Path Recognition for Outdoor Navigation Using Artificial Neural Networks: Case Study

Patrick Y. Shinzato, Leandro C. Fernandes, Fernando S. Osorio, Denis F. Wolf
Mobile Robotics Laboratory
Institute of Mathematics and Computer Science
University of Sao Paulo
Sao Carlos - SP, Brazil
shinzato, lnd, denis, fosorio@icmc.usp.br

***Abstract*—Navigation is a broad topic that has been receiving considerable attention from the mobile robotic community. In order to execute a safe navigation on outdoors it is necessary to identify parts of the terrain that can be traversed by the robot and parts that should be avoided. This paper describes an analyses of an image-based terrain identification based on different visual information using a multi-layer perceptron neural network. Experimental tests using an outdoor robot and a video camera have been conducted in real scenarios to evaluate the proposed methods.**

I. Introduction

One of the most desirable features from a mobile robot is the autonomous navigation capability. In order to perform such task, it is necessary the robot be able to obtain information about the environment using sensors and to identify a safe region of the environment [15]. Most research in navigation algorithms is directed to robots that act in structured indoor spaces like offices, where the main focus is obstacle avoidance.

Outdoor space navigation in real scenarios and unknown terrain are certainly more complex problems. Beyond the obstacles avoidance, is necessary that the mobile robot can identify the region where it can navigate safely. The irregularity of the terrain and dynamics environment are some of the factors that make difficult the robot navigation [12].

Several approaches for vision-based navigation have been discussed in the literature. Detect road boundaries through the use of gradient-based edge techniques are describes in [1], [2], [3]. These algorithms assume that road edges are clear and fairly sharp. In [4], Zhang developed an approach that extracts the texture in road images and uses it as a feature for the segmentation of unmarked roads. The approach presented by [5] divide the images in slices and tries to detect the path on each one. A work related with artificial neural network is the Autonomous Land Vehicle In a Neural Network (ALVINN) of [16], where a network is trained to classify the entire image.

Usually it is desirable that the mobile robot can move along the walkway on outdoor environments, rather than move along the grass or gravel. Therefore, these regions composed by grass and gravel are considered non-navigable regions and must be avoided. Since these elements are different from the walkway in color and texture, we can see camera as a good option to identify navigable regions. In this work we present an analysis of classification using neural networks that evaluate different features obtained from image. Also, we tests combinations of different features as input in order to obtain better results.

The next topics of this paper are organized as follows. Section II presents techniques and features used to identify the navigable region in the image. Section III shows the experimental results obtained from tests in real environment. At last, Section IV presents conclusion and future work.

II. Methodology

Navigation in outdoor spaces is considerable more complex than in structured indoor spaces. The terrain is composed by a variety of elements like grass, gardens, sidewalks, streets and gravels. These elements may have different colors and textures allowing to develop a vision-based system. The first step to build a vision-based outdoor navigation system is to classify outdoor spaces into two classes: navigable region and non-navigable region. The navigable region is the surface where a mobile robot can travel safely. After its detection, other algorithms can perform path planning and reactive obstacle avoidance.

Our earlier work [21] focused on determining that the pixels of an image that shows a sidewalk could be linearly distinguishable with an acceptable error rate. This indicates that artificial neural networks can produce acceptable results in the task of land classification. Some simple image features were used as input for artificial neural networks, like RGB data, HSV data, entropy and its combinations. For each feature it has been evaluated four artificial neural network configurations.

A. Block-based Classification Method

A block-based classification method consists in grouping pixels to generate a unique feature value that represents the group. In the grouping step, a frame resolution ($WIDTH$ x $HEIGHT$) pixels was divided in pixel-blocks with (N x N) pixels. For each pixel-block, a feature value is calculated depending on the feature chosen. This strategy has been used to reduce the amount of data and allowing faster processing.

