FACTA UNIVERSITATIS
Series: **Automatic Control and Robotics** Vol. 11, N° 2, 2012, pp. 129 - 140

INTELLIGENT CONTROL OF DaNI ROBOT BASED ON ROBOT VISION AND OBJECT RECOGNITION*

UDC 007.52 62-52:004.932.72'1

Ivan Ćirić, Žarko Ćojbašić, Miša Tomić, Milan Pavlović, Vukašin Pavlović, Ivan Pavlović, Vlastimir Nikolić

Department of Mechatronics and Control, University of Niš,
Faculty of Mechanical Engineering, Niš, Serbia

Abstract. *In order to use the mobile robot platform for tracking particular moving objects it is necessary to develop intelligent top level control algorithm. In this paper a computational intelligence based object recognition algorithm in robot vision system is presented. The main goal of this research was to enable DaNI robot to recognize the particular Lego NXT robot among other differently shaped Lego NXT robots and objects, and to localize them with accuracy high enough to allow following the chosen one. The necessary robustness of the 2D object recognition is achieved by a novel robust robot vision systems that introduces the closed-loop control of image segmentation without the use of extensive previous knowledge and computational intelligence as an important part of the vision system. Reliable feature extraction is necessary to fully exploit intelligent classifiers, which are the core of the proposed 2D object recognition method. Two different types of classifiers were developed and then compared, the ANFIS neuro-fuzzy classifier and neural network classifier. The supervisory control algorithm was tested through experiments and the results are showing that this kind of approach in robot vision for object recognition provides good results. Besides that, future use of computational intelligence techniques in robotic vision object recognition system is also briefly discussed.*

Key words: *robot vision, object recognition, computational intelligence, artificial neural network, neuro-fuzzy classifier*

Received November 17, 2012
Corresponding author: Ivan Ćirić
Department of Mechatronics and Control, University of Niš, Faculty of Mechanical Engineering, Aleksandra Medvedeva 14, 18000 Niš, Serbia • E-mail: ciric.ivan@masfak.ni.ac.rs
* **Acknowledgement**: This research was supported by the Ministry of Education, Science and Technological Development of Republic of Serbia, and DAAD through the projects HDTR and TR35005.

1. INTRODUCTION

Nowadays, recognizing objects in a scene is of significant interest and thus is one of the most active research areas in the field of artificial intelligence, because of its applications to such areas as video surveillance, robotics, autonomous navigation, etc.

Many research works have been conducted in the object recognition field. Belongie et al. [1] presented an approach to measuring similarity between shapes and exploited it for object recognition, while Juwei et al. [2] and Turk et al. [3] used the linear discriminate analysis (LDA) and the principal component analysis (PCA) to recognize the object. The proposed algorithms of these researchers focused on just one object recognition and required a cleaned background, and because of that they are not useful in real environment. Yun et al. [4] proposed a real-time object recognition using the relational dependency among the objects that is represented by the graphical model.

Neural networks have been widely used for robotic control and object recognition because of their ability to approximate arbitrary linear or non-linear systems in a compact set. Jung et al. [5] proposed a new Hybrid Hopfield Neural Network (HHN) algorithm for machine vision applications, which combines the advantages of both a Continuous Hopfield Network (CHN) and a Discrete Hopfield Network (DHN) that was applied for partially occluded object recognition in multi-context scenery. Lee et al. [6] proposed a mean field annealing neural network (MFANN) for three-dimensional object recognition. In paper [7] moving object recognition by a shape-based neural fuzzy network is proposed. The moving objects considered in this paper include pedestrians, vehicles, motorcycles, and dogs, while in paper [8] driver assistance system for both the motorway and inner city environment was presented. This system, used for detection of overtaking vehicles and recognition of pedestrians, relied on the adaptable time delay neural network (ATDNN) algorithm.

In this paper the aim is to teach DaNI robot [9] to recognize one from two Lego NXT robots [10], which are assembled in different shapes, and to follow it. We proposed robust robot vision systems, in which the necessary robustness of the 2D object recognition is achieved by introduction of the closed-loop control of image segmentation without use of extensive previous knowledge and also by introduction of computational intelligence as important part of the vision system [11]. The main idea behind this is the automatic adjustment of the segmentation parameters instead of using their default values to provide reliable feature extraction and consequently the full use of all benefits of computationally intelligent classifiers which are the core of the recognition methods proposed [12]. Besides the proposed reliable computationally intelligent object recognition in robotic vision, in this paper the future use of computational intelligence techniques in robotic vision object recognition system is also briefly discussed.

