Learning and Fast Object Recognition in Robot Skill Acquisition: A New Method

I. Lopez-Juarez[1], R. Rios-Cabrera[1], M. Peña-Cabrera[2], and R. Osorio-Comparan[2]

[1] Centro de Investigación y de Estudios Avanzados del I.P.N. (CINVESTAV)
Ramos Arizpe. CP 25900. Coah. México
{ismael.lopez,reyes.rios}@cinvestav.edu.mx
[2] Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas,
Universidad Nacional Autónoma de México (UNAM)
Apdo. Postal 20-726, México DF, México
{mario,roman}@leibniz.iimas.unam.mx

Abstract. Invariant object recognition aims at recognising an object independently of its position, scale and orientation. This is important in robot skill acquisition during grasping operations especially when working in unstructured environments. In this paper we present an approach to aid the learning of manipulative skills on-line. We introduce and approach based on an ANN for object learning and recognition using a descriptive vector built on recurrent patterns. Experimental learning results using a fast camera are presented. Some simple parts (i.e. circular, squared and radiused-square) were used for comparing different connectionist models (Backpropagation, Perceptron and FuzzyARTMAP) and to select the appropriate model. Later during experiments, complex figures were learned using the chosen FuzzyARTMAP algorithm showing a 93.8% overall efficiency and 100% recognition rate with not so complex parts. Recognition times were lower than 1 ms, which clearly indicates the suitability of the approach to be implemented in robotic real-world operations.

Keywords: ART Theory, Artificial Neural Networks, Invariant Object Recognition, Machine Vision, Robotics.

1 Introduction

Grasping and assembly operations using industrial robots is currently based on the accuracy of the robot and the precise knowledge of the environment, i.e. information about the geometry of assembly parts and their localization in the workspace. Techniques are sought to provide self-adaptation for robots. This document reports a neural-based methodology for invariant object recognition applied to self-adapting industrial robots which can perform assembly tasks. New objects can also be learned quickly if certain clues are given to the learner, since the methodology uses only two on-line patterns for learning complex objects. The architecture is firstly trained with clues representing different objects that the robot is likely to encounter (and with others that represent complex objects) within the working space to form its initial

J.A. Carrasco-Ochoa et al. (Eds.): MCPR 2010, LNCS 6256, pp. 40–49, 2010.

knowledge base. The main idea suggests that it is possible to get fast and reliable information from a simple but focused analysis of what an object might show. The very important aspects of the scene (we have called "clues"), can be used later to retrieve memorized aspects of the object without having to recall detailed features. By using neural networks it is possible to learn manipulative skills which can be used by an industrial manipulator. In someway we humans do that process once an object has been seen and learned for the first time.

The article describes a robust method for very fast learning, perimeter and centroid calculations, object functions and pose estimation. The remainder of this paper is structured as follows. Section 2 reviews related work and state our contribution to the field of self-adaptive industrial robots for assembly and object recognition. In Section 3, the analysis of three ANN's is presented for selecting the appropriate connectionist model while in section 4 the methodology is explained. Experimental results from several object learning and recognition tasks are given in section 5. Finally, conclusions and future work is described in section 6.

2 Background and Related Work

2.1 Related Work

Many authors considered only constraint motion control during assembly; however, to complete the autonomy of the assembly system a machine vision system has also to be considered. Hoska, introduced the concept of "Robotic Fixtureless Assembly" (RFA), that eliminates the need of using complex and rigid fixtures, which involves new technical challenges, but allows potential solutions [1]. Ngyyuen and Mills, [2] have studied RFA of flexible parts with a dynamic model of two robots with a proposed algorithm, which does not require measurements of the part deflections. The goal of RFA is to replace these fixtures with sensor-guided robots which can work within RFA workcells. Using ANNs, an integrated intelligent vision-guided system can be achieved as it is shown by Langley, [3]. Many authors had come with descriptor vectors and image transformations, used as general methods for computer vision applications in order to extract invariant features from shapes. Aguado, et al. [4] developed a methodology for including invariance in general form of the Hough transform, Chin-Hsiung, et al. [5] designed a technique for computing shape moments based on the quadtree representation of images. Best and McKay [6] described a method for registration of 3D shapes in minutes. Bribiesca, [7] developed a new chain code for shapes composed of regular cells, which has recently evolved even to represent 3D paths and knots. But the methods require a multiple-pattern input for training. Some authors use multiple cameras/views to extract information, to perform invariant object recognition and determine object's position and motion, Gonzalez-Galvan, et al. [8] developed a procedure for precision measure in 3D rigid-body positioning using camera-space manipulation for robot operation. Dickmanns [9], Kollnig and Nagel, [10] have shown solutions to facilitate the use of vision for real world-interaction. Applications of guided vision used for assembly are well illustrated by

Bone and Capson, [11] which developed a vision-guide fixtureless assembly system using a 2D computer vision for robust grasping and a 3D computer vision to align parts prior to mating.

