Vision-based Waypoint Following Using Templates and Artificial Neural Networks

Jefferson R. Souza, Gustavo Pessin, Patrick Y. Shinzato,
Fernando S. Osorio and Denis F. Wolf

Mobile Robotics Laboratory, University of Sao Paulo (USP)
Av. Trabalhador Sao-Carlense, 400 - P.O. Box 668 - 13.560-970, Sao Carlos, Brazil
{jrsouza, pessin, shinzato, fosorio, denis}@icmc.usp.br

Abstract

This paper presents a learning-based vehicle control system capable of navigating autonomously. Our approach is based on image processing, road and navigable area recognition, template matching classification for navigation control, and trajectory selection based on GPS way-points. The vehicle follows a trajectory defined by GPS points avoiding obstacles using a single monocular camera and maintaining the vehicle in the road lane. Different parts of the image, obtained from the camera, are classified into navigable and non-navigable regions of the environment using neural networks. They provide steering and velocity control to the vehicle. Several experimental tests have been carried out under different environmental conditions to evaluate the proposed techniques.

Keywords: Robotic Vehicle Navigation, Artificial Neural Networks, Computer Vision, Compass, GPS way-points, and Templates.

1. Introduction

Research in autonomous vehicles have reached significant progress and improved the experimental results over the last years. Some of them have focused on autonomous navigation, which is a fundamental task in this area [11]. Lately, several works have improved navigation in urban environments. Competitions, like DARPA Grand [8] and Urban [9] Challenges and ELROB [3] have been pushing the state of the art in autonomous vehicle control. In these competitions, several different solutions that combine information

from a large number of complex sensors were proposed. Some approaches use five (or more) laser range finders, video cameras, radar, differential GPS, and inertial measurement units [2], [11]. Although there is a large number of interesting applications for autonomous vehicle technology, its cost is still very high, hampering commercial applications.

This paper proposes a GPS-oriented and vision-based autonomous navigation approach for urban environments. The system uses a single monocular camera to acquire data from the environment, a compass (orientation) and a GPS (localization) to obtain the necessary information for the vehicle to reach destination through a safe path. It also detects the navigable regions (roads) and estimates the most appropriate maneuver. Several Artificial Neural Networks (ANNs) topologies have been evaluated in order to keep the vehicle in a safe path, and finally control the vehicle's steering and acceleration.

Our approach uses two different systems based on ANN. The first identifies road lane regions in the image. After that, our system verifies several possible directions are free to navigate. The value of the azimuth (difference between the current and target positions) as well as a set of values that correspond to free or obstructed directions on the road are obtained. These values are used as the inputs of the second system based on ANN, which aims to learn rules for vehicle control, providing steering and velocity values. The system is detailed in section 3. Figure 1 presents our CaRINA test platform.

Figure 1: Intelligent Robotic Car for Autonomous Navigation (CaRINA) test platform.

2. Related Works

ALVINN [12] is an ANN-based navigation system that calculates a steer angle to keep an autonomous vehicle in the road limits. The gray-scale levels of a 30 x 32 image were used as the input of an ANN. The original road image and steering were provided to improve the training, allowing ALVINN to learn how to navigate in new roads. After several upgrades, this system

was able to travel on many road types like single-lane and unlined paved roads at speeds of up to 55 mph. However, it is important to emphasize that this system was designed and tested to drive on well-maintained roads like highways under favorable traffic conditions. According to [6], the problem of ALVINN is the lack of ability to learn features, which would allow the system to drive on different road types other than that on which it was trained.

Hamner et al. [5] presented an outdoor mobile robot that learns to avoid collisions based on human behavior. The test vehicle is equipped with laser, GPS and IMU to produce a map. This method automatically learns the parameters of the control model. The model input is a goal point and a set of obstacles defined as points in the x-y plane of the vehicle. Random, Genetic Algorithm (GA) and Simulated Annealing (SA) were used in the learning step. GA and Random performed similarly well, and SA was worse than the other techniques. The authors also compared the randomly learned parameter set with the principal component analysis and observed that the random generalized well. The issue was the dependency of a goal point and the set of obstacles, and all the parameters were limited to a fixed range.

Intelligent Speed Adaptation and Steering Control [1] allows the vehicle to anticipate and negotiate curves safely. It uses Generic Self-Organizing Fuzzy Neural Network (GenSoFNN-Yager), which includes the Yager inference scheme [10]. GenSoFNN-Yager can induce knowledge from low-level information in the form of fuzzy *if-then* rules. Results have shown the robustness of the system in the task of learning from examples of human driving, negotiating new unseen roads. The disadvantage is that the autonomous driver demonstrates that anticipating is not always sufficient. Moreover, large variations in the distribution of the rules were observed, implying a high complexity of the system.

Stein and Santos [18] system computes the steering of an autonomous robot, moving in a road-like environment. It uses ANNs to learn behaviors based on examples from a human driver, replicating and sometimes even improving human-like behaviors. To validate the created ANNs, real tests were performed and the robot successfully completed several laps of the test circuit showing good capacities for both recovery and generalization with relatively small data sets.

Driving School [7] learns driving skills based on a human teacher. It is implemented as a multi-threaded, parallel architecture in a real car and trained with real driving data to generate steering and acceleration control for road following. Furthermore, it uses an algorithm to detect independently-

moving objects (IMOs) to spot obstacles with a stereo camera, demanding a more complex hardware. A predicted action sequence is compared to the driver's actions and a warning is issued if there are many differences between the two actions. The IMO detection algorithm is more general as it will respond not only to cars, but also to any sufficiently large (11 x 11 pixels) moving object. The steering prediction is very close to the human signal, but the acceleration is less reliable.

