Vision and GPS-based Autonomous Vehicle Navigation Using Templates and Artificial Neural Networks

Jefferson R. Souza, Gustavo Pessin, Gustavo B. Eboli, Caio C. T. Mendes,
Fernando S. Osório and Denis F. Wolf
Mobile Robotics Laboratory, University of São Paulo (USP)
Av. Trabalhador São-Carlense, 400 - P.O. Box 668 - 13.560-970 - São Carlos, Brazil
{jrsouza, pessin, caiom, fosorio, denis}@icmc.usp.br
gbuzogany@grad.icmc.usp.br

ABSTRACT

This paper presents a vehicle control system capable of learning to navigate autonomously. Our approach is based on image processing, road and navigable area identification, 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. The images obtained from the camera are classified into navigable and non-navigable regions of the environment using neural networks that control the steering and velocity of the vehicle. Several experimental tests have been carried out under different environmental conditions to evaluate the proposed techniques.

Categories and Subject Descriptors

I.2.9 [**Artificial Intelligence**]: Robotics—*Autonomous Vehicles.*

General Terms

Algorithms, Performance, Design, Experimentation.

Keywords

Robotic Vehicles Navigation, Compass, GPS, Trapezoidal Algorithm, Neural Networks and Urban Environments.

1. INTRODUCTION

Human driving errors are a major cause of car accidents on roads. These errors are caused by a series of in-car distractions, such as using mobile phones, eating while driving, or listening to loud music. Other human errors include drunk driving, speeding, and fatigue.

People often get injured or even die due to road traffic collisions (RTC). Also, bad road and weather conditions increase the risk of RTC. Autonomous vehicles could provide safer conditions in roads, as they can use sensors and actuators to detect and avoid dangerous situations. They could also improve the efficiency in freight transportation and traffic flow in large cities and provide transportation to physically handicapped or visually impaired people.

Figure 1: CaRINA test platform.

Research in mobile robotics 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 the area [9]. Lately, several works have improved navigation in urban environments. Competitions, like DARPA Grand [6] and Urban [7] Challenges and ELROB [1] have been pushing the state of the art in autonomous vehicle control.

The relevant results obtained in such competitions combine information from a large number of complex sensors. Some approaches use five (or more) laser range finders, video cameras, radar, differential GPS, and inertial measurement units [3], [10]. Although there are several interesting applications for such a technology, its cost is 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 the vehicle to reach destination through a safe path. It also detects the navigable regions (roads) and estimates the most appropriate maneuver. Different Artificial Neural Networks (ANNs) topologies are evaluated in order to keep the vehicle in a safe path, and finally control the vehicleŠs steering and acceleration. Figure 1 presents our Intelligent Robotic Car for Autonomous Navigation (CaRINA).

SAC'12 March 25-29, 2012, Riva del Garda, Italy.

