Research article

Computer vision and fuzzy rules applied to an industrial desktop robot

Cristina Santos
Industrial Electronics Department, University of Minho, Guimaraes, Portugal

Abstract

Purpose – Desktop robots are suitable for production line systems in industrial applications. Despite their capabilities to meet diverse requirements, they have to be programmed off-line using waypoints and path information. Misalignments in the workspace location during loading cause injuries in the workpiece and tool. Further, in flexible industrial production, machinery must adapt to changing product demands, both to the simultaneous production of different types of workpieces and to product styles with short life cycles. In this paper, visual data processing concepts on the basis of fuzzy logic are applied to enable an industrial desktop robot to raise its flexibility and address these problems that limit the production rate of small industries.

Design/methodology/approach – In this paper, a desktop robot performing dispensing tasks is equipped with a computer vision system. Visual information is used to autonomously change previously off-line stored robot programs for known workpieces or to call worker's attention for unknown/misclassified workpieces. A fuzzy inference classifier based on a fuzzy grammar, is used to describe/identify workpieces. Fuzzy rules are automatically generated from features extracted from the workpiece under analysis.

Findings – Different types of workpieces were tested and a good rate performance, higher than 90 per cent, was achieved. The obtained results illustrate both the flexibility and robustness of the proposed solution as well as its capabilities for good classification of workpieces.

Practical implications – The overall system is being implemented in an industrial environment.

Originality/value – The paper reports a piece of solid work which indicates clearly that the work is suitable for industrial utilization.

Keywords Computers, Fuzzy control, Industrial performance, Robotics

Paper type Research paper

Introduction

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.

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. 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 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.

In this paper, 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, the aim of this paper is to propose a flexible, adaptive and low-cost solution to address some problems which often limit the production rate of small industries.

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. The system is 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. 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,

The current issue and full text archive of this journal is available at
www.emeraldinsight.com/0144-5154.htm

Emerald

Assembly Automation
26/1 (2006) 59–68
 [ISSN 0144-5154]
[DOI 10.1108/01445150610645675]

equipped with a vision system, is performing precision dispensing tasks. 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.

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. Thus, by additional capabilities the robot can autonomously adapt to changing production needs, compensate misalignments and avoid injuries in the work tool and piece. As a result of this approach is that computation grounded on information derived from sensation enables the achievement of autonomy by starting some conceptualisation process of high level.

The dispensing application

Normally, in a 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 order to do so, a CCD Camera has been mounted over the robot.

In this study, a JR2000 Series Desktop robot from *I&J* Fisnar Inc (Janome, 2004) with simultaneous control of three axis is used as the test bed. The robot performs 3D linear and arc interpolation to include compound arcs in a working area of 150×150 mm. The overall experimental set-up is shown in Figure 1.

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. We have chosen to apply front lighting. The CCD Camera is a TRC TC5173 color camera with a resolution of 768×576 pixels. Image digitalization is done on a general purpose image acquisition board, National Instruments PCI1411, mounted inside a 100 MHz Pentium III PC. The PC is connected to the robot by a serial RS-232 C protocol.

System architecture

Figure 2 shows 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 Preprocessing 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.

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.

Preprocessing

To perform a robust industrial application, the following aspects must be minimized:

- random noise in the acquisition process;
- lack of light uniformity, which depends of the illumination technique; and
- 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.