3. Proposed Method

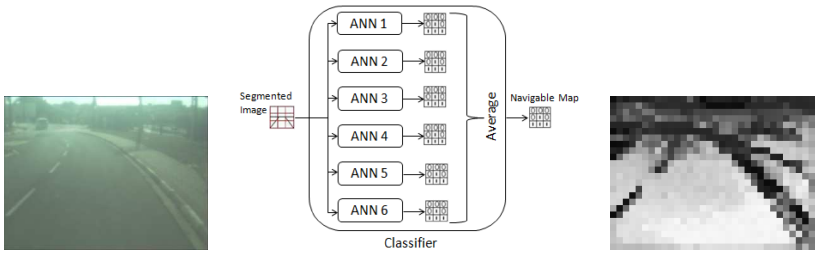
Our approach is composed by three steps. In the first, our system identifies the road on a color image obtained by a video camera attached to the vehicle. The result from first step is a visual navigation map that is used in the next step. In the second step, a template matching algorithm identifies possible geometries of the road ahead of the vehicle (straight line, soft and hard turn to left/right) and calculates an occupation percentage for each of them. Applying a threshold in these occupation values, the system distinguishes free areas from obstructed areas based on the visual navigation map and road geometry. Finally, a supervised learning technique is used to define the action that the vehicle should take to remain in the safe navigation regions (road). These steps are described in the next subsections.

3.1. Road Identification Step

In order to solve the road identification task, we apply a method based on [14]. This approach creates a system composed of several ANNs (committee classifier) that identifies the road into the image which is decomposed in many regions also called sub-images. Therefore, the classification of each sub-image is an average of the results of all ANN outputs (committee machine).

Initially, an image is transformed into a set of sub-images of $(K \times K)$ pixels, where a set of features are extracted from each of them.

According to the approach described in [14], an image-feature is a statistical measure computed for a group of pixels considering a specific color channel. The following statistical measures are used: average, entropy, variance and energy. These measures can be associated with some channel from four different color spaces in order to define a feature: RGB, HSV, YUV and normalized RGB. The normalized RGB is composed by $(R/(R+G+B))$, $(G/(R+G+B))$, $(B/(R+G+B))$.



(a) (b) (c)

Figure 2: Real image (a), Road identifier (b) and VNMap result (c).

The ANN used in this step consist of a Multi-Layer Perceptron (MLP) with Resilient-Propagation (RProp) training algorithm. The ANN topology consists in, basically, three layers, one input layer, one hidden layer with five neurons and the output layer. The ANN has been trained to return the value 1.0 when receives patterns belonging to a road class and to return 0.0 when receives patterns that does not belong to road class. In this work, the step of road identification uses six ANNs - as shown in Fig. 2(b) - and each one uses some, not all, image-features as input. In other words, these ANN are distinguished by the combination of image-features used as input. Table 1 shows the set of image-features adopted in each ANN. Their choice was based on the latest results of [14]. Since the number of neurons is small, the training time is also reduced, enabling the instantiation of multiple ANN.

After the features are generated, all the sub-images are classified by the six ANN, the six outputs are combined for each sub-image in order to generate a $Matrix$. This $Matrix$ is composed by real numbers where $Matrix(i, j)$ is the classification of sub-image(i, j). This matrix is called Visual Navigation Map (VNMAP) and it is used by next step to calculate the occupation values. Fig. 2 shows a sample of an image classified by this system, where the Fig. 2(c) shows the VNMap in gray-scale - black represents non-road class, white represents road class and the gray represents the intermediate values.

3.2. Occupation Value Step

In this step, our system identifies five possible geometries of the road ahead of the vehicle into VNMAP, as shows Fig. 3, and calculates an occupation value for each of them. The Equation 1 shows how to calculate the occupation value:

$$score(T) = \frac{\sum_{x \in T} Matrix(x.row, x.col)}{|T|}, \quad (1)$$

Table 1: Input attributes of the ANNs (R, G, B = red, green, blue components; H, S, V = hue, saturation, value components; Y, U, V = Luminance-Chrominance, average = av, normalized = norm, entropy = ent, energy = en and variance = var).

ANNs	Input attributes
ANN1	U av, V av, B norm av, H ent, G norm en and H av
ANN2	V av, H ent, G norm en, G av, U av, R av, H av, B norm av, G norm av and Y ent
ANN3	U av, B norm av, V av, B var, S av, H av, G norm av and G norm ent
ANN4	U av, V av, B norm av, H ent, G norm en and H av
ANN5	V av, H ent, G norm en, G av, U av, R av, H av, B norm av, G norm av and Y ent
ANN6	U av, B norm av, V av, B var, S av, H av, G norm av and G norm ent

where T is a set of all sub-image(i, j) that belongs to detemined geometry, $|T|$ is the number of cells into geometry or template and finally, $Matrix(x.row, y.col)$ is the real value into VNMAP that corresponds to sub-image$(x.row, y.col)$. The final result $score(T)$ is a value ranging from 0 to 1. A good performance was obtained in the navigation defined by the urban path avoiding obstacles using the best score [17].

(a) (b) (c) (d) (e)

Figure 3: Five possible geometries defining the vehicle directions it can follow. (a) Hard left turn, (b) Soft left turn, (c) Straight, (d) Soft right turn, (e) Hard right turn.

After obtaining the *score* for all templates, the system verifies if the occupation value is lower than a threshold level. If the occupation value from template is lower than threshold level, this template is considered as obstructed (value 0), if larger, this template is considered as free (value 1). The values free or obstructed are part of the system input for the next step.

3.3. Learning-Based Navigation

We have already analyzed several levels of memory of the templates based on examples obtained from human drivers using neural networks [16]. The

results of these neural networks have also been compared with different supervised learning techniques for the same purpose [15].

In this step, the basic network structure used is a feedforward MLP. The activation functions of the hidden neurons are logistic sigmoid and hyperbolic tangent, and the ANN learning method is the Resilient-Propagation (RProp). The inputs are represented by the azimuth (difference between the current and target positions of the vehicle) and the values for each template obtained by the occupation value step - obstructed or free occupation areas (OA). The outputs of the proposed method are the steering angle and speed (Figure 4).

