Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Computer Science 18 (2013) 240 – 249

Procedia Computer Science

2013 International Conference on Computational Science

Comparing support vector machines and artificial neural networks in the recognition of steering angle for driving of mobile robots through paths in plantations

Danilo S. Jodas[a,*], Norian Marranghello[a], Aledir S. Pereira[a], Rodrigo C. Guido[a]

[a]Sao Paulo State University, Rua Cristóvão Colombo, 2265, São José do Rio Preto, 15054-000, Brazil

Abstract

The use of mobile robots turns out to be interesting in activities where the action of human specialist is difficult or dangerous. Mobile robots are often used for the exploration in areas of difficult access, such as rescue operations and space missions, to avoid human experts exposition to risky situations. Mobile robots are also used in agriculture for planting tasks as well as for keeping the application of pesticides within minimal amounts to mitigate environmental pollution. In this paper we present the development of a system to control the navigation of an autonomous mobile robot through tracks in plantations. Track images are used to control robot direction by pre-processing them to extract image features. Such features are then submitted to a support vector machine and an artificial neural network in order to find out the most appropriate route. A comparison of the two approaches was performed to ascertain the one presenting the best outcome. The overall goal of the project to which this work is connected is to develop a real time robot control system to be embedded into a hardware platform. In this paper we report the software implementation of a support vector machine and of an artificial neural network, which so far presented respectively around 93% and 90% accuracy in predicting the appropriate route.

Keywords: Mobile Robotics, Image Processing, Support Vector Machine, Artificial Neural Networks

1. Introduction

In the recent years, there was a significant increase in the use of mobile robots in several areas. This increase is due to the execution of activities in places of difficult access or in situations that are potentially harmful to humans. The appearance of intelligent algorithms was a factor that significantly contributed to the advancement of mobile robots due to the possibility of development of intelligent agents that autonomously and reliably perform the activities for which they were projected.

* Danilo S. Jodas. Tel.: 00-55-17-8131-4987
E-mail address: danilojodas@gmail.com

doi:10.1016/j.procs.2013.05.187

Several papers on mobile robots report the use of artificial neural networks (ANNs) as the chosen intelligent system [1][2]. ANNs determine a result based on environmental input data captured by appropriate sensors. These results are used to control the mobile robot.

The combination of intelligent algorithms with image processing has provided good results for those applications in which the mapping of actions on the environment's plan is required to determine possible mobile robots' routes, avoiding collisions [3]. In such situations the images are captured by a video camera, being submitted to pre-processing algorithms for the extraction of important image features, which are then input to the intelligent algorithm.

Mobile robotics benefits from more powerful processors reaching the market as several associated tasks require real time responses without which the implementation of the corresponding control algorithms is not viable. Hardware implementation of such algorithms is also an important issue as it allows for faster execution. Field Programmable Gate Array (FPGA) devices are frequently used due to the possibility of internal reconfiguration of logical devices, meaning that it is possible to adapt the hardware to specific algorithmic requirements, and eventually leading to an increase in the robot's performance.

The goal of this paper is to present the comparison between support vector machine and artificial neural network in the recognition of the steering angle of plantation's paths. A navigation system based on image processing algorithms, support vector machine and artificial neural network was developed to realize the comparison. In the future this system will be embedded in hardware to drive a mobile robot through tracks in plantations. The navigation system is used to control the robot's direction, keeping it within the appropriate plantation track based on images of the terrain. Image processing algorithms are used to improve image quality as well as to extract relevant features from the image. The track thus identified is then processed and transferred to both support vector machine and artificial neural network to determine the angle from the current direction of movement that the robot is required to deviate, if any. The outcomes of the structures were compared to determine which one is most appropriate and advantageous to be used in the navigation system.

The correct navigation control is an essential task for the success of applications related to autonomous robotics. Those we are presently most concerned with are related to agriculture, such as weed detection and pesticide application in plantations.

The remaining of this paper is organized as follows: in section 2 the related works are discussed; in section 3 an overview of system is presented; in section 4 test results for the recognition in software are presented; and in section 5 some conclusions are drawn.

2. Related works

Several works in mobile robotics were already presented by various authors, mainly with respect to navigation and obstacle avoidance.