2. EXPERIMENTAL ROBOT PLATFORM

Robotics and automation are becoming an essential component of engineering and scientific systems and consequently they are very important topics for study by engineering and science researchers. Furthermore, robotics is built on fundamentals like transducer characterization, motor control, data acquisition, mechanics of drive trains, network communication, computer vision, pattern recognition, kinematics, path planning, and others that are also fundamental to other fields, manufacturing, for instance.

For the development of reliable computationally intelligent object recognition in robotic vision, National Instruments Robotics Starter Kit 1.0 has been used. NI Robotics Starter Kit 1.0, known as DaNI robot, (Fig. 1) is a mobile robot platform that features sensors, motors, and NI Single-Board RIO hardware for embedded control. DaNI is a four-wheel robot, powered with two motors and equipped with ultrasonic distance sensor for distance measurements.

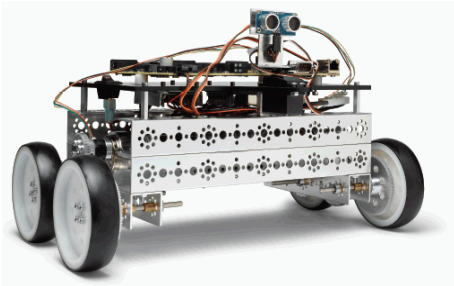

Fig. 1. DaNI robot

We built DaNI-C system (DaNI with camera) for object recognition [15]. DaNI-C is equipped with various sensors, where a web camera providing visual information about the robotic system's environment is one of the most important. Web camera was connected to laptop, which was linked with DaNI robot via Ethernet cable (Fig. 2). DaNI-C was controlled by LabVIEW 2011.

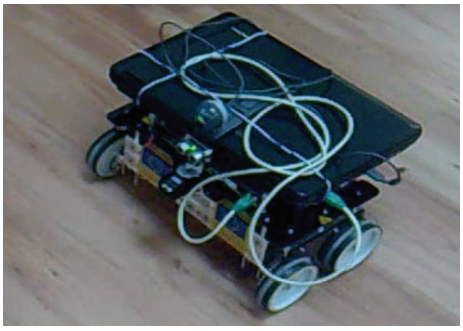

Fig. 2. DaNI-C

3. Computationally Intelligent Object Recognition

In this study, the effectiveness of the DaNI-C vision system due to the inclusion of feedback control of image segmentation and computationally intelligent classification has been tested for the next working scenario. The scenario was to teach DaNI-C robot system to recognize one from two Lego NXT robots, which are assembled in different shapes, and to follow it. One NXT robot is assembled as robogator, and other as shooterbot (Fig. 3). To allow the DaNI-C system to perform these tasks, it is crucial for the robotic system to perceive its environment visually. In particular, the robot vision system must be able to recognize the robogator and shootherbot among the other objects and to localize them with a high enough accuracy.

Fig. 3. NXT robots

3.1. Feature extraction from segmented image

The segmentation of the objects from the background is done by segmentation of the colored regions in the so-called Hue image, which contains the pure colour information of the original RGB (red, green, blue) image of a DaNI-C scene. Once the objects have been segmented, different features describing segmented object regions are calculated. The following chosen two features have been defined for satisfying required criteria.

▪ **Connectivity.** The spatial connectivity of a segmented object pixel and its neighbourhood segmented pixels can be expressed by the following measure:

$$I = -\log_2 p_{8,}\ I(0) = 0 \tag{1}$$

where p8 is a segmented pixel probability estimate, surrounded with 8 segmented pixels in its 8-pixel neighbourhood:

$$p_8 = \frac{\text{no of seg pixels surrounded with 8 seg pixels}}{\text{total no of segmented pixels}}. \tag{2}$$

It is evident from (2) that a small probability p8, which reflects scattered unconnected segmented pixels, corresponds to a large *I*. On the other hand, p_8 close to 1 gives a small *I* reflecting a well segmented image region of connected pixels.

