Visual Road Recognition Based in Multiple Artificial Neural Networks

Patrick Y. Shinzato and Fernando S. Osorio and Denis F. Wolf
Mobile Robotic Laboratory,
Institute of Mathematics and Computer Science,
University of Sao Paulo - ICMC-USP,
Sao Carlos, SP Brazil
Email: shinzato@icmc.usp.br, fosorio@icmc.usp.br, denis@icmc.usp.br

***Abstract*—The development of autonomous vehicles is a highly relevant research topic in mobile robotics. Road recognition using visual information is an important capability for autonomous navigation in urban environments. Over the last three decades, a large number of visual road recognition approaches have been appeared in the literature. This paper propose a novel visual road detection system based on multiple ANN that can identify the road based on color and texture. Several features are used as inputs of the ANN such as: average, entropy, energy and variance from different color channels (RGB, HSV, YUV). As a result, our system is able to estimate the classification and the confidence factor of each part of the environment detected by the camera. Experimental tests have been performed in several situations in order to validate the proposed approach.**

I. Introduction

Visual road recognition is one of the desirable skills to improve autonomous vehicles systems. As a result, visual road recognition systems have been developed by many research groups since the early 1980s, such as [1] [2] [3]. Details about these and others researches can be found in several surveys [4] [5] [6] [7].

Most of works developed before the last decade were based on certain assumptions about specific features of the road, such as lane markings [8] [9], geometric models [10] and road boundaries [11]. These systems have limitations or some kind of weakness because in most cases they showed satisfactory results only in autonomous driving on paved, structured and well-maintained roads. Furthermore they needed for favorable conditions of weather and traffic. Autonomous driving on unpaved or unstructured roads, and adverse weather conditions have been well-studied in the last decade [12] [13] [14] [15]. We can be highlight developed systems for the DARPA Grand Challenge [16] [17] like [18] [19] [20] focusing on desert roads.

One of the most representative works is the NAVLAB project [3]. Systems known as SCARF [21], UNSCARF [10], YARF [22], ALVINN [23], MANIAC [24] and RALPH [25] were developed by the same research group. Among these systems, the most relevant for this paper are the ALVINN and MANIAC because they also use artificial neural networks (ANN) to road recognition.

The idea of ALVINN consists of watching a human driver in order to learn the steering of wheels when a driving on roads on varying conditions. The learning step takes a few minutes [26] and the authors mention that if is necessary a retraining then this is a shortcoming [25]. Other aspect pointed out by the authors is that the retraining process invariably requires human intervention. According to [24], the major problem of ALVINN is lack of ability to learn features which would allow the system to drive on road types other than that on which it was trained.

In order to improve the autonomous control, MANIAC (Multiple ALVINN Networks In Autonomous Control) [24] has been developed. In this system, several ALVINN networks must be trained separately on their respective roads types that are expected to be encountered during driving. Then the MANIAC system must be trained using stored exemplars from the ALVINN training runs. If a new ALVINN network is added to the system, MANIAC must be retrained. Both systems trained properly, ALVINN and MANIAC, can handle nonhomogeneous roads in various lighting conditions. However, the approachs only works on straight roads [12].

Other group that developed visual road recognition based on ANN was the Intelligent Systems Division of the National Institute of Standards and Technology [27] [28]. They developed a system that make use of a dynamically trained ANN to distinguish between areas of road and nonroad. This approach is capable of dealing with nonhomogeneous road appearance if the nonhomogeneity is accurately represented in the training data. In order to generate training data, three regions from image were labeled as road and three others regions as nonroad, i.e., the authors made assumptions about the location of the road in the image, which causes problems in certain traffic situations. A later work [29] proposed dynamic location of regions labeled as road in order to avoid these problems. However, in shadows situations the new system less accurately than the previous system.

In this work, we present a visual road detection system that use multiple ANN, similar to MANIAC, in order to improve the robustness. However, our ANN learn colors and textures from sub-images instead of all road appearance that ALVINN learn. In our system, each ANN receives differents image features as input. Features like averages, entropy, energy and variance from differents color channels (RGB, HSV, YUV) from sub-images. In the final step of algorithm, we combine a set ANN's outputs to generate only one classification for each sub-image. This classification provides confidence factor for each sub-image classification of image that can be used by control algorithm. Unlike [27], our system does not need to be retrained all the time, therefore we don't need assume the location of the road.