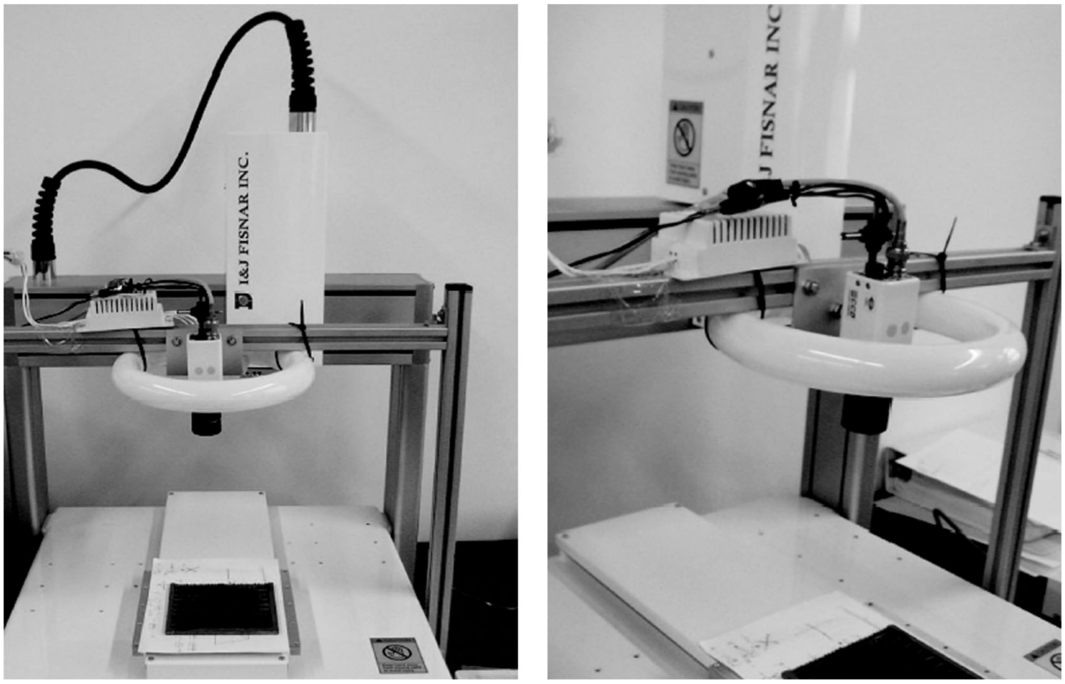
A light calibration procedure was developed (Russ, 1995) and employed to cope with the lack of light uniformity. A black and white object, covering all the working area, are acquired. Each of these images is divided in non-overlapping windows of 7×7 pixels and the mean of the gray-levels within each 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 shown in Figure 3.

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 (Figure 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.

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

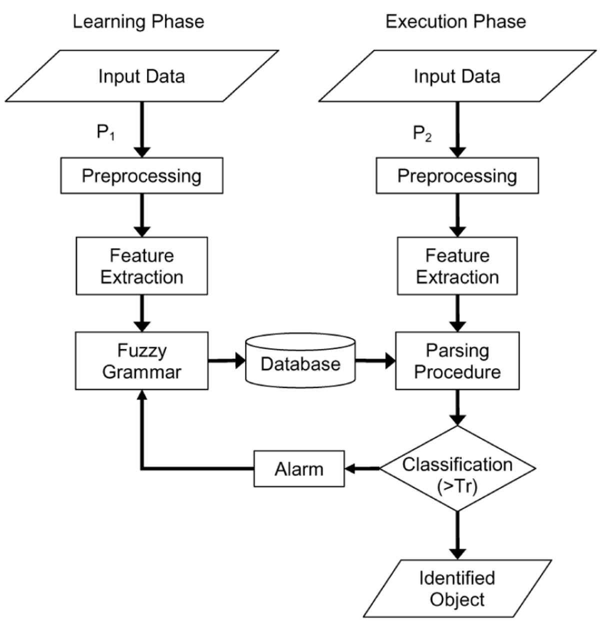
$$wi \begin{pmatrix} x_{ndi} \\ y_{ndi} \\ 1 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} x_{di} \\ y_{di} \\ 1 \end{pmatrix} \tag{1}$$

Figure 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

Figure 2 Architecture of the processing system



where

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

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

The extraction of the blobs that represent the objects is accomplished through a binarization with a fixed threshold

Figure 3 Light calibration procedure

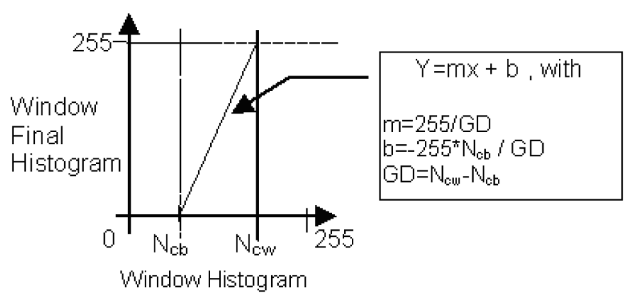
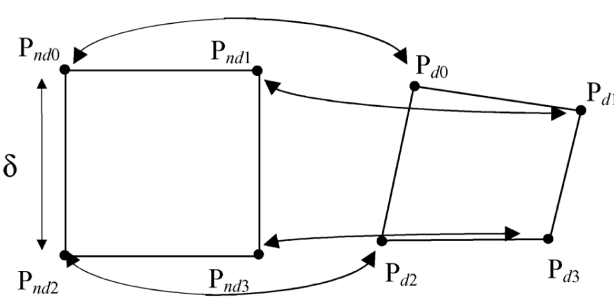