Young-Jae Ryoo [4] presented a visual navigation system for mobile robot based on image processing and artificial neural network. The system receive as input eight points from the lines that represent the road's edge and these points are passed to the artificial neural network, which calculates the output that is the steering angle of mobile robot. The inputs are eight points that cross transversely with the road's edge. The neural network was trained with the backpropagation algorithm.

Mandow et al [5] presented a mobile robot named AURORA, which navigate by corridors in greenhouses to pulverize plants. Information needed to keep the robot aligned in corridor or to determine the steering angle when reach the end of the corridor is captured by ultrasonic sensors. Images are used by experts to control the robot remotely. In addition, it is not used intelligent algorithms for navigation of AURORA mobile robot.

Astrand e Baerveldt [6] presented a mobile robot to detect weed in sugar beets. Detection of weed and path navigation is made based on images of terrain captured by video cameras. Steering angle is calculated by a modified line detection algorithm based on Hough transform, which detect lines identifying the plant row. Mobile robot system does not use artificial intelligence to detect weed or to calculate the steering angle.

A considerable amount of related works can be found in [1], [2], [3] and others. However we are not discussing them because the above mentioned ones summarize the state of the art.

3. The navigation system

In Fig. 1 a diagram of the system operation is presented.

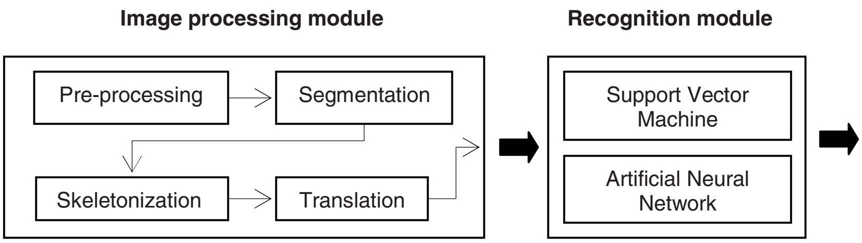


Fig. 1. Diagram of system operation

The image processing module aims at improving image quality as well as extracting features from the plantation images. In a pre-processing submodule filters for image smoothing are implemented to minimize the noise present in the image. A segmentation submodule is used to separate the track from planted area. A skeletonization submodule is used to extract the corresponding central path. A translation submodule is used to move the identified path to the center of image in order to provide a standard data representation for the recognition module. The recognition module is composed by a support vector machine and an artificial neural network, which receives the pixels of the translated skeleton and determines an output that represents the steering angle to the pattern of pixels presented. The use of support vector machines is due to the faster training and similar generalization capacity to the multilayer perceptron. The navigation system was developed in software for realization of initial tests. The development of image pre-processing algorithms was accomplished in C programming language together with the OpenCV library. The development of support vector machine and artificial neural network algorithms also was accomplished in C programming language.

3.1. Image processing module

For the initial tests an image bank composed by peanut and soybean plantation images was formed. The images were shot with a Kodak digital camera, M531 model, with a resolution of 6 Mega Pixels, under various illumination conditions. The bank has 1186 images, with 570 peanut images and 616 soybean images. Each image was resized to 300x225 pixels to reduce the pre-processing and segmentation algorithms' runtime. In the Fig. 2 the processing steps of a plantation image are shown. The image pre-processing is the step where techniques are applied to improve the image quality. The images of the bank were submitted to salt and pepper noise for testing the smoothing filters. The median filter was used to remove the noise present in the images. The median filter presented better results as compared to the low pass filter. The median filter was much efficient in removing noise and preserving edges in the images. In Fig. 2(b) the result of the median filter applied with 7x7 template is presented.

The extraction of tracks' features consists in obtaining a binary image in which the path is characterized by the black color and the area corresponding to the plant is represented in white. The images of the bank are represented in the RGB (Red, Green, Blue) color model and were converted to the HSI (Hue, Saturation, Intensity) color model so as to obtain the hue, saturation and intensity channels. The HSI color model was chosen due to the possibility of color identification regardless of illumination intensity or degree of saturation of the color image. The advantage in the use of HSI color model is the separation of the color information from intensity information. In Fig. 2(c) the hue component of an image is shown.