II. Algorithm Description

The overall visual system consists of two major stages: First is "(A) Generate features of all sub-images", where the image is transformed into a sub-images set. For each sub-image, several features was generated that will be used in classification stage. The second stage is "(B) Classification of sub-images", where we use a set of ANN that receives features as input to decide if the sub-image represents a navigable region or not. After that, we combine all ANN outputs to generate a matrix that is the "Visual Navigation Map" (VNMap). A control algorithm use VNMap in order to control the vehicle autonomously. Upon completion of the action, the system capture another image from environment and returns to stage (A). The Fig. 1 show how the system works for a sample with five ANN. As our system uses ANN, is necessary a stage of training. This step is detailed in Section II-C after the system description.

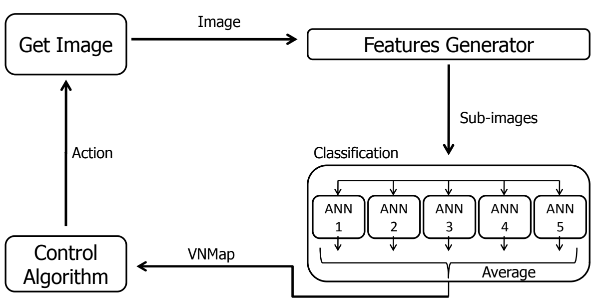
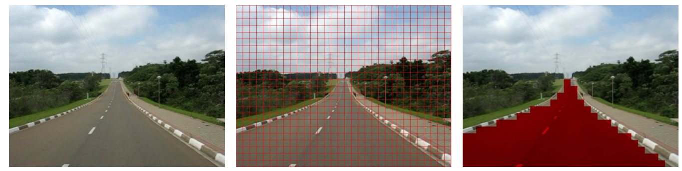


Fig. 1. The System Architecture after the training step. This paper addresses only the generate features and classification. Given an image, it is transformed into a set of sub-images that will be classified separately by multiple ANN. Combining all ANN outputs, is generated the classification for one sub-image. The classifications from all sub-images compose the VNMap that is used by a control algorithm.

A. Generate Features Of All Sub-images

In this part, the system generate features for each sub-image from set of images. This method consists to process and evaluating a collection of pixels directly connected, neighbors, as a group. In other words, a image resolution (M x N) pixels was decomposed in groups with ($K \times K$) pixels, as show Fig. 2(a) and Fig. 2(b). More specifically, suppose an image represented by a matrix I of size ($M \times N$). The element $I(m,n)$ corresponds to the pixel in row m and collumn n of image, where ($0 \leq m < M$) and ($0 \leq n < N$). Therefore, group $G(i,j)$ contains all the pixels $I(m,n)$ such that $(\ (i * K) \leq m < ((i * K) + K)\ )$ and $(\ (j * K) \leq n < ((j * K) + K)\ )$. For each group, many image features are calculated. These features will be used as input for the ANN that determining whether the sub-image represents a region navigable or non-navigable. If the sub-image is classified as navigable then all pixels from group are considered as navigable, as show Fig.2(c). This strategy has been used to reduce the amount of data, allowing faster processing and obtaining information like texture.



(a) (b) (c)

Fig. 2. In generate features stage, the image (a) is transformed into set of sub-images that represents each square from the image (b). After the classification, we can obtained results like (c), where all pixels from the square receive the same classification.

We use several statistical measures as image features. Simple measures, like mean and more complicated, like entropy and variance, were used. We calculated each statistical measure with each color channel from image. We use 3 color spaces: RGB, HSV, YUV and also we generate RGB normalized. The Table I show all features calculated for this work. The choice of these attributes was based on a feature selection method based on ANN.

TABLE I
Features calculated in this work. Note that RN, GN, BN are RGB channels normalized.

Measure	Channels from several color spaces										
	R	G	B	H	S	V	Y	U	RN	GN	BN
Mean	×	×		×	×	×		×		×	×
Entropy				×			×			×	
Variance			×								
Energy										×	

B. Classification Of Sub-images

The classification stage consists in run each sub-image in several ANN, where each ANN receives differents features. In this work, we have used a multilayer perceptron (MLP) [30], which is of a feedforward neural network model that maps sets of input data onto specific outputs. We uses the back propagation technique [31], which estimates the weights based on the amount of error in the output compared to the expected results.

