22

Computer Vision and Fuzzy Rules Applied to a Dispensing Application in an Industrial Desktop Robot

Cristina P Santos, Manuel J Ferreira
Industrial Electronics Department, University of Minho
Portugal

1. Introduction

In an era when new product styles are being introduced with ever-shortening life cycles, the cost of high preparation times for automation equipment every time a new part or product must be produced is becoming prohibitive, both in terms of time and money. In modern flexible industrial production, the capabilities of the machinery to adapt to changing production demands are a key factor for success.

Desktop and Scara Robots are universal tools for various industrial applications like dispensing, soldering, screw tightening, pick'n place, welding or marking. This type of robots is suitable for various production line systems (e.g. cell-line, in-line), and can be adapted to meet diverse requirements. They are easy to use, can be applied economically and, nowadays, a complex programming in robot language is unnecessary, thus reducing installation time and providing added convenience. These robots are typically programmed off-line by using waypoints and path information. However, the coordinates and types of waypoints have to be entered manually or taught. Typically, small workpieces with a high complexity of linear paths raise programming efforts.

Once a robot has been programmed off-line for a workpiece, the system should be able of identifying it and autonomously initiate the correct working procedure. Further, a semi-automated system has to be capable to autonomously deal with misalignments and compensate small deviations during loading, which may result in a bad execution of the robot off-line stored programs.

The concept of sensing for robotics is essential for the design of systems where the real time operation, flexibility and robustness are major design goals. The ability of a system to sense its surroundings and perform the task according to the existing conditions is an effective way to reduce costs and raise flexibility. Highest precision and minimum amount of programming time is the result. Advanced sensor systems are now emerging in different activities which will significantly increase the capabilities of industrial robots and will enlarge their application potential.

In this work, sensor data processing concepts on the basis of fuzzy logic are applied, to enable a robot to deal autonomously with typical uncertainties of an industrial working environment. Specifically, it is proposed a flexible, adaptive and low-cost solution to address some problems which often limit the production rate of small industries. Thus, by

Source: Industrial Robotics: Programming, Simulation and Applicationl, ISBN 3-86611-286-6, pp. 702, ARS/plV, Germany, December 2006, Edited by: Low Kin Huat

additional capabilities the robot can autonomously adapt to changing production needs, compensate misalignments and avoid injuries in the work tool and piece.
As a case study, consider a desktop robot executing dispensing applications on workpieces/fixtures. For each workpiece, the robot is programmed off-line. In order to improve the performance and flexibility of these industrial systems, we equipped the robot with a CCD Camera. The process is divided into two phases: a learning and an execution phase. On the learning phase, the worker programs the robot off-line such that it executes the dispensing operation over the workpiece. At this stage, an image of the workpiece is acquired, a set of descriptors describing it are computed, a fuzzy rule describing the workpiece is generated and included in a database together with the robot's program. On the execution phase, the worker loads a set of workpieces onto the robot's working table. An image is acquired, the corresponding descriptors are computed and, through a parsing and classification procedure, the workpieces are identified.
Alignments and offset values are calculated fully automatically which allows the robot to ensure accurate placement of tools. Workers stay busy loading and unloading workpieces/fixtures while a desktop robot, equipped with a vision system, is performing precision dispensing tasks. The system is also capable of autonomously starting a learning phase in case an unknown workpiece is shown to the system, and robust to deal with common errors such as a missing fixture. This significantly reduces development time for these tedious processes. Further, reduces costs by compensating misalignments in the workpiece location during loading avoiding injuries both in the workpiece and tool. Another result of this approach is that computation grounded on information derived from sensation enables the achievement of autonomy. Autonomy implies that through sensation and action it might be possible for a system to start some conceptualization process of high level.
This chapter is organized as follows. Section II describes the vision system and the software architecture for the learning and execution phases. The representation and description schemes are also described in this section as well as calibration and general image processing procedures. Section III describes the experimental results along with the hardware requirements of the system. A complete cycle of the execution phase is also depicted in this section. Finally, Section IV outlines the main conclusions and some future work is discussed.

2. The dispensing application

In a typical dispensing application the procedure is to program off-line the robot such that it executes the work over the workpiece. For each type of workpiece, a robot program is stored. This is the learning phase. During the production stage, the worker sets to run the program for the particular type of workpiece, loads the workpiece, issues a command to the robot which starts to run the dispensing application and, finally, unloads the workpiece. These procedures (execution phase) implement a full working cycle. However, two main problems may arise. Firstly, misalignments during loading may result in injuries both in the workpiece and tool. Secondly, in case other known types of workpieces are introduced in the production line, requires the worker to identify the corresponding stored robot program and load it onto the robot. This introduces delays and sometimes serious injuries due to worker failure. Further, it requires a worker able to directly interact with the robot. Such typical problems limit the production rate of small manufacturing industries.

Herein, we propose a cheap solution which improves the overall flexibility of a typical dispensing application and minimizes the two problems discussed above. Similarly to the procedure described, for each type of workpiece the robot is firstly programmed off-line and the program is stored. The main difference is that during the production stage, it is the system that autonomously identifies the loaded workpieces using visual information and a fuzzy inference classifier.

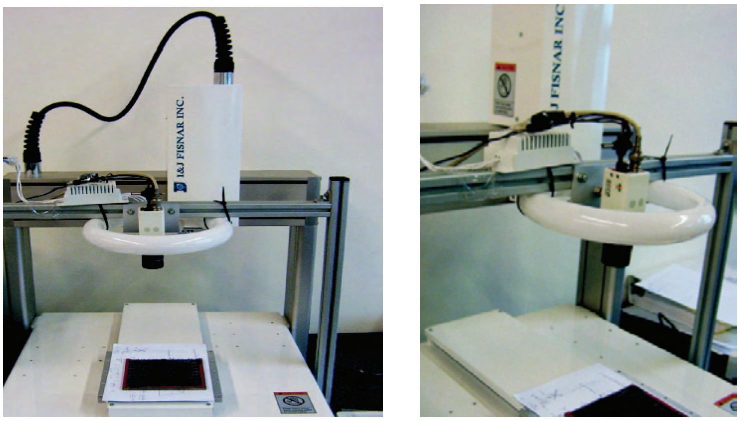
In this study, a JR2000 Series Desktop robot from I&J Fisnar Inc (Janome, 2004) with simultaneous control of 3 axis is used as the test bed. The robot performs 3D linear and arc interpolation to include compound arcs in a working area of 150x150mm. The overall experimental setup is shown in Fig. 1. A CCD camera has been mounted over the robot, and despite the applied algorithm to improve light uniformity, a fluorescent light was placed around the CCD Camera to assure that the scene is well illuminated. A front lighting technique was applied.

The CCD camera is a TRC TC5173 colour camera with a resolution of 768x576 pixels. Image digitalization is done on a general purpose image acquisition board, National Instruments PCI1411, mounted inside a 100MHz Pentium III PC. The PC is connected to the robot by a serial RS-232C protocol.