Figure 4 Mapping between a distorted ($P_{\mathrm{di}}(i = 0, 1, 2, 3)$) and a non-distorted element ($P_{\mathrm{ndi}}(i = 0, 1, 2, 3)$) of the grid. These elements have (x,y) coordinates



and through a blob-coloring like algorithm (Ballard and Brown, 1982). The segmentation of the image into regions could be achieved applying line finding or region growing techniques (Ballard and 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 shown in Figure 5(a) and (b) illustrates the corresponding extracted blob.

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 and Brown, 1982; Williams, 1999; Micheli-Tzanakou, 2000; Costa and Cesar, 2001; Gonzalez and 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.

Figure 6 shows some Feret diameters for object shown in Figure 5(a).

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

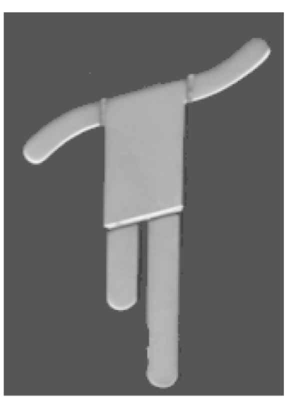
This type of external representation scheme is very useful in the computation of descriptors and is very adequate because the primary focus is on morphological features. However, this feature vector is highly dependent on the object's orientation, which poses a difficulty in the identification process. 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

$$\mathrm{NFD}(\theta) = \frac{\mathrm{FD}(\theta)}{\mathrm{FD}_{\max} - \mathrm{FD}_{\min}} - \frac{\mathrm{FD}_{\min}}{\mathrm{FD}_{\max} - \mathrm{FD}_{\min}} \tag{4}$$

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

Equation (4) is also independent of the size factor. For this particular application, this is a drawback since objects with

Figure 5 A sample object and the extracted blob

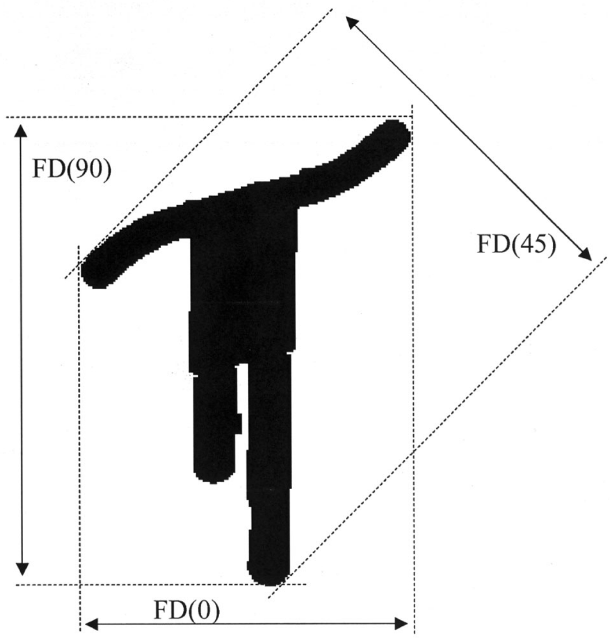


(a) Object sample

(b) Extracted blob

Assembly Automation
Volume 26 · Number 1 · 2006 · 59–68

Figure 6 The Feret diameters for object depicted in Figure 5(a) at angles 0, 45 and 90°



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.

Parsing

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 (Table I) that describe the objects shown in Figure 7.

If we consider a database only made with the objects' rules shown in Figure 7, the output results of the parsing procedure are as in Table II.

Classification

This module uses the output of the parsing and verifies which rule produces the higher value for the feature vector. If this value is less than a defined threshold 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.

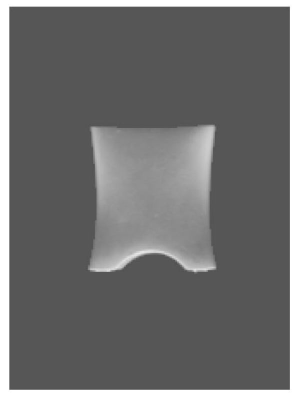
Fuzzy grammar

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:

- to deal with a high diversity of objects;
- to recognize simultaneously several different type of objects; and
- 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.