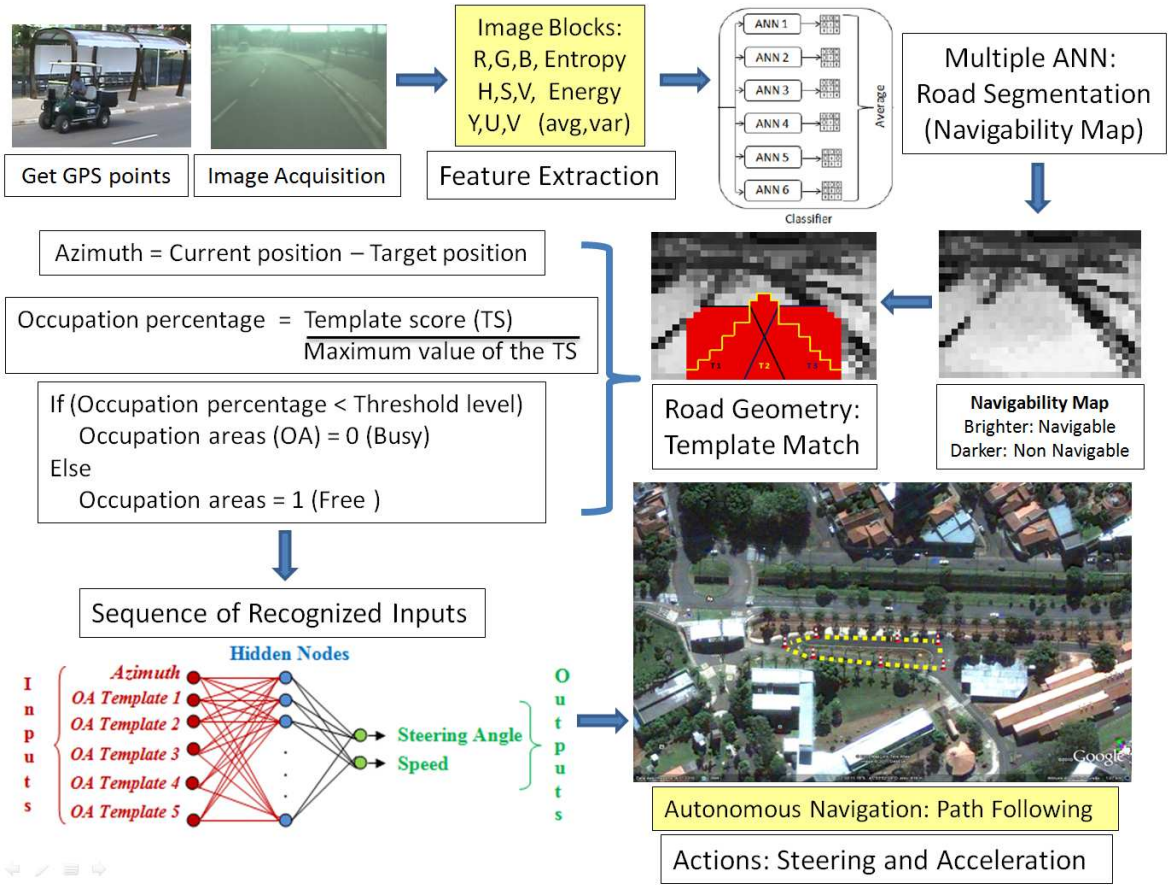


Figure 2: General outline of the vision and GPS based autonomous navigation system.

Our approach is based on two ANNs. The first identifies navigable regions using a template-based algorithm. It classifies the image and identifies the actions that should be taken by CaRINA. Images are acquired and then processed using the ANNs, which identify the road ahead of the vehicle. The value of the azimuth (difference between the current and target positions) as well as a set of values that correspond to navigable and non-navigable regions (free or obstructed) of the road is obtained. These values are used as the input of the second ANN, which aims to learn the control rules (supervised training data) for vehicle control, providing steering and velocity values. Several ANN topologies have been evaluated in order to obtain the minimum training error. The system is detailed in section 3.

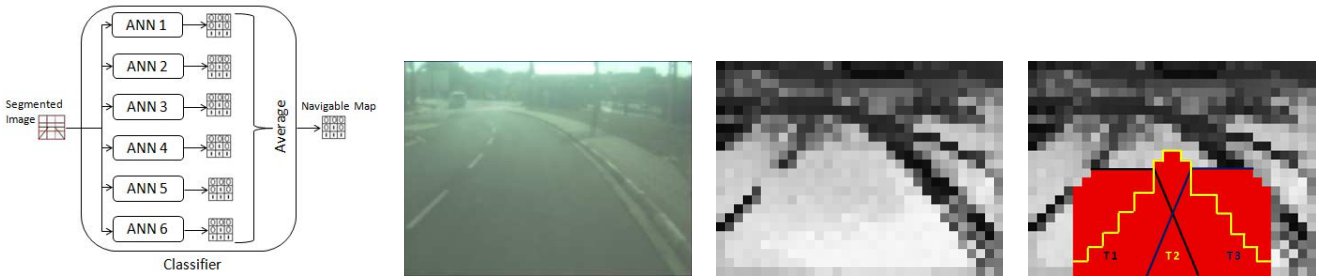
2. RELATED WORKS

Autonomous Land Vehicle in a Neural Network (ALVINN) [11] 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 generated to improve the training, allowing ALVINN to learn how to navigate in new roads. Low resolution of a 30 x 32 image and high computational time are some of the problems found. The architecture comprises 960 input units fully connected to a hidden layer by 4 units, also fully connected to 30 units in an output layer. As this approach requires real time decisions, this topology is not efficient.