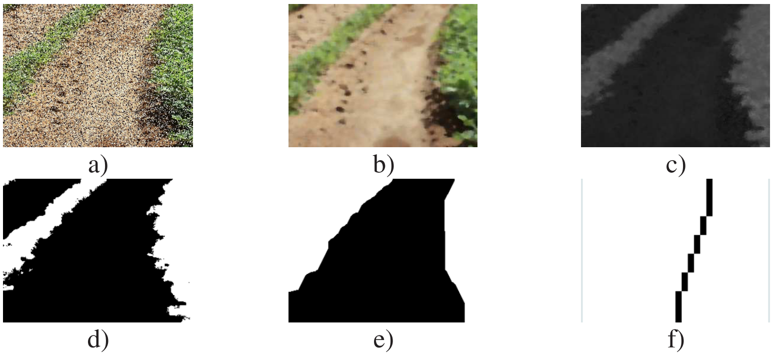


Fig. 2. Steps of the image processing (a) Noisy image (b) Noise removed with median filter (c) Hue component of the image (d) Segmentation of path. (e) Edge Smoothing and elimination of double path. (f) Skeletonized image

A drawback of this technique may be observed when an image has shadows. A shadow is a dark region in the image formed by blocking the light from an object. Shadows may impair the segmentation of an image due to either the similarity with black regions belonging to the areas of interest or their processing as an extension of an object present in the image. In Fig. 3(c) the incorrect path identification in an image with shadows is presented.

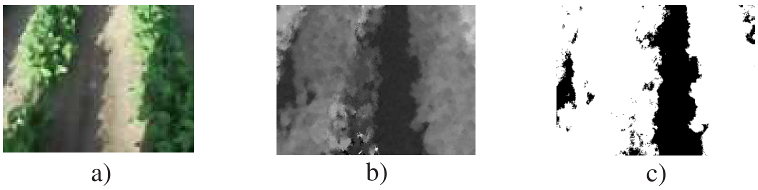
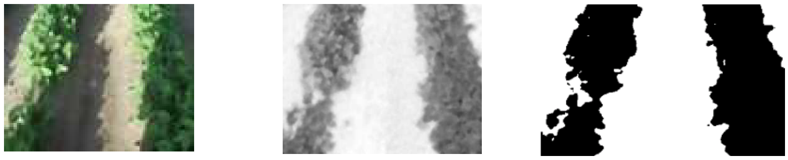


Fig. 3. Segmentation of image with shadows (a) Image with shadow. (b) Hue component (c) Path incorrectly identified

It is possible to see the partial identification of the path in Fig. 3(c), which is represented by the black color. This is because the shadow was processed as part of the plantation area due its similarity with the left path of the plantation. When represented in the hue component the shadow region shows the same values as those observed for the plantation area. Given this drawback the shadows must be disregarded during the path extraction procedure. A technique proposed by Finlayson, Drew, and Cheng [7] consists in obtaining images with pixels values in logarithmic space. Such images are said to be in the log-chromatic space, which is a gray scale image invariant to the illumination and free from shadows. Xu, Qi, and Jiang [8] rewrote the method proposed by Finlayson, Drew, and Cheng as presented in Eq. (1).

$$inv = \cos(\theta).\ln\left(\frac{r}{g}\right) + \sin(\theta).\ln\left(\frac{b}{g}\right) \tag{1}$$

In Eq. (1) r, g, and b are the image color values corresponding to the RGB model, and θ is the orthogonal projecting of points generated in the 2D log-chromaticity space. In Fig. 4 the image resulting from the shadow elimination process is presented. Note that the shadow's effect was minimized. The value used for θ parameter was 43.58, which was empirically determined from tests. The invariant image is used together with the hue component to generate an image with the extracted path. For the correct identification of the path the values of pixels in the hue component must be between 60º and 180º, which corresponds to the green and yellow colors in HSI model, and the corresponding pixels' values in the invariant image must be zero. The invariant image can be used to eliminate the shadows of original color image, but we decided not to follow this procedure because the invariant image was sufficient to correctly extract the plantation path.



a) b) c)

Fig. 4. Procedure to obtain invariant image (a) Image with shadow (b) Invariant image (c) Binarized invariant image

To smooth the border of the identified path, i. e., to remove small holes and links to adjacent paths, the morphological operation of closure was used. This procedure helps in the better representation of path's skeleton and minimizes the prominences effect in the image border. In Fig. 5 the result of application of the closure is presented.

a) b)

Fig. 5. Morphological operation of closure (a) Path identified with irregular edges (b) Edges smoothed with morphological operation of closure

The structuring element has the shape of an ellipse with 20x20 pixels. 20 iterations of the operation of closure upon the segmented image were used to obtain a smooth enough border. The choice of structuring element and of iteration's count was made empirically based on various tests. As ellipse shaped elements present smoother borders than other structuring elements' options they leave the image border smoother too.