▪ **Proportionality.** The proportionality of a region of connected segmented pixels can be expressed as follows:

$$p_{\mathrm{r}} = h \cdot w^{-1} \tag{3}$$

where h and w are respectively height and width of the bounding box of the segmented region. The bounding box is the smallest rectangle containing the segmented region.

To achieve good object segmentation it is necessary to adjust the object segmentation interval as illumination conditions change. Finding the optimal object segmentation interval that provides a segmented object image of good quality, appropriate for subsequent object feature extraction for object recognition and reconstruction, can be interpreted and converted to the problem of finding the minima of input-output characteristics where input (actuator variable) is segmentation increment and output (controlled variable) is connectivity *I*. The optimal value of the chosen controlled variable, that satisfies the input-output controllability condition, can be achieved by an optimization process using an appropriate extremum seeking algorithm through a closed-loop control structure.

Closed-loop segmentation results are optimal segmentation intervals for differently colored objects from the robot environment [12]. Hence, the segmentation finally results in as many binary images as there are objects present in the image.

The recognition of robots, using computationally intelligent classifiers, was done based on the robot's connectivity and proportionality descriptors extracted from the resulting binary segmented images. For real world application it is of particular interest to perform object classification reliably and effectively. This means that the applied classifier has to satisfy goals of high recognition accuracy and small computing time. For recognition purposes, two computationally intelligent classifiers were considered, namely an neuro-fuzzy (Adaptive Neuro-Fuzzy Inference System, ANFIS) classifier and a neural network classifier.

3.1. ANFIS neuro-fuzzy classifier

Neural networks are well known as an alternative to statistical techniques for pattern classification, which have high performance in terms of recognition accuracy but often do not meet the requirement for small response time. On the other hand, neural networks are superior in classification time to statistical techniques with similar or even better recognition accuracy. ANFIS can be seen as structure equivalent to a radial basis function (RBF) neural network. However it is a hybrid algorithm of both fuzzy logic and artificial neural network. Therefore, the ANFIS classifier has all advantages of these systems and, besides, its hybrid learning algorithm offers superior training results in comparison to other methods. ANFIS classification has strong potential of being very robust against system uncertainties and has capability to efficiently dealing with significantly increased number of object classes. This is of particular importance for further development of the mobile robotic system able to function autonomously in a complex unstructured environment. Also, there is a possibility to increase dimension of object feature space vector by adding more features without significant changes in classification strategy.

Consider a first-order TSK fuzzy inference system that consists of two rules

Rule 1: If X is A_1 and Y is B_1 then $f_1 = p_1x + q_1y + r_1$

Rule 2: If X is A_2 and Y is B_2 then $f_2 = p_2x + q_2y + r_2$

If f_1 and f_2 are constants instead of linear equations, we have a zero-order TSK fuzzy model. Figures 4(a) and (b) illustrate the fuzzy reasoning mechanism and the corresponding ANFIS architecture, respectively.

Node functions in the same layer of ANFIS are of the same function family, as described below. Note that O_i^j denotes the output of the ith node in layer j.

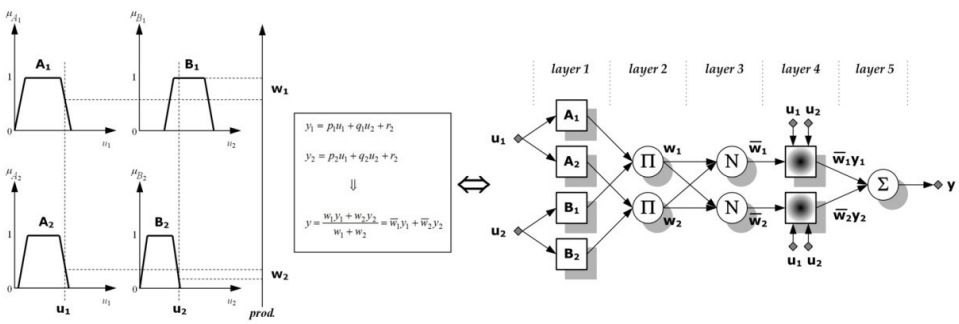

Fig. 4. First-order TSK fuzzy model using trapezoidal membership functions and the corresponding ANFIS architecture