3. System Architecture

Fig. 2 presents the architecture of the processing system, in which two paths were specified: one for the learning phase (P1) and another for the execution phase (P2).

The first two modules are identical for both P1 and P2 and deal with object analysis. The Pre-processing module enhances the image quality and extracts the blob objects of the image. This module is necessary because the acquired images are not perfect for analysis. The Feature Extraction module extracts the feature vector that best characterizes each object. P1 has a Fuzzy Grammar module which generates the fuzzy rules that describe the objects. These rules are stored in a database.



a) b)

Fig. 1. Experimental setup showing the desktop robot with the mounted CCD Camera, the fluorescent lamp and a mould. a) General front view. b) Detailed view of the front lighting technique.

In the execution phase P2, the feature vectors extracted for each object are submitted to a Parsing Procedure module developed with the compilers yacc and lex (Appel, 1998; Bumble-Bee, 1999). These vectors are submitted to each rule stored in the database and a value is obtained for each of them. Finally, the Classification module verifies which rule has a higher value thus identifying the workpiece under analysis. A threshold is specified such that an alarm sounds when an unknown or misclassified workpiece is detected. Further, a learning phase is automatically initiated and an appropriate fuzzy rule is generated for that workpiece.

4. Preprocessing

To perform a robust industrial application, and prior to apply segmentation procedure to extract the diferent objects in the image, the following aspects must be minimized: 1) random noise in the acquisition process; 2) lack of light uniformity, which depends of the illumination technique; and 3) image distortions due to the type of lenses.
Noise was reduced by calculating the final image to process as an average of five consecutive acquired images.

4.1 Light calibration procedure

A light calibration procedure was developed (Russ, 1995) and employed to cope with the lack of light uniformity. A black and a white object, covering the all working area, are acquired. Each of these images is divided in non-overlapping windows of 7x7pixels and the mean of the gray-levels within each of the windows is calculated (N_{cb} and N_{cw} for the black and white windows respectively). The final histogram is calculated by the histogram stretching of each window as depicted in Fig. 3.

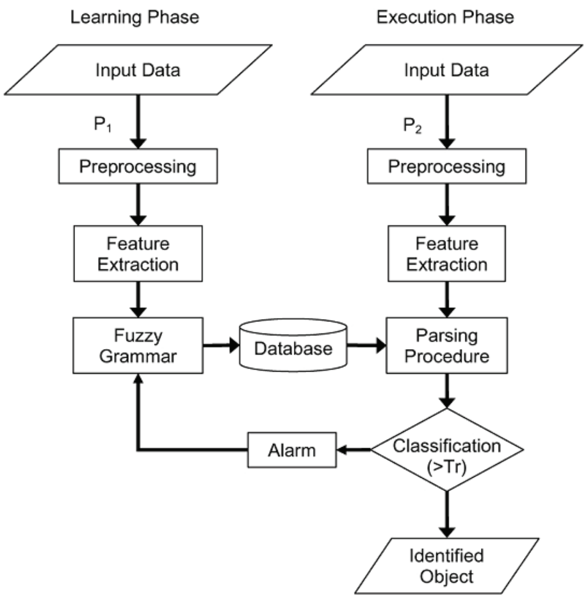


Fig. 2. Architecture of the processing system.

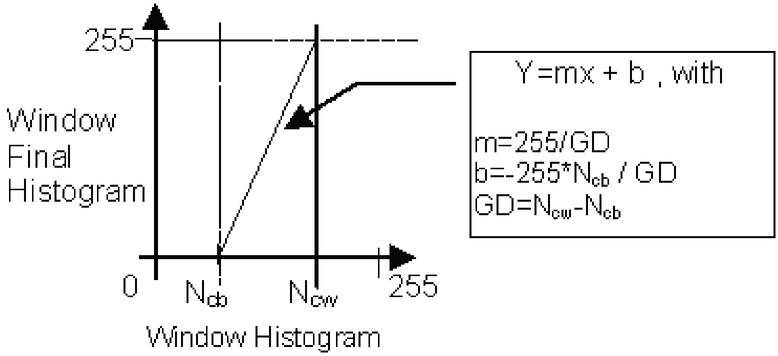


Fig. 3. Light calibration procedure.

4.2 Image distortion procedure

Image distorting is solved by applying an image correction using a well-known grid calibration procedure (Matrox, 1998). An image of a grid with size, δ_i, is acquired. Image correction is done according to the mapping between distorted, P_{di} (i=0,1,2,3), and non-distorted, P_{ndi} (i=0,1,2,3), elements of the grid (see Fig. 4). δ_i is chosen such that the sides of the distorted elements of the grid are straight lines and depends on the magnitude of distortion.

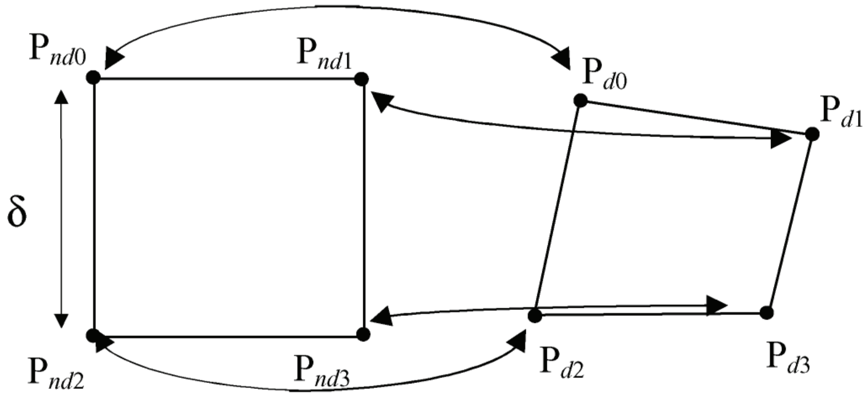


Fig. 4. Mapping between a distorted (P_{di} (i = 0, 1, 2, 3)) and a nondistorted element. (P_{ndi} (i = 0, 1, 2, 3)) of the grid. These elements have (x, y) coordinates.