Due the position of the camera with respect to the terrain it is possible that some images include paths adjacent to the main path. The elimination of those paths reduces the number of information to be processed by both support vector machine and artificial neural network. To remove such paths the region growth by pixels aggregation algorithm was used. The algorithm starts with selection of seed pixels, which are represented by black pixels belonging to the path. The number of seed pixels is dependent on the amount paths identified. For example, in the image of Fig. 5(b) two paths were identified, so it is possible to select two seed pixels, one for each path, and each path is processed as a region. Counting the pixels in each path is simultaneous with executing the region growth procedure. The regions are labeled with an integer value and in the end of the process it is possible to determine the region that has more pixels, which prevail in the image. The other paths are removed just altering the color for white. In Fig. 2(e) the result of the region growth algorithm execution is presented.

The next step is the thinning of the identified path. This is done by applying the thinning algorithm proposed by Zhang and Suen [9]. This algorithm reduces the number of information to be transmitted to the support vector machine and to the artificial neural network. In Fig. 2(f) the result of the application of the thinning algorithm is shown. The extracted path may be displaced due to the position of the camera, which can be prejudicial during classification because each displacement may be classified as a different angle. To make classification independent of the path displacement it was used the invariant translation algorithm proposed by Yuceer and Oflazer [10]. It is used to compute the center of the object in the image and make it coincide with center of the image itself. The center of object is obtained by averaging their coordinates in the x and y axes, which are computed by using Eq. (2) and Eq. (3). In equations (2) and (3), f(x,y) is the intensity of gray in the coordinates (x,y) and M and N are the amount of lines and columns in the image, respectively. A mapping of the pixels of the object to the center of the image is performed by computing the difference between the central column y of the object and the central column y of the image.

$$x_{av} = \frac{1}{\sum_{i=1}^{M}\sum_{j=1}^{N} f(x_i, y_j)} \sum_{i=1}^{M}\sum_{j=1}^{N} f(x_i, y_j).x_i \tag{2}$$

$$y_{av} = \frac{1}{\sum_{i=1}^{M}\sum_{j=1}^{N} f(x_i, y_j)} \sum_{i=1}^{M}\sum_{j=1}^{N} f(x_i, y_j).y_j \tag{3}$$

This difference is used to move the pixels of the object to the center image, as displayed in Eq. (4).

$$f_t(x, y) = (x, (N/2) - y_{av}) \tag{4}$$

In Eq. (4), N is the amount of columns of image. In Fig. 6 an image where the skeleton was displaced to the center is presented.

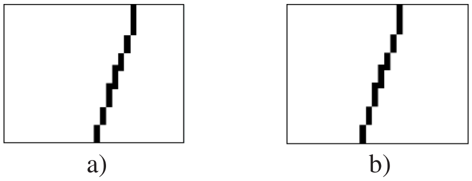


Fig. 6. Skeletonized image (a) outside center (b) skeleton centralized

The final step was resizing the image to the resolution of 30x23 pixels to reduce the size of the input layer of both support vector machine and artificial neural network.

3.2. Recognition module

The Hough transform is a technique through which one transposes the lines of an image described in cartesian space (x, y) to the polar coordinate space (ρ, θ), as shown in the Fig. 7, where ρ represents the distance from origin represented by the $\overline{AB}$ segment to C line and θ is the angle of the $\overline{AB}$ segment with respect to the x axis.

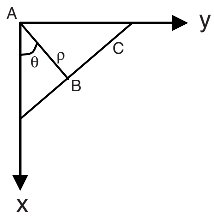


Fig. 7. Polar coordinate space [11]

As the training of both support vector machine and artificial neural network is supervised, the angles of each skeletonized image are needed for this procedure. So, the Hough transform and the Least Square method [12] are applied to each image to obtain the value θ to be used during the supervised training of the SVM and ANN.