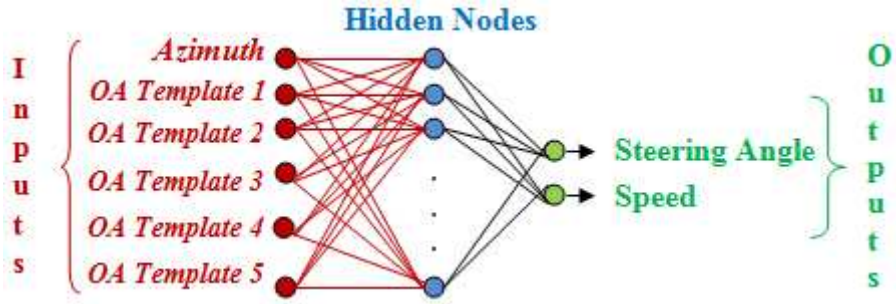


Figure 4: Structure of the second ANN used to steering and acceleration commands.

4. Results

Experiments were performed using CaRINA (Figure 1), an adapted electric vehicle capable of autonomous navigation in an urban road. It is equipped with a VIDERE DSG camera, a ROBOTEQ AX2580 motor controller for steering, a TNT Revolution GS Compass (orientation) and a GARMIN 18X-5Hz GPS (localization). The image acquisition resolution was set to (320 x 240) pixels.

The road identification system converts an image of 320 x 240 pixels in a matrix of 32 x 24 blocks (K = 10). Each block is individually classified. During the learning step the user manually classifies parts of one or more scenes as navigable or non-navigable to serve as learning data to the ANN. In the control learning step, the system uses the classified image as input and the desired controls as output of the second ANN, as it is described in the Section 3.3.

Next subsections describe the performed experiments, in addition to an analysis of the neural networks used in the road identification step. The

experiments have been performed in an urban environment in two scenarios with distinct characteristics. Experiment 1 shows a closed area with several objects such as buildings, trees, walls, transit plates, bikes, cars and buses, which influence the road identification step. The experiment 2 presents a more open environment with few objects in the scene.

4.1. Experiment 1

Seven GPS waypoints were used to define the desired trajectory. In order to navigate, the vehicle used a monocular camera to avoid obstacles. The experiment was successfully performed as it followed the GPS points and avoided obstacles. In some of the straight paths, the vehicle had to avoid curbs at the same time it was attracted by the GPS goal points, resulting in some oscillation in the final trajectory. Figure 5 shows the vehicle trajectory obtained by GPS coordinates.

Figure 5: GPS coordinates performed by CaRINA (environment 1).

Table 2 shows the ANNs classification results obtained during the vehicle navigation. Five different ANNs topologies have been analyzed using two neural transfer functions (logistic sigmoid and hyperbolic tangent). These topologies represent the architecture of the second ANN used in our proposed system. Several topologies of ANNs were tested to obtain the minimum training error and the optimal neural network, with a well-defined architecture.

The ANN architecture was selected considering the number of neurons in the hidden layer using the RProp supervised learning algorithm in MLP networks and considering the values of MSE (Mean squared error) and the best epoch (Optimal point of generalization [OPG], i.e., minimum training error and maximum capacity of generalization).

We also evaluated the ANNs modifying the neural transfer functions. The main difference between these functions is that logistic sigmoid produces positive numbers between 0 and 1, and the hyperbolic tangent (HT), numbers between -1 and 1. After the analysis of Table 2, the 6x15x2 architecture using the hyperbolic activation function showed the lowest MSE for the 10600 epoch. Five different runs were executed changing the random seed used in the neural networks (Tests 1 to 5).

Table 2: Results of the ANNs for each topology.

Learning Functions [OGP and MSE (10^{-3})]										
	Logistic Sigmoid					Hyperbolic Tangent				
ANN Topology	Test 1	Test 2	Test 3	Test 4	Test 5	Test 1	Test 2	Test 3	Test 4	Test 5
6x3x2	600	3000	1400	700	1000	1400	2300	1000	2000	2600
Topology 1	5.266	3.922	3.747	5.407	4.856	4.119	3.418	3.737	3.393	3.310
6x6x2	7600	1200	8300	1700	5600	2200	700	1900	9700	26400
Topology 2	2.490	3.257	3.250	3.670	3.414	3.271	3.413	2.148	5.404	3.217
6x9x2	800	600	500	900	2300	1100	1100	800	800	400
Topology 3	3.774	5.013	4.584	5.050	3.190	3.162	3.318	3.650	2.352	4.292
6x12x2	91800	15200	1000	23600	1400	400	1400	800	300	7400
Topology 4	2.836	3.797	3.746	2.048	5.645	4.040	3.899	2.959	3.595	3.895
6x15x2	400	600	1300	400	700	2600	76100	10600	2500	1000
Topology 5	4.226	4.597	4.000	5.319	3.188	3.232	1.999	**1.881**	3.510	2.989

Figure 6 shows the histograms of the errors based on the best ANN topology obtained in Table 2. Topologies 4 and 5 were accepted based on the results. Figure 6(b) presents the error concentrated on zero and with a lower dispersion than the one in Figure 6(a). The same is true for Figure 6(d), which shows a lower dispersion compared to Figure 6(c), therefore encouraging the use of topology 5.

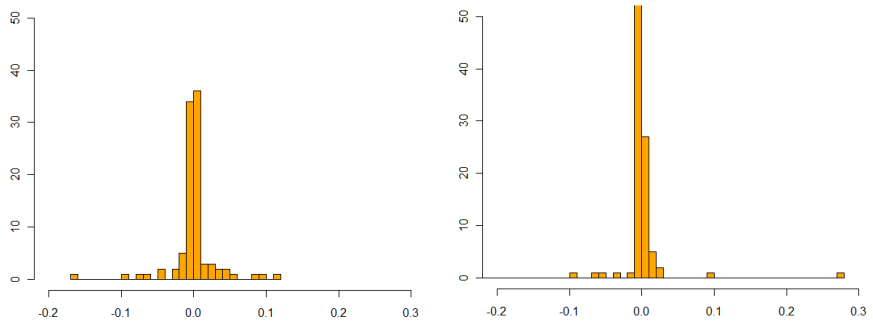
The statistical inference of the results was evaluated using Shapiro method [13]. We observed that the null hypothesis (normal adequacy), had not been satisfied (Table 3) using the best ANN topology (6x15x2) for the test data.