B. Texture - Shannon Entropy

In this work, texture analysis consists on calculate pixels entropy value. In a simple way, entropy can be defined as

978-1-4244-5697-0/10/$25.00

being the regularity degree of a data set [6]. Mathematically, Shannon entropy can be defined as follow:

$$E(X) = -\sum_{x \in X} \mathrm{p}(x) log \mathrm{p}(x)$$

where $\mathrm{p}(x)$ is the probability of x being in the set. So, in this case, x corresponds to the pixel and the pixel-block corresponds to the set. Calculation depends on the space colors used and number of channel used.

C. RGB Color Spaces

The RGB color is a space where each color can be defined by the quantities variation of R (red), G (green) and B (blue) components [7]. The classification based in the color space generates a feature with RGB pixel format. This feature is the weighted average of the pixel occurrence in pixel-block.

Another way of evaluation is based on RGB entropy in the set. In order to obtain the entropy value, it is calculated the frequency of each pixel into the pixel-block. For each pixel with value x, $\mathrm{p}(x)$ is calculated by dividing the frequency of x by the total number of pixel into pixel-block. Note that x and y are a pixel in format RGB, $x = y$ if and only if:

- red of x equals red of y and,
- green of x equals green of y and,
- blue of x equals blue of y

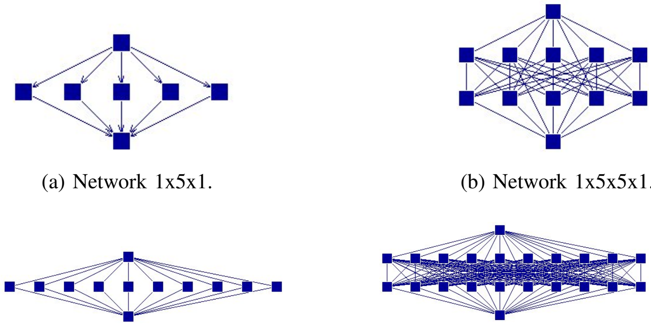
D. HSV Color Spaces

The HSV color space is a space that contains hue (H), saturation (S) and value (V)(brightness) [8]. If the component saturation is equals zero, then hue loses its sense. By maintaining value component to describe the image, it is obtained a gray scale image. The HSV based method uses only hue component or saturation component, generating the pixel-block feature with weighted average based on the hue values or saturation values of the HSV pixels. Also, entropy value has been calculated for channels H and S of HSV.

E. Artificial Neural Networks

Artificial Neural Networks (ANN) are notorious for presenting very own properties such as: adaptability, ability to learn by examples and ability of generalization. In this work, we have used a multilayer perceptron (MLP) [19], which is of a feedforward neural network model that maps sets of input data onto specific outputs. A MLP learning algorithm is the back propagation technique [20], which estimates the weights based on the amount of error in the output compared to the expected results.

In this work, we evaluated four different configurations. One layer with five neurons as show the Fig. 1a, one layer with ten neurons as show the Fig. 1c, two layers, each layer with five neurons and two layers, each with ten neurons, as show the Fig. 1b and Fig. 1d. All networks tested have only one neuron on output layer, enough to classify the pixel-block as navigable (returning 1) or non-navigable (returning 0). The input layer varies depending of combination of evaluated features. If RGB is the evaluated feature then the input layer has three neurons, one neuron for each channel. If the combination of RGB and H entropy are evaluated then the input layer has four neurons.



(a) Network 1x5x1.

(b) Network 1x5x5x1.

(c) Network 1x10x1.

(d) Network 1x10x10x1.

Fig. 1: Diferent topologies of neural network used in terrain classification.

III. EXPERIMENTS AND RESULTS

In order to validate the approach proposed in this paper, several experiments have been carried out at the university campus. We collected data in realistic environments under different texture floor. More specifically, different scenarios composed by grass and walkways with different textures and sun light effects (like shadows) were examined, as shown the Fig. 2. The first one (Fig. 2a) is a straight and flat sidewalk, uniform surface, covered by some leaves of trees planted on the lawns of side of the sidewalk and a building in the background. The Fig. 2b is characterized by presenting a texture generated by the arrangement of sidewalk's bricks, not have a linear trajectory and have a grid of cement on the way. The Fig. 2c is the same environment of Fig. 2a, but in a sunny day creating shadows on the surface of the sidewalk.

