OBJECT DETECTION AND LOCALIZATION SYSTEM BASED ON NEURAL NETWORKS FOR ROBO-PONG

Reza Sabzevari

Islamic Azad University of Qazvin
Department of Mechatronics, Qazvin, Iran
Member of Young Researchers' Club
sabzevari@gmail.com

A. Shahri

Iranian University of Science and Technology
Electrical Engineering Department
Tehran, Iran
shahri@iust.ac.ir

A.R. Fasih

Islamic Azad University of Qazvin
Department of Mechatronics, Qazvin, Iran
ar fasih@yahoo.com

Saeid Masoumzadeh

Islamic Azad University of Qazvin
Department of Computer Science, Qazvin, Iran
saeid@masoumzadeh.com

ABSTRACT

This paper presents a vision system for a Ping-Pong player robot, called Robo-Pong. The robot employs color object detection techniques based on neural networks in its vision system. In this approach a quite simple architecture is employed to detect and localize objects in robot's work space. The architecture is designed to be very easy-implement and also surprisingly fast to work on such a real-time system. Also a mapping system is attached to the object detection one, in order to estimate object locations. To increase the real-time in-field train capabilities of the system some early stopping methods were exploited to deal with such vast train data.

1. INTRODUCTION

In recent years, many scientists and students have their robots compete with each other in robotic competitions in a variety of ways. Micromouse maze contest, robocup robot soccer, and AAAI robot competition are the most famous contests that are held each year [1]. Early works of researchers, show that humanoid robots have been attracted them in this field, which Ping-Pong player robots could be classified in this category [2]. In [3] Miyazaki et al. worked on a method of generating stroke movements for table tennis player robot. Also in [4] Matsushima et al. presented a method for controlling the paddle to return the ball to a desired point on the table with specified flight duration.

In our paper the proposed vision system designed for Robo-Pong is expected to detect and localize the ball and some markers on the table and robot's work space. To have the ball location in 3-dimentional coordinate, two BASLER 310A cameras were used as side and top view eyes. For this reason a wide 4mm lens is employed for top view camera to cover all parts of environment. Cameras are connected through a fire-wire link to an IEEE 1394 computer plug and capturing video in 60 frames per second in the size of 640x480 pixels with 24bit depth. Once the capture started, the robot can localize the ball and every visible marker. The Robo-Pong is shown in Figure 1.

Recently, there has been a tremendous amount of development in color recognition. Color recognition can be fulfilled in a variety of methods such as Spatial-Color Joint Probability [5], Color Profile [6], Histogram Threshold [5], Similarity of Color Histograms [7] and state-of-the-art methods based on Neural Networks [8], [9].

Stoksik, et al. [10] designed a color recognition system which was based on artificial neural network, whose generalization capabilities enabled the system to overcome limitations of traditional algorithm or microprocessor based systems. They described a new design using a combination of an artificial neural network to replace the database and an "expert system" to interpret the color. Their back-propagation neural network had three input neurons plus 35 hidden layer neurons and 10 output neurons. The three input neurons are for the three input signals (R, G and B) from a probe. The ten output neurons were each designated red, green, blue, brown, gray, purple, yellow, orange, white and black. More than one output neuron could be activated at any time, so colors such as light greenish-blue would be represented by outputs on the white, green and blue output neurons. They obtained some erroneous results for some lighter and darker shades of the representative colors. To alleviate this problem, they decided to train lighter and darker shades of the representative colors, by partially activating the white or black output unit in the training set.

Lalanne and Lempereur [11] used color information obtained from thermal paints to measure the temperature of combustion chamber or on high speed components under combustion test which is used by gas turbine manufacturers. At the first stage, to train the system they pass temperature and the corresponding color pattern information to the network. After training procedure the color-to-temperature function, constructed by trained network, can be applied to images of interest to generate temperature maps giving color information of each pixel.

Pomerleau [12] chose a multi-layer perceptron (ALVINN – Autonomous Land Vehicle in a Neural Network) with a single hidden layer for mobile robot guidance. The input layer of his network consists of a 30x32 unit retina which receives images from a sensor. Each of 960 units in the input retina is fully connected to the hidden layer of 4 units, which in turn is fully connected to 30-unit-ouput layer. ALVINN learns to guide mobile robots using the back propagation training algorithm. He presented real time training "on-the-fly" techniques to train his network. He applied his network on a CMU Navlab which is a modified Chevy van equipped with forward and backward facing color cameras and scanning laser range finder for sensing the environment.

