presenting relevant results on autonomous vehicles navigation.

There are many other relevant and known works on autonomous robotics being developed worldwide. Some of them are powered by government initiatives as for example the DARPA Grand Challenges [7][8]. The first two editions (2004 and 2005) were held in desert, and 2007 edition in an urban environment.

Autonomous mobile robots usually perform three main tasks: localization, mapping and navigation control [17]. Mapping is the creation of an environment model using the sensorial data, representing the environment structure. Localization task must occur simultaneously to navigation control. It consists on estimate the robot´s position in a well-known environment, using its sensorial data. The Navigation task is therefore the ability to obtain enough information about the environment, process it, and act, moving safely through the navigable area.

In order to develop an intelligent autonomous vehicle able to navigate through environments composed by streets and highways, it must be assumed that the robot knows its approximate position, the environment map and the path to be followed (origin/destination). This way, navigation in this environment consists on following a well-defined path, considering the navigable areas.

This work focuses on this navigation task, and describes the proposal of a Vision-Based Perception System, able to recognize the navigable area of a structured outdoor environment (streets) processing the captured frame and classifying it into states which represent structural features of the environment, allowing the robot to detect its current context.

The proposed approach does not need a very detailed environment map (metric map), only a graph to represent the main elements, in a simpler path representation. Furthermore, accurate pose estimation is not necessary. The approximate robot´s position is enough to navigate. So, the main objective is to detect the current node in a Topological map, being useful to autonomously decide when and how to proceed in order to go straight, turn left or right, even if these three situations are detected simultaneously (at an intersection, for example).

The developed system uses Artificial Neural Networks (ANN) [19] in two steps: first to classify the frame obtained from camera, resulting in a navigability map and second to detect patterns on these navigability maps, representing the current context. For this second step, a Finite State Machine (FSM) is used to represent the chosen paths. The ANN is trained to recognize the different kinds of states at the environment and then a FSM generator converts any chosen path into a sequence of these states. Each state has its own reactive navigation control sub-system. This way, the system combines the high-level deliberative behavior with a reactive control, allowing a safe navigation control.

The next topics of this work are organized as follows: Section 2 presents some previous related works; Section 3 presents the Topological Navigation System overview, Section 4 presents the experiments and results and Section 5 presents the conclusion and possible future works.

II. 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 outdoor 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].

*Research supported by FAPESP.

The authors are with the Mobile Robotics Laboratory, in University of São Paulo – São Carlos, SP - Brazil (e-mail: dsales@ icmc.usp.br, fosorio@ gmail.com).

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, resulting in reactive models for navigation control. Works such as ALVINN [13] and RALPH [14] were the first to apply neural networks for this reactive control in outdoor environments. In [15], a reactive control was implemented by combining a neural classifier for navigable area detection with template matching for steering wheel angle control.

For our autonomous navigation system development, purely reactive models are not totally adequate, since immediate reaction to sensors data is not enough to guarantee a correct control in more complex environments. A more robust system must be implemented, providing sequence and context information that are absent in purely reactive models.

In robotics, FSM-based approaches [5] are very often used. FSMs are useful because the system can be easily described as a sequence of states (context changes), considering the inputs (sensors) and specific actions for each state. This way, for each detected state the robot can assume a different behavior. This work uses this idea as principle, so the path is described as a FSM in which current state is detected after processing sensors data.

The use of a Machine Learning technique such as Artificial Neural Networks is a very interesting way to process input data, identifying and classifying the states and transitions to determine the best actions to be performed [6]. ANNs are tolerant to noise and imprecision on input data, and able to detect the states and transitions between these states. ANNs are also very efficient to generalize knowledge (adjusting the outputs to many inputs, even the ones not explicitly taught to the net). So, this technique is very useful for state detection through path features recognition.