The ANN's topology consists, basically, two layers, where the hidden layer has five neurons and the output layer has only one neuron, as show the Fig. 3. We chose this topology based on previous evaluation [32]. The ANN was trained for returns value "0.0" when receives patterns representing non-navigable regions and return "1.0" when receives patterns representing navigable regions. The input layer depends of features chosen for each ANN instance. Since the number of neurons is small then the training time is also reduced, enabling the instantiation of multiple ANN.

After evaluate the sub-image in all ANN availables, we calculate the average of all outputs. This value is used to

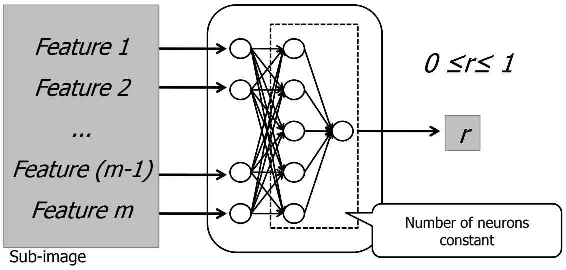


Fig. 3. ANN topology: The ANN use some features, not all, to classify the sub-image between navigable and non-navigable. The output is a real value ranging from 0.0 to 1.0, where if the value is closer to 1 then greater will be the confidence factor about the navigability of pixels. If the value is closer to 0 then greater will be the confidence factor about the non-navigability of pixels.

compose the VNMap, which consists of a matrix of probabilities. The system appears to have more certainty regarding the classification if the value is closer to the extremes, i.e., 0 or 1. Fig.4 shows a sample of image classified, where the right image show in gray-scale the VNMap - Black represents non-navigable, white represents navigable and the gray represents the intermediate values. This VNmap that can be used by some control algorithm.

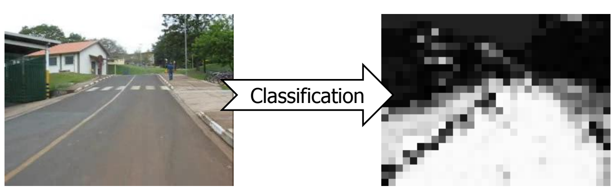


Fig. 4. Results from a classification sample. Black represents non-navigable, white represents navigable and the gray represents the intermediate values.

C. Training Step

The training multiple ANN is offline and supervised, i.e., the supervisor selects navigable and non-navigable parts of training images. Based on marks defined, the system generate all image features that will be used by ANN with the expected output. These patterns are the training data.

Regarding ANN convergence evaluation , two metrics are frequently used: "MSE" and "HIT Hate". The MSE is "Mean-Square Error" and usually the taining step stops when the "MSE" converges to zero or some acceptable value. However, a small mean-square error does not necessarily imply good generalization [33]. Also this metric does not provides how many patterns are missclassified, i.e., if the error is high in some patterns or if the error is evenly spread in all patterns. Other way is check how many patterns were classified correctly (Hit Rate). In this case, the problem is define a good precision to interpret the ANN output, since the output not is exactly the value expected.

Seeking a more adequate assessment to the proposed problem, we used a method that assigns a weight to each error precision and calculates a score for the ANN. In other words, a pattern with a error of 0.1 has a greater weight than an error of 0.2 in the score. Therefore, a higher score indicates few errors or several small errors. Based on this, VNMaps with more certainty (more white and black blocks) will assign high scores to ANN that spread it. The Equation 1 show how to calculate the score:

$$score = \frac{\frac{1}{N*p(0)}\left(\sum_{i=0}^{hmax} h(i)*p(i)\right)+1}{2.0}, \tag{1}$$

where N is the number of evaluated patterns, $hmax$ is the size of $h()$ and $p()$ is the weight function. The value $hmax$ determines a precision for the interpretation of ANN's output. In other words for instance, if $hmax = 10$ then the output that is a real value ranging 0 and 1 will be divided in 10 classes of errors, one class for errors between 0 and 0.1, other class for errors between 0.1 and 0.2, and so on. Also, each class is directly related with a weight.