Layer 1: Each node in this layer generates membership grades of a linguistic label. For instance, the node function of ith node might be

$$O_i^1 = m_{A_i}(x) = \max\left[\min\left(\frac{x-a}{b-a},\ 1,\ \frac{d-x}{d-c}\right),\ 0\right], \tag{4}$$

where x is the input to node i; A_i is the linguistic label (small, large, etc.) associated with this node; and $\{a, b, c, d\}$ is the parameter set that changes the shape of the trapezoidal membership function. Parameters in this layer are referred to as the premise parameters.

Layer 2: Each node in this layer calculates the firing strength of each rule via multiplication

$$O_i^2 = w_i = \mu_{A_i}(x) \times \mu_{B_i}(y), \;\; i = 1, 2 \tag{5}$$

Layer 3: The ith node of this layer calculates the ratio of the ith rule's firing strength to the sum of all rules firing strength

$$O_i^3 = \overline{w}_i = \frac{w_i}{w_1 + w_2}, \;\; i = 1, 2 \tag{6}$$

Layer 4: Node i in this layer has the following node function:

$$O_i^1 = \overline{w}_i f_i = \overline{w}_i (p_i x + q_i y + r_i) \tag{7}$$

where w_i is the output of layer 3 and $\{p_i, q_i, r_i\}$ is the parameter set. Parameters in this layer will be referred to as the consequent parameters.

Layer 5: The single node in this layer computes the overall output as the summation of all incoming signals overall output

$$O_i^3 = overall \;\; output = \sum_i \overline{w}_i f_i = \frac{\sum w_i f_i}{\sum w_i} \tag{8}$$

The hybrid learning algorithm of ANFIS consists of two alternating parts:

1) BP/GD which calculates error signals (defined as the derivative of the squared error with respect to each node output) recursively from the output layer backward to the input nodes, and

2) the RLSE method, which finds a feasible set of consequent parameters. We observe that, given fixed values of premise parameters, the overall output can be expressed as a linear combination of the consequent parameters

$$f = \overline{w}_1 f_1 + \overline{w}_2 f_2 = (\overline{w}_1 x) p_1 + (\overline{w}_1 y) q_1 + (\overline{w}_1) r_1 + (\overline{w}_2 x) p_2 + (\overline{w}_2 y) q_2 + (\overline{w}_2) r_2 \tag{9}$$

Equation (9) can be recast as a matrix equation

$$\mathbf{AX} = \mathbf{B}\,, \tag{10}$$

where $\mathbf{X}$ is an unknown vector whose elements are the consequent parameters. Least-squares estimate (LSE) of $\mathbf{X}$, namely $\mathbf{X}^*$, is sought to minimize the squared error $\|\mathbf{AX} - \mathbf{B}\|^2$. Sequential formulas are employed to compute the LSE of $\mathbf{X}$. Specifically, let the ith row vector of matrix $\mathbf{A}$ defined in (10) be a_i^T and the ith element of $\mathbf{B}$ be b_i^T. Then

$$\mathbf{X}_{i+1} = \mathbf{X}_i + \mathbf{S}_{i+1} a_{i+1} (b_{i+1}^T - a_{i+1}^T \mathbf{X}_i), \;\; \mathbf{S}_{i+1} = \mathbf{S}_i - \frac{\mathbf{S}_i a_{i+1} a_{i+1}^T \mathbf{S}_i}{1 + a_{i+1}^T \mathbf{S}_i a_{i+1}}, i = 0, \; 1, \; \ldots, \; P-1\,, \tag{11}$$

where $\mathbf{S}_i$ is often called the covariance matrix and the least-squares estimate $\mathbf{X}^*$ is equal to $\mathbf{X}_p$. The initial conditions to bootstrap (11) are $\mathbf{X}_0 = 0$ and $\mathbf{S}_0 = \gamma\mathbf{I}$, where γ is a positive large number and $\mathbf{I}$ is the $M \times M$ identity matrix, where M is the number of consequent parameters. For given fixed values of premise parameters, the estimated consequent parameters are known to be globally optimal.