Our setup for the experiments was a Pioneer 3 AT robot (Fig. 3) equipped with an A610 Canon digital camera. The image resolution was (320 x 240) pixels and the video has 30 FPS. The robot and camera were used only for data collect. In order to execute the experiments with ANNs, we used a Stuttgart Neural Network Simulator (SNNS) which is a software simulator for neural networks developed in University of Stuttgart. The pixel-block size used was $N = 10$. The openCV [10] library has been used in the image acquisition and to visualize the processed results from SNNS.

Fig. 3: Pioneer 3 AT robot used during the experiments.

Three *Test Sets* have been executed to validate our approach, all of them using the three scenarios. The first *Test*

(a) Scenario 1, leaves on the way. (b) Scenario 2, texture of bricks. (c) Scenario 3, shadows on the way.

Fig. 2: Three diferents enviroments used as scenarios in experiments.

Set compared the performance of different combinations of features, unique features and its combinations two by two. In this *Test Set*, all pixel-blocks from the frame were used. The *Test Set 2* evaluated the performance of neural networks of *Test Set 1* and tested other combinations of input using pixel-blocks below horizon line. The third and last *Test Set* evaluated all combinations of *Test Set 2*, differing in number and arrangement of pixel-blocks used in the training step.

A. Networks Evaluated - Test Set 1

We have used several types of entries: RGB Value, RGB entropy, HSV hue value, HSV hue entropy, HSV saturation, HSV saturation entropy and its combinations two by two. For each input we analyzed the four network topology previously presented. The networks were evaluated at each 100 training cycles until reach 10,000 cycles. For each frame, all blocks were used, except for those that contain margin, resulting in an amount of 660 patterns per frame. The patterns of the first frame were used in training step. The evaluation step was executed using the frames 200, 500, 800 and 1100. The real distance traveled by the robot between frames of evaluation step is approximately 5 meters.

The ANN has been trained to return the value 0 if the feature value was classified as non-navigable and value 1 if classified as navigable. However, the networks provided responses in decimal values between 0 and 1. For this reason we defined responses as follow:

- if result $\leq$ 0.4 then the region is classified as non-navigable;
- if result $\geq$ 0.6 then the region is classified as navigable;
- if result $>$ 0.4 and result $<$ 0.6 then is classified as unknown; Notice that the unknown classification is actually an error value.

During the experiments, some rare times the network didn't converged before than 10,000 cycles. In these cases the answer was a value between 0.4 and 0.6 for all tests. In order to solve this problem we added a condition that verify if the network classified the entire frame as unknown. After adding this condition, if the classification of the entire frame is still unknown until an arbitrary chosen number of cycles then the network is restarted. This problem was associated to random initial weight values of neurons connections.

The experiments presented no significant difference between the different network topologies. More specifically, the results showed that increasing number of neurons in hidden layer brings no or little improvement to the classification results - around 3% - in hit rate. Among 84 experiments, we discuss only the results obtained from the network of one hidden layer with five neurons - 21 experiments - in this *Test Set*.

In Fig. 4, the columns descriptions are: (1) RGB as input; (2) H as input; (3) S as input; (4) S entropy as input; (5) H entropy as input; (6) RGB entropy as input; (7) RGB and H as input; (8) RGB and S as input; (9) RGB and S entropy as input; (10) RGB and H entropy as input; (11) RGB and RGB entropy as input; (12) H and S as input; (13) H and S entropy as input; (14) H and H entropy as input; (15) H and RGB entropy as input; (16) S and S entropy as input; (17) S and H entropy as input; (18) S and RGB entropy as input; (19) S entropy and H entropy as input; (20) S entropy and RGB entropy as input; (21) H entropy and RGB entropy as input;