III. Experiments and Results

In order to validate the proposed system, several experiments were perfomed. Our setup for experiments was a car equipped with an A610 Canon digital camera. The image resolution was (320×240) pixels with 30 FPS. The car and camera were used only for data collection. In order to execute the experiments with ANNs, we used a Fast Artificial Neural Network (FANN) [34] which is a C library for neural networks. The OpenCV library [35] has been used in the image acquisition and to visualize the processed results from system. The sub-image size used was $K = 10$, making each image has 768 sub-images.

The system instantiates five ANN of same type and selects the ANN with best score. In this experiment, the system trains each ANN five times until 5000 cycles. Our $hmax$ was "10" and we used a linear weight funtion where starts with "1" to "-1". Other important information is that three set of features are used as input for ANN. In other words the system has three types of ANN. For each type, it was instantiated two ANN. Thus, our system uses six ANN to classify the images. The features sets used are:

- Mean of U, V, H and Normalized B; Entropy of H; Energy of Normalized G.
- Mean of V, G, U, R, H, Normalized B, Normalized G; Entropy of H and Y; Energy of Normalized G.
- Mean of U, Normalized B, V, S, H, Normalized G; Variance of B; Entropy of Normalized G.

Many paths covered by a vehicle were taped using a video camera. These paths are composed by road, sidewalks, parking, buildings, and vegetation. Also, some stretchs presents adverse conditions such as dirt, traces of other vehicles and shadows (Fig. 5). Altogether, data were collected from eight scenarios. For each one, it was created a training database and evaluating database. Also, we combined elements of the eight training data in a single database and the eight evaluating data in other single database. Thus, we tested the system with nine databases.

In order to verify the cability to generalization, the system was been trained with each training database but we evaluated

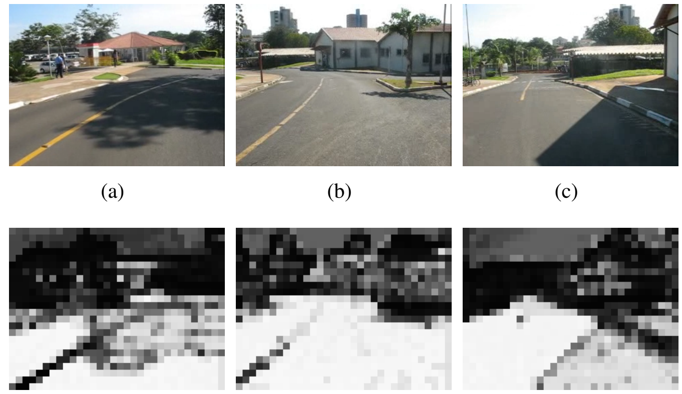
Fig. 5. Samples of scenarios used in this work. Note the occurrence of shadows, and how the colors of road changed in different lighting conditions.

with all evaluating databases. Table II show all normalized scores.

TABLE II
RESULTS OF SYSTEM FOR ALL DATABASES. EACH COLUMN REPRESENTS THE SCORE REACHED BY SYSTEM TRAINED WITH SCENERY DESCRIBED IN FIRST COLUMN. THE SCORE ARE NORMALIZED.

Training Model	Evaluation Sets S 1	S 2	S 3	S 4	S 5	S 6	S 7	S 8	S 9
S 1	0.94	0.89	0.89	0.89	0.92	0.92	0.89	0.90	0.91
S 2	0.99	0.97	0.84	0.96	0.88	0.96	0.93	0.89	0.92
S 3	0.92	0.87	0.89	0.87	0.90	0.89	0.89	0.89	0.89
S 4	0.84	0.94	0.79	0.96	0.84	0.94	0.91	0.87	0.89
S 5	0.91	0.86	0.87	0.87	0.92	0.89	0.88	0.87	0.88
S 6	0.84	0.94	0.82	0.95	0.84	0.95	0.90	0.86	0.89
S 7	0.75	0.73	0.68	0.72	0.78	0.77	0.96	0.93	0.77
S 8	0.85	0.76	0.79	0.76	0.85	0.82	0.95	0.95	0.83
S 9	0.92	0.93	0.87	0.93	0.91	0.93	0.91	0.90	0.91

