Control of an Industrial Desktop Robot Using Computer Vision and Fuzzy Rules

Cristina M. Peixoto Santos*, Manuel João Ferreira*
*University of Minho/Industrial Electronics Department, Guimarães, Portugal

***Abstract*— Desktop robots are suitable for various production line systems in industrial applications like dispensing, soldering, screw tightening, pick'n place, welding or marking. 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 modern 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. Specifically, a desktop robot performing dispensing tasks is equipped with a CCD camera. 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.

Regarding the evaluation of the system performance, different types of workpieces were tested and a good rate performance, higher than 90%, 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. The overall system is being implemented in an industrial environment.

I. 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 (like dispensing sealing material onto cast motor parts) 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. 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. 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 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. 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. The system is adaptive, versatile and 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, equipped with a vision system, is performing precision dispensing tasks. This significantly reduces development time for these tedious processes.

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. 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 paper 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.

II. System Architecture

Fig. 1 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 *Preprocessing* module enhances the image quality and extracts the blob objects of the image. This module is necessary because the obtained images are not perfect. 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 from the *Feature Extraction* module are submitted to a *Parsing Procedure* module developed with the compilers yacc and lex [3], [4]. 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 an 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.

A. Preprocessing

To perform a robust industrial application, 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.

A light calibration procedure was developed [5], [6] and employed to cope with the lack of light uniformity. A black and a white object, covering the all working area,

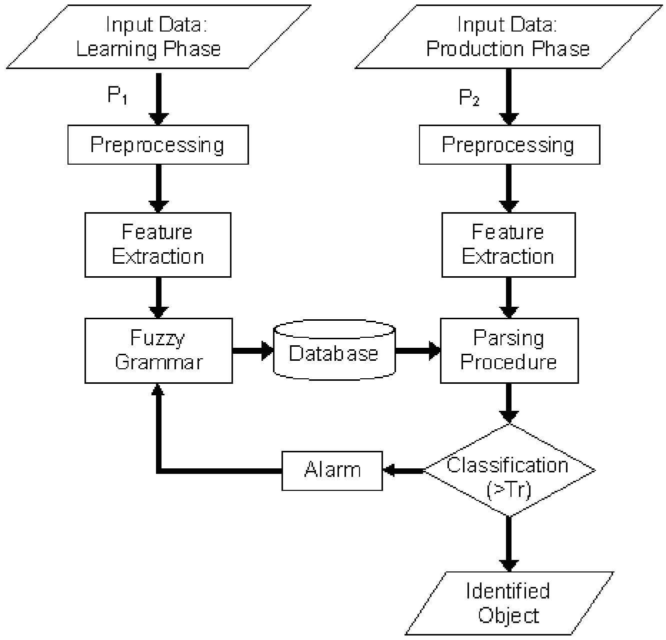


Fig. 1. Architecture of the processing system.

are acquired. Each of these images are 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. 2. This procedure ensures a high degree of light uniformity.

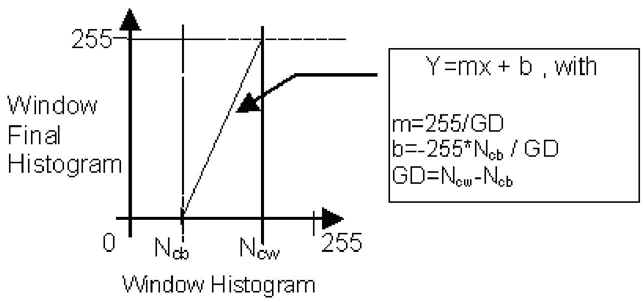


Fig. 2. Light calibration procedure.

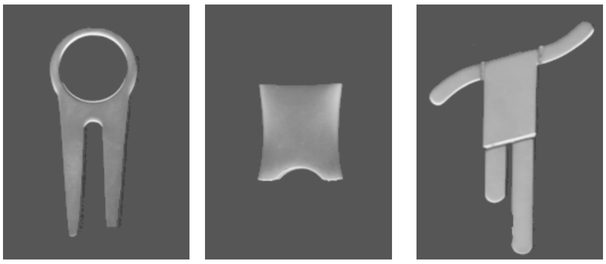
Image distorting is solved by applying an image correction using a well-known grid calibration procedure [5].

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 [7], [8].

B. Feature extraction

After the image segmentation into regions, 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. Several 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 [9], [10]. The best results were obtained using the Feret diameters at different rotation angles, θ, of the object.

By trial and error, we have chosen an increment between



(a) Object 1. (b) Object 2. (c) Object 3.