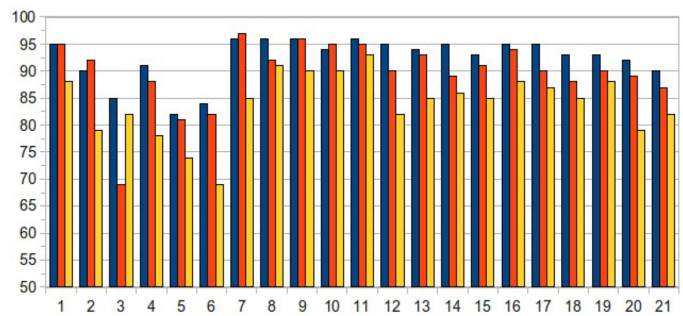
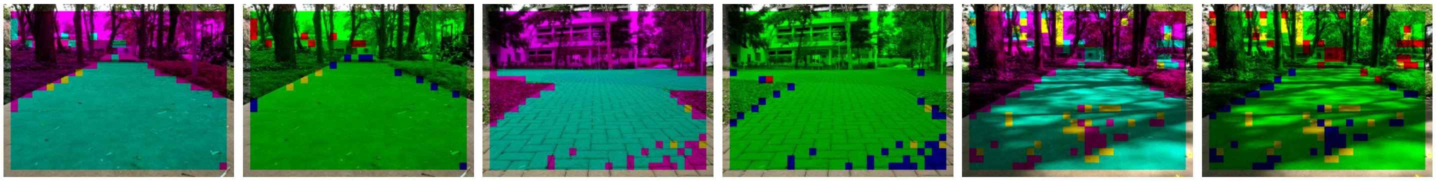


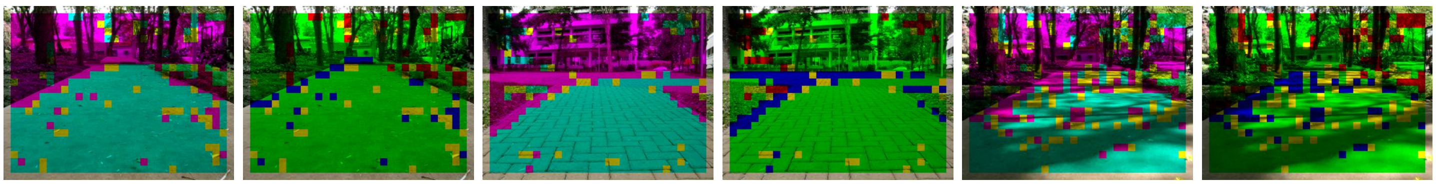
Fig. 4: Graphic Hit Rate for training and evaluation using the entire frame.

Fig. 4 illustrates the hit rate from each method in the different tests scenarios. The blue column represents results in scenario 1, red column represents results in scenario 2 and yellow column represents results in scenario 3. The best result was obtained by the RGB and RGB entropy combination (Fig. 5a), column 11 in Fig. 4 with 92% hit rate. The pixel in cyan color is navigable region, pixel in magenta color is non-navigable region and pixel in yellow color represents unknown region.



(a) Result of Scenario 1. (b) Analysis of Output for Scenario 1. (c) Result of Scenario 2. (d) Analysis of Output for Scenario 2. (e) Result of Scenario 3. (f) Analysis of Output for Scenario 3.

Fig. 5: The Best Result has been obtained to RGB entropy and mean. The thumbs 5a, 5c and 5e shows non-navigation - as magenta - and navigation - as cyan - network responses for each scenario, while 5b, 5d and 5f, shows correct, false-negative, false-positive and unknown classification in green, blue, red and yellow, respectively.



(a) Result of Scenario 1. (b) Analysis of Output for Scenario 1. (c) Result of Scenario 2. (d) Analysis of Output for Scenario 2. (e) Result of Scenario 3. (f) Analysis of Output for Scenario 3.