Table 3: Results of the Shapiro-Wilk Test.

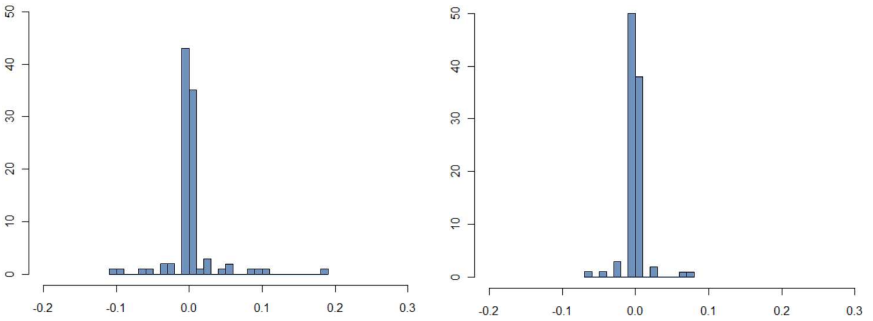
	Shapiro-Wilk Normality Test	
	p-value	
	Steering	Velocity
Topology 1	**0.2468**	3.84e-09
Topology 2	5.43e-14	1.97e-15
Topology 3	6.81e-14	2.68e-15
Topology 4	6.69e-13	8.60e-15
Topology 5	2.20e-16	1.59e-15

Value in boldface - Accepted as normal
Other values - Not accepted as normal

The nonparametric method of Man-Whitney [4] was used to check the differences between the topologies (see Table 4). For the velocity, the method does not reject the null hypothesis (equality) (the p-values are higher than



(a) Steering of topology 4 (Test 4 LS) (b) Steering of topology 5 (Test 3 HT)



(c) Velocity of topology 4 (Test 4 LS) (d) Velocity of topology 5 (Test 3 HT)

Figure 6: Error histogram considering different hidden layers of the best ANNs (MSE = 2.048 [topology 4] and MSE = 1.881 [topology 5]) (a) using steering angle data of topology 4 (logistic sigmoid [LS]), (b) steering angle data of topology 5 (hyperbolic tangent [HT]), (c) velocity data of topology 4 (LS), and (d) velocity data of topology 5 (HT). The x axis represents the error used in the first experiment (Test 1).

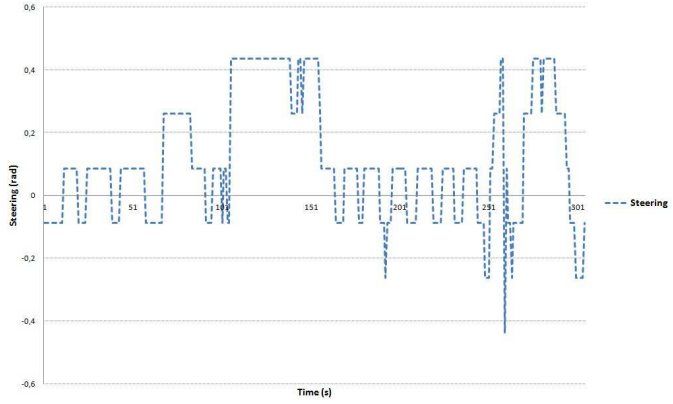
0.05) and for the steering, the null hypothesis is rejected only between topologies (T) 1 and 5 (the p-values are lower than 0.05). According to this method and using a confidence level of 95%, there is no evidence of statistical differences between the best results, except for T1 and T5 concerning the steering.

Table 4: Results of the Man-Whitney Test.

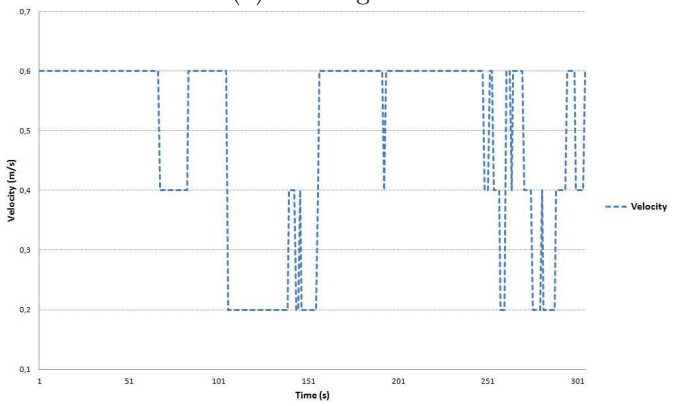
Man-Whitney Test					
		p-value			
		T1	T2	T3	T4
Steering	T5	0.031	0.115	0.397	0.114
Velocity	T5	0.464	0.403	0.715	0.585

Figures 7(a) and 7(b) illustrate the steering angle and velocity of CaRINA using the training data for the Experiment 1.

Figures 8(a) and 8(b) illustrate the steering angle and velocity of CaRINA



(a) Steering wheel



(b) Velocity

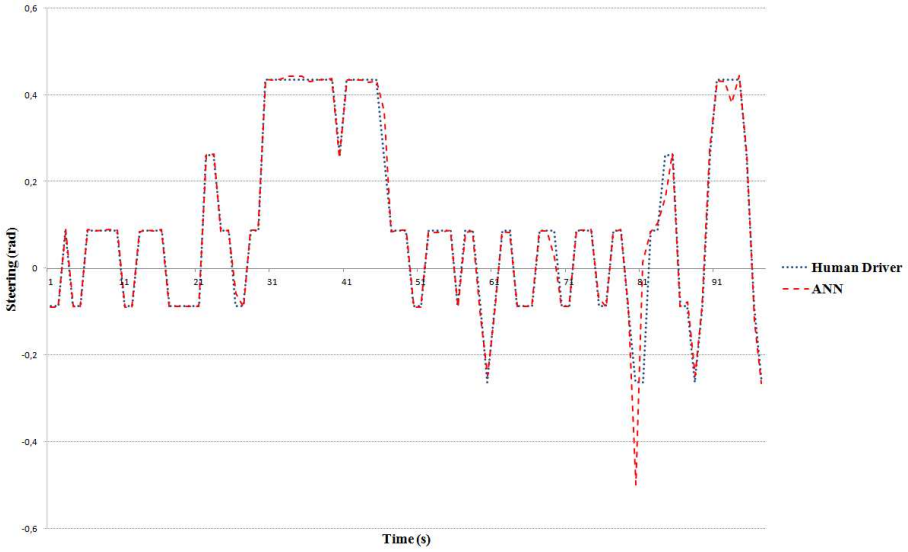
Figure 7: The steering wheel and velocity using the training data.