A previous method proposed by Peña-Cabrera [12] reported 100% of object recognition; however, there were used simple geometry (i.e. circular, squared and radiused-square pegs) and 72 patterns for training each of the three objects. The testbed was formed basically by a 6 DOF KUKA KR15 robot, master computer and a JR3 F/T sensor attached to the robot's wrist to form a feedback loop while assembling components. The F/T sensor was used only for alignment during part mating after the assembly component was recognised by the vision system (the reader is referred to [13] for details). The vision system was composed by a CCD Pulnix 6710 camera with 640x480 pixel resolution. POSE information was provided by the vision system to the master computer to generate the robot motion commands for component grasping. Once the assembly component was held by the robot, then the vision system also determined the assembly block location for assembly. Figure 1 shows the assembly components used for recognition and learning of robot manipulative tasks (i.e. assembly operations).

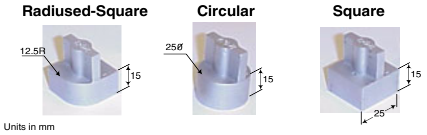

Fig. 1. Assembly Components

2.2 Original Work

The research presented in this paper is focused on learning and object recognition of simple and complex 2D objects in order to enable industrial robots to learn manipulative tasks (i.e. assembly operation). In this area moment invariants are still popular descriptors for image regions and boundary segments, but computation of moments of a two dimensional (2D) image involves a significant amount of multiplications and additions in a direct method. The computation of moments can be simplified since it contains only the information about the shape of the image as proposed by Chen, [14]. This paper proposes recurrent pattern vector descriptors, using collections of 2D images to obtain a very fast feature data of an object by using image projections and mirror images. Fast learning is achieved on-line considering only two patterns of simple –geometrical- and complex objects to achieve invariance to rotation, translation and scale. The fast algorithm allows calculation of a boundary object function (BOF) and centroid which defines object information, and also considers variance normalized grey-colour intensity properties, which in conjunction with an ANN forms a system called SIRIO which recognizes, learns and performs pose estimation of assembly components in the order of milliseconds what constitutes a practical tool for real-world robot applications.

3 Neural Networks Evaluation

The next subsections show the result in experimental comparisons among BP (Backpropagation), P (Perceptron), and FAM (Fuzzy ARTMAP) in order to select an appropriate model for the vision system. For training and testing purposes, the shapes of the patterns shown in figure 1 were used. Figure 2 shows the patterns.

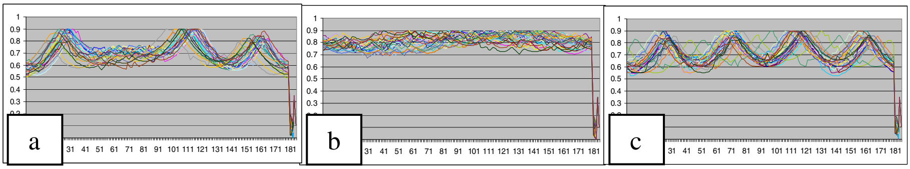

Fig. 2. a) radiused-square; b) Circle, c) square. Total=216 patterns (72 from each shape)

All ANN's were programmed in Visual C++ .NET using a Pentium D PC @ 2.8 GHz with 512 MB RAM.

3.1 Backpropagation

BP is a stochastic steepest descent learning rule used to train single or multiple layer nonlinear networks. The algorithm overcomes some limitations of the perceptron rule by providing a framework for computing the weights of hidden layer neurons, but also it takes more time for training/testing. The configuration was: layer_In=185, layer_hidden=200, layer out=4, the weights were selected layer 1,2: Random weight [-2.0,2.0], and layer 3, Random weight [-1.0,1.0]. The learning rate α=0.7 and maximum error allowed was 0.12. (Patterns used are in figure 2). Other experiments were done using, topology 185-300-4, learn rate= 0.85, 185-250-4 but they showed less efficiency.