Fig. 6: The Worse Result has been obtained to H entropy. Misclassification distributed into interest area and below the horizon.

Figs. 5b, 5d and 5f represent the visualization for the neural network output, differing the blocks erroneously classified - false positives (red), false negatives (blue), unknown (yellow) - from the correctly classified (green). In order to evaluate qualitatively results of 10% to 20% of miss rate is acceptable, it is necessary to know the false positives, false negatives and unknown localization and arrangement. If the errors are dispersed, isolated or out of the interesting regions, like the false positives being above the skyline (Fig. 5b), then the miss rate is acceptable. Notice that the false negatives (blue) and unknowns (yellow) occur in the edges, the majority of the cases.

However, if the blocks erroneously classified are grouped and in of the interesting region, like in the Fig. 5d - bottom right region - or as show in Fig. 6d, where the left curve is erroneously classified, generating a failure on terrain classification. Figs. 5c, 5d, 6c and 6d show the results obtained from scenario 2, while Figs. 5e, 5f, 6e and 6f show the results obtained from scenario 3.

B. Networks Evaluated - Test Set 2

Based on the results obtained on *Test Set 1*, we decided to evaluate only the networks with one hidden layer with five neurons. This time, we evaluated the 21 previous combinations and new combinations 3 by 3. Besides that, we combined a channel of RGB with other features, totaling 49 experiments. We evaluated networks trained using the same number of cycles used in *Test Set 1*. The pixel-blocks of the first frame were used in training step, except for those that contain margin, resulting in an amount of 660 patterns per frame. The evaluation step has been executed using the same frames used in *Test Set 1*. However only blocks localized below skyline were used for the evaluation, totaling 516 blocks per frame.

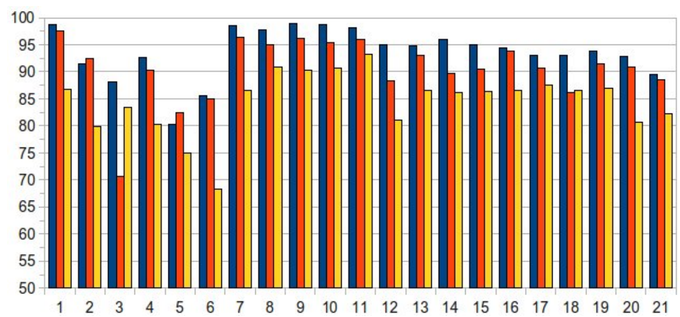
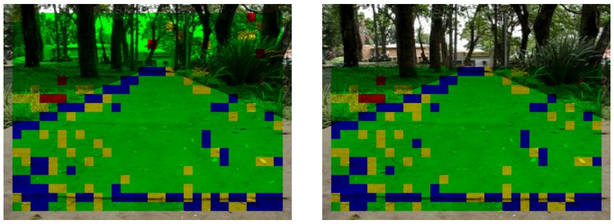


Fig. 7: Graphic Hit Rate for training using the entire frame and evaluation using the region below the horizon line.

Test Set 2 determines the degree of confidence of a classification based on ANN. For the same input sets, in this approach we got a little general increase in hit rate - as show in Fig. 7 - due to the elimination of false positives in region above skyline, where most of the errors classification case. However, considering this test area, a 20% miss rate can mean an unacceptable error.

The most significant improvement happens at first scenario with special gain to (7) and (11) - around 1% to 4%. Note that (5) and (17) got a reduce in correctness percentage, decrease of 2%, both using H entropy. This happened because misclassification occur in blocks that was located below the horizon, as show the Figs. 8a and 8b. The same happens for the other scenarios, with different combinations and proportions.

Based on images 8a and 8b, it can be concluded that learning neural networks is efficient, since appeared many false negatives along the region of the sidewalk, where a fissure



(a) Output from *Test Set 1*, classication of all frame.

