ABCM Symposium Series in Mechatronics - Vol. 2 - pp.661-668

A Computer Vision System based on Multi-Layer Perceptrons for Controlling Mobile Robots

Marcos Gonçalves Quiles
Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo
quiles@icmc.usp.br

Roseli Aparecida Francelin Romero
Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo
rafrance@icmc.usp.br

Abstract. *Computer Vision System (CVS) is an important tool for robot navigation. In this paper, a CVS is proposed for controlling mobile robots which is based on Multi-Layer Neural Networks. This system is constituted by two main modules: vision module and navigation control module. The first module processes images that can contain several objects with different colors, returning the position of one of these objects, which corresponds to a given color. The control module utilizes the results obtained from the vision module providing the robot the action to be performed. The aim of the proposed system is to enable a robot to navigate in an environment following a color object and avoiding obstacles. Some experiments in a real environment with the robot Pioneer I are presented to show the good performance of proposed system.*

Keywords: *Mobile Robot Control, Computer Vision, Neural Networks, Potetial Fields*

1. Introduction

The development of robots, which can help or replace humans in risk tasks or even where the precision of the movements to be done is essential, has been taking the attention of several parts of the society. In the last few years, with the modernization and the increasing of computer resources, the researchers have been concerned in how to construct intelligent robots which are able to interact with the humans in a friendly manner and be able to act in unstructured environments (Pagello et al., 2002). This idea differs of the former mechanical robots, which are usually installed in factory environments.

For many real applications, as the task of delivering pieces in a industry, fixed robots are not able to perform this task. Thus, the development of mobile robots is primordial as for solving simple tasks as for more complex ones, such as: ocean and universe exploration, in a nuclear disaster where people can not go because of the contamination danger. In this environment, mobile robots will be able to replace humans, or at least, perform a prior condition analysis before the humans take any action.

For the mobile robot control, many single mechanisms or even the combination of them have been used, such as sonar, vision devices, laser, infrared, etc. Several works have been developed using multi-layer perceptrons (MLP) for image analysis applied to robot control in many tasks. Pomerleau (1995) proposed a system to drive a car in an autonomous way. In (Blank and Ross, 1997), it was developed a system to enable a robot to follow one ball on a surface. In (Jonsson, Wiberg and Wickström, 1997), a robot learned to drive itself in a corridor avoiding obstacles. In (Tyler and Czarnecki, 1999), an 1D vision system was applied for robot soccer. Another work was proposed by Waldherr, Thrun and Romero (2000), in which an interface for a mobile robot control based in gesture was proposed.

However, the most of the works cited below do not deal with color in the vision system. Color can be fundamental in several domains, since this information can represent features which cannot be treated when just the intensity (gray levels) is used. For example, for robot soccer domain, the color analysis can be very important (Simões e Costa, 2000). According to Cheng et al. (2001), image segmentation is the first step in image analysis and pattern recognition. It is considered a critical and essential with immediate impact at the final processing result. In the development of a vision system that deals with colors, the image segmentation can be essential to separated the interested objects from the background. This processing can facilitate the realization of some tasks such as face recognition. Although, one of the major difficulties of the development of a color image segmentation system is the variations of intensities observed in real environment, this problem is related to the way that the colors are represented (color space) and the method which makes the segmentation (Cheng et al., 2001).

Many works related to color image segmentation has been developed using several techniques and color spaces, such as: Neural Network, Fuzzy Systems, Multithresholding, etc. However, none of them is able to produce satisfactory results for all domains (Cheng et al., 2001). Moreover, another problem find in great part of the segmentation methods is that most of them cannot act in real-time, which is essential for controlling of mobile robots.

This paper presents a Computer Vision System (CVS) based on both techniques: artificial neural networks and potential fields, for controlling a mobile robot with an onboard camera. For the development of this system two modules was

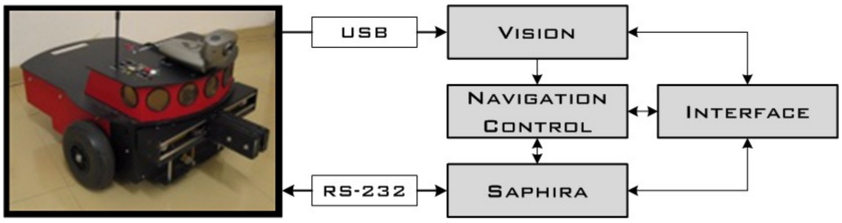

Figure 1. The Proposed System Architecture

implemented: Vision Module and Navigation Control Module.