Chan et al. [2] shows an Intelligent Speed Adaptation and Steering Control that allows the vehicle to anticipate and negotiate curves safely. It uses Generic Self-Organizing Fuzzy Neural Network (GenSoFNN-Yager) which include the Yager inference scheme [8]. GenSoFNN-Yager has as feature their ability to induce from low-level information in form of fuzzy *if-then* rules. Results show the robustness of the system in learning from example human driving, negotiating new unseen roads. The autonomous driver demonstrates that to anticipate is not always sufficient. Moreover, large variations in the distribution of the rules were observed, which imply a high complexity of the system.

Stein and Santos [18] system's 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. One of the issues found is the impossibility of validating network training without testing it with the real robot.

Markelic et al. [5], proposes a system that learns driving skills based on a human teacher. Driving School (DRIVSCO) is implemented as a multi-threaded, parallel CPU/GPU 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-



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

Figure 3: Classifier structure (a), real image (b), image processing step (c) and template matching (T) (d).

moving objects (IMOs) to spot obstacles with a stereo camera. 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 of 4 steps. In the first step an image is obtained and through ANNs another image is generated, identifying where the road (navigable region) is (Figure 3 (c)). In the second step, a template matching algorithm is used to identify the geometry of the road ahead of the vehicle (straight line, soft turn, or hard turn). In the third step, an occupation percentage is calculated to obtain the occupation areas, verifying if the road is obstructed or free based on the classified image. 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 Image Processing Step

The proposed method by Shinzato [13] was used here, as it applies ANNs to a road identification task. A system composed of six Multilayer Perceptron (MLP) ANNs was proposed to identify the navigable regions in urban environments (Figure 3 (a)). The real image can be seen on Figure 3 (b). The result of this ANNs output combination is a navigability map (Figure 3 (c)). The brighter blocks represent the more likely area to be considered navigable. This step divides an image into blocks of pixels and evaluates them as single units. The advantage is that it can train the ANNs to identify different types of navigable and non-navigable regions (e.g. paved and non-paved roads, sidewalks).

Initially, the image is divided into blocks of pixels, which are individually evaluated. Several features, such as pixel attributes, like RGB average, image entropy and other features obtained from this collection of pixels are calculated for each block. In the grouping step, a frame with (MxN) pixels resolution is sliced into groups with (KxK) pixels. Suppose an image represented by a matrix I of size (MxN). Element $I(m,n)$ corresponds to the pixel in row m and column n of the image, where ($0 <= m < M$) and ($0 <= n < N$). Therefore, group $G(i,j)$ contains all the pixels $I(m,n)$ such that ((i*K) $<= m <$ ((i*K)+K)) and ((j*K) $<= n <$ ((j*K)+K)). This strategy has been used to reduce the amount of data, allowing faster processing.

Once a block has been processed, its attributes are used as inputs of the ANNs. The ANNs are used to classify the blocks considering their attributes (output 0 to non-navigable and 1 to navigable). Each ANN contains an input layer with neurons according to the image input features (see Table 1), one hidden layer with five neurons, and an output layer with only one neuron (binary classification). However, after the training step, the ANN returns real values between 0 and 1 as outputs. These real values can be interpreted as the classification certainty degree of one specific block. The main difference between the six ANNs is the set of image attributes used as input. All these sets of attributes (see Table 1) are calculated during the block-segmentation of the image. Their choice was based on the results of Shinzato [13].

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, 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

After obtaining the six outputs of the ANNs referring to each block, the classifier calculates the average of these values to compose a single final output value. These values represent each block obtained from the original image together with the navigability map matrix (Figure 3(c)) used to identify the most likely navigable region. It is important to mention that the ANN is previously trained using examples of navigable and non-navigable regions selected by the user using the initial image frames. Next, the trained ANN is integrated into the vehicle control system and used as the main source of information for the autonomous navigation control system.

3.2 Template Matching Step

After obtaining the ANN classification, different road templates are placed over the image in order to identify the road geometry. One of them identifies a straight road ahead, two identify soft turns, and two identify hard turns. Each template is composed of a mask of 1s and 0s, as proposed in [17]. The value of each mask is multiplied by the correspondent value into the navigability matrix. The total score for each template is the sum of products. A good performance was obtained in the navigation defined by the urban path avoiding obstacles using the best template [16].

In this step, templates are selected for use in the next step of the system. The template that obtains the higher score is selected as the best match to the road geometry.

3.3 Occupation Percentage

In this step, the templates of the previous step are used to calculate the occupation percentage of the road regions (navigable and non-navigable). This calculation is performed by dividing the template score by its maximum value. This is carried out for each template, so several values normalized between $[0, 1]$ are obtained.