The purpose of clustering is to distil natural groupings of data from a large data set, producing a concise representation of system behaviour. The quick subtractive or MMC clustering technique was developed by Yager/Filev and modified by Chiu [13]. The clustering of I/O data produces a set of cluster centers, and each cluster center acts as a prototypical data point that describes a characteristic mode of the system, and can be considered as the nucleus of a fuzzy if-then rule. In that way partitioning of the inputs and determination of the initial minimal fuzzy rule base can be performed.

Namely, if a collection of n-normalized data points $\{x_1, x_2, ..., x_n\}$ in an M-dimensional space is considered, measure of the potential of data point can be defined as

$$P_i = \sum_{j=1}^{n} \exp\left(-\alpha \left\| x_i - x_j \right\|^2\right), \ \alpha = 4 / r_a^2 \tag{12}$$

The constant r_a is effectively the radius defining a neighbourhood. After the potential of every data point has been computed, the data point with the highest potential is selected as the first cluster center. If x_1^* is the first cluster center with potential P_1^*, the potential of each data point is revised by

$$P_i \leftarrow P_i - P_i^* \exp\left(-\beta \left\| x_i - x_j \right\|^2\right), \beta = 4 / r_b^2 \ , \tag{13}$$

where r_b is positive constant larger than r_a in order to avoid high density of the cluster centers [14].

The ANFIS architecture with 5 layers, the hybrid learning algorithm (back-propagation for nonlinear parameters and least squares method for linear parameters) [13] and Modified Mountain Clustering (MMC) [14] technique for the initial neuro-fuzzy classifier structure determination have been used. The initial classifier structure with 14 rules, and with 12 to 14 primary fuzzy sets in the spaces of input and output variables has been selected as optimal, after experimenting with both clustering based and arbitrary grid partitioning of the input spaces for initial classifier structure determination. This has been done with roughly 464 training features pairs. The same training set has been used for the training of the classifier, i.e. for the parameters adjustment. The training features were extracted from the segmented regions of different objects belonging to classes “robogator”, “shooterbot” and “no robot”. Segmented objects images were obtained using the proposed closed-loop segmentation of images of the DaNI-C environment captured at different time instances in a wide range of illumination conditions. Different illumination conditions as well as different cameras views cause the difference in the values of the descriptors of the same objects. The fuzzy surface of the trained classifier is presented in figure 5.

The testing of the developed classifier was done using another 232 objects feature descriptors pairs. These test features were also obtained from the experiments in DaNI-C

working scenarios. They were imaged in different illumination conditions, corresponding to a variation from a dark room lighted with candles (15lx) to the lighting level of an office according to the European law UNI EN 12464 (500lx).

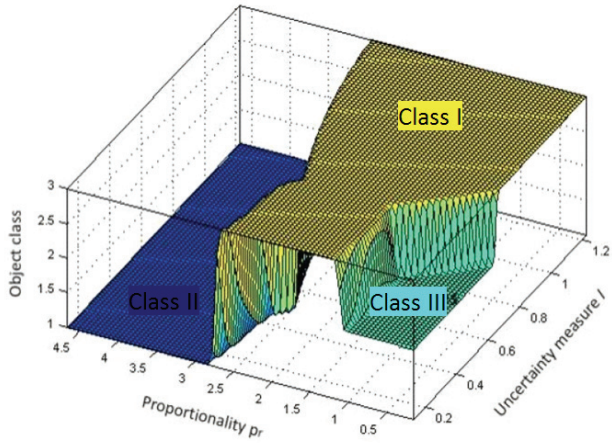

Fig. 5. Trained fuzzy surface of classifier

The obtained classification results are very good, as indicated by the fact that the classification performance rate was above 95%. Misclassification happened only in cases of significant object occlusion, which resulted in the improper value of connectivity measure of a segmented object region.

3.2. Artificial neural network classifier

Feed forward networks often have one or more hidden layers of nonlinear neurons followed by an output layer of linear neurons. Multiple layers of neurons with nonlinear transfer functions allow the network to learn nonlinear relationships between input and output vectors. These multilayer feed forward networks were trained for function approximation (nonlinear regression). The training process required a set of examples of proper network behaviour - network inputs u and target outputs y.