The homogeneous coordinate transformation between $P_{di}=(x_{di}, y_{di})$ and $P_{ndi}=(x_{ndi}, y_{ndi})$, for i=0,1,2,3, is given by

$$w_i \begin{pmatrix} x_{ndi} \\ y_{ndi} \\ 1 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} x_{di} \\ y_{di} \\ 1 \end{pmatrix} \tag{1}$$

where

$$x_{ndi} = \frac{a_{11}x_{di} + a_{12}y_{di} + a_{13}}{a_{31}x_{di} + a_{32}y_{di} + a_{33}} \tag{2}$$

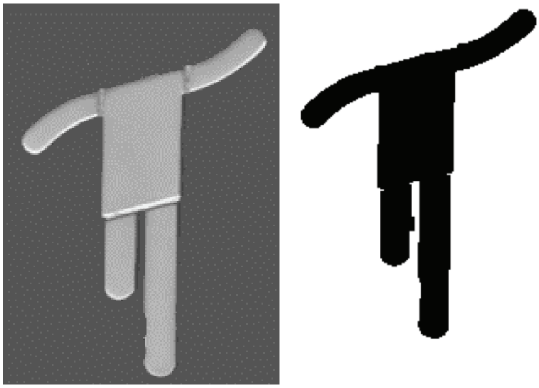
$$y_{ndi} = \frac{a_{21}x_{di} + a_{22}y_{di} + a_{23}}{a_{31}x_{di} + a_{32}y_{di} + a_{33}} \tag{3}$$

4.3 Segmentation procedure

The extraction of the blobs that represent the objects is accomplished through a binarization with a fixed threshold and through a blob-coloring like algorithm (Ballard & Brown, 1982). The segmentation of the image into regions could be achieved applying line finding or region growing techniques (Ballard & Brown, 1982). However, line finding techniques if followed by a floodfill procedure may produce incorrect results in non-simple connected regions. Region growing techniques commonly use only properties of local groups of pixels (local techniques). Another possibility would be split and merge techniques, however these are more complex and time consuming.

In case the image is highly contrasted and consists of dark (or white) objects in a white (or black) background, as in our case, simple local techniques can be used. In such conditions, a blob-coloring like algorithm is time effective. In the final result regions are geographically separated, meaning that each blob can be addressed in an efficient manner.

Consider the sample object illustrated in Fig. 5(a). Fig. 5(b) illustrates the corresponding extracted blob.



a) b)

Fig. 5. A sample object and the extracted blob. a) Object sample. b) Extracted blob.

5. Feature extraction

After image segmentation, it is necessary to choose a representation and a description scheme for the resulting aggregate of segmented pixels in a form suitable for further computer processing.

As pointed out by Williams (1999), although description of a 2D shape is one of the foremost issues in pattern recognition, a satisfactory general theory does not exist. Ideally, the selected features must be invariant to the shape scaling and rotation and should support clustering or grouping; resulting in the generation of a robust representation. Low order geometric moments are arguably the most common features. Diameter features, like Feret diameters and distance-versus-angle signatures, tend to lead to topological and symmetry considerations and are more robust to noise and small changes of shape (Ballard & Brown, 1982; Williams, 1999; Micheli-Tzanakou, 2000; Costa & Cesar, 2001; Gonzalez & Woods, 2002; Kindratenko, 2004).

The perimeter, the first and second central moments and the Feret diameter representations were tested in order to verify those that allow maximum flexibility, meaning to allow the coexistence of objects with different shapes in the same database. The best results were obtained using the Feret diameters (longest distance between any two points along the object boundary) at different rotation angles, θ, of the object, and thus were chosen to build the feature vectors of the representation scheme in our work.

Fig. 6 presents some Feret diameters for object depicted in Fig. 5(a).

By trial and error, we have chosen an increment between rotation angles of 10 degrees.

This type of external representation scheme is very useful in the computation of descriptors and is very adequate because the primary focus is on morphological features. However, this feature vector is highly dependent on the object's orientation, which poses a difficulty in the identification process. To solve this, we first orient the object by setting the axis of higher inertia always in a predefined position. Further, the fuzzy inference system implies that the magnitude of each element of the feature vector must be in the interval [0,1] and thus a normalization of the obtained feature vector is required. Therefore, normalization is achieved simply by normalizing the obtained curve to unit maximum value as given by

$$NFD(\theta) = \frac{FD(\theta)}{FD_{\max} - FD_{\min}} - \frac{FD_{\min}}{FD_{\max} - FD_{\min}} \quad (4)$$

where $FD(\theta)$ is the Feret diameter at angle θ and FD_{max}, FD_{min} are the maximum and minimum value of the Feret diameters for the feature vector, respectively. The normalized Feret diameters for the objects depicted in Fig. 7 are illustrated in Fig. 8.

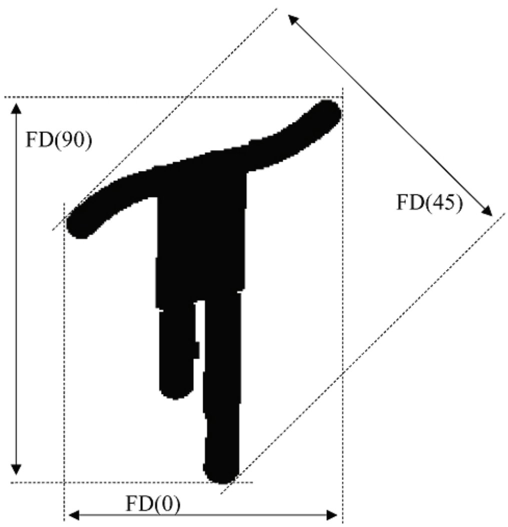
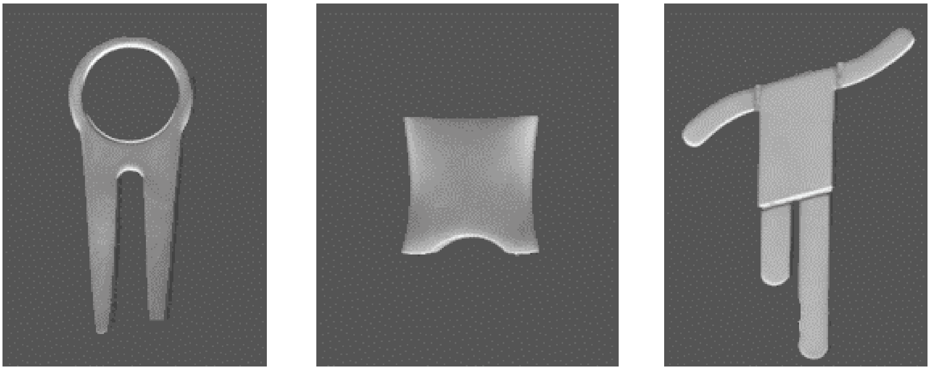


Fig. 6. The Feret diameters for object depicetd in Fig. 5 at angles 0, 45 and 90º



a) b) c)

Fig. 7. Some objects used in the choice of the external representation type. a) Object 1. b) Object 2. c) Object 3

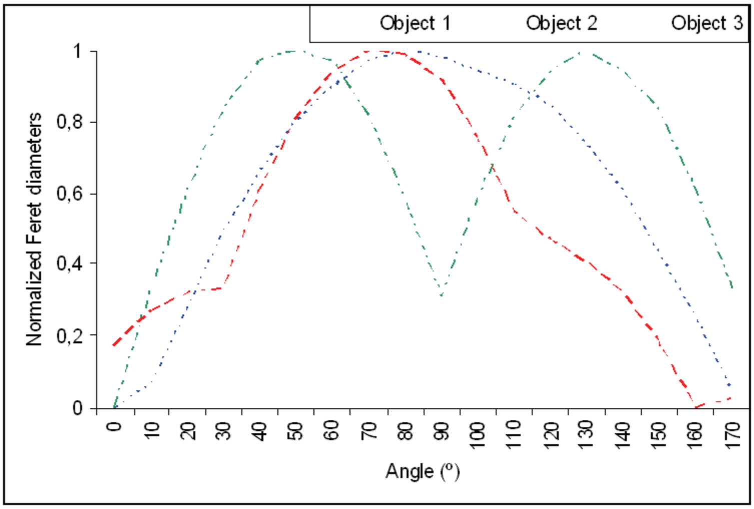


Fig. 8. Normalized Feret diameters for objects depicted in Fig 7. Slod, dash and dash-dot traces represent objects 1, 2 and 3, respectively.