The line "S 1" represents the system trained with training database of Scenery 1. In general, we obtained a good result since the worst result of this line is 0.89. The result on a unknow scenery never presented in training step was 0.89. The system trained on Scenery 2 also had a good performance with worst result of 0.84. Another important detail that could be observed in the Table II, is that scenario 3 is the most difficult to classify among all other scenarios. In this case the best score in column "S 3" is 0.89 when the system was trained with patterns of same scenario. This is because the high occurrence of shadows on the road which causes the system to decrease the degree of certainty of the classification of road-pixel. Another issue is also the variation of the colors of the sky that also decreased the degree of certainty about the non-navigability. Fig 6 shows same images of scenery 3 and the VNMaps generated. Even with a high degree of certainty on the road, note that the certainty of non-navigability of the sky is still low.



(a) (b) (c)

(d) (e) (f)

Fig. 6. The images (a)-(c) are the original images from Scenery 3 and (d)-(f) are VNMaps generated by system. Note that the certainty of non-navigability of the sky is low. These VNMaps are generated by system trained with patterns of same scenery.

In general, an indication of good generalization of the system is the results of the last column from Table II, which is the database of all patterns of evaluation. In this column, almost all trained systems reached good results. Also, good results appear in the last line, where the system was trained with patterns of all databases. In order to ilustrate the results obtained, Fig. 7 shows original images, the VNMap and the VNMap over the image. Note that in the third row of images, a large shadow appears on the road, nevertheless, the system could correctly identify the road. It is important to note that the classification obtained in the shadow region has a lower degree of certainty than the the rest of the image. Also note that the degree of certainty about non-navegability of sky is relatively high.

Based on the experiments, we concluded that results of 0.85 or more represents a satisfatory classification. We can see a poor result in the Fig. 8, which is same scenery of Fig. 7. Except that the system has been trained with patterns of scenery 7. This classification has low degree of certainty, failing to classify areas affected by shadows and made some mistakes in the sky. This poor result happens due to the simplicity of scenery 7 that not contains shadows, buildings or lightings changes. The images from Fig. 9 show difference between scenarios 1 and 7. Based on theses observations, the system could not maintain a good classification if the path change drastically or if the system receives a incomplete training database. When the system do not reaches the score of "0.85" then it is necessary that the supervisor rebuilds the training database. This necessity is clearly demonstrated by looking at the last row of the Table II because if the training database is rich in many situations then the system reaches good performance in many scenarios.

Other experiment performed used others scenarios that showed that the proposed system is able to learn characteristics

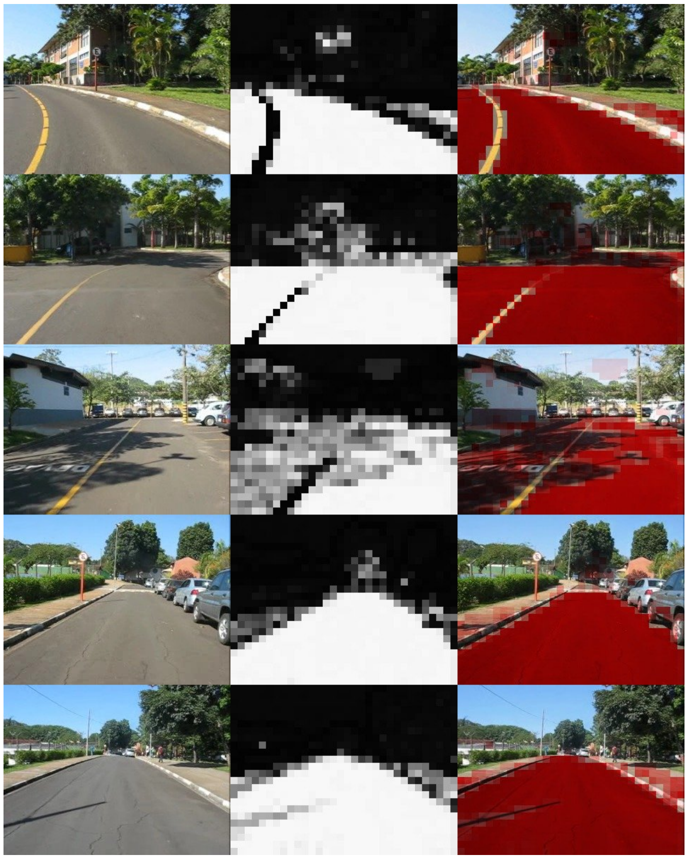


Fig. 7. Samples of a good results. Results in scenery 1. Left images are the original image; Middle images are the VNMap; Right images are the VNMap on the image. Note in the third row of images much of the street is covered by a shadow of a tree. Despite the adverse conditions, the system could identify the street but with a lower degree of certainty that the the rest of the image. In the training step, this system obtained a score 0.94.