Each image was transformed into a 690-element vector that corresponds to the image pixels. These values are inputs to both support vector machine and artificial neural network used for the recognition of steering angle. The images are classified as one of 19 possible patterns according to the angle they form to an imaginary straight line on the axis of movement. Such angles range from minus 45 degrees to plus 45 degrees with respect to the above mentioned straight line, and are discretized in 5 degrees intervals. As the values of the steering angles are discretized, the values produced by both Hough transform and Least Square method are also discretized by using a nearest neighbor criterion, meaning that if the value θ of the Hough transform is 42, the two possible neighbors are 40 and 45. Thus, the desired output for the image will be 40 as this is the nearest value to 42.

The SVM kernel functions used in this work are shown in Table 1.

Table 1. Types of Kernel Functions

Type	Kernel Function
Radial Basis Function	$\exp\left(\frac{-1}{2\sigma^2}\lVert x-y\rVert^2\right)$
Hardware Friendly Kernel	$2^{-\gamma\lVert x_i - x_j\rVert_1}$

The last kernel function presented in the Table 1 is named hardware-friendly kernel (HFK) and was proposed by Anguita et al. [13] to reduce the implementation complexity of kernel functions in hardware. The use of this kernel is more suitable for implementation in hardware than the Gaussian kernel because it is not necessary to calculate divisions and exponentials, which makes the so called hardware-friendly kernel faster for several pattern recognition procedures. The parameter γ is an integer value defined as two to the power p (i.e. $\gamma = 2^p$, with p = 0, 1, 2 …). The L1-norm of the distance between the inputs and the support vector (i.e. $\lVert x\rVert$) is represented as $\lVert x_i - x_j\rVert$. By using the L1-norm instead of the Euclidean distance we avoid the computation of square root and exponentiation, which makes the hardware implementation less complex.

4. Experimental results

The tests were performed with 1128 images of peanut and soybean plantations. 71 images were used for training and 1057 images were used for verification. We have removed 58 images of the tests because they are inappropriate to test.

For the tests one SVM and one ANN were used for recognition of the steering angle. The topology of the SVM is composed by 690 inputs, 71 kernels and 1 output. The output is a real value between -45 and 45 that represents the steering angle for the presented image. As kernel functions, the radial basis function and the hardware-friendly kernel were used. As discussed in the previous section an attempt was made to use a simpler kernel that is easier to implement in hardware, which also presents results close to those of the RBF kernel. The so named hardware-friendly kernel was tested, being these results compared to those of the RBF kernel. The value chosen for the parameter γ was 0.0625, which was determined empirically through tests showing the best results.

The topology of the ANN is composed by 690 input neurons, 71 hidden neurons and 19 output neurons. Each output neuron represents a steering angle and their output is 0 or 1. An output neuron has outcome 1 if the input image belongs to the steering angle represented by the neuron. Otherwise, the neuron's output is 0. The learning algorithm used in the training of the ANN was the backpropagation one.

The following information was considered in the tests:

- Amount of exact hits
- Amount of approximate hits
- Errors
- Hit percentage

The amount of exact hits indicates the number of images for which the required steering angle and the one computed by both the SVM and the ANN exactly match. The amount of approximate hits indicates the number of images for which the difference between the required steering angle and the one computed by both the SVM and the ANN is 5°. The errors represent the number of images for which the difference between the required steering angle and the one computed by both the SVM and the ANN is above 5º. The hit percentage was computed as the ratio of the number of exact hits to the number of approximate hits. The amount of approximate hits was considered due to the similarity among the skeletons whose angles differ by 5º or less, causing both the SVM and the ANN to classify the corresponding image to an approximate angle figure. Some tolerance may be acceptable if the processing time is low enough to prevent the next exit to be made after a long distance can be travelled by the mobile robot, thus minimizing the off route displacement.

4.1. Test with SVM

In this test 71 training images and 1057 verification images were used, which belong to the plantation images' bank. Thus, the SVM's topology was represented by 690 input elements, 71 hidden elements and 1 output element. The results of this test are shown in the table 2.

Table 2. Results of the test with SVM

Information	RBF	HFK
Exact hits	776	810
Approximate hits	197	175
Errors	84	72
Hit percentage	92,05%	93,19%

We can notice that the results of the hardware-friendly kernel were better than those of the radial basis function kernel.