In this paper, first the training data for the color recognition network is prepared through a statistical method. The neural network model put forth a robust vision system since a properly trained back-propagation network tends to give reasonable answers when presented with inputs that they have never seen. Therefore, an extremely easy method to implement a neural network is presented and the results are applied on the robot in order to have the highest performance. Moreover, a mapping network is designed to find the relevant distance of detected objects, ball and markers, from the robot. Finally, a GUI is employed to make different modules work with each other and make a user friendly environment.

Figure 1. *The Robo-Pong and field markers.*

2. TRAINING DATA REPRESENTATION

A training data base containing approximately 128000 different colors are prepared through statistical method. The training data is prepared by r, g, b of different colors and the number of times that this color occurs in each object. Therefore, if a color occurs in object one 50 times and in object two 80 times and in object three 1100 times, this color is considered the color of object three. These colors are put in arrays of [r,g,b,ObjType] and ObjType is in binary format for training of the network.

3. SYSTEM ARCHITECTURE

In order to recognize objects, the following system architecture has been designed (Figure 2). At first, camera is precisely calibrated and some filters are applied to improve lighting conditions. Afterward, the object detection neural network is applied to captured frames and used to segment parts according to their colors. On the basis of color, a binary image for each object is obtained and the noises are removed by median filter. By exploiting statistical information on solidity and convex area of each object, different objects are detected and their centriod are extracted. Finally, by a mapping neural network the objects absolute coordinates are extracted.

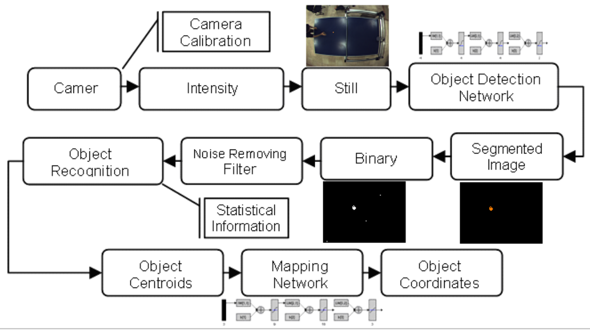

Figure 2. *System Architecture.*

4. OBJECT DETECTION NEURAL NETWORK

A truly autonomous robot must sense its environment and react appropriately. Previous robot perception systems have relied on hand-coded algorithms for processing sensor information. Artificial neural networks can learn the visual processing required to provide information for robot reactions.

It is extremely challenging for the Robo-Pong to be fast and precise and having low sensitivity to different lighting conditions; therefore, it was considered that neural network be applied on the Robo-Pong robot. However, it is an immensely time consuming and tiresome process to implement a neural network in generic programming languages like Delphi, C# and any other languages; consequently, we have taken advantage of MATLAB neural network toolbox to obtain optimum weight and bias matrices. Afterward, these matrices are exploited in a generic programming language such as C# which is much faster than MATLAB in multiplying neural network matrices for object detection. In other words, training of the neural network is done in MATLAB and operations which give us the color of objects are done in C#. It should be mentioned that color features, shape features, and combination of color and shape features can be used for classification. However, overlapped objects in 2D image, lighting condition that cause deformations on objects' boundary view, similarity in shape of table and net in some views, and expensive computational cost, forced us to employ a neural network on the basis of color classification.

Real-time reaction is a vital ability for Robo-Pong, so it is essential to have a neural network which has the minimum number of layers and neurons as it dramatically affects the calculation time. As an example, if one row is added to the weight matrix of a perceptron network which has architecture of 2×4, 1228800 more multiplication and 1228800 more summation in a calculation for a 640×480 pixel image have to be done. On the other hand, the network should provide us with the desired results. Accordingly, different variety of neural network with different training parameters and different types of output are implemented and tested. Networks such as perceptron and feed-forward back propagation that are suitable for classification are tested with outputs in the form of two binary digits and four binary digits in order to detect four types of objects viewed as ball, table and markers in the match field. The networks are provided with different image formats with different lighting conditions in order to make the robot vision more stable. If we compare different image formats, we find out the best input format for our network is rgbI. Several reasons support this choice. First of all, if a comparison is made between RGB and HIS, we find out that RGB is much more sensitive to lighting condition than HIS. Besides, adding the intensity feature to RGB will give a more robust response to lighting conditions. If rgb (RGB ratios) are applied, a more robust response to different lighting conditions will be gained. Also, in different lighting conditions rgb remains constant but in HIS model only hue and saturation remains constant and intensity differs.