of different types of roads in different weather conditions. This scenarios, never tested during the developed stage of system, are dirty road and road under the rain. The images in Fig. 10 shows good results obtained in these cases. Note that the degree of certainty about the non-navegability of background is very high and degree of certainty about the navegability is high for all road-blocks in scenery (a) and high for almost road-blocks in scenery (b).

IV. Conclusion

Visual road recognition is one of the desirable skills to improve autonomous vehicles systems. We present a visual road detection system that use multiple ANN. Our ANN is capable to learn colors and textures instead of totally road appearance. Also, our training evaluation method is a more adequate assessment to the proposed problem, since many results with many low degrees of certainty leads a low score. Finally, the system classification provides confidence factor for each pixel-group classification of image that can be used by a control algorithm.

In general the results obtained in the experiments were satisfactory, since the system reached good classification results when the training step obtains good score. Furthermore, the

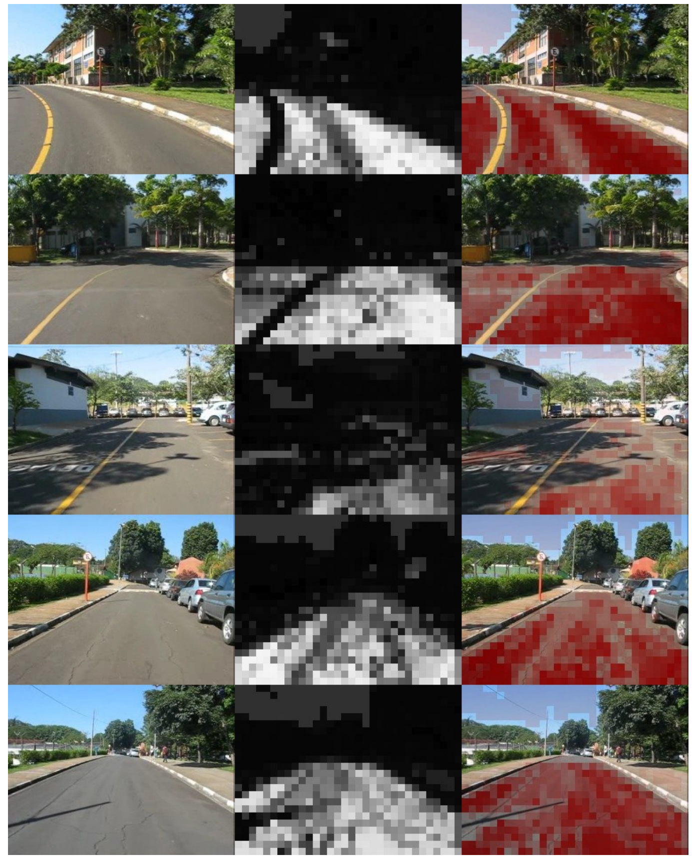
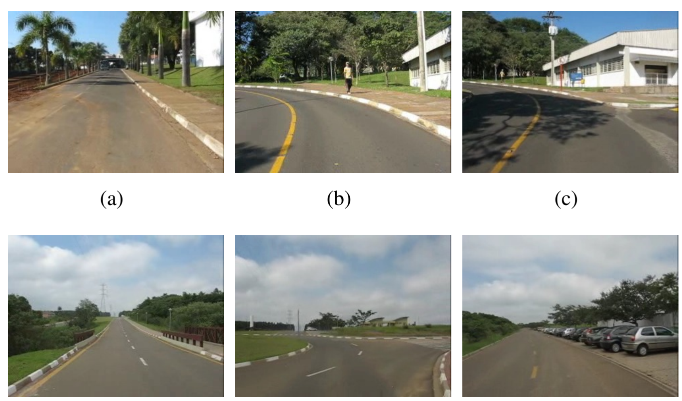


Fig. 8. Samples of a poor results. Results in scenery 1. In the training step, this system obtained a score of 0.75 due to system be trained over patterns of scenery 7.