Since this ANN depends on weights randomly selected, five experiments where done, and the average of the experiments is considered for comparison purposes. The figure 3a, shows the performance of the ANN in learning. Since there were only 3 objects to recognize, the classification starts with a 64.16% error (the error means the % of patterns that are not recognized from the universe of 216). The best case in 675 epochs reached 0% error in 57.593 s (training), in the worse case 1332 epochs in 116.375 s.

3.2 Perceptron

It is a feedforward network with one or more outputs that learn the position of a separating hyperplane in pattern space (a layer for nonlinearly separable pattern pairs). The first layer has fixed weights and the second ones change according to the error in the output. If a neuron shows no error, its weights are not modified. This architecture was considered, because it can reach always a no-linear patterns classification properly with the enough number of neurons in layer 1. In [15] it was demonstrated that

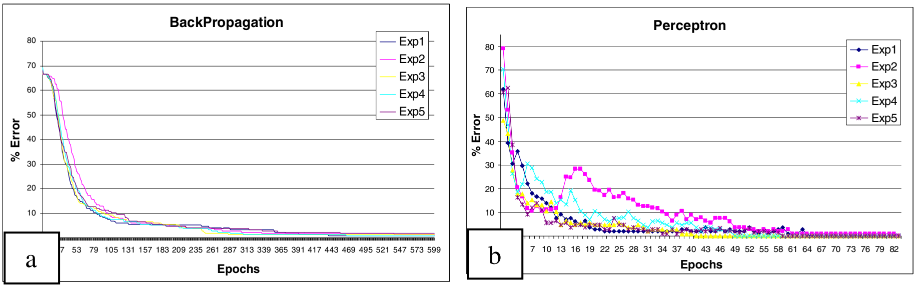

Fig. 3. a) Backpropagation performance (epochs are showed only until 600), b) Perceptron

one hidden layer is sufficient for performing any arbitrary transformation, using enough nodes. This model was considered for comparison also because of its high speed in training/testing compared with BP. Five experiments were done. The configuration used was: Inputs: 185, Outputs: 4, Layer1: 450 neurons (10 C/N=Connections per neuron), Random weights [-2.0, 2.0], Layer2: 4 neurons, Threshold: 0.0 (both layers, signum threshold device), α= 0.85. Other experiments were done using 450 (6, 8, 10, 12 C/N)-4, 350 (6, 8, 10, 12 C/N)-4, showing very similar results. The behaviour starts with an average error of 63.8%. (See figure 3b) This ANN showed a much better performance than the BP, in the best case reached 0% error in 41 epochs and a training time: 0.8112 s, the worse case it took 83 epochs, with 1.734 s in training.

3.3 Fuzzy ARTMAP

Fuzzy ARTMAP developed by Carpenter. G. A, [16] is one of the architectures based on ART (Adaptive Resonance Theory) in which supervised learning is carried out. This ANN creates several neurons according to the number of patterns and the differences among them. It has several advantages compared with traditional neural networks such as Backpropagation, since it does not suffer catastrophic lose of knowledge. The configuration for this architecture was, 2 epochs, rho_map=0.8; β= 1.0; ρ_a = 0.7; ρ_b = 1.0; aF1Size= 185; bF1Size=4. For all experiments α was set to 0.1. Five experiments were carried out, but since this ANN does not depend on random values, the same results were obtained in epochs/%error.

This ANN starts from an error of 100%, but reached fast 0% error in one epoch. The best training time was 0.172 s and the worst 0.188 s. See figure 4a. For all cases, the generated Knowledge Base (internal representation of the network) showed the same behaviour: For the circle one neuron was generated, and for the square and radiused-square, 2 neurons. This is because it was configured to a maximum data compression. Figure 4b, is a 100 epochs experiment with the same patterns, varying parameters β, ρ_a, α and rho_map. This experiment was done for testing the stability of the network and for choosing the best parameters for the proposed methodology. The graphic shows that the combinations of the parameters in the middle part of the graph do not reach stability and the network continues creating new neurons. Generally, β and ρ_a determine the stability. The property of FAM encoding critical features is a key