using the 6x15x2 architecture (Test 3 HT), showing the values based on the control rules (supervised training data) and the results obtained by the learning of the ANN. Small oscillations are present in the data learned by the ANN, since the rules maintained the steering wheel and velocity constant, resulting in the linearity of data (the problem for the ANN was to learn a rigid curve and high velocity, but this fact did not interfere in the results).

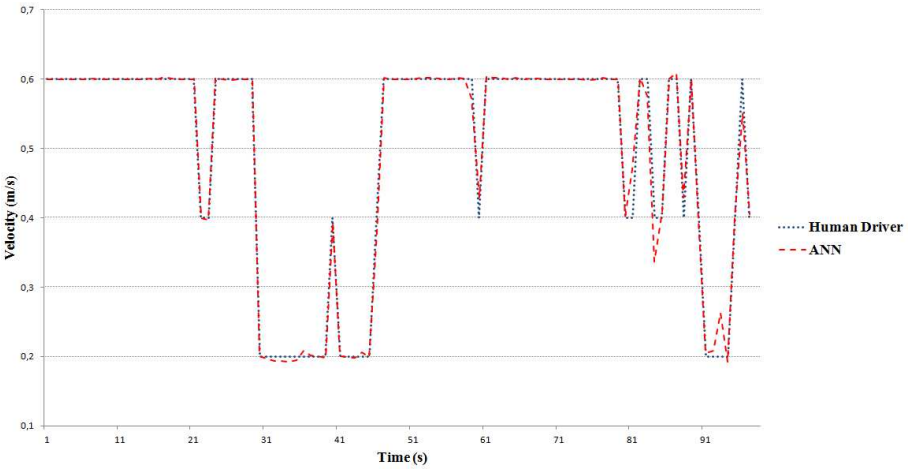
4.2. Experiment 2

Seven GPS waypoints were used. They were more than 100 meters away of each other and separated by sidewalks between the points and non-navigable areas (grass fields).

Figure 9 shows the vehicle trajectory obtained by GPS coordinates (Experiment 2). Waypoints P0 to P6 represent the GPS points. The red dashed line represents the orientation obtained by both GPS and compass, which



(a) Steering wheel



(b) Velocity

Figure 8: The human driver and ANN values using the test data.

should be traversed by the vehicle. The yellow line is the real path followed by the vehicle, showing the obstacles avoidance by using the camera images. The path performed was approximately 1.08 km. We only show the orientation lines between P0 and P1 (red line) and between P1 and P2 to keep the GPS coordinates image more clearly.

Figures 10 (a) and 10 (b) show the histogram of the error, where the main goal in these images is to observe its dispersion. In both images, which have

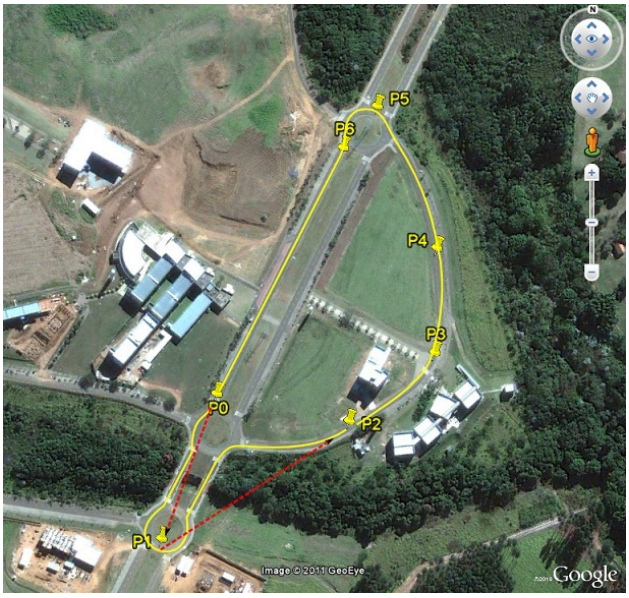
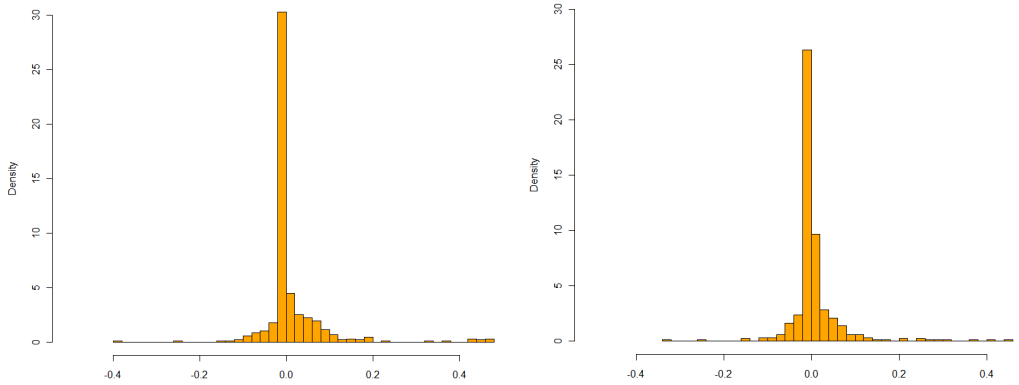


Figure 9: GPS coordinates performed by CaRINA (environment 2).

similar outliers, the error is concentrated near zero. Using Shapiro-Wilk test of normality, we obtain p-values lower than 0.05, and with 95% confidence, the distribution of the error is not accepted as normal. Therefore, we compare the distributions using the non parametric Man-Whitney test, and, with 95% confidence, there is no difference between the groups.