The process of training a neural network involves tuning the values of the weights and biases of the network to optimize network performance, as defined by the network performance function. There are generally four steps in the training process: assemble the training data, create the network object, train the network, and simulate the network response to new inputs. The common performance function for feed forward networks is the mean square error [13] between the network outputs $\hat{y}$ and the target outputs y, defined as follows:

$$F = mse = \frac{1}{N}\sum_{i=1}^{N}(e_i)^2 = \frac{1}{N}\sum_{i=1}^{N}(y-\hat{y})^2 \tag{14}$$

Optimization methods for performance function use the Jacobian of the network errors with respect to the weights. The gradient and the Jacobian are calculated using a back propagation algorithm, which involves performing computations backward through the network.

Back propagation [13,14,15] is the generalization of the Widrow-Hoff learning rule to multiple-layer networks and nonlinear differentiable transfer functions. Input vectors and the corresponding target vectors are used to train a network until it can approximate a function, associate input vectors with specific output vectors, or classify input vectors in an appropriate way as defined by the user. Networks with biases, a sigmoid layer, and a linear output layer are capable of approximating any function with a finite number of discontinuities.

Standard back propagation is a gradient descent algorithm, as is the Widrow-Hoff learning rule, in which the network weights are moved along the negative of the gradient of the performance function. The term back propagation refers to the manner in which the gradient is computed for nonlinear multilayer networks. There are a number of variations on the basic algorithm that are based on other standard optimization techniques, such as conjugate gradient and Newton methods.

Properly trained back propagation networks tend to give reasonable answers when presented with inputs that they have never seen. Typically, a new input leads to an output similar to the correct output for input vectors used in training that are similar to the new input being presented. This generalization property makes it possible to train a network on a representative set of input/target pairs and get good results without training the network on all possible input/output pairs.

The performance of a trained network can be measured to some extent by the errors on the training, validation and test sets, but it is often useful to investigate network response in more detail. One option is to perform a regression analysis between the network response and the corresponding targets. The ANN used in this study was a standard feed-forward, back propagation neural network with three layers: an input layer, a hidden layer consisting of 10 hidden neurons and an output layer. Input network variables were two object features, while three network outputs represent probabilities of object belonging to each of the three classes. The ANN classifier is shown in Figure 6. For training, the back propagation scaled conjugate gradient algorithm that updates weight and bias states according to Levenberg-Marquardt optimisation was used, while the mean squared error was used as a performance measure during training.

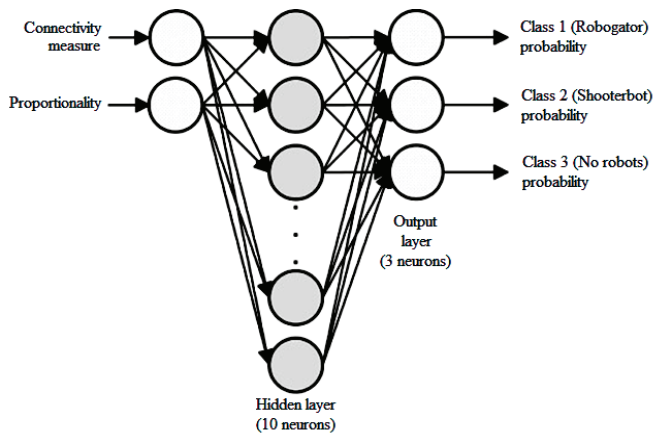

Fig. 6. Neuro classifier structure

For ANN classifier more demanding dataset of 1040 data triplets was used containing more severely occluded object features, that was randomly divided into training, validation and testing sets. The training set (728 samples) was presented to the network during training, and the network was adjusted according to its error. The validation dataset (156 samples) was used to measure network generalisation, and to halt training when generalisation stopped improving. Finally, the testing dataset (156 samples) had no effect on training and so provided an independent measure of network performance. Classification accuracy was above 97%.