Since there are nearly 47000 samples (pixels) to be trained to the network and we only had two options for our output (two or four binary digits), a perceptron network is not suitable for this purpose and the desired outcome could not be reached. Hence, different kinds of feed-forward neural networks with different number of layers and neurons are tested. The optimum neural network is a feed-forward back propagation network that has four layers. As properly trained feed-forward back propagation networks tend to give reasonable answers when presented with inputs they have never seen [13], a feed-forward network has been employed. For pattern recognition problems, if a comparison is made between different kinds of function approximation and pattern recognition problems in MATLAB [13], we find out that for color segmentation which is a pattern recognition problem a four-layer network will be suitable; however, in some other applications it may not be suitable. The input layer has four neurons and the output layer has two neurons. A sigmoid (logsig) function is applied for output layer in as much as our output is in binary form. The transfer function for all layers of feed forward network is logsig since it compresses input range into a finite output range. The inputs for the network are r,g,b,I.

Appropriate training function for pattern recognition problems are trainrp (resilient back propagation) or conjugate gradient algorithms such as trainscg (scaled conjugate gradient) or traincgb. Our designed network is trained by trainrp. In order to prevent over-fitting of the network, the training data which is large enough to provide an adequate fit is considered. Moreover, the output of the network is rounded for the reason that the desired output should be in binary format. The network architecture is shown in Figure 3.

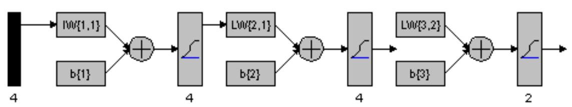

Figure 3. *The neural network architecture.*

Later on, the weight and bias matrices are extracted from MATLAB and the results are passed to C#. Figure 4 shows the output of the neural network after converting the binary output format to a color image.

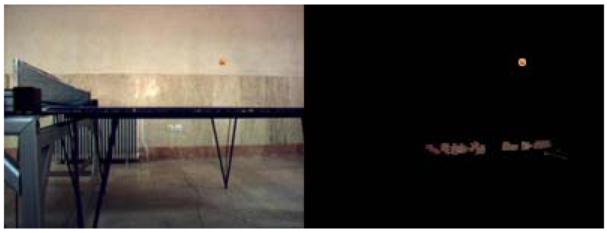

Figure 4. *Input image / output image.*

5. REGION SEGMENTATION AND DECISION MAKING

Having determined pixels related to each object, we have three binary images. What needed are images that are segmented with regions according to each object (Figure 5). To this end, sharp noises should be removed. Therefore, a pixel surrounded by pixels with a different color is not related to the desired object; consequently, these pixels can be assumed as noise. To remove these noises from our images, a median filter with 3×3 neighborhood is used.

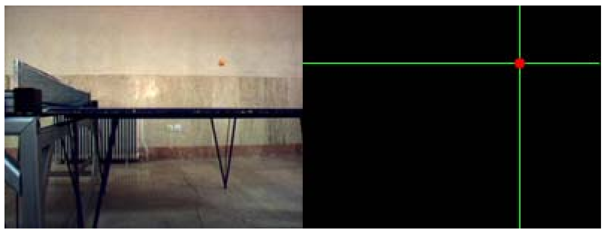

Figure 5. *Input image / output binary image.*

Having extracted regions related to each object, Convex-Hall of each region is calculated. Those regions that have solidity and convex area obtained according to statistical information are accepted as objects. It should be mentioned that solidity is the ratio of area to convex-area. Afterward, objects' center of area is extracted. The procedure is shown in Figure 6.