The vision module is responsible for the preprocessing of acquired images, color image segmentation and the image recognition. The segmentation process is implemented by using a color classification method based on MLP model. For the image recognition has been also used MLP model, which is responsible to perform to determine the position of object in the robot visual field.

Based on the information from the vision module, the navigation control module is responsible to drive the robot into the environment avoiding obstacles and tracking an object of a specific color. For the development of this controller, the potential field technique (Arkin, 1998) has been used.

After the development of the system, several experiments have been performed with a Pioneer I Robot (ActivMedia Robotics) in a real environment for validating the proposed CVS.

2. The Computer Vision System

The computer vision system proposed in this paper is constituted by two main modules. The first is responsible for the image preprocessing, segmentation and recognition. The second implements the navigation control module responsible to drive the robot into the environment. In Figure 1, the system architecture is shown, where the arrows denote the dataflow. A RS232 connection is used to transmit and receive information between the robot and the laptop. The USB interface was used to set up a connection between the laptop and the camera.

The proposed system architecture can be classified as belonging to reactive paradigm (Arkin, 1998), which means that no prior information of the environment is known and the acquired information is not stored by the robot. The reactive paradigm has been shown to be a very attractive way to implement controllers for robots acting in real and dynamic environment. This can be observed by the fact that the robot behaviors (in this paradigm) are explicit relations of how the robot must react about the information obtained from the sensors. Moreover, the behavior defined in this paradigm can be used as a base for the implementation of controllers based on the hybrid paradigm (Arkin, 1998). In this case, the robot could perform more complex tasks such as, explore the environment creating a map representation and avoiding obstacles.

2.1 The Robotic Hardware

All of the experiments of the proposed vision system have been performed and tested on a Pioneer I Robot (Figure 1) from ActivMedia Robotics[1]. Seven Polaroid sonars 6500 compose the original perception of the robot. The external communication can be performed by radio or through the RS232 interface.

For the development of the proposed vision system, a Creative WebCam Go Plus camera was incorporated to the system for realizing the visual perception of the environment. This camera has been set on the top of the robot and turned to the floor. This configuration enables the robot to perceive the visual information from the base of the robot up to $4.8m$ ahead.

2.2 Saphira Environment

The communication between the developed system and the Pioneer I robot is performed by the Saphira (Konolige, 1997). The Saphira is an environment constituted by many libraries used to develop robotics applications. It is maintained by the Artificial Intelligence Center of Stanford Research Institute. It has a library with routines that permit the user to build up programs in C/C++ language for the control of mobile robots as the Pioneer I, providing an abstract level from the robot hardware.

2.3 The Interface

The interface is a simple module responsible for the interaction between the user and the system. It has all the parameters necessary to control the experiments such as: maximum velocity, rotation and translation angles, the behavior

[1] http://www.activmedia.com/

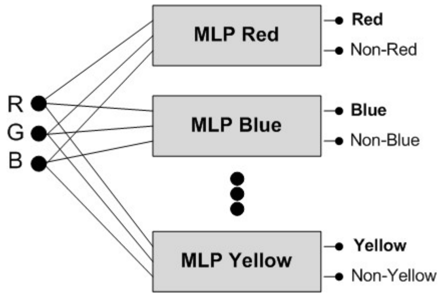

Figure 2. The proposed Color classification System

weights and the color selection for the image segmentation (the color that must be followed). The interface also has the connection procedures which are responsible to open and close the connection with the Pionner I robot or the simulator.

With the interface the user can program the configuration of an experiment and visualize all the processing information by graphics and data generated during the experiment.

2.4 Vision Module

The vision module is responsible by acquiring and processing the information and by sending it to control module. We can say that the vision module acts as a translator who receives as input a picture from the environment and to follow generates as output a high level interpretation of this picture. For example: a red object is in position XY. Based on the vision module output, the control module can generate the necessary commands for the navigation of the robot.