Figure 7 Some objects used in the choice of external representation type

(a) Object 1



(b) Object 2

(c) Object 3

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

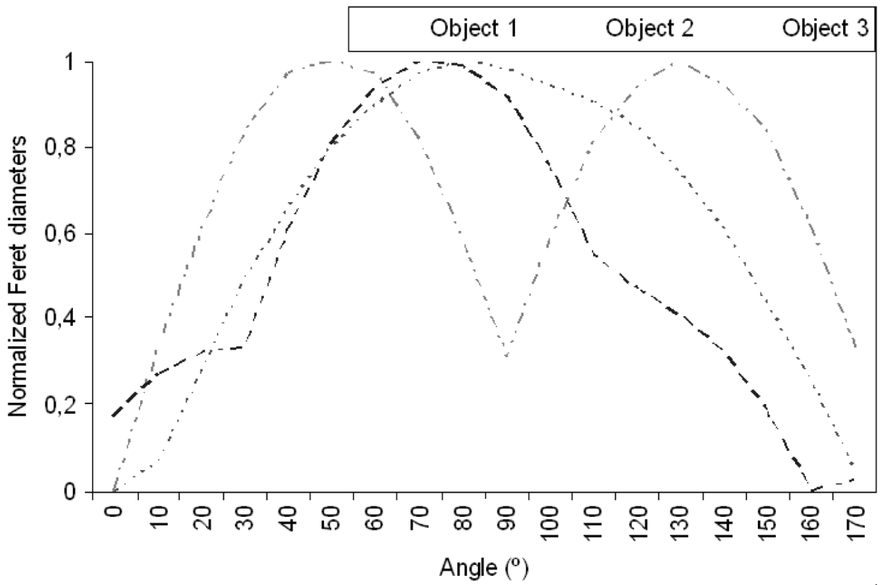


Table I Feature vectors for the objects depicted in Figure 7

Angle (°)	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 II Results of parsing procedure

	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

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 and Pal, 1992; Kerre and Nachtegael, 2000; Micheli-Tzanakou, 2000; Costa and 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 used in industry, they do not fulfil the constraints of the dispensing application.

In this paper, 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 and 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 and Booth, 1986a, b; Bezdek and Pal, 1992; Malaviya, 1996; Stanchev and 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 and Zadeh (1969), Pal and Majumber (1986), Bezdek and Pal (1992) and Yager and 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). $S_0 \in V_N$ is the starting symbol.

μ 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 shown in Figure 9 and includes four different steps:

1. The codification of the features to primitives. In this paper, 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 III. The membership function $\prod$ is shown in Figure 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.

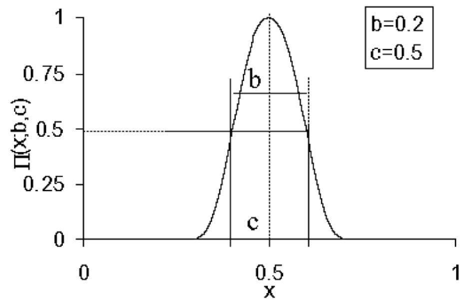
Figure 9 Syntax of the developed fuzzy language L(GF)

```
Language -> L(GF) = {x,μ(x)|x∈V*T, S⇒x}
GF=(VN,VT,P,S0,{μ})
VN={S0, Name, ElementSet ,Primitive, TermSet, Element, Term}
VT={SN, FD0, .. FD170, HistVar:1,..HistVar:11 (Table I),+,..,#}
S0→ 'Rule' RuleName' 'ElementSet
        ElementSet  →  ElementSet '&' ElementSet
                       '('ElementSet {'|' ElementSet }')'
                       '('ElementSet { '+' ElementSet } ')'
                       Element
                       λ
        Element     →  TermSet '#' Primitive
                       Primitive
        TermSet     →  '>' Term
                       '<' Term
                       '('Term '||' Term')'
        RuleName    →  Obj1
                       other
        Primitive   →  SN, FD0, ..., FD170
                       other
        Term        →  'HistVar:1' ... 'HistVar:11'
```

Table III Linguistic terms

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

Figure 10 Membership function PI



3. The definition of fuzzy modifiers (FM): "More than", "Less than" and "Between". The FM "More than" LT is defined by

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

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

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

The FM "Between" LT_1 e LT_2, is given by

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

where w_1 and w_2 are threshold values (Bezdek and Pal, 1992; Malaviya, 1996).