(a) Topology 4 (6x12x2 - LS) (b) Topology 5 (6x15x2 - HT)

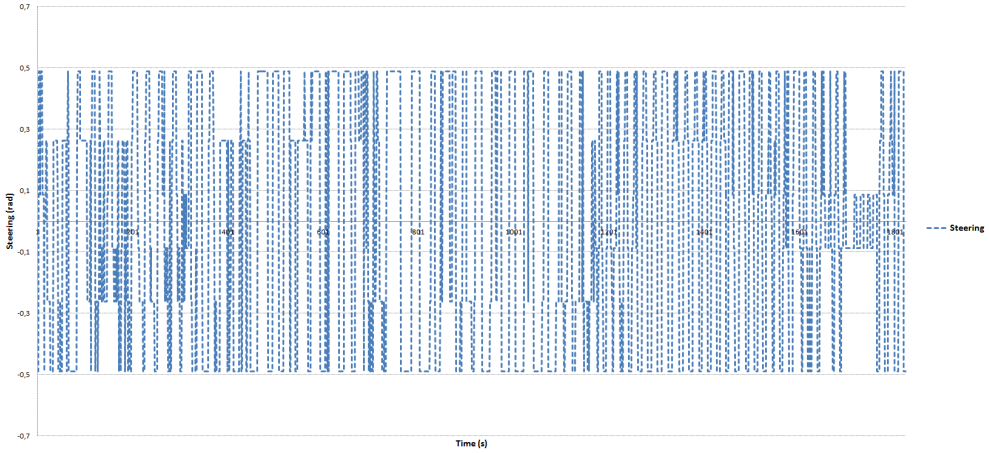
Figure 10: Error histogram considering the topologies 4 and 5 (steering angle data - it was used in the path 2). (a) Topology 4. (b) Topology 5. The x axis represents the error.

Table 5 shows the steering and velocity data of the best ANNs, where the ANN with 15 neurons in the hidden layer has smaller mean and standard deviation error than the ANN with 12 neurons. Therefore, even if both are statistically equivalent, topology 5 should be used in the system.

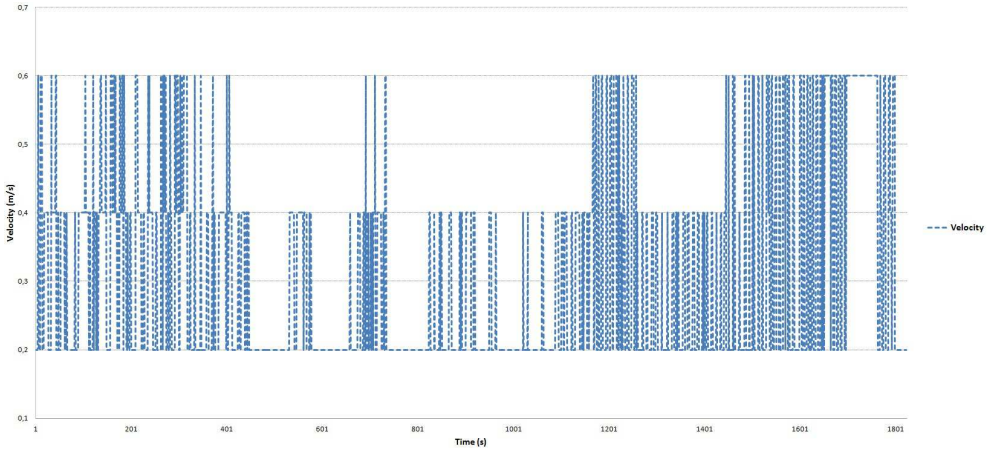
Table 5: Results of the mean and standard deviation error using steering and velocity.

	Topologies	Steering	Velocity
Mean	ANN 15	**0.007629**	**0.000528**
	ANN 12	0.014762	0.002138
Std	ANN 15	**0.055944**	**0.035246**
	ANN 12	0.073536	0.038567

Figures 11(a) and 11(b) illustrate the steering angle and velocity of CaRINA using the training data for the Experiment 2.