The vision module is divided in three tasks: the preprocessing, image segmentation and recognition task.

The first task is responsible by the image acquisition from the camera with a resolution of 320X240 pixels and by its reduction for 80X60 pixels. The image reduction is accomplished by the average of the points. A window constituted by 4X4 pixels goes through the captured image generating a point with the average value of the 16 pixels contained in window. This level resolution (80X60 pixels) has been defined empirically based on some experimental tests with real images of environment. It has been adopted the minimum resolution necessary to perform the recognition of the objects carried out by this system.

The image segmentation process is performed by a color classification system. The classifier system is constituted by a set of MLP neural networks (Figure 2), in which each net has the role to classify a defined color, separating it from other colors of the image. For example, when the red color is requested, each pixel of the image is classified as red or non-red. The classification is done through the RGB components of each pixel in the image. In this case, the requested neural network receives as input the three components (RGB) and provides as output, the classification result.For example, the neural network output can be red or non-red color. As the segmentation process implemented here is to isolate an specified color, the resultant segmented image is constituted by only two colors: the specified color (represented by the white color) and the other colors (represented by the black color).

The third task of the vision module must recognize of the segmented images. For this purpose, another MLP neural network receives as input signals all the pixels (80X60) that compose the segmented image and provides as output one of the both information: either the object is in the image or it is not. In the affirmative case, the neural network also informs the object position.

A MLP neural network, used in this case, is constituted by 3 neurons in the output layer, which the first and the second outputs represent the position X and Y, respectively, of the object in the image and the third output (P) informs if the object is or not present in the image (Figure 3). The outputs X and Y were empirically discretized based on experiments performed in a real environment, where the images was labeled second the values below:

- X : 0.0, 0.1, 0.2, ... , 1.0
- Y : 0.0, 0.1, 0.2, ... , 1.0

where X and Y represent respectively the measure values according to the object position in the visual field (not based in the real environment). However, in the tests performed, it has been noted that the values: 0.0, 0.4, 0.7, 0.8, 0.85, 0.88, 0.9 and 0.91 for the Y output could be matched with intervals of $0.5m$ in the real environment. It means, when Y outputs 0.0, the object is close to the robot; when Y outputs 0.4, the object is about $0.5m$ from the robot, and go on with steps of $0.5m$. This estimation function of the robot position in the real environment has been developed considering that is not possible to get the real object 3D position without using a stereo vision.

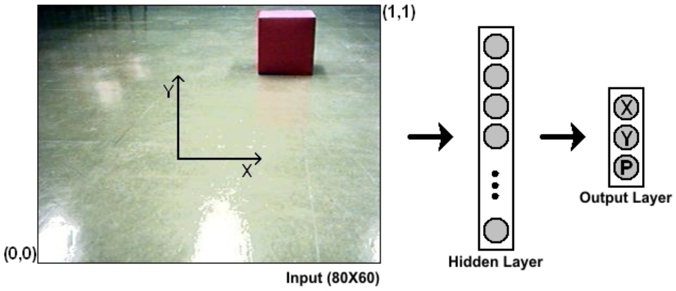

Figure 3. The recognition MLP neural network

$$SpaceFunction(X,Y) = \begin{cases} ([(0.5 - X) * 200], [0000]), & \text{if } Y \approx 0.0 \\ ([(0.5 - X) * 400], [0500]), & \text{if } Y \approx 0.4 \\ ([(0.5 - X) * 750], [1000]), & \text{if } Y \approx 0.7 \\ ([(0.5 - X) * 1024], [1500]), & \text{if } Y \approx 0.8 \\ ([(0.5 - X) * 1500], [2000]), & \text{if } Y \approx 0.85 \\ ([(0.5 - X) * 1875], [2500]), & \text{if } Y \approx 0.88 \\ ([(0.5 - X) * 2250], [3000]), & \text{if } Y \approx 0.9 \\ ([(0.5 - X) * 2625], [3500]), & \text{if } Y \approx 0.91 \end{cases} \tag{3}$$

Figure 4. Function for computing the 3D coordinates: X, Y

2.5 Navigation Control Module

The control module was developed allowing the robot to navigate into the environment avoiding obstacles and searching for a specified color object. With this controller the robot is able to keep navigating into the environment in a random way searching for the specified color object. When the robot finds the object, the controller drives the robot in the direction of this object. This controller was developed using the field potential technique (Arkin, 1998), which has the aim of generating the real path that the robot should follow based on the information generated by three implemented behaviors: GoAhead(), Follow() and AvoidCollision().