Equation 4 is also independent of the size factor. For this particular application, this is a drawback since objects with different sizes require different robot programs. In order to identify objects with the same shape but with different sizes, we established a size dependent feature. We have introduced the feature S, which classifies the object's shape relatively to its size and is given by the FD_{max}, normalized to the maximum size allowed for an object.

6. Fuzzy Grammar

After the extraction of the feature vector that characterizes an object, it is necessary to classify the object according to its attributes. Specifically, our application deals with the following constraints: a) to deal with a high diversity of objects; b) to recognize simultaneously several different type of objects and c) to autonomously detect a new type of objects during the execution phase and thus initiate a learning phase, using the intervention of the human operator only to program the robot. To accomplish these goals the learning phase of the recognition process must be done with a unique sample of each type of object.

Regarding the classifiers and recognizers, there are different approaches based on statistical and artificial intelligence methods (Bezdek & Pal, 1992; Kerre & Nachtegael, 2000; Micheli-Tzanakou, 2000; Costa & Cesar, 2001; Looney, 2002). The most common solutions commercially available use recognizers based on the calculus of metrics like Euclidean, Minkowsky e Mahalanobis distance measures (Williams, 1999). However, these recognizers, as well as the ones based on neural, fuzzy logic and neurofuzzy networks, demand a great amount of samples from the population to perform learning. Despite the fact that these modern technologies are now firmly established as leading advanced control techniques in use in industry, they do not fulfil the constraints of the dispensing application.

In this work, a fuzzy system modelling approach was developed in which a fuzzy inference system identifies the fuzzy rules representing relationships among 2D shape features. There are several approaches that generate these fuzzy rules. The most often applied are based on statistics, neural networks and genetic algorithms (Ivancic & Malaviya, 1998; Peters et al., 1998; Looney, 2002).

However, none of these methods satisfy the needs of this specific application. Therefore, we decided to apply a fuzzy grammar approach. Fuzzy grammar is a pattern classification syntactic model used to represent the structural relations of patterns (Fu & Booth, 1986ab; Bezdek & Pal, 1992; Malaviya, 1996; Stanchev & Green, 2000) and describe the syntax of the fuzzy languages that generate the fuzzy rules. This inference system fulfils the project demands due to its linguistic description approach, which keeps the number of rules small, and due to its capability to generate a fuzzy rule using only one sample of a pattern.

Herein, we briefly review some basic concepts of fuzzy grammar (for a full discussion see (Lee & Zadeh, 1969; Pal & Majumber, 1986; Bezdek & Pal, 1992; Yager & Zadeh, 1992)). Fuzzy grammar GF is a quintuple $GF=(V_N,V_T,P,S_0,\mu)$, in which V_N and V_T are finite disjoint sets of non-terminal and terminal vocabulary respectively, such that $V=V_N\cup V_T$ is the total vocabulary of the grammar. P is a finite set of production rules of the type $\alpha\rightarrow\beta$, with $\alpha\in V_N$ and β is a member of the set V^* of all strings (including the null string ε). $S0\in V_N$ is the starting symbol. $\mu$is the mapping of $P\rightarrow[0,1]$, such that $\mu(p)$ denotes the possibility of the current language sentence $p\in P$.

The syntax of the developed language L(GF) is depicted in Fig. 9 and includes 4 different steps:

1) The codification of the features to primitives. In this work, the features are the Feret diameters (NFD(θ)) and the size S, which are coded to the primitives FDθ and SN, respectively. When more than one sample of an object is presented to the system the mean value of each feature is used.
2) The definition of linguistic terms HistVar:#. This setting is done according to Table 1. The membership function $\prod$ is illustrated in Fig. 10 for $\prod(x,b,c)$. The parameter c is chosen such that the eleven membership functions cover the all universe of discourse X and have disjoint maximums.
3) The definition of fuzzy modifiers (FM): “More than”, “Less than” and “Between”. The FM “More than” LT is defined by

$$\mu_{MT}\langle LT\rangle = \begin{cases} 1 & x \geq L \\ S(x, L-lb, L-lb/2, L & x < L \end{cases} \tag{5}$$

where L is a threshold value and lb is the bandwidth value of the S membership function (Bezdek & Pal, 1992; Malaviya, 1996). The FM “Less than” LT is given by

$$\mu_{LT}\langle LT\rangle = \begin{cases} 1 & x \leq L \\ 1-S(x, L, L+lb/2, L+lb & x > L \end{cases} \tag{6}$$

The FM “Between” LT1 e LT2, is given by

$$\mu_B < TL_1 >< TL_2 > \;\; = \begin{cases} 1-S(x, w_1, w_1+lb/2, w_1+lb) & x > w_1 \\ 1 & w_2 \leq x \leq w_1 \\ S(x, w_2-lb, w_2-lb/2, w_2) & x < w_2 \end{cases} \tag{7}$$

where w1 and w2 are threshold values (Bezdek & Pal, 1992; Malaviya, 1996).

Language -> $L(G_F) = \{x, \mu(x) | x \in V^*_T, S \Rightarrow x\}$

$G_F = (V_N, V_T, P, S_0, \{\mu\})$

V_N={S_0, Name, ElementSet ,Primitive, TermSet, Element, Term}

V_T={SN, FD0, .. FD170, HistVar: 1,..HistVar: 11 (Table I),+,..,#}

$S_0 \rightarrow$ 'Rule' RuleName' 'ElementSet

ElementSet	$\rightarrow$	ElementSet '&' ElementSet
		'('ElementSet {'|' ElementSet }')'
		'('ElementSet { '+' ElementSet } ')'
		Element
		λ
Element	$\rightarrow$	TermSet '#' Primitive
		Primitive
TermSet	$\rightarrow$	'>' Term
		'<' Term
		'('Term '| |' Term')'
RuleName	$\rightarrow$	Obj1
		other
Primitive	$\rightarrow$	SN, FD0, ..., FD170
		other
Term	$\rightarrow$	'HistVar: 1' ... 'HistVar: 11'

Fig. 9. Syntax of the developed fuzzy language L(GF).