The association of ANNs with FSMs is being developed and evolved since the 1990´s [21][22][23][24], and recent researches are focused on the application of this approach in robotics. In [12], an autonomous car parking system was developed, using recurrent neural networks for FSM learning. The sensors data and current state were used as ANN inputs, so the system could detect when a context change was needed. This work inspired the development of our FSM-based topological navigation control.

So, in [20] a Vision-based autonomous navigation system was implemented for indoor environments using a simple FSM control. In that work, a neural vision system was responsible to provide a navigability map of a captured frame in an indoor environment, generating a matrix with values between 0 and 1 to represent the certainty about the navigability of a block of pixels. Then, an algorithm was developed to analyze interest areas of this matrix in order to detect the current state of a FSM (robot context), navigating through a set of straights and turns.

The main idea of this FSM-Control was evolved, so in [4] an autonomous navigation system was developed with an ANN as control unit. A LIDAR sensor was adopted as the main sensor, so an ANN was trained to recognize the “laser signatures” associated to every possible state of a FSM. The environment was represented by a Topological map, so each state was related to a specific node (part of the indoor environment). This way, all possible states could be learned by the ANN, and all possible paths (subgraphs) represented by FSMs. This work introduced the Topological approach adopted in this paper. The topological navigation system was also successful with other sensorial systems, such as 3D-vision in indoor environments, as implemented on [16].

The proposed approach comes from these previous works, combining the visual navigation designed in [20] with the neural FSM learning for Topological Navigation developed in [4]. A newer (and more robust) version of the classifier used in [20] is adopted. The developed system components are described in next section.

III. MATH

The proposed system is composed by two main steps performed by ANNs: the first one is the generation of a navigability map after processing the captured frame; the second one is the pattern recognition in the navigability maps, resulting on state detection for Topological Navigation control. Figure 1 shows the full-system overview.

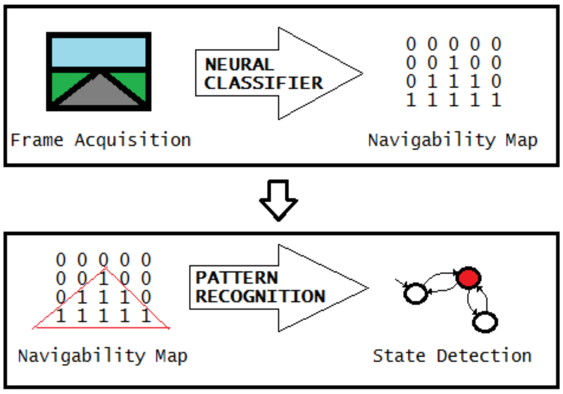

Figure 1 – System overview.

A. *Navigability Map Generation*

This step consists on process the captured frame in order to know which pixels are representing a navigable area in front of the robot (autonomous vehicle).

The neural classifier adopted was proposed by Shinzato [3] as an improvement to the model used in [20]. It is composed by an ANN committee with five ANNs; each one is responsible to classify the pixels into navigable (1) or non-navigable (0), based on different sets of features of the image. The ANN attributes are image features such as RGB average, HSV entropy, variance and energy. The mean of the five values is the final classification, ranging from 0 to 1.

In order to increase system efficiency, the pixels are not classified individually; the original frame is sliced into blocks of 10x10 pixels. This way, the navigability map dimensions are not the [320x240] of the original frame, but [32x24] only. Figure 2 shows the classifier structure. This classifier was chosen due to its excellent results in many environments and climate conditions, as shown on Figure 3. No modifications were done to the original classifier.

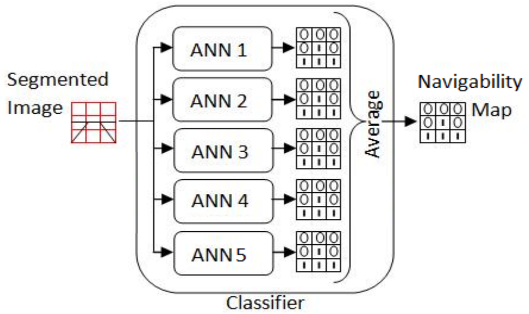

Figure 2 – Classifier Structure [3].