4.2. Test with ANN

We have performed four tests with ANN. In each test different numbers of neurons in the hidden layer as well as different training parameters were used. In the table 3 the training parameters used in the tests are shown and in the table 4 the results of the verification step are presented. It's possible to notice in table 4 that the best result was provided by the third test. However, such result is inferior to the test performed with SVM. Besides, the amount of neurons in the hidden layer of the ANN is greater than those used in the SVM topology. It's a disadvantageous characteristic because in this case twice as much hardware space would be needed to implement the ANN's topology.

Table 3. ANN's parameters in each test

Test	Learning rate	Momentum rate	Mean error	Number of hidden neurons
First test	0.7	0.1	0.0001	71
Second test	0.7	0.1	0.0001	115
Third test	0.7	0.1	0.00006	142
Fourth test	0.7	0.1	0.0001	213

Table 4. Results of the recognition using ANN

Information	First test	Second test	Third test	Fourth test
Exact hits	725	644	748	681
Approximate hits	215	252	206	233
Errors	117	161	103	143
Hit percentage	88,9%	84,8%	90,3%	86,5%

4.3. Performance analysis

The tests were performed on a computer with 4 Gigabytes of RAM memory and an Intel Dual Core processor of 2200 MHz. The SVM’s execution performance for each kernel is presented in Fig. 8. As can be seen in the graphics, the runtime of the HFK was approximately 50% faster as compared to the RBF kernel in both training and execution. Besides, the HFK presented better results than the RBF kernel in this test.

In Fig. 9 the graphics of ANN’s runtime is presented. In the graphic of Fig. 9 it is possible to see that the ANN’s training time was rather long, taking approximately 1 hour. In this case to train the network several times until an ideal solution is reached could take hours or days, particularly if it is needed to include more training patterns. A circular graphic was used in Fig. 9 to improve the visualization of the runtime, mainly the recognition time.

Independently of the kernel function used, the SVM proved more advantageous than artificial neural networks concerning to the training time because there are not backpropagation cycles like those occurring in ANN. It could enable the implementation of online training to aggregate the classification of new patterns in different plantation fields.