Designation	Function
HistVar:1	$\Pi(x,0.2,0.0)$
HistVar:2	$\Pi(x,0.2,0.1)$
HistVar:3	$\Pi(x,0.2,0.2)$
HistVar:4	$\Pi(x,0.2,0.3)$
HistVar:5	$\Pi(x,0.2,0.4)$
HistVar:6	$\Pi(x,0.2,0.5)$
HistVar:7	$\Pi(x,0.2,0.6)$
HistVar:8	$\Pi(x,0.2,0.7)$
HistVar:9	$\Pi(x,0.2,0.8)$
HistVar:10	$\Pi(x,0.2,0.9)$
HistVar:11	$\Pi(x,0.2,1.0)$

Table 1. Linguistic terms used on the fuzzy grammar.

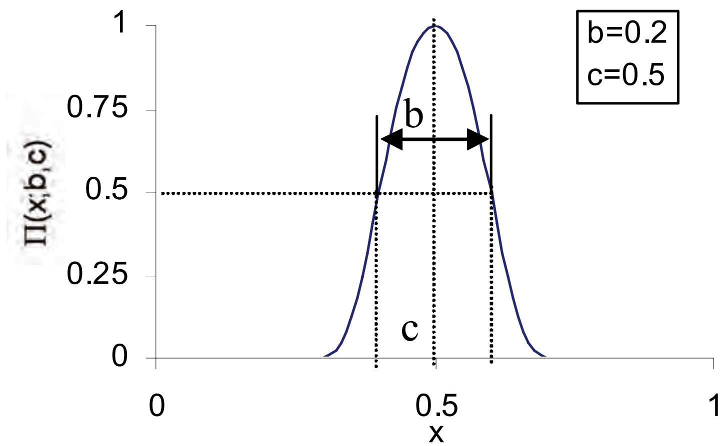


Fig. 10. Membership function PI.

4) The definition of fuzzy operators (FO) which define the relations between the linguistic terms and primitives. We defined the following FO:
   a) &, representing the AND of two primitives. It is given by the Yager intersection (Pal & Majumber, 1986).
   b) >, representing "More than" LT and is given by µMT<LT>.
   c) <, means "Less than" LT and is given by the function µLT<LT>.
   d) | |, describes "Between two" LT and is given by µB<LT1><LT2>.
   e) #, means a "Separator between a" primitive and a LT.
   f) ( ), imposes a hierarchy in the rule.

Consider as an example object 2 depicted in Fig. 7. Fig. 11 illustrates the primitive FD20, obtained from the Feret diameter feature, NFD(θ)=0.6, when θ=20 degrees. This primitive has non-zero degrees of membership for LT HistVar:6, LT HistVar:7 and LT HistVar:8 (Fig. 11). The highest fuzzy value is obtained using LT HistVar:7. Thus, HistVar:7#FD20 is part of the fuzzy rule which characterizes object 2. Finally, the rule created by the fuzzy grammar is:

HistVar:1#FD0&HistVar:4#FD10&HistVar:7#FD20&#HistVar:9#FD30&HistVar:11#FD40&HistVar:1#FD50&HistVar:11#FD60&HistVar:9#FD70&HistVar:7#FD80&HistVar:4#FD90&#HistVar:7#FD100&HistVar:9#FD110&>HistVar:10#FD120&HistVar:11#FD130&>HistVar:10#FD140&HistVar:9#FD150&HistVar:7#FD160&HistVar:4#FD170&HistVar:2#SN.

If more than one linguistic term gives a fuzzy value superior to 0.75; we apply fuzzy modifiers like “More than”, “Less than” and “Between”, to combine the obtained results. Fig. 12 illustrates the procedure for fuzzy modifier “More than” HistVar:10 for the primitive FD140. The final fuzzy value results from the combination of LT HistVar:10 and LT HistVar:11. Similar procedures apply for fuzzy modifiers “Less than” (Fig. 13) and “Between” (Fig. 14).

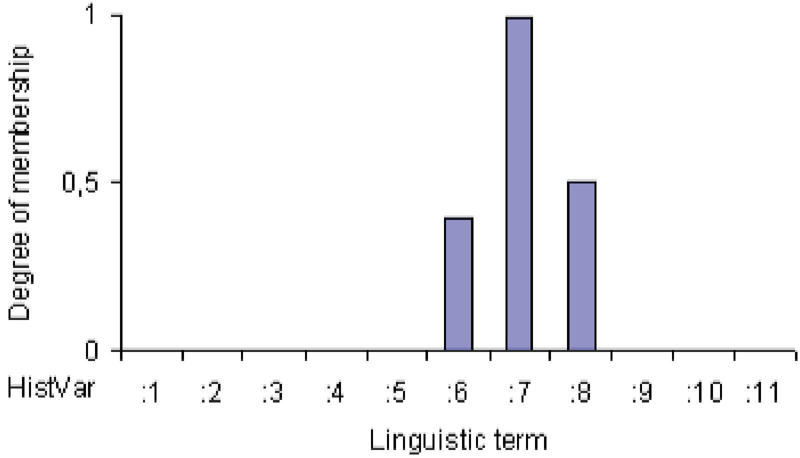


Fig. 11. The highest fuzzy value for LV FD20 is obtained using LT HistVar:7.

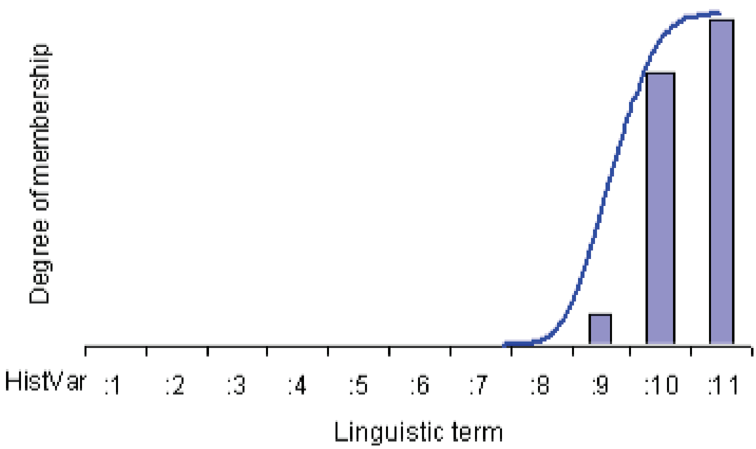


Fig. 12. Linguistic term for the primitive FD140 – Fuzzy Modifier „More than“ HistVar:10.