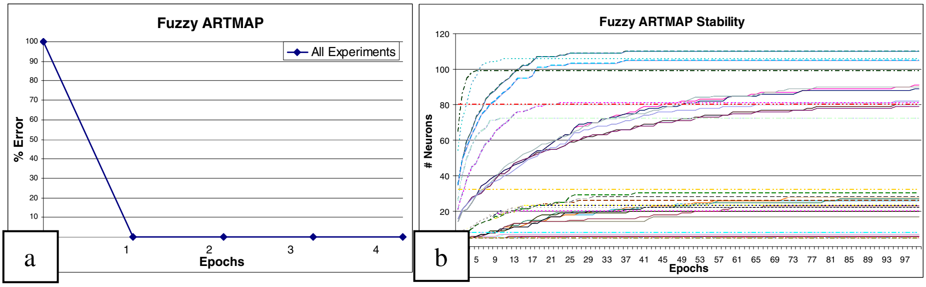

Fig. 4. a) FuzzyARTMAP, b) FAM Stability-sensibility parameters

to code stability. This learning strategy makes the difference of ART networks and MLP's (Multi Layer Perceptrons), which typically encode the current input, rather than a matched pattern, and hence, employ slow learning across many input trials to avoid catastrophic forgetting. [17].

3.4 Evaluation

Table 1 shows the results of the experiments. Time average is showed in figure 5.

Table 1. Results of the ANN experiments. Time is given in seconds and it is training/testing all patterns.

ANN	Exp1 Train	Exp1 Test	Exp2 Train	Exp2 Test	Exp3 Train	Exp3 Test	Exp4 Train	Exp4 Test	Exp5 Train	Exp5 Test	Average Train	Average Test
BACKPROPAGATION	79.625	0.047	116.375	0.047	68.734	0.047	57.593	0.047	74.657	0.047	79.3968	0.047
PERCEPTRON	1.391	0.031	1.734	0.031	0.8112	0.016	1.109	0.047	1.203	0.047	1.24964	0.0344
FUZZY ARTMAP	0.188	0.015	0.187	0.015	0.187	0.016	0.172	0.016	0.172	0.016	0.1812	0.0156

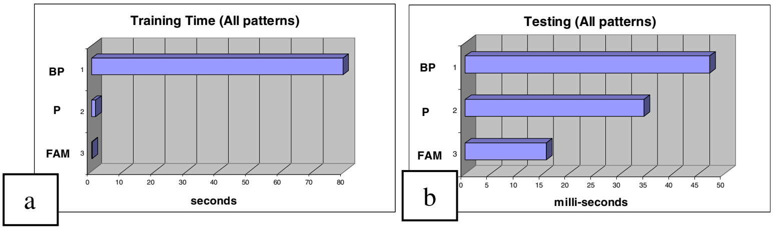

Fig. 5. a) ANN's Training time - b) ANN's Testing time. **BP** showed a Train-Test time of 367.577ms-0.217 ms per pattern, the **P,** 5.78ms-0.159 ms per pattern, and **FAM** 0.838 ms -- 0.0722 ms per pattern.

FAM was chosen because of its incremental knowledge capabilities and stability, but mostly because of the fast recognition and geometrical classification responses.

4 Object Recognition Methodology

4.1 First Approach

The Boundary Object Function (BOF) is the function that describes a specific shape (distances perimeter-centroid). In [12] a vector called CFD&POSE was proposed, eq (1), where***:*** ***D_i*** distances centroid-perimeter, ***X_C, Y_C,*** centroid, ***ϕ***, orientation, ***Z*** is the object's height (for this experiments, 3D objects were used), ***ID*** is a code number related to the geometry of the components.

$$[CFD\,\&\,POSE] = [D_1, D_2, D_3, D_n, X_c, Y_c, \phi, Z, ID]^T \tag{1}$$

This method was tested with 3 shapes for on-line recognition, the results showed training/testing time of few ms and a 100% classification. However, the methodology needed several patterns for training, then adding a new object would not be fast.

The hypothesis of the new SIRIO is: *With only two patterns from a regular or irregular object, it is possible to learn and recognise the object on-line achieving invariance to: rotation, location and scaling of a 2D image representation of an object.* The basic idea is to imitate the human learning/recognition ability, since humans need only to observe the object once or twice irrespective of its size/orientation/position in order to remember it later.