The first behavior, GoAhead(), has the objective to keep the robot in constant movement. The second behavior, Follow(), is responsible for drive the robot in the direction of the target using the information deriving from the vision module. The last behavior, AvoidCollision(), takes care of keeping the robot in the opposite direction of the obstacles. All the behavior are represented by a 2D vector, which represents the direction (θ_F) and the magnetude ($|\overrightarrow{F}|$).

In the behavior GoAhead() is not considered the perception of the world. The aim of this behavior is to keep the robot in movement. So, a vector with constant direction and magnitude, (C_g), is used to represent this behavior (uniform field).

$$|\overrightarrow{F_g}| = C_g \tag{1}$$

$$\theta_{\overrightarrow{F_g}} = 0 \tag{2}$$

On the other hand, the Follow() behavior considers perception through vision module for driving the robot to the target.

As described previously, the neural network output provides the coordinates X and Y corresponding the position of the object. Then, the output Y has been discretized with values which increment is equal to $0.5m$ in the real environment (3D). In this way, it is possible to calculate an approximated values for the 3D coordinates, denoted here by X_{3D} and Y_{3D}, respectively, by using the function showed in Figure 4. This function receives as input the signal X and Y received of the recognition MLP network and returns, as output, the pair $([X_{3D}], [Y_{3D}])$ (expressed in mm), which are estimates of the object position in the real environment . After that, the Follow() behavior can be performed.

As the behavior GoAhead(), the Follow() behavior works with a atraction force, represented by a vector with constant module, (C_a) and a rotation angle that indicates the robot's direction to reach the target. This direction is calculated from the data obtained from the vision module.

$$|\overrightarrow{F_a}| = C_a \tag{4}$$

$$\theta_{\overrightarrow{F_a}} = \arctan(X_{3D}, Y_{3D}) \tag{5}$$

The AvoidCollision behavior is responsible for keeping the robot so far of the obstacles. The sonar signals are considered in this case. Considering that the robot is equipped with seven sonars, each one generating a distinct vector (repulsion forces), the resultant force provided by AvoidCollision() behavior is given by the sum of the seven vectors.

The repulsion force of the AvoidCollision() behavior is generated by a radial field whose decay function is describe as follows:

$$|\overrightarrow{F_{ri}}| = \exp((-D + L)/T) \tag{6}$$

where D is the distance between the robot and the obstacle, L is the proximity limit and T is a constant that defines the degree of decay of the exponential function. The direction of the repulsion force (vector), denoted by θ angle, is define by:

$$\theta_{\overrightarrow{F_{ri}}} = \arctan(-XS_i, -YS_i) \tag{7}$$

where XS_i and YS_i are the coordinates X and Y, in mm, obtained from each sonar i, $i = 1, , 7$. The $-$ signal is used to invert the direction, since this behavior must keep the robot far from the obstacles.

The resultant repulsion force is give by a vector that is the sum of the individual vectors generated by each sonar, as shown in the Equation 8.

$$\overrightarrow{F_R} = \sum_i \overrightarrow{F_{ri}} \tag{8}$$

After the calculation of the forces for each defined behavior, the resultant force (robot trajectory) is define by:

$$\overrightarrow{F} = P_1\overrightarrow{F_G} + P_2\overrightarrow{F_A} + P_3\overrightarrow{F_R} \tag{9}$$

where $P_i, i = 1, 2, 3$ are the weights of each force of the system. In the experiments, the values of P_i was defined as:

- $P_1 = 0.5$
- $P_2 = 1.0$
- $P_3 = 0.7$

After the resulting force has been calculated, from the integration of the behaviors, the robot have to navigate based on the following values:

$$Vel_T = |\overrightarrow{F}|Vel_{Max} \tag{10}$$

and,

$$\theta_{\overrightarrow{F}} = \arctan(F_X, F_Y) \tag{11}$$

where Vel_T is the translation velocity of the robot, Vel_{Max} is the maximum velocity of the robot, and the direction of the robot is defined by $\arctan(F_X, F_Y)$.