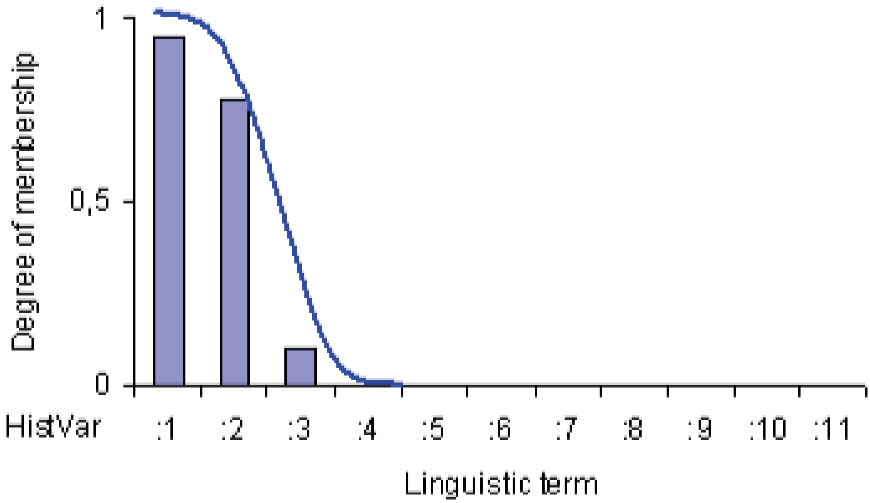


Fig. 13. Fuzzy Modifier „Less than“ HistVar:2.

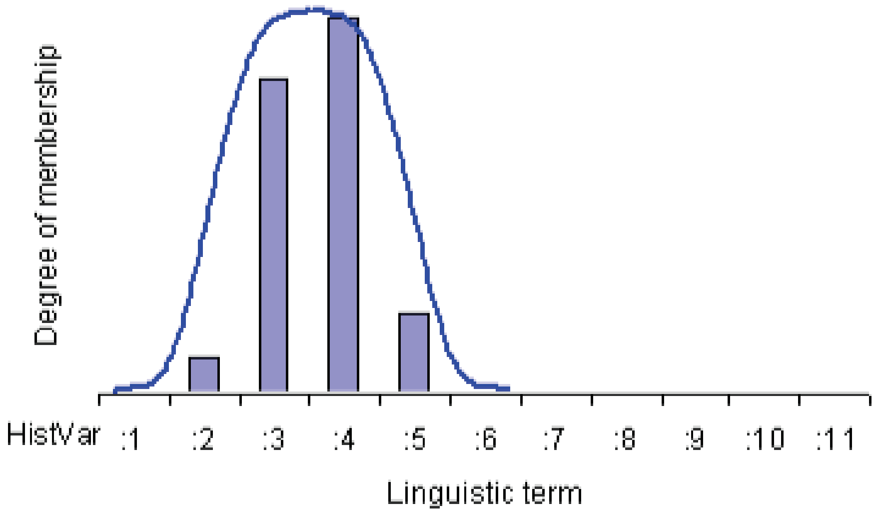


Fig. 14. Fuzzy Modifier „Between“ HistVar:3 and HistVar:4.

7. Parsing procedure

The parsing procedure was developed for the fuzzy grammar. The inputs are the feature vectors extracted for an object from the Feature Extraction module and the rules stored in the database. The feature vectors are submitted to each rule of the database. The output of

the parsing procedure is a value in the interval [0,1] reflecting the grade of membership of the object for each class.
Consider as a simple example the feature vectors which are presented in Table 2. They describe the objects depicted in Fig. 7.

Angle (º)	Object 1	Object 2	Object 3
0	0,00	0,00	0,17
10	0,06	0,33	0,27
20	0,28	0,61	0,32
30	0,48	0,83	0,33
40	0,65	0,97	0,60
50	0,79	1,00	0,80
60	0,89	0,97	0,93
70	0,97	0,80	1,00
80	1,00	0,58	0,98
90	0,98	0,30	0,91
100	0,94	0,58	0,75
110	0,90	0,80	0,54
120	0,84	0,94	0,46
130	0,73	1,00	0,40
140	0,60	0,94	0,32
150	0,43	0,8	0,18
160	0,25	0,61	0.00
170	0,04	0,33	0,02

Table 2. Feature Vectors for the Objects depicted in Fig. 7.

If we consider a database only made with the objects' rules depicted in Fig. 7, the output results of the parsing procedure are presented in Table 3.

	Rule Object 1	Rule Object 2	Rule Object 3
Object 1	0,89	0,00	0,00
Object 2	0,00	0,9	0,00
Object 3	0,00	0,00	0,89

Table 3. Results of parsing procedure.

8. Classification

This module uses the output of the parsing and verifies which rule produces the higher value for the feature vector. For the example of Table 3 the result of the classification procedure is shown in Table 4.

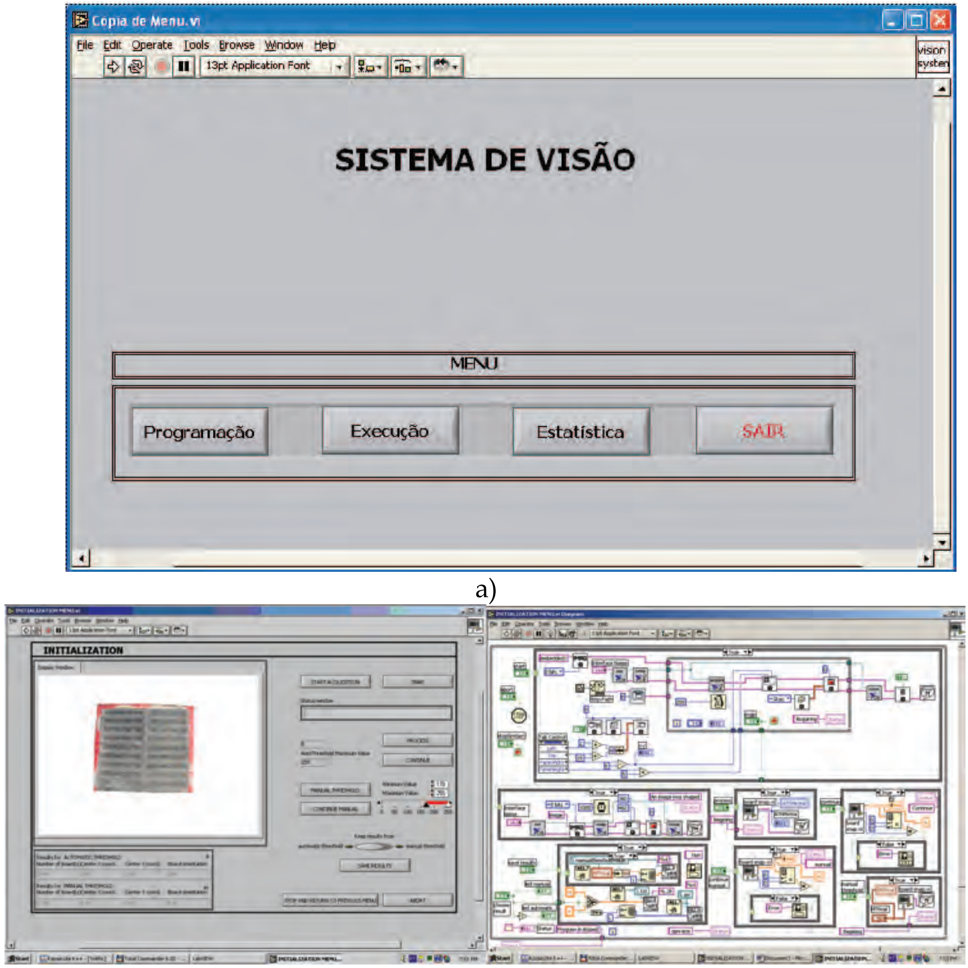
	Classification	
	Rules higher result	Object Type
Object 1	0.89	1
Object 2	0.9	2
Object 3	0.89	3