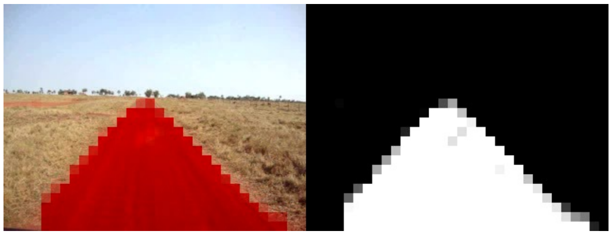


(a) (b) (c)

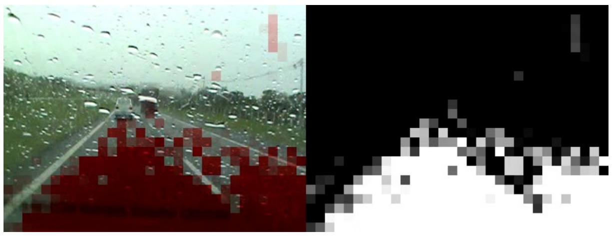
(d) (e) (f)

Fig. 9. The images (a)-(c) are the Scenery 1 and (d)-(f) are scenery 7. Note that the scenery 7 is very simple if compared with scenery 1 because doesn't have shadows, dirty and buildings.

set of features presented in this paper can be used in differents road types and weather conditions. It may be possible to get a better classification if we add a preprocessing to reduce the influences of the shadows in the image and maybe use more ANN in the system. As future work we plan to integrate with other visual system like "lane detection" in order to improve the system in urban scenarios. We also plan to integrate our approuch with laser mapping in order to make conditions



(a) Dirty road.



(b) Road in the rain.

Fig. 10. These scenarios never was tested during the system development. Note that the degree of certainty is very high.

to retrain the ANN without human intervention and without making assumptions about the image. Finally, as the system classifies each block independently, we intend to improve the processing efficience using a GPU.

Acknowledgment

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] A. Broggi, M. Bertozzi, and A. Fascioli, “Argo and the millemiglia in automatico tour,” *Intelligent Systems and their Applications, IEEE*, vol. 14, no. 1, pp. 55 –64, January 1999.

[2] R. Gregor, M. Lutzeler, M. Pellkofer, K. Siedersberger, and E. Dickmanns, “Ems-vision: a perceptual system for autonomous vehicles,” in *Intelligent Vehicles Symposium, 2000. IV 2000. Proceedings of the IEEE*, 2000, pp. 52 –57.

[3] C. Thorpe, M. Hebert, T. Kanade, and S. Shafer, “Vision and navigation for the carnegie-mellon navlab,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 10, no. 3, pp. 362 –373, May 1988.

[4] F. Bonin-Font, A. Ortiz, and G. Oliver, “Visual navigation for mobile robots: A survey,” *Journal of Intelligent & Robotic Systems*, vol. 53, pp. 263–296, 2008, 10.1007/s10846-008-9235-4.

[5] G. Desouza and A. Kak, “Vision for mobile robot navigation: a survey,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 2, pp. 237 –267, Feb. 2002.

[6] M. Bertozzi, A. Broggi, and A. Fascioli, “Vision-based intelligent vehicles: State of the art and perspectives,” *Robotics and Autonomous Systems*, vol. 32, no. 1, pp. 1–16, 2000.

[7] E. D. Dickmanns, “Vehicles capable of dynamic vision: a new breed of technical beings?” *Artif. Intell.*, vol. 103, pp. 49–76, August 1998.

[8] A. Takahashi, Y. Ninomiya, M. Ohta, and K. Tange, “A robust lane detection using real-time voting processor,” in *Intelligent Transportation Systems, 1999. Proceedings. 1999 IEEE/IEEJ/JSAI International Conference on*, 1999, pp. 577 –580.

[9] M. Bertozzi and A. Broggi, “Gold: a parallel real-time stereo vision system for generic obstacle and lane detection,” *Image Processing, IEEE Transactions on*, vol. 7, no. 1, pp. 62 –81, Jan. 1998.

[10] J. Crisman and C. Thorpe, “Unscarf, a color vision system for the detection of unstructured roads,” in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 3, April 1991, pp. 2496 – 2501.