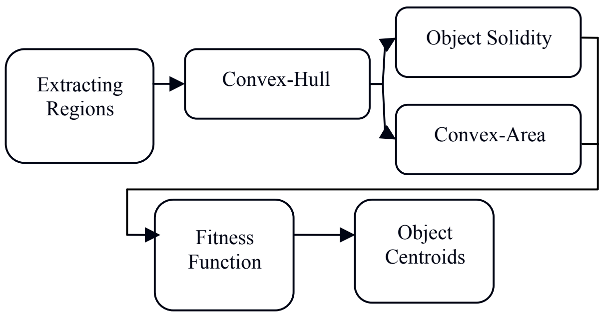

Figure 6. *Procedure for detecting objects' centre of area.*

Since we have spotted noise caused by shading effects which results in some regions bigger than could be removed by normal noise reduction filter, the convex area of segmented regions is considered to find those big enough to construct our desired objects. Another advantage of convex area is the improvement in the accuracy of coordinates reported for detected objects. Some critical deformations in shape of detected objects could be compensated making decision based on convex area to avoid these situations. As expected, in case of utilizing area of region instead of convex area the center of object does not sit on the center of the shape.

6. DETECTING POSITION OF OBJECTS

Through region growing procedure the centers of convex-hulled regions is also calculated which these kinds of data for the detected objects is sent to mapping procedure to produce commands for motion control module of the robot. These commands contains the distances of objects from robot and also the angles that robot should turn to reach each object. For example for the image shown in Figure 4, which has the size of 640×480, the centers of objects are such described in Table 1.

Table 1. *Centre of detected balls and markers.*

Ball Positions					
X	180.82	305.78	301.97	309.25	468.06
Y	99.96	216.62	79.50	167.05	303.40

Markers' Positions			
X	155.42	385.93	565.23
Y	220.58	86.82	109.17

7. DETECTING POSITION OF OBJECTS

Having determined the objects centers in the image, now we have to convert these coordinates into distances and angles that the robot can put them into action. To this end, another neural network is employed. The training data for this network is prepared through experimental tests in Mechatronics Research Laboratory of IAUQ and a 60-data-set is prepared. This network which is used for function approximation gets x, y coordinates and type of the object as input and hands in (r, θ) which is distance and angle as output. Again we had to find optimum network. Likewise, networks with biases, a sigmoid layer, and a linear output layer are capable of approximating any function with a finite number of discontinuities. Eventually, a network with the architecture shown in Figure 7 is an appropriate network. The inputs are x, y coordinates in pixel unit in addition to object type and the outputs are r in cm , θ in degrees and the object type. The network is trained with the 40-data-set and validated with a 10-data-set through training and tested with a 10-data-set. The training method is Levenberg Marquardt with early stopping, in order to prevent over fitting and to improve generalization of the network.

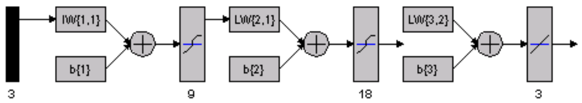

Figure 7. *Mapping neural network.*

Prediction error for distance is 2.6% and prediction error for orientation is 3.4% for the 10 test data set and is gained by Equation (1).

$$\delta = \frac{1}{N}\sum\left|\frac{M-P}{M}\right| \times 100\% \qquad (1)$$

Where δ is mean prediction error, M is the measured error value, P is the predicted error value and N is the number of test data set.

Figure 8 shows measured values vs. predicted value for test data set.

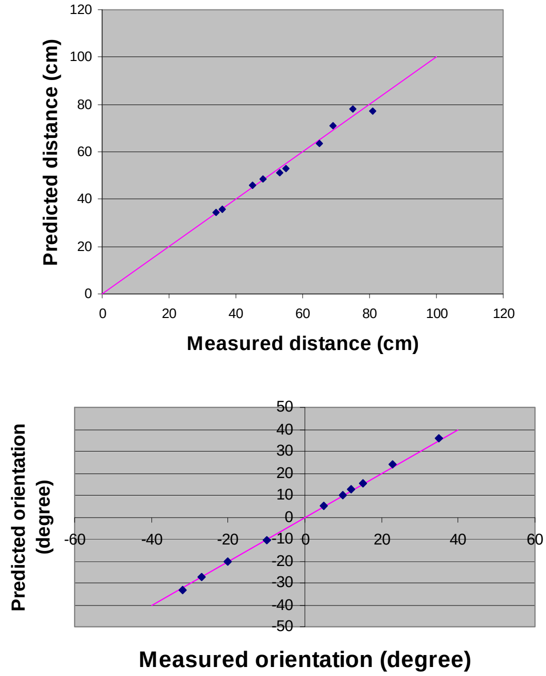

Figure 8. *Measured vs. predicted values for test data set.*

The Results of above network for the picture of Figure 4 is given in Table 2.