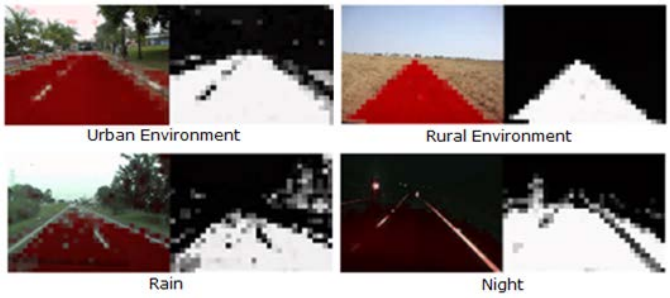

Figure 3 – Classification results on different scenarios and conditions [3].

The navigability map generated by this classifier is the input considered for ANN training indeed, in Topological Navigation step. This way, climate and luminosity conditions have no direct effect on state detection, since pattern recognition is done over navigability maps only.

B. *Topological Navigation System*

This step consists on train an ANN to efficiently recognize patterns on the input data, allowing the detection of environment features used to determine the current state (context).

The environment is mapped as a topological map in which each node is related to a specific part of the environment, described by its structural features such as straight path, turn or intersections. All possible kinds of states (environment features) are taught to the net, so any possible path on this environment can be seen as a sequence of these states, being represented by a FSM, as represented on Figure 4.

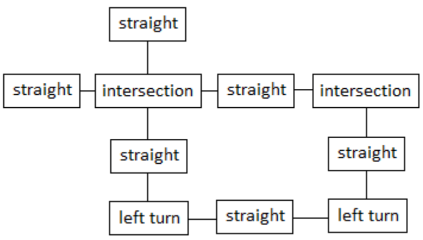

Figure 4 – Example of Topological Map.

After selecting a destination, a FSM with expected states sequence (subgraph) can be generated. For each state, an adequate reactive control must be activated, for example keeping the robot aligned at a straight path state. This way, the system combines this deliberative control (path planning) with a reactive control to keep a correct motion at the navigable areas only according to current state.

The ANN input is the data obtained from sensorial system (navigability matrix), and the output is the current state (environment feature) detected. Each different class must be related to a different track shape or condition (straight, left and right turns, bifurcation and obstacle ahead, for example).

In order to begin the training process, a database must be generated. This is done collecting a set of frames for each possible situation, then processing the collected frames with Shinzato´s neural classifier, for navigability maps generation. As the learning process is supervised, a specialist must classify these navigability maps in one of the possible states before ANN training, resulting in a set of input/desired-output pairs.

The system setup is shown on Figure 5 with its two main stages: the environment classification, in which the ANN is trained to recognize the possible situations; and navigation control, which consists on follow a well-defined path, generated after route selection.

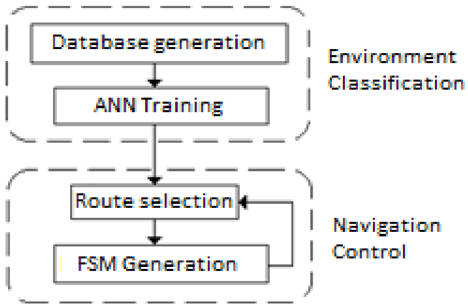

Figure 5 – System setup.

Despite the speed benefits obtained after processing the pixels in 10x10 blocks with Shinzato´s classifier on previous step, the total amount of elements in navigability map (768) is still too large for processing as ANN input. As the first 384 elements (upper half of matrix) are related to elements above the horizon line, they are not relevant for track analysis, so only the bottom half is used as ANN inputs. This way, the ANN input layer has 384 neurons only.

Some empirical tests were realized in order to determine the number of hidden layer neurons. The best results were achieved by a feed-forward MLP, with the 384 input neurons, 384 neurons on hidden layer and 5 neurons on output layer (1 neuron per defined class). Figure 6 shows a simplified representation of this MLP.