4. Implementation and Results

DaNI-C recognized shooterbot through algorithm shown in Fig. 7. After the start of the program, image processing was first performed. DaNI-C took the image of the environment and then performed segmentation. From segmented image, program extracted features. Using those features, classifier recognized the objects in the image. If there was not a shootherbot in the image, a new image was acquired, and the mentioned steps were repeated. If the classifier recognized the shooterbot, then "Follow him" sequence of the program was initiated, and DaNI-C started to follow the shooterbot (Fig. 8).

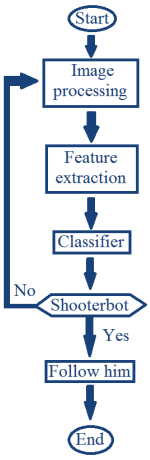

Fig. 7. Control algorithm

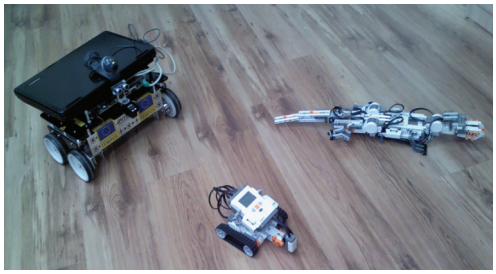

Fig. 8. DaNI-C in action

To analyze the performance of the recognition and following of shooterbot we used results received from a LabVIEW. These results are obtained from DaNI-C mobile robot platform.

Fig. 9 shows change of x coordinate (measured in pixels) vs. time, obtained by the robot vision system. Horisontal axis represents time, and vertical axis represents "x coordinate of the center of the mass of the shooterbot". Based on that, following of shooterbot is performed by DaNI-C, by deciding of which side to go in order to minimize the absolute value of x coordinate.

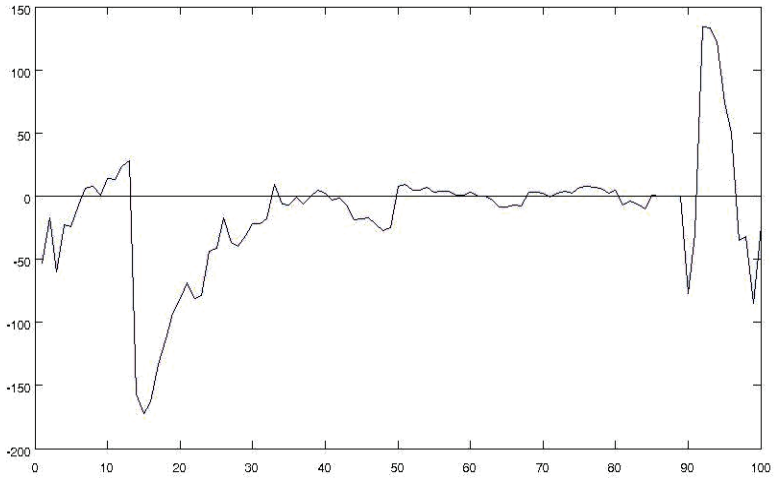

Fig. 9. Robot tracking of shooterbot in time measured in pixels

5. CONCLUSION

Robustness of the complete robot vision system against variable illumination is achieved by including feedback control at the image segmentation level, providing reliable feature extraction for neuro-fuzzy and neural classifiers, which are important parts of the proposed object recognition method. Experimental results show that this kind of approach in robot vision for object recognition gives good results.

Using the fact that both neuro-fuzzy and neural classifiers can handle different types of objects, and on the other hand classifiers made decisions which are reliable and accurate, the algorithm used in the paper can be implemented in different tasks for robot vision recognition mobile platform.

REFERENCES

1. S. Belongie, J. Malik, J. Puzicha, "Shape matching and object recognitionusing shape contexts", *IEEE Trans. Pattern Anal. Mach. Intell. 24 (24) (2002)*.
2. J. Lu, K.N. Platanoiotis, A. Venetsanopoulos, *Face recognition using LDA based algorithms*, Neural Networks (2002).
3. M.A. Turk, A.P. Pentland, "Face recognition using eigenfaces", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1991.
4. W.H. Yun, S. Y. Bang, D. Kim, "Real-time object recognition using relational dependency based on graphical model", *Pattern Recognition*, vol. 41, pp. 742 – 753, 2008.
5. J. H. Kim, S. H. Yoon, and K. H. Sohn, "A robust boundary-based object recognition in occlusion environment by hybrid hopfield neural networks", *Pattern Recognition,* vol. 29, No. 12, pp. 2047-2060, 1996.
6. Y.L. Lee, R.H. Park, "A surface-based approach to 3-D object recognition using a mean field annealing neural network", *Pattern Recognition* vol. 35, pp. 299 – 316, 2002.
7. C.F. Juang, L.T. Chen, "Moving object recognition by a shape-based neural fuzzy network", *Neurocomputing* vol. 71, pp. 2937–2949, 2008.