4. The definition of fuzzy operators (FO) which define the relations between the linguistic terms and primitives. We defined the following FO:
   - &, representing the AND of two primitives. It is given by the Yager intersection (Pal and Majumber, 1986).
   - $>$, representing "More than" LT and is given by $\mu_{\mathrm{MT}}\langle \mathrm{LT}\rangle$.
   - $<$, means "Less than" LT and is given by the function $\mu_{\mathrm{LT}}\langle \mathrm{LT}\rangle$.
   - ||, describes "Between two" LT and is given by $\mu_B\langle \mathrm{LT}_1\rangle\langle \mathrm{LT}_2\rangle$.
   - #, means a "Separator between a" primitive and a LT.
   - ( ), imposes a hierarchy in the rule.

Consider as an example object 2 is shown in Figure 7. Figure 11 shows the primitive FD20, obtained from the Feret diameter feature, NFD(θ) = 0.6, when $\theta = 20°$. This primitive has non-zero degrees of membership for LT HistVar: 6, LT HistVar: 7 and LT HistVar: 8 (Figure 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&

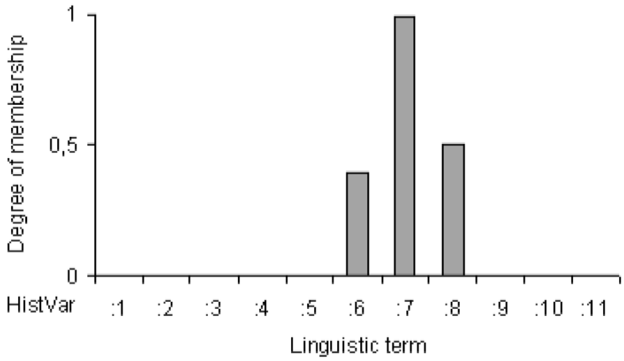


Figure 11 The highest fuzzy value for LV FD20 is obtained using LT HistVar:7

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. Figure 12 shows 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" (Figure 13) and "Between" (Figure 14).

Experimental results

In order to increase the processing speed and reduce the development time, we used the commercial software, LabView 6.1 with IMAQ 6.0. This was a requirement from the company that supports the development of the dispensing application. The fuzzy grammar was developed in C++. Figure 15 shows three different panels of the developed application.

The robot program that the robot performs over the workpiece, is sent to the robot via rs232 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 rs232 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.

Figure 12 Linguistic term for the primitive FD140 – Fuzzy Modifier "More than" HistVar:10

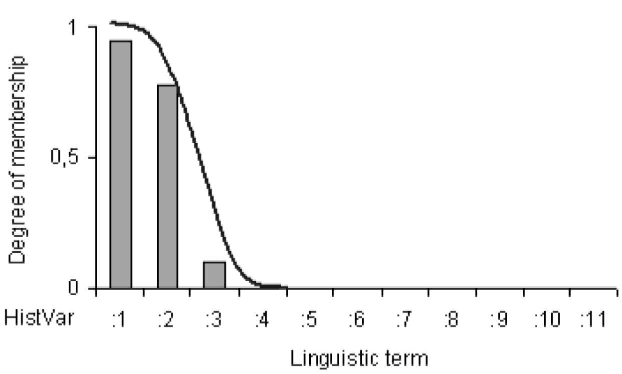


Figure 13 Fuzzy Modifier "Less than" HistVar:2

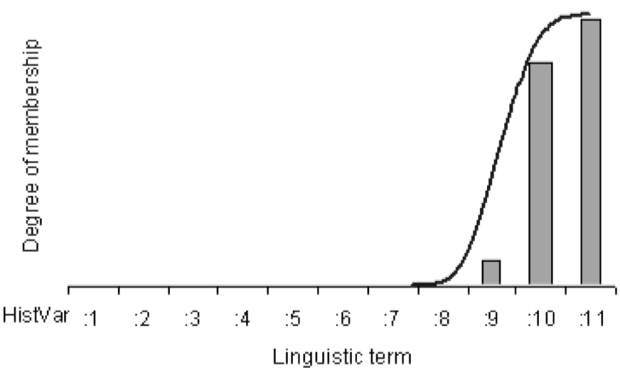
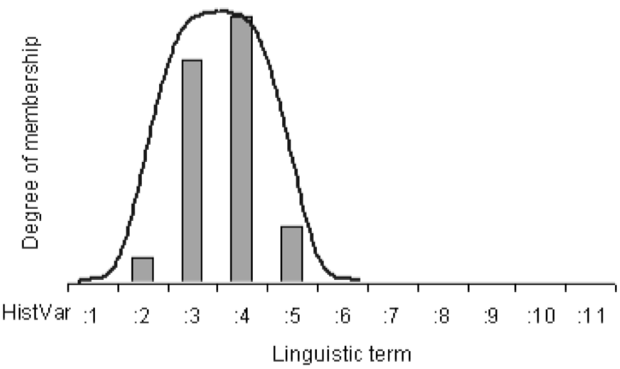