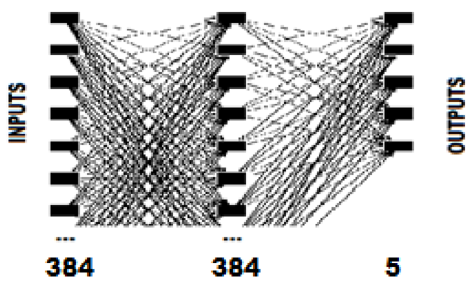

Figure 6 – ANN Topology.

Each output neuron must correspond to one of the implemented states. The ANN is trained just once e must be able to work for any possible path at the environment.

Once the Topological Map of an environment is given, it becomes easy to establish a route between two points, “by hand” 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 a well-defined route.

An algorithm must be responsible to convert any possible path into a sequence of states and related actions (also considering that one single state can have different associated actions). So, after selecting a path to be covered, the FSM must be stored on memory to be used by the control unit.

As mentioned earlier, the proposed hybrid control combines the deliberative control resulting from FSM-based topological navigation with a reactive control to guarantee a safe driving, avoiding collisions.

The Topological Navigation previously validated in indoor environments allows the robot to follow its planned path and also know its approximate position, but doesn´t control the motion “into” every situation (state). When the autonomous vehicle is in a straight path for example, a reactive control must be activated to keep it centered and aligned in the track. This is the main benefit of this hybrid/hierarchical control: take advantage of deliberative model for path control and simultaneously guarantee a safe navigation and obstacle avoidance with reactive control.

For this implementation, it is assumed that vehicle´s initial position is always known, as the topological map also. The current position can be estimated through current state detection. So, it isn´t necessary to estimate the exact current position, and navigation and self-localization tasks are performed together.

Input data processing makes it possible to determine if the vehicle still at the same state (part of the path) or if a context change is needed. State transitions must occur only if the detected state is compatible with next expected state (this information is related to the stored FSM). Figure 7 shows the topological navigation control flowchart.

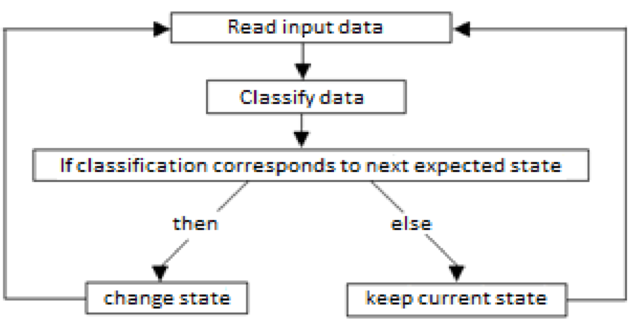

Figure 7 – Navigation control flowchart.

IV. EXPERIMENTS AND RESULTS

The experiments were carried out in a real structured urban environment, using a standard monocular RGB camera as the only sensor. This environment was composed by straights, turns and intersections, so its properties could be represented with four states: “straight path”, “left turn”, “right turn”, and “bifurcation”.

Some dynamic elements such other moving vehicles were included on database examples as “obstacle ahead” state, so if this situation is detected, an adequate behavior must be activated. Examples of input frames for the five implemented states are shown next, on Figure 8.