3. Results

The results are presented in to parts: in the first one describes the results of the system tuning (parameters definition and neural network learning process) and the second part describes the experiments carried out with the Pioneer I Robot in a real environment.

3.1 System Tuning

As mentioned above, this first part describes the experiments performed and the obtained results during the system tuning. More specifically, we present results of the learning process of the neural networks responsible for the image segmentation task and for the recognition task.

For the learning process two algorithms was used: the backpropagation (Haykin, 1998) and the Rprop (Riedmiller and Braun, 1992). The backpropagation algorithm was implemented in two ways: on-line and the batch mode.

3.1.1 Image Segmentation Neural Network

For the image segmentation neural network learning was created database with 100 pattern. For example, to perform the learning process of the neural network responsible for recognize the red color, a database with 50 patterns of red pixels and 50 patterns with other colors was created. Three colors was considered in this work: red, blue and yellow.

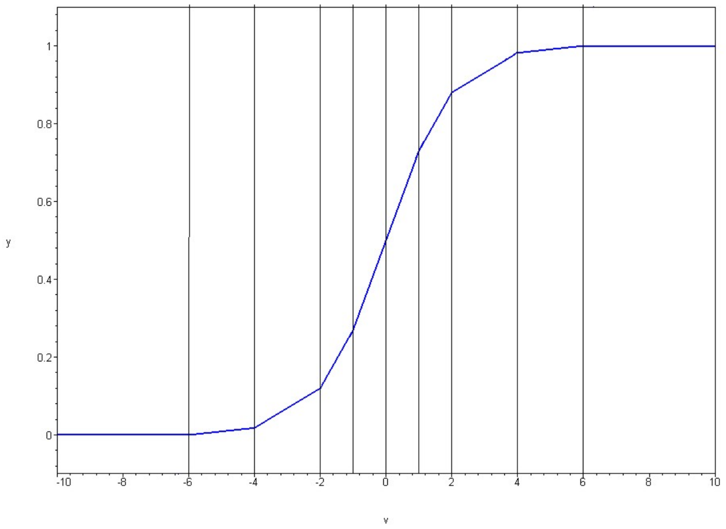

Figure 5. Discretized Sigmoid Function

Table 1. The learning process results for the image segmentation neural network: red color

	Iterations	MSE	SD
Rprop	298	0,039053	0.03100381
Batch BP	5000	0.069356	0.02955651
On-Line BP	477	0.044432	0.05420535

To find a suitable neural network topology some tests was carried out dividing the database in two parts: 80% for the learning process and 20% to test the set topology. In these tests were observed that the topology 3X3X2, it means, a neural network with 3 input neurons (RGB), 3 neurons in the hidden layer and 2 neurons in the output layer was appropriated to the blue and the red color and to the yellow color the topology 3X5X2 showed better results.

After finding a suitable topology, tests using the 10-Fold Cross-Validation technique was carried out. The Table 1 shows the obtained results from the learning process using the algorithms: Rprop, Batch Backpropagation and on-line Backpropagation for the red color (for the blue and the yellow color, the results was almost the same). In this table is presented the number of iteration necessary to the convergence of the neural network, the average of mean square error (MSE) and the standard deviation (SD) for the 10-fold execution of the learning process. It can be observed in the results that the precision of the algorithms was almost the same, differing just in the number of iteration used by each learning algorithm. The computational time spent in the learning process was almost the same for all the algorithms.