Table 4. Results of classification procedure.

If this value is less than a defined threshold (Tr of Fig. 2) it is assumed that a new type of object is present. In such case, the feature vector that characterizes this new object is submitted to the fuzzy grammar module in order to generate the new appropriated fuzzy rule.

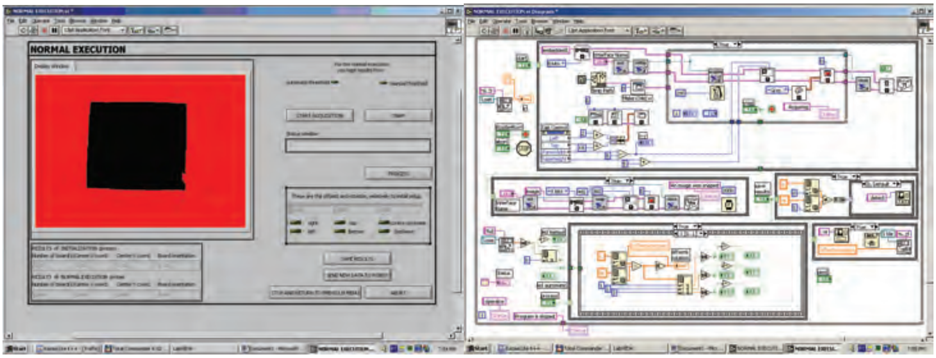
9. Experimental Results

The commercial software LabView 6.1 with IMAQ 6.0 was used, in order to increase the processing speed and to reduce the development time. This was also a requirement from the company that supports the development of the dispensing application. The fuzzy grammar was developed in C++. To call the fuzzy grammar from the LabView a DLL was created to encapsulate the C++ code. Fig. 15 illustrates the three principal LabView panels and vi diagrams of the application.



a)

b) c)



d) e)

Fig. 15. Different panels of the developed application. A selection must be done among: learning, execution phase and a statistical option for showing statistical data. a) Main panel. Learning phase: b) front panel, c) vi diagram. Execution phase: d) front panel, e) vi diagram.

The robot program that the robot performs over the workpiece, is sent to the robot via RS-232C protocol under the control of the JR software (trademark). However, the formats of the file that contains both the robot program and the information send to the RS-232 port are not known. This constraint was overcome through the development of a DLL that establishes the communication between LabView and the JR software. This DLL sends Microsoft Windows messages to the JR software, providing the appropriate commands to change the robot software. Finally, a message with the start command is sent to the JR software in order to initiate the robot program. JR software must be running during learning and execution phases. This development enables the robot to be controlled by the computer vision software.

9.1 A complete cycle

The feasibility and efficiency of the used approach have been studied by performing a set of experiments using 10 different types of objects (see Fig. 16).

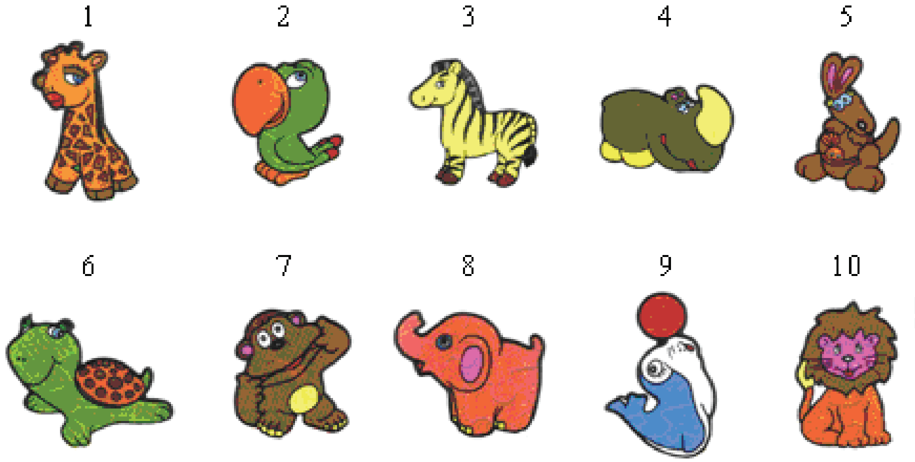


Fig. 16. Objects used in final experiment.

During the learning phase, for each object, the used robot program, the generated fuzzy rule, the orientation and the position are stored together in the database. On the execution phase, these workpieces are presented to the system having different positions and orientations relatively to the learning phase. The developed approach was able to identify each workpiece. Rotations, R, alignments and offset values in x, y were calculated, the robot's stored programs were adjusted accordingly and sent to the robot. Finally, the robot executed the changed program over each workpiece. The minimum offset that the system was able to calculate was as small as 0.2mm. The minimum rotation was 3 degrees.
Second column of Table 5 shows the generated linguistics terms for each primitive in the learning phase for object 10 of Fig. 16. An identical object but rotated of 40 degrees and with an offset in position of (x,y)=(10,15)mm was processed during the execution phase. Third and fourth column of Table IV show the obtained primitives and Linguistic terms, respectively. The classification result for the object 10 rule is 0.90, whereas for the other objects is 0.0. The calculated offset and orientation was of (10.0,15.1)mm and 39.3 degrees, respectively.
Table 6 shows the percentage of good classifications when each object is placed with 20 different locations and orientations. In some cases, objects were classified as Not Available in Database (NAD).