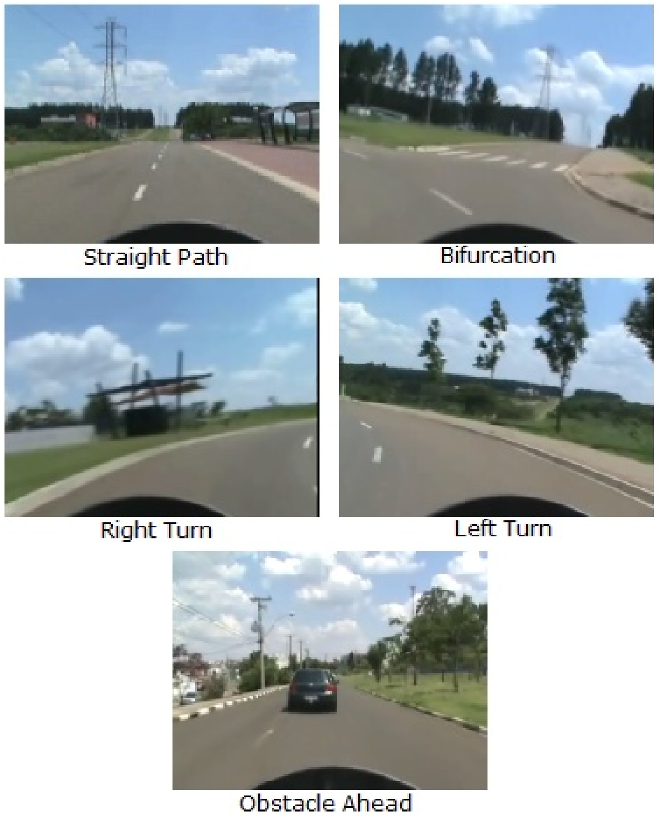

Figure 8 – Possible states for detection.

The ANN was implemented and trained with Stuttgart Neural Network Simulator (SNNS) software, then it was converted to C language using SNNS2C tool in order to be integrated to vehicle´s control unit.

ANN training database was generated collecting about 1’25” video at 30fps for each class in a run through the environment, resulting in about 2500 examples per class. The final database was composed by 11186 input/output pairs.

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 = 500.

Three different topologies were tested, with different number of hidden layer neurons. The tests were held with 96, 192 and 384 neurons on hidden layer. These amounts were considered after empirical tests. As mentioned earlier, the input layer is composed by 384 neurons (bottom elements of navigability matrix), and output layer is composed by 5 neurons (1 neuron for each possible class).

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 5 classes on the datasets).

All networks presented great results (higher than 98%). The network with best results was 384-384-5, with 99.44% accuracy, as can be observed at Table 1. The training time for this network is about two hours at a dual core PC.

TABLE I. ANN´S ACCURACY AFTER 500 TRAINING CICLES

ANN	*Test 1*	*Test 2*	*Test 3*	*Test 4*	*Test 5*	*Mean*
384-96-5	0,98	0,98	0,99	0,98	0,99	**0,984**
384-192-5	0,99	0,98	0,99	0,98	0,99	**0,986**
384-384-5	0,995	0,995	0,993	0,993	0,994	**0,994**

The confusion matrices for 384-384-5 net tests are shown next, on Figure 9. The error per class is close to zero, so it is easy to note that classification errors occur very few times.

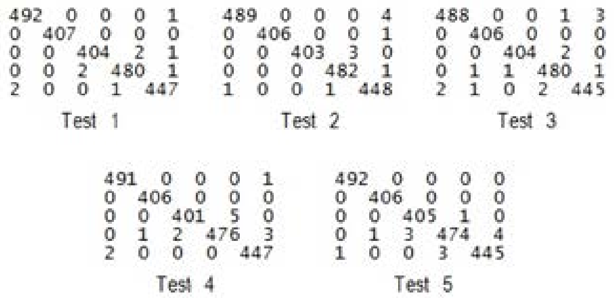

Figure 9 – Confusion matrices for 384-384-5 net.

The recall rate, precision rate and F1 measure for each class in the five tests carried out are shown next, on Table 2, Table 3 and Table 4 respectively. In order to estimate these values for each class independently, the following assumptions were taken:

- TP (True Positive) rate is the amount of examples rightly classified as the target class;
- FP (False Positive) rate is the amount of examples classified as target class but their real class is one of the other 4 classes;
- FN (False Negative) rate is the amount of target class examples classified as other classes examples.