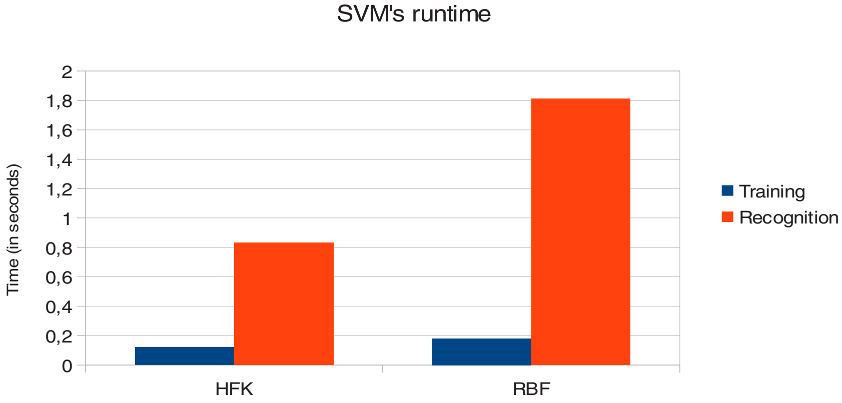


Fig. 8. SVM's runtime for both training and recognition

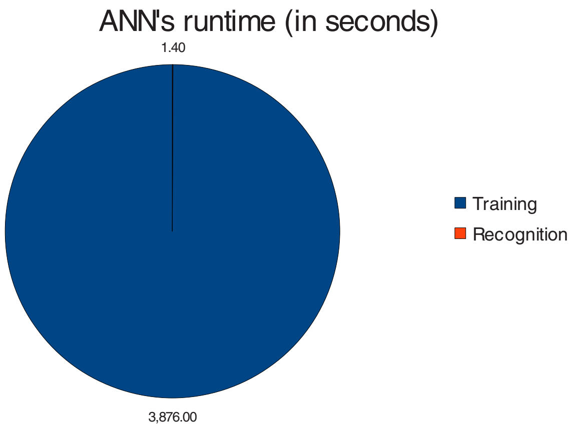


Fig. 9. ANN's runtime for both training and recognition

5. Conclusion

This work presented a navigation system for mobile robots handling paths in plantations, based on computer vision, support vector machine and artificial neural network. The main objective of the navigation system is to obtain a satisfactory success rate in the prediction of steering angle using a small amount of image information, i. e., only the skeleton of the plantation pathways. Besides was presented the comparison in the recognition of both SVM and ANN. The results obtained from SVM were better and the SVM's execution time was less than the ANN in training step. The ANN's training can to take hour or days as a SVM can perform it in just some minutes. It enables the execution faster of several training using different parameters and patterns to adjust the connection's weight and to achieve an appropriate recognition.

The major difficulty at this stage of the work was to achieve an exact percentage of correct recognition, a fact that is due to similarity of the skeletons obtained from those images having similar values of angles. However, for most images it has been verified that the difference between the angles computed by both SVM and ANN to that one computed using both Hough transform and Least Square method is about 5º. Analyzing the possible deviations from the expected route when such an error occurs we decided to accept this approximation as a correct guess. Nonetheless, to adopt such an approach the SVM execution time must be improved. Moreover, possible improvements on the algorithm's execution time can lead to the update of the driving angle at smaller distance intervals.

The tests presented allow us to conclude that the use of support vector machines may properly determine the mobile robot direction of movement with the same capacity of generalization of artificial neural network.

Acknowledgements

The first author acknowledges CAPES – Brazilian Federal Agency for the Improvement of Higher Education – for the scholarship provided during this research.

References

[1] M. Meng, A.C. Kak. Mobile Robot Navigation Using Neural Networks and Nonmetrical Environmental Models. IEEE Control Systems, vol. 13, nº 5, p. 30-39, October 1993

[2] D. Janglová. Neural Networks in Mobile Robot Motion. International Journal of Advanced Robotic Systems, vol. 1, nº 1, p. 15-23, 2004

[3] A. Howard, H. Seraji. An Intelligent Terrain-Based Navigation System for Planetary Rovers. IEEE Robotics & Automation Magazine, vol. 8, nº 4, p. 9-17, December 2001

[4] Young-Jae Ryoo. Neural Network Control for Visual Guidance System of Mobile Robot. Lecture Notes in Computer Science, vol. 4432, p. 685-693, 2007

[5] A. Mandow et al. The Autonomous Mobile Robot AURORA for Greenhouse Operation. IEEE Robotics & Automation Magazine, vol. 3, nº 4, p. 18-28, December 1996

[6] B. Åstrand, A.-J. Baerveldt. An Agricultural Mobile Robot with Vision-Based Perception for Mechanical Weed Control. Journal of Autonomous Robots, vol. 13, nº 1, p. 21-35, July 2002

[7] G. D. Finlayson, M. S. Drew, L Cheng. Intrinsic Images by Entropy Minimization. Proceedings of the 8th European Conference on Computer Vision, p. 582-595, 2004

[8] L. Xu, F. Qi, R Jiang. Shadow Removal from a Single Image. Proceedings of the 6th International Conference on Intelligent Systems Design and Applications, vol. 2, p. 1049-1054, October 2006

[9] T. Y. Zhang, C. Y. Suen. A Fast Parallel Algorithm for Thinning Digital Patterns. Communications of the ACM, vol. 27, nº 3, p. 236-239, March 1984

[10] C. Yüceer, K. Oflazer. A Rotation, Scaling, and Translation Invariant Pattern Classification System. Journal of Pattern Recognition, vol. 26, nº 5, p. 687-710, May 1993

[11] R. C. Gonzalez, R. E Woods, S. L. Eddins. *Digital Image Processing using MATLAB*. 2nd ed. Gatesmark Publishing , 2009

[12] A. Bjõrck. Numerical Methods for Least Squares Problems. Society for Industrial and Applied Mathematics, April 1996.

[13] D. Anguita, S. Pischiutta, S. Ridella, D. Sterpi. Feed-Forward Support Vector Machine Without Multiplier. IEEE Transactions on Neural Networks, vol.17, nº 5, p. 1328-1331, September 2006.

[14] S. Haykin. *Neural Networks: a Comprehensive Foundation*. Prentice Hall, 1999