Some tests were performed in Pentium IV 2.26 GHz to check the real-time performance of the image segmentation process. For the red and the blue colors, the neural network spent $0.0125s$ to segment a frame with 80X60 pixels and for the yellow color it spent $0.0187s$ for each frame. To improve the real-time response of these segmentation task we also test a discretized sigmoid function ($Sl(x)$) (Figure 5 - Equation 12) that approach the sigmoid function used as a activation function of the neurons.

$$Sl(x) = \begin{cases} 0, & \text{se } x < -6 \\ 0.054 + 0.009 * x, & \text{se } x \geq -6 \text{ e } x < -4 \\ 0.222 + 0.051 * x, & \text{se } x \geq -4 \text{ e } x < -2 \\ 0.420 + 0.150 * x, & \text{se } x \geq -2 \text{ e } x < -1 \\ 0.500 + 0.230 * x, & \text{se } x \geq -1 \text{ e } x < 1 \\ 0.580 + 0.150 * x, & \text{se } x \geq 1 \text{ e } x < 2 \\ 0.778 + 0.051 * x, & \text{se } x \geq 2 \text{ e } x < 4 \\ 0.946 + 0.009 * x, & \text{se } x \geq 4 \text{ e } x < 6 \\ 1, & \text{se } x \geq 6 \end{cases} \qquad (12)$$

With this new sigmoid function the time spent to carry out the frame segmentation for the blue and red colors was $0.0062s$ and for the yellow color $0.0125s$. These results demonstrate the advantage of use this discretized function in contrast with the original sigmoid function and also show the performance of the proposed system to act in a real-time environment.

No lost of quality in the image segmentation process was observed when using this function. The Figure 6 presents

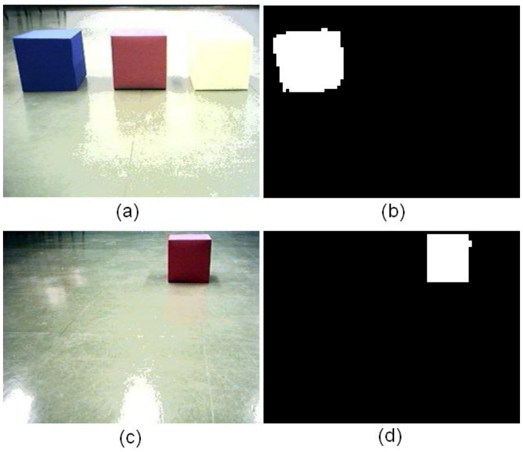

Figure 6. Result of the image segmentation using the RGB color space. (a) Original image 01 (b) Preprocessed and segmented image 01 for the blue color (c) Original image 02 (d) Preprocessed and segmented image 02 for the red color

	Iteration	MSE	SD
Rprop	6	0.004809	0.000883
Batch BP	503	0.012650	0.003577
On-Line BP	2	0.005850	0.001403

Table 2. The learning process results for the recognition neural network

the segmentation result for the blue and red color. These segmentation result were the same for both function: original sigmoid function and discretized sigmoid function $S1(x)$.

3.1.2 Recognition Neural Network

For the learning process of the MLP responsible for the recognition task, a database with 700 patterns was created. Each pattern is compose of a frame preprocessed and segmented (binary image with 80X60 pixels) and its respective label which means the position of the object into the visual field.

To establish the topology of this neural network, we also divide the database in to parts: 80% for the learning process and 20% to test the set neural network. By using the three implemented algorithms: Rprop, Batch Backpropagation and On-Line Backpropagation; the topology 4800X10X3 showed to be a suitable topology to perform this recognition task. This topology means: 4800 neurons in the input layer (80X60 binary pixels), 10 neurons in the hidden layer and 3 neurons in the output layer (X, Y and P)

Table 2 present the results of the application of the cross-validation technique (10-folds) with the set neural network (4800X10X3) using the three implemented algorithms. In the Table we can observed that the result of both algorithms was almost the same with advantages to the Rprop algorithm that was faster than the others.

3.2 Robot Experiments

After the system tuning, several experiments was carried out with the Pionner I robot to test the quality of the CVS in a real environment. To run the CVS a laptop Pentium III $500MHz$ and a Creative WebCam Go Plus were used.

The first experiments were performed to find good parameters for the behavior force weights (as mentioned in Section 2.5). Such experiments were carried out in a simple environment just to adjust the system to be able to avoid obstacles without collision and also to follow a target.

After the first experiments, which was performed to set some system parameter, we built in a real environment several sceneries with random obstacles and a target also randomly positioned into these sceneries (as show in Figure 7). For all performed experiments using the found parameters the robot was able to navigate avoiding the obstacles and following a target (represented by a defined color) when it was present into the visual field of the robot. We can observe in Figure 7, when the robot gets close to obstacles, it was able to avoid them without collision. Besides, when the robot find the target, it goes to the target position, which demonstrate the capacity of the system of distinguish color in a real and unstructured environment.