TABLE II. RECALL RATE FOR EACH CLASS IN THE FIVE TESTS

Recall	*Test 1*	*Test 2*	*Test 3*	*Test 4*	*Test 5*	*Mean*
Class 1	0,998	0,991	0,987	0,998	1	**0,995**
Class 2	1	0,997	1	1	1	**0,999**
Class 3	0,992	0,992	0,995	0,987	0,997	**0,993**
Class 4	0,993	0,997	0,993	0,987	0,983	**0,991**
Class 5	0,993	0,995	0,988	0,995	0,991	**0,992**
Total	0,995	0,995	0,993	0,993	0,994	**0,994**

TABLE III. PRECISION RATE FOR EACH CLASS IN THE FIVE TESTS

Precision	*Test 1*	*Test 2*	*Test 3*	*Test 4*	*Test 5*	*Mean*
Class 1	0,996	0,998	0,995	0,995	0,998	**0,996**
Class 2	1	1	0,995	0,997	0,997	**0,998**
Class 3	0,995	1	0,997	0,995	0,992	**0,996**
Class 4	0,993	0,991	0,989	0,989	0,991	**0,991**
Class 5	0,993	0,986	0,991	0,991	0,991	**0,990**
Total	0,995	0,995	0,993	0,993	0,994	**0,994**

TABLE IV. F1 MEASURE FOR EACH CLASS IN THE FIVE TESTS

F1 Measure	*Test 1*	*Test 2*	*Test 3*	*Test 4*	*Test 5*	*Mean*
F1 Measure 1	0,997	0,994	0,991	0,997	0,999	**0,995**
F1 Measure 2	1	0,998	0,997	0,998	0,998	**0,998**
F1 Measure 3	0,993	0,996	0,996	0,991	0,995	**0,994**
F1 Measure 4	0,993	0,994	0,991	0,988	0,987	**0,991**
F1 Measure 5	0,993	0,991	0,99	0,993	0,991	**0,991**
Total	0,995	0,995	0,993	0,993	0,994	**0,994**

Despite the excellent results considering the accuracy mean, a 100% safe navigation is not guaranteed with a good classifier only. Furthermore, 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, an iteration counter was implemented, removing oscillations resulting from possible classification errors. This means that a state transition will occur only after a determined amount of consecutive detections of the expected state, indicating confidence on transition detection.

V. CONCLUSION

The developed classifier achieved good results, with high accuracy level for the ANN individually, and 100% accuracy after iteration counter implementation on the experiments carried out. This shows that the association of ANN and FSM – an already successful implementation for indoor environments – can be a suitable approach for autonomous vehicles navigation in structured outdoor environments.

The use of a camera as main sensor proved to be an efficient and reliable solution for state detection, allowing the development of low-cost autonomous driving systems, since it can also be used for reactive control.

The main challenge for future works is to apply this same methodology for autonomous urban navigation with dynamic elements, avoiding accidents and obeying the traffic laws. We also consider using other sensorial systems (also fusing sensors and techniques) to detect new features and landmarks useful for navigation control.

ACKNOWLEDGMENT

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

REFERENCES

[1] Zingg, S., Scaramuzza, D., Weiss, S., and Siegwart, R. MAV Navigation through Indoor Corridors Using Optical Flow , 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, vol. 24, issue 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. (1979) “Introduction to Automata Theory, Languages and Computation”. Addison - Wesley, 1979.

[6] Marino, A.; Parker, L.; Antonelli, G. and Caccavale, F. Behavioral Control for Multi-Robot Perimeter Patrol: A Finite State Automata approach. In: ICRA, 2009.

[7] Thrun, S. et al. (2006) "Stanley: The Robot that Won the DARPA Grand Challenge," Journal of Field Robotics, Vol. 23, No. 9, June 2006, p.661-692.

[8] Urmson, Chris et al. (2008). “Autonomous driving in urban environments: Boss and the Urban Challenge”. In: Journal of Field Robotics. Vol. 25 , Issue 8 (August 2008). Special Issue on the 2007 DARPA Urban Challenge, Part I. Pages 425-466.