4.2 Patterns Generation –Proposed Approach

In order to subtract only parameters from the object, we take the BOF and grey-scale characteristics: the average grey colour and the object-histogram. We suppose a robotic fixtureless scenario in which there is no specific guide or line to find the orientation of a piece. We can approach a line through the piece and determine the orientation angle, but we can obtain only a value 0-180° (in both cases 0° to 180° or 180° to 360°). This behaviour will generate two different patterns for the same object that can not be rotated and rearranged automatically for the lack of the real angle. In the conveyor belt, it is also possible for the robot to find an object that is the mirror image of the trained one (the other object's side). In order to classify those mirror images also in the ANN, it is necessary to reproduce those in the inverse order during the training stage in such a way to have a new pattern to classify (2 on-line images and 2 mirror images) see Fig 6a. With these initial patterns, it is possible now to rearrange all other not trained patterns with a rotation function based on the found orientation and to generate very similar patterns. For managing the size invariance, normalization is applied. The grey-scale intensity average is an important characteristic of an object, and it is included in the object-properties vector as well as its histogram. The histogram no matter the size, orientation or light conditions (within a working range), has implicit information about the object and behaves similarly. It is proposed the vector of eq. (2), where **I** is the grey-scale intensity average value redundantly copied N times in the vector, and **H** is the object-histogram considering only K points of its behaviour and **P** is another possible invariant property. The figure 6b shows the generation of a vector with the properties of a wrench. The descriptor vector uses 180 BOF data, 10 Intensity data, and 30 data coming from function histogram, all data is normalized to [0,1].

$$[BOF + I_{1\ldots n} + H_{1\ldots K} + P] \tag{2}$$

For the first experiments, the whole pattern sets were made of [BOF + I], and the histogram was not used in order to test part of the vector only. Later other experiments where done, using the H, for the group of similar figures.

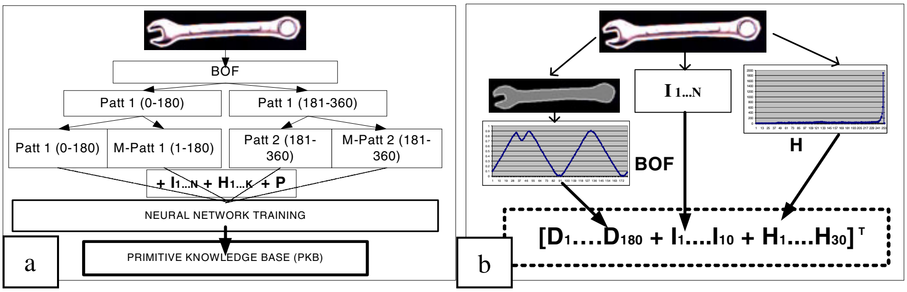

Fig. 6. a) PKB Base generation using two simple patterns. b) Descriptor vector generation

Figure 7 shows the universe of images used for the experiments and some of the patterns generated.

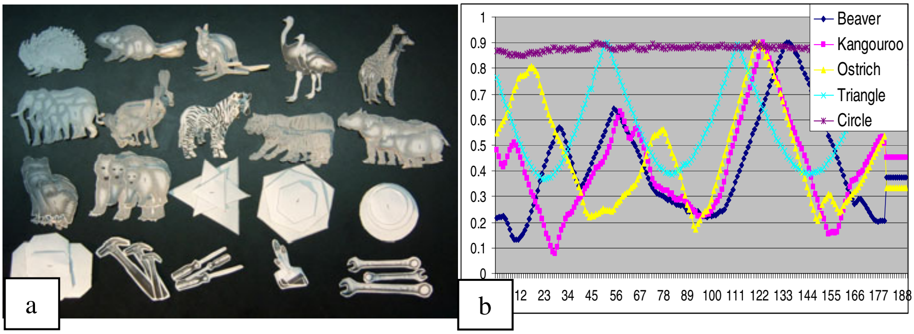

Fig. 7. a) 20 different figures: small, medium and big (total = 60), b) Patterns for 5 figures

One of the advantages of the algorithm is that while finding the perimeter and marking the object, the centroid, mass, histogram, etc, are calculated at the same time. Once this is completed, then the descriptor vectors are generated. All data is normalized to the range [0, 1], and the number of points fixed to vector size.