Table 2. *Position of objects relative to camera.*

Ball Positions					
Distance from camera	34.5	52	55	63.5	71
Angles from camera	-1	-23	-20	-40	-13

Markers' Positions			
Distance from camera	53	68	63
Angles from camera	-43	-9	15

The above mentioned algorithms designed in modular format; therefore, an integrated GUI is designed in MATLAB workspace. Capturing image, applying filters, calculating convex-hall and solidity, and sending data to robot are done in MATLAB; however, color segmentation is done with a C# program.

8. CONCLUSIONS

In this paper, a vision system based on neural network was established for a Ping-Pong player robot, called Robo-Pong, which can be seen in Figure 9. The training data on the basis of statistical method which is the number of times that a specified color occurs in an object was prepared. A feed-forward back-propagation network with an optimum number of neurons was designed in MATLAB. The optimal number of neurons was gained through trial and error for having the best performance (minimum number of neurons) and quality in detecting the color of objects. This process is very easy and can be done in a short time in comparison with implementing this method with generic programming languages. The weight and bias matrices of the trained network was extracted and utilized in C# in order to reduce computation time in practice which is critical in robocup competition. Moreover, for finding the location of the objects, a mapping network was designed and for over-fitting prevention early stopping method was applied for the training.

This method is very fast and reliable method for defining the color of objects, not only in the robotic field, but also in other automation processes in which color of the objects are needed to be defined.

Figure 9. *The ping-pong player robot, Robo-Pong.*

9. REFERENCES

[1] T. Braunl, “Research relevance of mobile robot competitions,” *Robotics & Automation Magazine, IEEE*, vol. 6, 1999, pp. 32-37.

[2] R.L. Andersson, *A robot ping-pong player: experiment in real-time intelligent control*, MIT Press, 1988.

[3] F. Miyazaki et al., “Realization of the table tennis task based on virtual targets,” *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, vol. 4, 2002.

[4] M. Matsushima, T. Hashimoto, and F. Miyazaki, “Learning to the robot table tennis task-ball control & rally with a human,” *Systems, Man and Cybernetics, 2003. IEEE International Conference on*, vol. 3, 2003.

[5] J. Luo and D. Crandall, “Color object detection using spatial-color joint probability functions,” *Image Processing, IEEE Transactions on*, vol. 15, 2006, pp. 1443-1453.

[6] J.A.T. da Costa and A.P. Dias, “Color profile function for morphological image segmentation,” *Proceedings of SPIE*, vol. 5667, 2005, p. 275.

[7] L.S. Wang et al., “Image segmentation based on optimal histogram threshold by improved genetic algorithms,” *J. Data Acquis. Process*, vol. 20, 2005, pp. 130-134.

[8] M. Korayem et al., “A Vision System Based on Neural Network for a Mobile Robot and Statistical Analyses of Its Errors,” Feb. 2007, pp. 2351-2356.

[9] X. Zhang and A.L.P. Tay, “Fast Learning Artificial Neural Network (FLANN) Based Color Image Segmentation in RGBSV Cluster Space,” *Neural Networks, 2007. IJCNN 2007. International Joint Conference on*, 2007, pp. 563-568.

[10] M. Stoksik, D.T. Nguyen, and M. Czernkowski, “A neural net based color recognition system,” *Artificial Neural Networks, 1991., Second International Conference on*, 1991, pp. 86-90.

[11] T. Lalanne and C. Lempereur, “Color recognition with a camera: a supervised algorithm forclassification,” *Image Analysis and Interpretation, 1998 IEEE Southwest Symposium on*, 1998, pp. 198-204.

[12] D.A. Pomerleau, *Neural Network Perception for Mobile Robot Guidance*, Kluwer Academic Publishers Norwell, MA, USA, 1993.

[13] H. Demuth, M. Beale, and I. MathWorks, *Neural Network Toolbox for Use with MATLAB: User's Guide*, MathWorks, Inc, 2006.