[9] Buehler, Martin; Iagnemma, Karl; Singh, Sanjiv (Editors). The 2005 DARPA Grand Challenge: The Great Robot Race (Springer Tracts in Advanced Robotics). Springer; 1st. edition (October, 2007).

[10] Nefian, A.V.; Bradski, G.R. (2006) “Detection of Drivable Corridors for Off-Road Autonomous Navigation”. ICIP-06: Proceedings of the IEEE International Conference on Image Processing. pp. 3025-3028.

[11] J.M. Álvarez, A. M. López, and R. Baldrich. (2008) “Illuminant Invariant Model-Based Road Segmentation”. IEEE Intelligent Vehicles Symposium, Eindhoven, Netherlands, June 2008. http://www.cvc.uab.es/adas/index.php?section=publications.

[12] Heinen, Milton Roberto ; Osório, Fernando S. ; Heinen, Farlei ; Kelber, Christian . SEVA3D: Using Artificial Neural Networks to Autonomous Vehicle Parking Control.. In: IJCNN - IEEE Intenational Joint Conference on Neural Networks, 2006, Vancouver. Proceeding of the WCCI (World Congress on Computational Intelligence) - IJCNN. Vancouver, Canadá : IEEE Press, 2006. v. 1. p. 9454-9461.

[13] Pomerleau, D. ALVINN: An Autonomous Land Vehicle In a Neural Network. Advances in Neural Information Processing Systems 1, 1989.

[14] Pomerleau, D. RALPH: Rapidly Adapting Lateral Position Handler. IEEE Symposium on Intelligent Vehicles, September, 1995, pp. 506 - 511.

[15] Souza, J.; Sales, D. O.; Shinzato, P. Y.; Osório, F. S.; Wolf, D. F. Template-based autonomous navigation in urban environments. In: Proceedings of the 2011 ACM Symposium on Applied Computing, TaiChung, China, 2011.

[16] Sales, D. O.; Correa, D. S. O.; Osório, F. S.; Wolf, D. F. 3D Vision-based Autonomous Navigation System using ANN and Kinect Sensor. In: Conference Proceedings EANN 2012 – CCIS: Volume number 311., London, UK, 2012.

[17] Wolf, Denis F.; Osório, Fernando S.; Simões, Eduardo; Trindade Jr., Onofre. Robótica Inteligente: Da Simulação às Aplicações no Mundo Real. [Tutorial] In: André Ponce de Leon F. de Carvalho; Tomasz Kowaltowski. (Org.). JAI: Jornada de Atualização em Informática da SBC. Rio de Janeiro: SBC - Editora da PUC. RJ, 2009, v. 1, p. 279-330.

[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.; Cheng Wang; Wuensche, H.J.; (2008) "Design and capabilities of the Munich Cognitive Automobile". IEEE Intelligent Vehicles Symposium, 2008. Page(s): 1101 – 1107.

[19] Haykin, S. Neural Networks: A Comprehensive Foundation. Prentice Hall PTR, Upper Saddle River, NJ, USA, 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. Lee; Horne, Bill G.; LIN, Tsungnan. Learning a class of large finite state machines with a recurrent neural network. Neural Networks 8(9): 1359-1365. 1995.

[22] Omlin, Christian W.; Giles, C. Lee. Constructing Deterministic Finite-State Automata in Recurrent Neural Networks. Jornal of the ACM 43(6): 937-972. 1996.

[23] Frasconi, Paolo; Gori, Marco; Maggini, Marco; Soda, Giovanni. Representation of finite state automata in Recurrent Radial Basis Function networks. Machine Learning 23:1, 5-32 1996.

[24] Cleeremans, Axel; Servan-Schreiber, David; McClelland, James L. Finite State Automata and Simple Recurrent Networks. Neural Computation, Vol. 1, No. 3, Pages 372- 381. 1989.