5 Experiments and Results

Algorithms were coded in Visual C++ .NET using a Pentium D PC @ 2.8 GHz with 512 MB RAM. For taking the images in real-time, a high speed, 100 fps, IEEE 1394 Basler f602c 640x480 resolution camera, was used. From the universe of 20 different

Table 2. For geometric pieces 100% recognition was achieved. An overall average recognition of 93.8% was achieved. Only for 6 out of 20 objects the percentage was not 100% (similar objects). If we consider only the No Mirror part (side 1) for testing, the achievement is 96.3%. Training time was 1.95 ms per pattern and 0.2 ms for testing. FAM parameters were: 2 epochs, rho_map=0.92; β= 1.0; ρa = 0.9; ρb = 1.0. Err=Error, T=total, NM= No Mirror, N=Neurons.

#	Object	Code ANN	Err Side 1	Err Side 2	T. Err	% OK	%OK NM	Generated N
1	Porcupine	1 0 0 0 0 0	0	0	0	100.0	100.0	{0, 1, 2, 3}
2	Beaver	1 0 0 0 0 1	0	0	0	100.0	100.0	{4, 5, 6, 7}
3	Kangouroo	1 0 0 0 1 0	0	0	0	100.0	100.0	{8, 9, 10, 11}
4	Ostrich	1 0 0 0 1 1	0	0	0	100.0	100.0	{12, 13, 14, 15}
5	Giraffe	1 0 0 1 0 0	1	3	4	83.3	91.7	{16, 17, 18, 19}
6	Elephant	1 0 0 1 0 1	2	4	6	75.0	83.3	{20, 21, 22, 23}
7	Hare	1 0 0 1 1 0	0	1	1	95.8	100.0	{24, 25, 26, 27}
8	Zebra	1 0 0 1 1 1	1	3	4	83.3	91.7	{28, 29, 30, 31}
9	Tiger	1 0 1 0 0 0	1	2	3	87.5	91.7	{32, 33, 34, 35}
10	Rhinoceros	1 0 1 0 0 1	2	4	6	75.0	83.3	{36, 37, 38, 39}
11	Raccoon	1 0 1 0 1 0	0	0	0	100.0	100.0	{40, 41, 42, 43}
12	Bear	1 0 1 0 1 1	2	4	6	75.0	83.3	{44, 45, 46, 47}
13	Triangle	1 0 1 1 0 0	0	0	0	100.0	100.0	{48, 49, 50}
14	Hexagon	1 0 1 1 0 1	0	0	0	100.0	100.0	{51}
15	circle	1 0 1 1 1 0	0	0	0	100.0	100.0	{52}
16	Parabola	1 0 1 1 1 1	0	0	0	100.0	100.0	{53, 54}
17	Hammer	1 1 0 0 0 0	0	0	0	100.0	100.0	{55, 56, 57, 58}
18	Drill	1 1 0 0 0 1	0	0	0	100.0	100.0	{59, 60, 61}
19	Brush	1 1 0 0 1 0	0	0	0	100.0	100.0	{62, 63, 64, 65}
20	Wrench	1 1 0 0 1 1	0	0	0	100.0	100.0	{66, 67, 68}
				Total	**30**	**93.8**	**96.3**	

objects (three sizes for each one) as observed in figure 7a, 20 different middle size were selected for training purposes, placing each object in two different angles (values within 1° to 180° and 181° to 360° range). Two patterns were generated, as well as their mirror images (40 images were captured on-line and 40 mirror images, for the 20 different figures) for creating the PKB in the ANN. For the first experiment only [BOF + I] was used in order to evaluate de performance. After training, the experiments consisted of testing the 60 pieces, even though they where not trained, with different location, and orientation. For testing: Size 1, 2, 3, Angle 45, 135, 225, 315, Sides: 1, 2, Total=480 patterns were used. The results are given in Table 2.

Most of the errors in table 2 were detected in the animals with similar bodies, (Tiger, Bear, Zebra, etc), this is because the ANN confused in some conditions this animals among them. The solution of this problem could be adding more patterns, but then the hypothesis will not be fulfilled. The other option is to add the object histogram to complete the descriptor vector. For this group of animals, the complete vector was generated, new experiments were done and the results were successfully achieved. We can conclude that if we have very similar objects it is necessary to use the complete vector and for simple or medium complex figures with the basic form is enough. We can see in objects 14 and 15 that the ANN created only one neuron for recognizing all patterns of its group, because of the object simplicity. 68 neurons were created for recognizing the universe of 20 objects, 3 sizes, all angles.