(b) Output from *Test Set 2*, classification of the region below the horizon.

Fig. 8: Comparison between *Test Set 1* and *Test Set 2* aswers.

in the walkway appears. On this particular case, the fissure must be regarded as a non-navigable region because its visual information does not match the pattern of the sidewalk visual information. This classification can be considered wrong only if was compared with a human classification, that had other related information.

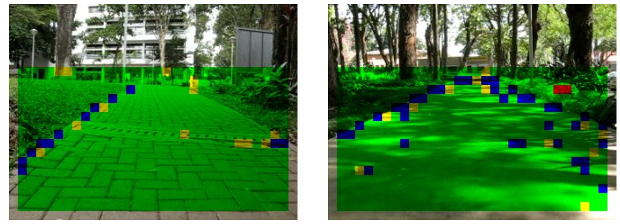
Thus, it is clear that the evaluation of the results of a neural network will be more careful if we use only the region below horizon. Based on this, we evaluated other inputs combinations expecting to increase hit rate.

In Fig. 10, the columns descriptions are: (22) R as input; (23) G as input; (24) B as input; (25) R and H as input; (26) R and S as input; (27) R and S entropy as input; (28) R and H entropy as input; (29) R and RGB entropy as input; (30) G and H as input; (31) G and S as input; (32) G and S entropy as input; (33) G and H entropy as input; (34) G and RGB entropy as input; (35) B and H as input; (36) B and S as input; (37) B and S entropy as input; (38) B and H entropy as input; (39) B and RGB entropy as input; (40) RGB, H and S as input; (41) RGB, H and S entropy as input; (42) RGB, H and H entropy as input; (43) RGB, H and RGB entropy as input; (44) RGB, S and S entropy as input; (45) RGB, S and H entropy as input; (46) RGB, S and RGB entropy as input; (47) RGB, S entropy and H entropy as input; (48) RGB, S entropy and RGB entropy as input; (49) RGB, H entropy and RGB entropy as input;

The results in Fig. 10 showed that components R, G and B alone are not good features for input for the networks. The network obtained good results in scenario 1 but in more complex scenarios (as scenario 2 and 3) , it did not get the same performance. The combination of a RGB channel with another feature was not better than the other combinations evaluated previously. Among experiments 22 at 39, the worst results were on the scenario 3. This is because the RGB incorporates lighting conditions - shade - in color. However, the combination of all channels RGB and others features got the best results - hit rate higher or equal 90% for the 3 scenarios, experiments 41 at 49, as show in Fig. 9a and 9b.

C. Networks Evaluated - Test Set 3

An important detail about algorithm road following that uses ANN is that the knowledge of the ANN must remain compatible even with changes of environment along the way.



(a) Experiment 48, Scenario 2.

(b) Experiment 48, Scenario 3.

Fig. 9: Best Results of *Test Set 2*.

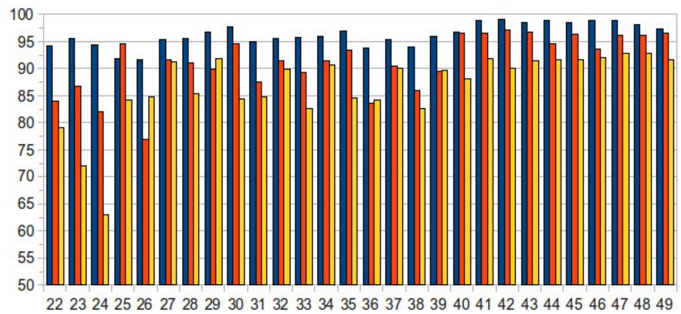
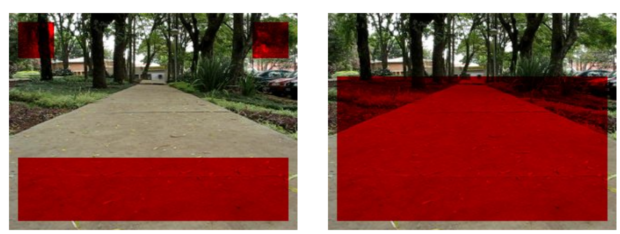


Fig. 10: Graphic of Hit Rate For New Combinations.