	LT	Primitive value	LT value
Primitive	(Learning phase)	(Execution phase)	(Execution phase)
FD0	HistVar:1	0.00	1.00
FD10	HistVar:2	0.13	0.92
FD20	HistVar:5	0.38	0.99
FD30	HistVar:7	0.60	1.00
FD40	HistVar:9	0.75	0.90
FD50	HistVar:10	0.91	0.98
FD60	HistVar:11	1.00	1.00
FD70	HistVar:11	1.00	1.00
FD80	HistVar:10	0.93	0.92
FD90	HistVar:9	0.84	0.90
FD100	HistVar:10	0.85	0.90
FD110	HistVar:10	0.94	0.90
FD120	HistVar:11	0.96	0,94
FD130	HistVar:10	0.91	0.98
FD140	HistVar:9	0.78	0.99
FD150	HistVar:7	0.59	0.99
FD160	HistVar:5	0.35	0.90
FD170	HistVar:3	0.17	0.96
SN	HistVar:5	0.36	0.91

Table 5. Example of execution data for object 10 from Fig. 16.

Object	1	2	3	4	5	6	7	8	9	10	NAD
1	95										5
2		100									0
3			100								0
4				95							5
5		5			90						5
6						95					5
7							90			5	5
8								95			5
9									100		
10										95	5

Table 6. Classifications of Objects (IN %).

As we can see from the above results, the developed approach can be applied when objects have different locations and orientations and only one sample was used during the learning phase. The advantage is that a high percentage of type of objects (greater than 90%), when submitted to rules of objects of other types, gives 0 as a result. This means that the system creates disjoints rules and assures a good classification.

10. Conclusion

In this work, we have used sensing technology to endow an industrial Desktop robot with a greater degree of flexibility in dealing with its environment. The goal was an adaptive, flexible, low-cost solution to maximize efficiencies in dispensing applications. Concretely, a CCD Camera was mounted over the robot and the visual information was used to autonomously change a previously off-line stored robot program to each workpiece.

The results illustrate the flexibility and robustness of the overall application. Further, the employed approach assures a good classification of workpieces and a minimum offset and rotation values of 0.2 mm and 3 degrees, respectively.

To further improve the classification procedure we intend to introduce new features in the rules and to experiment other methods than fuzzy logic. The overall application can be improved in such a way that the robot's program could also be automatically generated through the extraction of the relevant waypoints and path information.

The solution proposed can easily be extended to other type of machinery applications. For instance, to quality control inspection procedures including: dimensional measurement and gagging, verification of the presence of components, hole location and type of holes, detection of surface flaws and defects. It can also be easily extended to other categories of machine vision applications, in fact this approach was already applied to texture segmentation for tracking purposes (Ferreira, 2006). This application differs from the one that was presented here only by the type of features extracted from the images.

11. References

Appel, A.W. (1998), *Modern compiler implementation in C*, Cambridge University Press, ISBN: 0-521-60765-5, Cambridge.

Ballard, D.H. & Brown, C.M. (1982), *Computer vision*, Prentice Hall, ISBN:0131653164 New Jersey.

Bezdek, J.C. & Pal, S.K. (1992), *Fuzzy Models for pattern recognition*, IEEE Press, ISBN: 0780304225, New York.

Bumble-Bee (1999), *Parser Generator Manual* [online], Bumble-Bee Software, Available from http://www.bumblebeesoftware.com.

Costa, L.F. & Cesar Jr., R.M. (2001), *Shape Analysis and Classification*, Image Processing Series, Philip A. Laplante, (Ed.), CRC Press, ISBN 0-8493-3493-4, Florida.

Ferreira, M. J., Santos C, (2006), Tracking system using texture cue based on wavelet transform. *7th Portuguese Conference on Automatic Control*, Controlo 2006, September 2006, Lisboa, Portugal.

Fu, K.S. & Booth, T.L. (1986a), Grammatical inference: introduction and survey, Part I, *IEEE Transactions on pattern analysis and machine intelligence*, Vol. PAMI -8, No. 3, page numbers (343-359), ISSN: 0162-8828.

Fu, K.S. & Booth, T.L. (1986b), Grammatical inference: introduction and survey, Part II, *IEEE Transactions on pattern analysis and machine intelligence*, Vol. PAMI 8, No. 3, page numbers (360-375), ISSN: 0162-8828.

Gonzalez, R.C. & Woods, R.E. (2002), *Digital Image Processing*, Prentice Hall, ISBN: 0201180758, New Jersey.

Ivancic, F. & Malaviya, A. (1998), An Automatic Rule Base Generation Method for Fuzzy Pattern Recognition with Multi-phased Clustering, *Proceedings of KES'98 IEEE Conference of Knowledge Engineering System*, page numbers (66-75), Australia, April 1998, IEEE Press, Adelaide.

Janome (2004), *JR 2000 Software manual*, Series Desktop Robot, I&J Fisnar Inc.

Kerre, E.E. & Nachtegael, M. (2000), *Fuzzy techniques in image processing*, Studies in fuzziness and soft computing, Vol. 52, Physica-Verlag, ISBN: 3790813044, New York.

Kindratenko, V. (2004), Shape analysis, In: *CVonline: Compendium of Computer Vision* [online], R.Fisher, (Ed), Available from http://homepages.inf.ed.ac.uk/rbf/CVonline/.

Lee, E.T. & Zadeh, L.A. (1969), Note on fuzzy languages, *Information Sciences*, Vol. 1, (October 1969), page numbers (421-434).

Looney, C.G. (2002), Pattern Recognition, In: *Opto-Mechatronic Systems Handbook: Technical Applications - Handbook Series for Mechanical Enginnering*, Hyngsuck Cho, (Ed.), Vol 10, CRC Press, ISBN: 0849311624, New York.

Malaviya, A. (1996), *On-line handwriting recognition with a fuzzy feature description language*, PHD Thesis, Technische Universitat Berlin, Berlin.

Matrox (1998), *Matrox Imaging Library User guide*, Manual No. 10513-MN-0500, Matrox Electronic System Ltd, Canada.

Micheli-Tzanakou, E (2000), *Supervised and Unsupervised Pattern recognition, feature extraction and computational*, Industrial Electronics Series, J. David Irwin, (Ed.), CRC Press, ISBN: 0-8493-2278-2, Florida.

Pal, S.K. & Majumber, D.K. (1986), *Fuzzy mathematical approach to pattern recognition*, Halsted Press Book, John Wiley & Sons, ISBN: 0470274638, New Delhi.

Peters, L. & Leja, C., Malaviya, A. (1998), A Fuzzy Statistical Rule Generation Method for Handwriting Recognition, *Expert Systems Journal*, Elsevier Press, Vol. 15, No. 1, page numbers (1591-1604).

Russ, J.C. (1998), The image processing handbook, 3ed, CRC Press, ISBN: 0849325323 London.

Stanchev, P.L. & Green Jr, D. (2002), Formal languages for image primitives description, *International journal information theories and applications*, Vol. 9, No. 2, page numbers (51-60).

Williams, P.S. (1999), *The automatic hierarchical decomposition of images into sub-images for use in image recognition and classification*, PHD Thesis, University of Western Australia.

Yager, R.R. & Zadeh, L.A. (ed) (1992), *An introduction to fuzzy logic applications in intelligent systems*, Kluwer Academic, ISBN: 0792391918, Boston.