Figure 14 Fuzzy Modifier "Between" HistVar:3 and HistVar:4



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 (Figure 16).

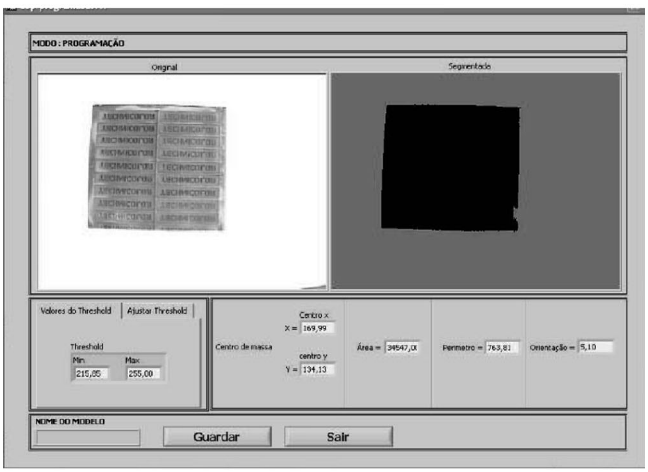
During the learning phase, for each object, the used robot program, the generated fuzzy rule, the orientation and the position are stored together in the database. On the execution phase, these workpieces are presented to the system having different positions and orientations relatively to the learning 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 0.2 mm. The minimum rotation was 3°.

Second column of Table IV shows the generated linguistics terms for each primitive in the learning phase for object 10 of

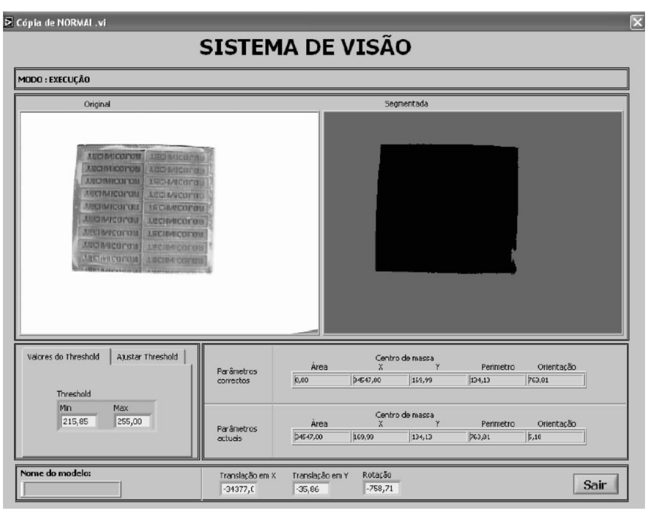
Assembly Automation
Volume 26 · Number 1 · 2006 · 59–68

Figure 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



(b) Learning phase front panel



(c) Execution phase front panel

Figure 16. An identical object but rotated of 40° 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°, respectively.

Table V 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).

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 (>90 per cent), when submitted to rules of objects of other types, gives zero as a result. This means that the system creates disjoints rules and assures a good classification.

Figure 16 Objects used in final experiments

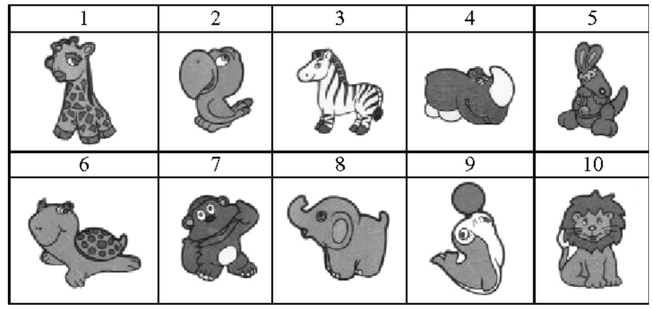


Table IV Example of execution data for object 10 from Figure 16

Primitive	LT (learning phase)	Primitive value (execution phase)	LT value (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 V Classifications of objects (IN per cent)

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

Conclusion

In this paper, we presented an adaptive, flexible, low-cost solution to maximize efficiencies in dispensing applications. We have used sensing technology to endow an industrial Desktop robot with a greater degree of flexibility in dealing with its environment. 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°, respectively.