Therefore, make a supervised training can be characterized in a limiting or complicating factor. One way for resolve this problem is to define areas where surely all blocks are not navigable or are navigable.

The Fig. 11a show two areas painted of red above horizon line that surely contain blocks considered non-navigable and one greater area below horizon line that only contain blocks considered navigable. This is possible because the camera position is static. Thus, we can make a unsupervised training. We used the same area of evaluation, showed in Fig. 11b, of *Test Set 2* for compare the results between the *Test Sets 2* and *3*. The Figs. 12 and 13 show the results obtained for compare with the Figs. 7 and 10.



(a) Training Area used in *Test Set 3*.

(b) Evaluate Area used in *Test Set 2* and *Test Set 3*.

Fig. 11: Regions used in Neural Network.

The results were good, since the number of training patterns used was less than the previous *test sets*. Around 32 patterns of region non-navigable and 210 patterns patterns of region navigable were used. The results were better or had little difference in the results obtained with the 660 training patterns, as show in Fig. 14b. In most cases evaluated (99%) the network converges in 500 cycles of training. As in *Test Set 2*, more than 20% error is strictly dangerous, as in test 17 (Fig. 14a), where the entire environment has been classified as navigable.

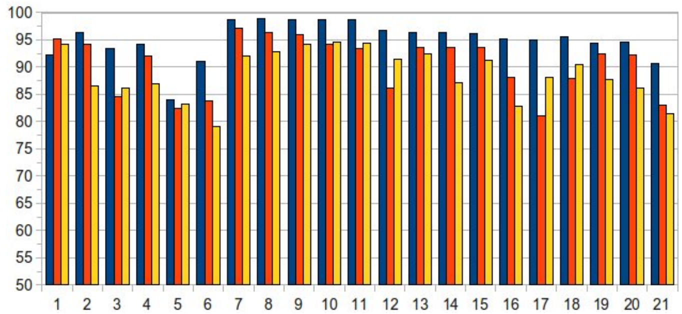


Fig. 12: Graphic Hit Rate for new training area, for the first input set.

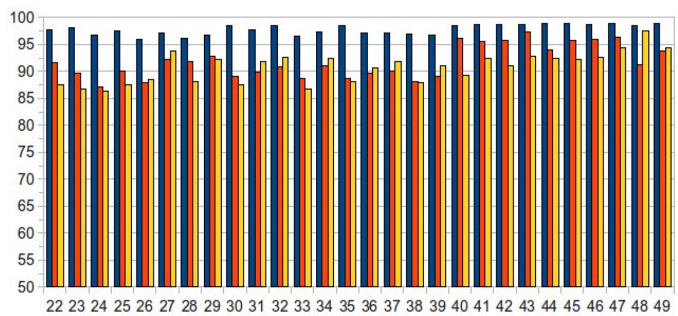
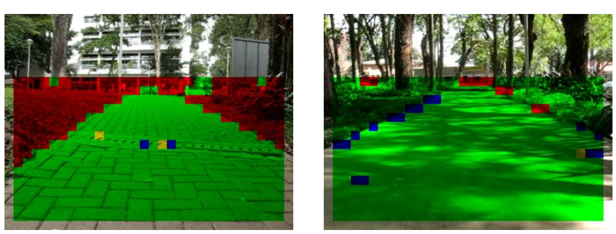


Fig. 13: Graphic Hit Rate for new training area, for second (combined) input set.



(a) Bad Classification. (b) Good Classification.

Fig. 14: Results of *Test Set 3*.