After obtaining these values based on the occupation areas resulting from the classified images (navigable and non-navigable - image processing step), we verify if the occupation percentage obtained is lower than a threshold level. It is then assigned as either obstructed (value 0) or free (value 1) for the occupation areas, which are part of the system input for the next step.

3.4 Learning Based Navigation

We have already developed works analyzing different levels of memory of the templates based on examples obtained from human drivers using neural networks [15]. The results of these neural networks have also been compared with different supervised learning techniques for the same purpose [14]. This work is based in the integration of GPS and compass, furthermore the occupation percentage on the autonomous navigation system.

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 backpropagation (RPROP). The inputs are represented by azimuth (difference between the current and target positions of the vehicle) and the five values obtained by the occupation percentage step (obstructed or free occupation areas). The outputs of the proposed method are the steer angle and speed (Figure 4).

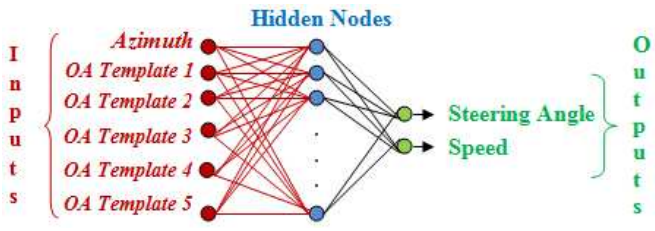


Figure 4: Structure of the second ANN used to generate steering and acceleration commands.

4. EXPERIMENTAL RESULTS

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

Figure 5: GPS coordinates performed by CaRINA.

Seven GPS waypoints (desired trajectory) were defined. 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.

Table 2 shows the values of the path performed by CaRINA. Five different ANNs topologies were analyzed using two learning 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 a near optimal neural 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 learning 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. Furthermore the HT is the activation function most commonly used in neural networks. 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).

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.

Table 2: Results of the ANNs for each topology.

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

The statistical difference of the results was evaluated using Shapiro method [12]. We observed that the null hypothesis, i.e., 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 Man-Whitney method [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 0.05) and for the steering, the null hypothesis (equality) is rejected only between topologies (T) 1 and 5 (the p-values are lower than 0.05). According to this statistical 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 and 8 illustrate the steering angle and velocity of CaRINA 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 original control rules maintained the steering wheel and velocity constant, resulting in the linearity of data (the only problem for the ANN was to learn a rigid curve and high velocity, but this fact did not interfere in the results).

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

5. CONCLUSION AND FUTURE WORKS

Autonomous vehicle navigation is a fundamental task in mobile robotics. This paper has showed a GPS oriented vision-based autonomous navigation system that can be trained to identify the road and navigable regions using ANNs, template matching classification, occupation percentage and learning based navigation. Our approach was evaluated using CaRINA in 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.

As future work, we plan to evaluate other classification methods and decision making algorithms. We also are planning to integrate camera and LIDAR in order to deal with bumps and depressions in the road.

6. ACKNOWLEDGMENTS

The authors acknowledge the support granted by 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] European land-robot (elrob), 2011. http://www.elrob.org/, Access on 26 Aug.

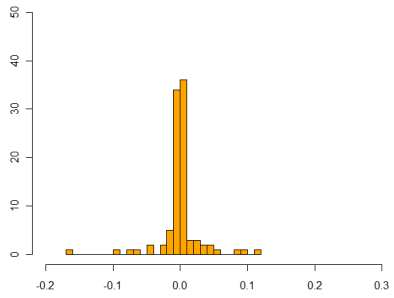
[2] M. Chan, D. Partouche, and M. Pasquier. An intelligent driving system for automatically anticipating and negotiating road curves. *Int. Conf. on Intelligent Robots and Systems*, pages 117–122, 2007.

[3] H. Dahlkamp, A. Kaehler, D. Stavens, S. Thrun, and G. Bradski. Self-supervised monocular road detection in desert terrain. In G. Sukhatme, S. Schaal, W. Burgard, and D. Fox, editors, *In RSS*, 2006.