We are currently implementing this solution in a real industrial environment.

We intend to further improve the classification procedure introducing new features in the rules and experiment other methods than fuzzy logic. Further, we intend to automatize the overall application such that the robot's program is also automatically generated through the extraction of the relevant waypoints and path information. The solution proposed in this paper can easily be extended to other type of machinery applications, as well as to other categories of machine vision applications. For instance, to quality control inspection including: dimensional measurement and gagging, verification of the presence of components, hole location and type of holes, detection of surface flaws and defects.

References

Appel, A.W. (1998), *Modern Compiler Implementation in C*, Cambridge University Press, Cambridge.

Ballard, D.H. and Brown, C.M. (1982), *Computer Vision*, Prentice Hall, Englewood Cliffs, NJ.

Bezdek, J.C. and Pal, S.K. (Eds) (1992), *Fuzzy Models for Pattern Recognition*, IEEE Press, New York, NY.

Bumble-Bee (1999), *Parser Generator Manual*, Bumble-Bee Software, available at: http://www.bumblebeesoftware.com

Costa, L.F. and Cesar, R.M. Jr (2001) in Philip, A. and Laplante (Eds), *Shape Analysis and Classification, Image Processing Series*, CRC Press, Boca Raton, FL.

Fu, K.S. and Booth, T.L. (1986a), "Grammatical inference: introduction and survey", *IEEE Transactions on pattern analysis and machine intelligence*, Vol. PAMI-8 No. 3, pp. 343-59, Part I.

Fu, K.S. and Booth, T.L. (1986b), "Grammatical inference: introduction and survey", *IEEE Transactions on pattern analysis and machine intelligence, PAMI*, Vol. 8 No. 3, pp. 360-75, Part II.

Gonzalez, R.C. and Woods, R.E. (2002), *Digital Image Processing*, Prentice Hall, Englewood Cliffs, NJ.

Ivancic, F. and Malaviya, A. (1998), "An automatic rule base generation method for Fuzzy Pattern recognition with multi-phased clustering", *IEEE Conference of Knowledge Engineering System, Proceedings of KES'98*, IEEE Press, Adelaide, pp. 66-75.

Janome (2004), "JR 2000 Software manual", Series Desktop Robot.

Kerre, E.E. and Nachtegael, M. (Eds) (2000), "Fuzzy techniques in image processing", *Studies in Fuzziness and Soft Computing*, Vol. 52, Physica-Verlag, New York, NY.

Kindratenko, V. (2004), "Shape analysis", in Fisher, R. (Ed.), *CVonline: Compendium of Computer Vision*, available at: http://homepages.inf.ed.ac.uk/rbf/CVonline/

Lee, E.T. and Zadeh, L.A. (1969), "Note on fuzzy languages", *Information Sciences*, Vol. 1, pp. 421-34.

Looney, C.G. (2002), "Pattern recognition", in Hyngsuck Cho (Ed.), *Opto-Mechatronic Systems Handbook: Technical Applications - Handbook Series for Mechanical Enginnering*, Vol. 10, CRC Press, Boca Raton, FL.

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", in David Irwin, J. (Ed.) Industrial Electronics Series, CRC Press, Boca Raton, FL.

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

Peters, L., Leja, C. and Malaviya, A. (1998), "A fuzzy statistical rule generation method for handwriting recognition", *Expert Systems Journal*, Vol. 15 No. 1, pp. 1591-604.

Russ, J.C. (1995), *The Image Processing Handbook*, 2nd ed., CRC Press, London.

Stanchev, P.L. and Green, D. Jr (2000), "Formal languages for image primitives description", *International Journal Information Theories and Applications*, Vol. 9 No. 2, pp. 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. and Zadeh, L.A. (Eds) (1992), *An Introduction to Fuzzy Logic Applications in Intelligent Systems*, Kluwer Academic, Boston, MA.

Corresponding author

Cristina Santos can be contacted at: cristina@dei.uminho.pt