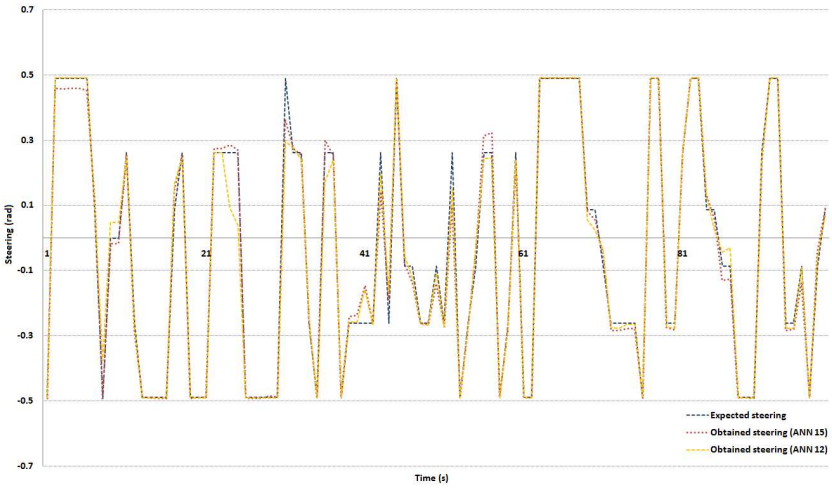
(a) Steering wheel



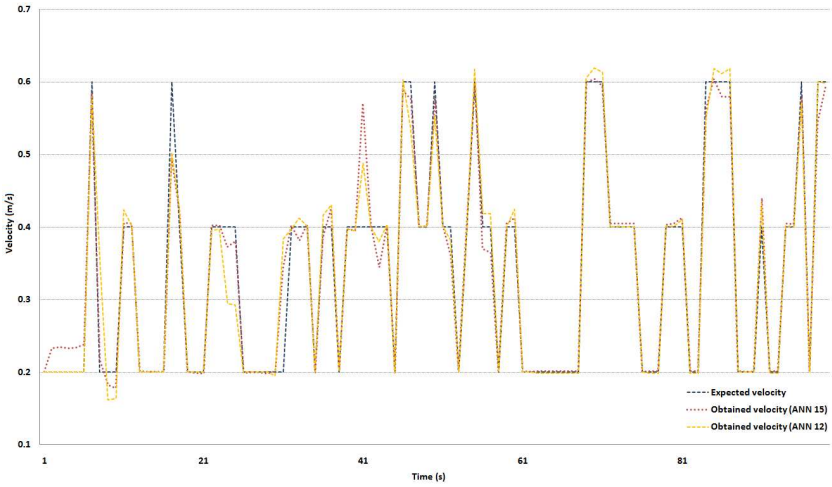
(b) Velocity

Figure 11: The steering angle and velocity using the training data.

Figures 12 (a) and 12 (b) illustrate the steering angle and velocity of CaRINA, showing the expected values and the values obtained by the learning of ANNs 15 and 12. Figure 12 (a) shows that both results are very similar, with small changes. According to Figure 12 (b), the results show small oscillations, which represent the high velocity peaks that the ANNs did not learn in some paths, but they do not interfere significantly in the vehicle control.



(a) Steering wheel



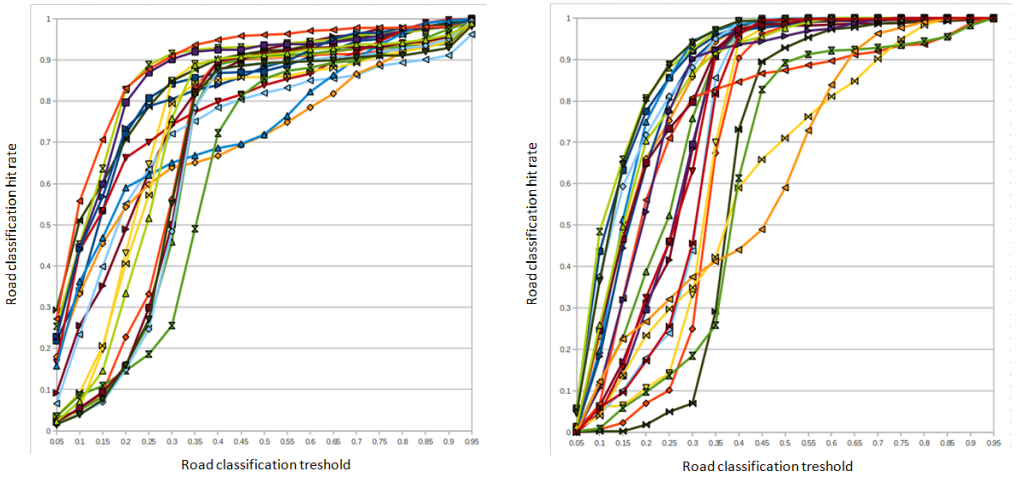
(b) Velocity

Figure 12: The steering angle and velocity using the test data.

CaRINA was able to safely navigate autonomously in an urban road; it did not get too close to the sidewalk or run over it (non-navigable areas) and also was able to track the GPS points (way-point) provided to the system.

4.3. Analysis of the Road Identification System

This subsection presents the hit rate from different frames over the path. These frames were manually classified to evaluate the road identification system. The graphics from Fig. 13(a) and (b) show the hit rate for the *Experiment 1* and *Experiment 2* respectively. The graphics show the hit rate varying from 0 to 1 by different threshold values and each line represents one frame. In general, the results of road identification system were satisfactory, since the hit rate is around 80% when the threshold value is larger than 0.4.



(a) Experiment 1 (b) Experiment 2

Figure 13: Hit rates by threshold value interpretationt. Different colored lines represent different analyzed frames from the experiments. Hit Rate is expressed into the interval from 0.0 to 1.0 (100% correct classification). X axis represents the Road classification treshold and Y axis represents the Road classification hit rate.

Is important to note that the evaluation occurs only below the horizon since the positions templates are fixed and always below the horizon line (skyline). Also, although the system presents several false positives in Fig. 14(b) and (d), the black blocks show that our system distinguish very well the non-navigable region from road regions. These black blocks decrease the occupation value for the templates in this area and this is satisfactory step to choose the best direction for the vehicle. The image classification approach can handle some degree of lighting variance in the scene like shadows. However a large variation in the illumination may require to repeat the learning step.

(a) (b) (c) (d)

Figure 14: Some samples of results from road identification step. (a) image raw from *Experiment 1*; (b) VNMap from *Experiment 1*; (c) image raw from *Experiment 2*; (d) VNMap from *Experiment 2.*

5. Conclusions and Future Works

Autonomous vehicle navigation is a fundamental task in mobile robotics. This paper has presented a monocular camera and a GPS-based autonomous navigation system that can be trained to identify the road and navigable regions. It uses a template matching classification to identify possible navigable directions, determines the occupation percentage in order to avoid obstacles, and controls the steering and velocity using an ANN learning scheme to follow GPS Waypoints, which is a novelty compared to our previous works. Our approach was evaluated using an Electrical Vehicle (CaRINA) in an urban road. CaRINA was able to navigate autonomously in this environment, successfully following the desired trajectory. Our quantitative analysis also obtained satisfactory results for the different ANNs topologies.

The integration of the monocular camera and GPS was one of the contributions of this paper, where CaRINA was able to safely navigate autonomously in different urban environments. The proposed method has been used to negotiate the curves and roundabouts, did not too close to the sidewalk (non-navigable areas) and tracked the GPS points (waypoint).

As future works, we plan to evaluate other classification methods and decision-making algorithms. We also intend to integrate a Stereo Camera and/or a LIDAR to deal with bumps and depressions in the road.