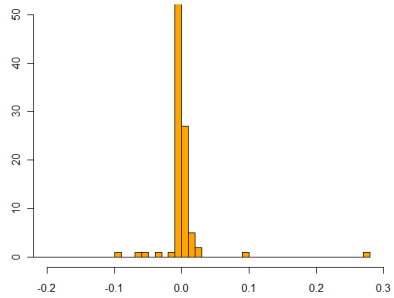
[4] M. P. Fay and M. A. Proschan. Wilcoxon-mann-whitney or t-test? on assumptions for hypothesis tests and multiple interpretations of decision rules. *Statistics Surveys*, 4:1–39, 2010.

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

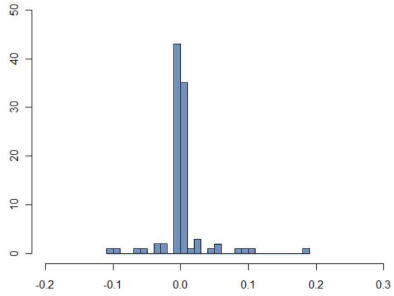
[6] B. Martin, I. Karl, and S. Sanjiv. *The 2005 DARPA Grand Challenge*, volume 36. Springer Tracts in Advanced Robotics, 2007.



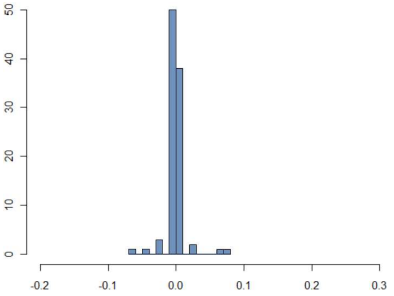
(a) Steering of topology 4 (Test 4 LS)



(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, and (d) velocity of topology 5. The x axis shows the error.

[7] B. Martin, I. Karl, and S. Sanjiv. *The DARPA Urban Challenge*, volume 56. Springer Tracts in Advanced Robotics, 2010.

[8] R. J. Oentaryo and M. Pasquier. Gensofnn-yager: A novel hippocampus-like learning memory system realizing yager inference. *In IJCNN*, pages 1684–1691.

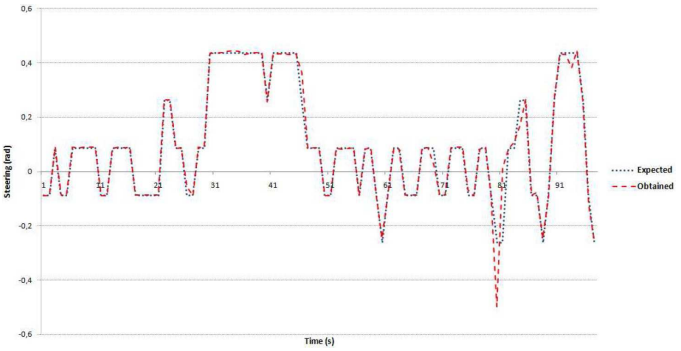


Figure 7: Steering wheel using the test data.

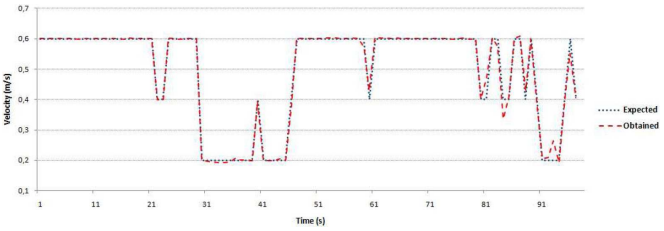


Figure 8: Velocity using the test data.

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

[10] A. Petrovskaya and S. Thrun. Model based vehicle tracking in urban environments. *IEEE ICRA*, 2009.

[11] D. A. Pomerleau. *ALVINN: An Autonomous Land Vehicle In a Neural Network*. Advances In Neural Information Processing Systems, 1989.

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

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

[14] J. R. Souza, G. Pessin, F. S. Osório, and D. F. Wolf. Vision-based autonomous navigation using supervised learning techniques. *In 12th Engineering Applications of Neural Networks (EANN'11)*, 2011.

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

[16] J. R. Souza, D. O. Sales, P. Y. Shinzato, F. S. Osório, and D. F. Wolf. Template-based autonomous navigation and obstacle avoidance in urban environments. *SIGAPP - Applied Comp. Rev.*, 2011.

[17] J. R. Souza, D. O. Sales, P. Y. Shinzato, F. S. Osório, and D. F. Wolf. Template-based autonomous navigation in urban environments. *In 26th ACM Symp. on Applied Computing*, pages 1376–1381, 2011.

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