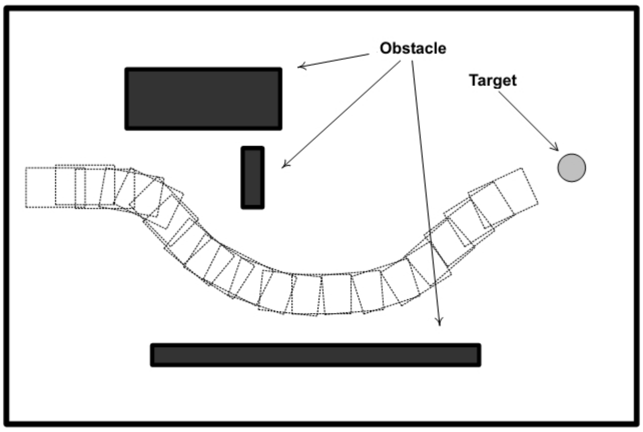

Figure 7. The robot performing a task in a real environment

4. Conclusions

In this paper was proposed a computer vision system applied to a mobile robot control, developed using MLP neural networks and the potential field technique. Two modules were implemented: the vision module and the navigation control module. In the vision module, the processing is made in three parts: preprocessing, image segmentation and recognition. The navigation control module was implemented using the potential field technique. This module had the function of drive the robot into a real environment avoiding obstacles and seeking for a target.

With the experiments, we could confirm the quality and the performance of the image segmentation methodology applied in a real-time task for the three colors considered by this work. Moreover, to improve the time response of the segmentation system, we also implement a discretized function to the sigmoid function of neurons. With this discretized sigmoid function, we improve the response of the system about 30 % without reduce the quality of the segmentation.

In the tests performed in a real environment, we could observe that even with limited devices such as the laptop and the webcam, the system was able to carry out the established tasks in real-time.

As a future work we intend to implement this system in a FPGA and also make a fusion of this CVS with some other systems developed by our research group.

5. References

Arkin, R., 1998 "Behavior-Based Robotics", MIT Press.

Blank, D. S. and Ross, O., 1997 "Incorporating a connectionist vision module into a fuzzy, behavior-based robot controller", In Proceedings of the 1997 Midwest Artificial Intelligence and Cognitive Science Society Conference.

Cheng, H. D., Jiang, X. H., Sun, Y. and Wang, J., 2001 "Color image segmentation: advances and prospects", Pattern Recognition, Num. 34, pp. 2259-2281.

Haykin, S., 1998 "Neural Networks: A Comprehensive Foundation (2ed)", Prentice Hall.

Jonsson, M., Wiberg, P. and Wickström, N., 1997 "Vision-Based Low-Level Navigation Using a Feed-Forward Neural Network", In International Workshop on Mechatronical Computer Systems for Perception and Action.

Konolige, K., 1997 "Saphira Manual Version 6.1", ActivMedia Inc.

Pagello, E., Arai, T., Dillmann, R. and Stentz, A., 2002 "Towards the Intelligent Autonomous Systems of the third millenium", Robotics and Autonomous Systems, Num. 40, pp. 63-68.

Pomerleau, D., 1995 "Neural Network Vision for Robot Driving", The Handbook of Brain Theory and Neural Networks.

Riedmiller, M. and Braun, H., 1992 "RPROP- A fast adaptive learning algorithm", In Proc. of ISCIS VII.

Simões, A. S. and Costa, A. H. R., 2000 "Using neural color classification in robotic soccer domain", In International Joint Conference IBERAMIA'2000.

Tyler, L. G. and Czarnecki, C. A., 1999 "A Neural Vision Based Controller for a Robot Footballer", Image Processing and its Application, Num. 465, pp. 77-81.

Waldherr, S. Thrun, S. and Romero, R., 2000 "A Gesture-Based Interface for Human-Robot Interaction", Autonomous Robots, Vol. 9, Num. 2, pp. 151-173.

6. Responsibility notice

The authors are the only responsible for the printed material included in this paper