8. C. Wöhler, J.K. Anlauf, "Real-time object recognition on image sequences with adaptable time delay neural network algorithm – applications for autonomous vehicles", *Image and Vision Computing,* vol. 19, pp. 593-618, 2001.
9. http://sine.ni.com/ds/app/doc/p/id/ds-217/lang/sr
10. http://mindstorms.lego.com/en-us/history/default.aspx
11. D. Ristic-Durrant., S. M. Grigorescu, A. Graser, Z. Cojbasic, V. Nikolic, "Robust Stereo-Vision Based 3D Object Reconstruction for the Assistive Robot FRIEND", *Advances in Electrical and Computer Engineering,* pp. 15-22, 2011.
12. D. Ristic-Durrant, *Feedback Structures in Image Processing*, Ph.D. thesis, Shaker-Verlag, Germany, 2007.
13. R. Jang, "ANFIS: Adaptive network based fuzzy inference systems", *IEEE Trans. on Syst., Man and Cyber.,* vol. 23, no. 3, pp. 665-685, 1993.
14. S. Chiu, "Fuzzy Model Identification Based on Cluster Estimation", *Journal of Intelligent & Fuzzy Systems*, vol. 2, no. 3, 1994.
15. I. Ćirić, Ž. Ćojbašić, M. Tomić, M. Pavlović, V. Pavlović, „Computationally Intelligent Object Recognition for DaNI Robot Vision", Proceedings of XI International SAUM Conference 2012., Niš, Serbia, , pp.132 – 135, 2012.

INTELIGENTNO UPRAVLJANJE ROBOTOM DANI ZASNOVANO NA ROBOTSKOJ VIZIJI I PREPOZNAVANJU OBJEKATA

Ivan Ćirić, Žarko Ćojbašić, Miša Tomić, Milan Pavlović, Vukašin Pavlović, Ivan Pavlović, Vlastimir Nikolić

U cilju primene mobilne robotske platforme kod praćenja pojedinih pokretnih objekata neophodno je razviti inteligentni upravljački algoritam na najvišem nivou upravljanja. U ovom radu predstavljen je algoritam za prepoznavanje objekata u robotskoj viziji zasnovan na računarskoj inteligenciji. Glavni cilj istraživanja je osposobiti robot DaNI da prepoznaje jednog Lego NXT robota među drugim drukčije sklopljenim Lego NXT robotima i ostalim objektima, a da zatim dovoljno precizno odredi poziciju odabranog robota u prostoru kako bi ga pratio. Neophodna robusnost 2D prepoznavanja objekata postignuta je primenom novog robusnog sistema robotske vizije koja uvodi upravljanje sa povratnom spregom u segmentaciji slike bez primene dodatnog prethodnog znanja o sceni kao i računarsku inteligenciju kao bitan deo robotske vizije. Pouzdano odredjuivanje parametara iz slike je neophodno kako bi se u potpunosti iskoristile mogućnosti inteligentnih klasifikatora, koji su u osnovi predloženog metoda za 2D prepoznavanje objekata. Dva različita tipa klasifikatora su razvijena, ANFIS neuro-fazi klasifikator i klasifikator zasnovan na neuronskoj mreži, a nakon toga su njihove performanse uporedjene. Inteligentni upravljački algoritam je eksperimentalno testiran i rezultati pokazuju da ovakav pristup prepoznavanju objekata daje dobre rezultate. Osim toga, dalja upotreba tehnika računarske inteligencije u prpoznavanju objekata u robotskoj viziji je ukratko predstavljena.

Ključne reči: *robotska vizija, prepoznavanje objekata, računarska inteligencija, veštačke neuronske mreže, neuro-fazi klasifikator*