6. Acknowledgments

The authors would like to acknowledge the support granted by CAPES, CNPq and FAPESP to the INCT-SEC (National Institute of Science and Technology - Critical Embedded Systems - Brazil), and FAPESP doctoral grant (process 2009/11614-4).

7. References

[1] Chan, M., Partouche, D., Pasquier, M., 2007. An intelligent driving system for automatically anticipating and negotiating road curves. International Conference on Intelligent Robots and Systems, 117–122.

[2] Dahlkamp, H., Kaehler, A., Stavens, D., Thrun, S., Bradski, G., 2006. Self-supervised monocular road detection in desert terrain. In Proc. Robotics Science and Systems Conference.

[3] ELROB, 2011. European land-robot (elrob). Http://www.elrob.org/, Access on 26 Aug.

[4] Fay, M. P., Proschan, M. A., 2010. Wilcoxon-mann-whitney or t-test? on assumptions for hypothesis tests and multiple interpretations of decision rules. Statistics Surveys 4, 1–39.

[5] Hamner, B., Scherer, S., Singh, S., 2006. Learning to drive among obstacles. IEEE/RSJ International Conference on Intelligent Robots and Systems, 2663–2669.

[6] Jochem, T., Pomerleau, D., Thorpe, C., 1993. Maniac: A next generation neurally based autonomous road follower. Proceedings of the International Conference on Intelligent Autonomous Systems.

[7] Markelic, I., Kjaer-Nielsen, A., Pauwels, K., Jensen, L. B. W., Chumerin, N., Vidugiriene, A., Tamosiunaite, M., Rotter, A., Hulle, M. V., Kruger, N., Worgotter, F., 2011. The driving school system: Learning automated basic driving skills from a teacher in a real car. IEEE Transactions on Intelligent Transportation Systems.

[8] Martin, B., Karl, I., Sanjiv, S., 2007. The 2005 darpa grand challenge. Springer Tracts in Advanced Robotics 36.

[9] Martin, B., Karl, I., Sanjiv, S., 2010. The darpa urban challenge. Springer Tracts in Advanced Robotics 56.

[10] Oentaryo, R. J., Pasquier, M., 2006. Gensofnn-yager: A novel hippocampus-like learning memory system realizing yager inference. International Joint Conference on Neural Networks, 1684–1691.

[11] Petrovskaya, A., Thrun, S., 2009. Model based vehicle detection and tracking for autonomous urban driving. Autonomous Robots Journal 26 (2-3), 123–139.

[12] Pomerleau, D. A., 1989. Alvinn: An autonomous land vehicle in a neural network. Advances In Neural Information Processing Systems.

[13] Shapiro, S. S., Wilk, M. B., 1965. An analysis of variance test for normality (complete samples). Biometrika 52 (3-4), 591–611.

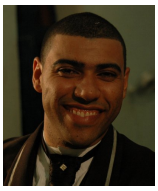
[14] Shinzato, P. Y., Wolf, D. F., 2010. A road following approach using artificial neural networks combinations. Journal of Intelligent and Robotic Systems 62 (3), 527–546.

[15] Souza, J. R., Pessin, G., Osório, F. S., Wolf, D. F., 2011. Vision-based autonomous navigation using supervised learning techniques. 12th Engineering Applications of Neural Networks 363, 11–20.

[16] Souza, J. R., Pessin, G., Shinzato, P. Y., Osório, F. S., Wolf, D. F., 2011. Vision-based autonomous navigation using neural networks and templates in urban environments. First Brazilian Conference on Critical Embedded Systems, 55–60.

[17] Souza, J. R., Sales, D. O., Shinzato, P. Y., Osório, F. S., Wolf, D. F., 2011. Template-based autonomous navigation and obstacle avoidance in urban environments. ACM - Applied Computing Review.

[18] Stein, P. S., Santos, V., 2010. Visual guidance of an autonomous robot using machine learning. 7th IFAC Symposium on Intelligent Autonomous Vehicles.



Jefferson Rodrigo de Souza is a PhD student in the Institute of Mathematics and Computer Science at the University of Sao Paulo. He obtained his MSc degree in Computer Science at the Federal University of Pernambuco in 2010. At ICMC/USP, he has been working in machine learning techniques for mobile robots and autonomous vehicles. More specifically, autonomous driving based on learning human behavior. His current research interests are Mobile Robotics, Machine Learning, and Hybrid Intelligent System.

Gustavo Pessin is a PhD student in the Institute of Mathematics and Computer Science at the University of Sao Paulo. He obtained a M.S. in Computer Science in 2008 from the University of the Sinos Valley. At ICMC/USP, he has been working on several research projects related to robotics and machine learning techniques. His current research interests are Mobile Robotics, Artificial Intelligence, and MultiAgent Systems.

Patrick Yuri Shinzato is a PhD student in the Institute of Mathematics and Computer Science at the University of Sao Paulo. He obtained a M.S. in Computer Science in 2010 from the University of Sao Paulo. At ICMC/USP, he has been working in computer vision, machine learning and neural networks. His current research interests are Computational Vision, Artificial Intelligence, Mobile Robotics and Sensors Fusion.

Fernando Santos Osrio is an Assistant Professor in the Department of Computer Systems at the University of Sao Paulo (ICMC/USP). He obtained his PhD degree in Computer Science (Informatique) at the INPG-IMAG (Institut National Politechnique de Grenoble - France) in 1998. Currently he is Co-Director of the Mobile Robotics Laboratory at ICMC/USP (LRM Lab.). His current research interests are Intelligent Robots, Machine Learning, Computer Vision, Pattern Recognition and Virtual Simulation. He has published an extensive number of journal and conference papers, focused in the above cited research areas.

Denis Fernando Wolf is an Assistant Professor in the Department of Computer Systems at the University of Sao Paulo (ICMC/USP). He obtained his PhD degree in Computer Science at the University of Southern California - USC in 2006. Currently he is Co-Director of the Mobile Robotics Laboratory at ICMC/USP. His current research interests are Mobile Robotics, Machine Learning, Computer Vision, and Embedded Systems. He has published over 50 journal and conference papers over the last years.