[11] A. Broggi and S. Bert, “Vision-based road detection in automotive systems: A real-time expectation-driven approach,” *Journal of Artificial Intelligence Research*, vol. 3, pp. 325–348, 1995.

[12] C. Tan, T. Hong, T. Chang, and M. Shneier, “Color model-based real-time learning for road following,” in *Intelligent Transportation Systems Conference, 2006. ITSC ’06. IEEE*, September 2006, pp. 939 –944.

[13] F. Diego, J. Alvarez, J. Serrat, and A. Lopez, “Vision-based road detection via on-line video registration,” in *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, September 2010, pp. 1135 –1140.

[14] J. Alvarez and A. Lopez, “Novel index for objective evaluation of road detection algorithms,” in *Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on*, October 2008, pp. 815 –820.

[15] M. Manz, F. von Hundelshausen, and H.-J. Wuensche, “A hybrid estimation approach for autonomous dirt road following using multiple clothoid segments,” in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, May 2010, pp. 2410 –2415.

[16] DARPA04, http://www.darpa.mil/grandchallenge04/, 2004, visited in February/2011.

[17] DARPA05, http://www.darpa.mil/grandchallenge05/, 2005, visited in February/2011.

[18] H. Dahlkamp, A. Kaehler, D. Stavens, S. Thrun, and G. Bradski, “Self-supervised monocular road detection in desert terrain,” in *Proceedings of Robotics: Science and Systems*, Philadelphia, USA, August 2006.

[19] D. Lieb, A. Lookingbill, and S. Thrun, “Adaptive road following using self-supervised learning and reverse optical flow,” in *Proceedings of Robotics: Science and Systems*, Cambridge, USA, June 2005.

[20] J. Lee, C. Crane, S. Kim, and J. Kim, “Road following in an unstructured desert environment using monocular color vision as applied to the darpa grand challenge,” in *International Conference on Control, Automation, and Systems (ICCAS)*, 2005.

[21] J. Crisman and C. Thorpe, “Scarf: A color vision system that tracks roads and intersections,” *IEEE Trans. on Robotics and Automation*, vol. 9, no. 1, pp. 49 – 58, February 1993.

[22] K. Kluge and C. Thorpe, “The yarf system for vision-based road following,” *Mathematical and Computer Modelling*, vol. 22, no. 4-7, pp. 213 – 233, 1995.

[23] D. A. Pomerleau, “Neural network vision for robot driving,” in *The Handbook of Brain Theory and Neural Networks*. University Press, 1996, pp. 161–181.

[24] T. Jochem, D. Pomerleau, and C. Thorpe, “Maniac: A next generation neurally based autonomous road follower,” in *Proceedings of the International Conference on Intelligent Autonomous Systems*, February 1993.

[25] D. Pomerleau and T. Jochem, “Rapidly adapting machine vision for automated vehicle steering,” *IEEE Expert*, vol. 11, no. 2, pp. 19 –27, Apr. 1996.

[26] D. Pomerleau, “Efficient training of artificial neural networks for autonomous navigation,” *Neural Computation*, vol. 3, no. 1, pp. 88–97, 1991.

[27] M. Foedisch and A. Takeuchi, “Adaptive real-time road detection using neural networks,” in *Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on*, October 2004, pp. 167 – 172.

[28] P. Conrad and M. Foedisch, “Performance evaluation of color based road detection using neural nets and support vector machines,” in *Applied Imagery Pattern Recognition Workshop, 2003. Proceedings. 32nd*, 2003, pp. 157 – 160.

[29] M. Foedisch and A. Takeuchi, “Adaptive road detection through continuous environment learning,” in *Applied Imagery Pattern Recognition Workshop, 2004. Proceedings. 33rd*, October 2004, pp. 16 – 21.

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

[31] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning internal representations by error propagation,” pp. 673–695, 1988.

[32] P. Shinzato and D. Wolf, “A road following approach using artificial neural networks combinations,” *Journal of Intelligent & Robotic Systems*, pp. 1–20, 2010, 10.1007/s10846-010-9463-2.

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

[34] FANN, http://leenissen.dk/fann/html/files/fann-h.html, 2011, visited in February/2011.

[35] G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library*. Cambridge, MA: O’Reilly, 2008.