For sending information to the robot once in the manufacturing cell, a secondary vector is considered, (based on the recognized object and a data base related to it) in eq. (3), where $\mathbf{C_x}$, $\mathbf{C_y}$ is the POSE of the object, $\mathbf{\Phi}$ is the orientation angle, **ID** is the identified object, and **OI** is Object Information related to the object grasping taken from the data base.

$$[C_x\, C_y + \Phi + \mathrm{ID} + \mathrm{OI}] \qquad (3)$$

6 Conclusions and Future Work

An invariant method using 2D image object representation suitable for part grasping during robot operations was presented. The method proposes to use of only two image patterns from a regular or irregular object instead of using multiple patterns for training. From the given results it was demonstrated that it is possible to learn and recognise the objects on-line achieving invariance to: rotation, location and scaling.

The method showed potential fast recognition and accuracy in the classification of simple and complex 2D objects. The superiority of FAM for this task was also experimentally demonstrated compared to other connectionist models. For future work it is intended to prove the algorithms in the robotic testbed using varying light conditions.

References

1. Hoska, D.: Fixturless assembly manufacturing. Manufacturing Eng. 100, 49–54 (1988)
2. Ngyuen, W., Mills, J.K.: Multirobot control for flexible fixturless assembly of flexible sheet metal autobody parts. In: IEEE Int. Conf. Robotics & Automation, pp. 2340–2345 (1996)
3. Langley, C.S., Eleuterio, D.: GMT, A memory efficient neural network for robotic pose estimation. In: Proc. of IEEE Int. Symp.on Computational Intelligence in Robotics and Automation, vol. 1, pp. 418–423. IEEE CIRA (2003)
4. Alberto, S., Aguado, E., Montiel, M.: Invariant characterization of the Hough Transform for pose estimation of arbitrary shapes. Pattern Recognition 35, 1083–1097 (2002)
5. Wu, C.-H., et al.: A new computation of shape moments via quadtree decomposition. Pattern Recognition 34, 1319–1330 (2001)
6. Best, P.J., McKay, N.D.: A Method for Registration of 3-D Shapes. IEEE Trans. on Pattern Analysis and Machine Intelligence 14(2) (1992)
7. Bribiesca, E.: A new Chain Code. Pattern Recognition 32, 235–251 (1999)
8. Gonzalez-Galvan, E.J., et al.: Application of Precision-Enhancing Measure in 3D Rigid-Body Positioning using Camera-Space Manipulation. The International Journal of Robotics Research 16(2), 240–257 (1997)
9. Dickmanns, E.: Vehicles capable of dynamic vision: a new breed of technical beings? Artifical Intelligence 103, 49–76 (1998)
10. Kollnig, H., Nagel, H.: 3d pose estimation by directly matching polyhedral models to gray value gradients. Int. Journal of Comp.Vision 23(3), 282–302 (1997)
11. Bone, G.M., Capson, D.: Vision-guided fixturless assembly of automotive components. Robotics and Computer Integrated Manufacturing 19, 79–87 (2003)
12. Peña-Cabrera, M., Lopez-Juarez, I., Rios-Cabrera, R., Corona-Castuera, J.: Machine vision approach for robotic assembly. Journal of Assembly Automation 25(3), 204–216 (2005)
13. Corona-Castuera, J., Lopez-Juarez, I.: Behaviour-based approach for skill acquisition during assembly operations, starting from scratch. Robotica 24(6), 657–671 (2006)
14. Chen, K.: Efficient parallel algorithms for computation of two-dimensional image moments. Pattern Recognition 23, 109–119 (1990)
15. Cybenko, G.: Approximation by superposition of a Sigmoidal Function. Mathematics of Control, Signals, and Systems 2, 303–314 (1989)
16. Carpenter Gail, A., et al.: Fuzzy ARTMAP: A Neural Network Architecture for Incremental Supervised Learning of Analog Multidimensional Maps. IEEE Transactions on Neural Networks 3(5) (1992)
17. Carpenter, G.A., Grossberg, S.: Adaptive Resonance Theory. In: The Handbook of Brain theory and neural networks, 2nd edn. MIT Press, Cambridge (2002)