Among all experiments performed, combining RGB with others features has obtained the best result. It has been the more robust classifier in different experiments, regardless of the training or evaluation. The network used was simple with 5 neurons in hidden layer and more or less 30 connections at all. Since it obtained a good convergence with a few training cycles, then the use of artificial neural network makes it suitable for road following techniques, provided that certain conditions are met: training set automatically defined, as showed in *Test Set 3* and define when to retrain.

IV. Conclusion and Future Work

Autonomous navigation is one of the main capabilities of autonomous robots. This paper addresses the problem of identification navigable areas in the environment using artificial neural networks and vision information. Different combinations of network topologies have been evaluated in realistic environments.

The results demonstrate that our approach it is able to detect safe navigation areas even in complex environments with shadows and blocks on the ground. As future work we plan to evaluate others image features and more complex environments. We also plan to integrate our approach with laser mapping, which provides depth information.

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), processes 573963/2008-9 and 08/57870-9.

References

[1] He, Y., Yang, H., Zhang, B., Color-based road detection in urban traffic scenes, *IEEE Trans. Intelligent Transportation Systems*, Dec. 2004.

[2] Broggi, A. and Berte, S., Vision-based road detection in automotive systems: a real-time expectation-driven approach, *Journal of Artificial Intelligence Research*, 1995.

[3] Rotaru, C., Graf, T., Zhang, J., Extracting road features from color images using a cognitive approach, *IEEE Intelligent Vehicle Symposium*, 2004.

[4] Zhang, J., Nagel, H., Texture-based segmentation of road images, *In Proc. IEEE Intelligent Vehicle Symposium*, 1994.

[5] Ghurchian, R., Takahashi, T., Wang, Z., D., Nakano, E., On robot self-navigation in outdoor environments by color image processing, *In Proc. International Conference on Control, Automation, Robotics and Vision*, pp.625-630 2002.

[6] Shannon, C., E., A mathematical theory of comunication, *Bell System Technical Journal*, vol27, 1948.

[7] Joblove, G., H., Greenberg, D., Color spaces for computer graphics. *SIGGRAPH*, Comput. Graph., v. 12, n.3, p.20-25, 1978.

[8] Reiter, C., With j: image processing 2: color spaces. *SIGAPL APL Quote Quad*, v.34, n. 3, p.3-12, 2004.

[9] Rosholm, A., *Statistical Methods for Segmentation and Classification of Images,* PhD thesis, Technical University of Denmark, 1997.

[10] Bradski, G., R., Kaehler, A., Learning OpenCV, O'REILLY, 2008.

[11] Kindermann, R. and Snell, J. L., Markov Random Fields and Their Applications, *American Mathematical Society* , 1980.

[12] Wolf, D. F. and Sukhatme, G. S. and Fox, D. and Burgard, W., Autonomous Terrain Mapping and Classification Using Hidden Markov Models, *International Conference on Robotics and Automation*, 2005.

[13] Ye, C. and Borenstein, J., A new terrain mapping method for mobile robots obstacle negotiation, *UGV Technology Conference at the 2003 SPIE AeroSense Symposium*, pp.52-62, 2003.

[14] YeWeon, I. and Kanade, T., J., High-Resolution Terrain Map from Multiple Sensor Data, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.14(2), pp.278-292, 2002.

[15] Arkin, R., Behavior-Based Robotics, *MIT Press*, 1998.

[16] Dean Pomerleau, Neural Network Vision for Robot Driving, The Handbook of Brain Theory and Neural Networks, 1995.

[17] C. M. Bishop. Neural networks for pattern recognition. Oxford University Press, Oxford, UK, 1996.

[18] S. Haykin. Neural Networks: A Comprehensive Foundation (2nd Edition). Prentice Hall, July 1998.

[19] P. S. Churchland and T. J. Sejnowski. The Computational Brain. MIT Press, Cambridge, MA, USA, 1994.

[20] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. pages 318-362, 1986.

[21] Shinzato, P. Y. and Wolf, D. F. . Path Recognition for Outdoor Navigation. Accepted In: Latin American Robotics Symposium, 2009, Valparaiso, Chile.