Fig. 3. Some objects used in the choice of external representation type.

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 the orientation drawback, the object is at first oriented 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]. Therefore, normalization is achieved simply by normalizing the obtained curve to unit maximum value as given by

$$NFD(\theta) = \frac{FD(\theta)}{FD_{max} - FD_{min}} - \frac{FD_{min}}{FD_{max} - FD_{min}}, \quad (1)$$

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. 3 are illustrated in fig. 4.

Equation 1 is also independent of the size factor. 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} Feret diameter, normalized to the maximum size allowed for an object in the system.

The chosen descriptors capture essential differences between objects, as required, while maintaining as much independence as possible to changes in factors such as location and orientation.

C. Fuzzy Grammar

Fuzzy grammar is a pattern classification syntactic model used to represent the structural relations of patterns [8], [12], [13], [14], and describe the syntax of the languages that generate the fuzzy rules. Herein, we briefly review some basic concepts of fuzzy grammar (for a full discussion see [15], [16]). 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 nonterminal 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 $p \in P$.

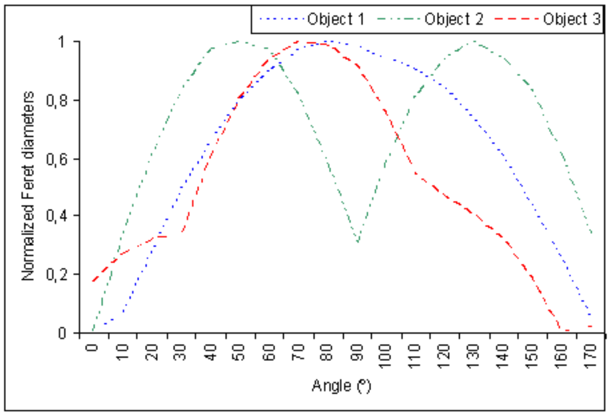


Fig. 4. Normalized Feret diameters for objects depicted in Fig. 3. Solid, dash and dash-dot-dot traces represent objects 1, 2 and 3, respectively.

The syntax of the developed language $L(GF)$ is depicted in Fig. II-C and includes 4 different steps: 1) The codification of the primitives to linguistic variables

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 → 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'

Fig. 5. Syntax of the developed fuzzy language $L(GF)$.

(LV). In this paper, the primitives are the Feret diameters ($NFD(\theta)$) and the size SN, which are coded to the linguistic variables $FD\theta$ and SN, respectively. 2) The definition of linguistic terms (LT) (Table I). 3) The definition of fuzzy modifiers (FM): "More than", "Less than" and "Between". The FM "More than" LT is defined by

$$\mu_{MT} < LT >= \begin{cases} 1 & x \geq L \\ S(x, L - lb, L - \frac{lb}{2}, L) & x < L \end{cases} . \quad (2)$$

where L is a threshold value and lb is the bandwidth value of the S membership function [13]. The FM "Less than" LT is given by

$$\mu_{LT} < LT >= \begin{cases} 1 & x \leq L \\ 1 - S(x, L, L + \frac{lb}{2}, L + lb) & x > L \end{cases} \quad (3)$$

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

$$\mu_B < LT_1 >< LT_2 >= \begin{cases} S(x, w_1, w_1 + \frac{lb}{2}, w_1 + lb) & x > w_1 \\ 1 & w_2 \leq x \leq w_1 \\ x \geq L & \\ S(x, L - lb, L - \frac{lb}{2}, L) & x < L \end{cases} \quad (4)$$

where w_1 and w_2 are threshold values [14].

4) The definition of fuzzy operators (FO) which define the relations between the linguistic terms and variables. We defined the following FO: a) &, representing the AND of two primitives, is given by the Yager intersection [17]. b) $>$, representing "More than" LT is given by $\mu_{MT} < LT >$. c) ¡, means "Less than" LT and is given by the function $\mu_{LT} < LT >$. d) $||$, describes "Between two" LT and is given by $\mu_B < LT_1 >< LT_2 >$. e) # means a "Separator between a" LV and a LT. f) (), imposes a hierarchy in the rule.

TABLE I
LINGUISTIC TERMS

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

Each LT has a member function, which gives the LT value for a particular LV, as depicted in table I.

Fig. 6 illustrates an example for the linguistic variable $FD20$, obtained from the Feret diameter, $NFD(\theta) = 0.6$, when $\theta = 20$ degrees for Object 2 (Fig. 3(b)). There are non-zero degrees of membership for this linguistic variable for LT HistVar:6, LT HistVar:7 and LT HistVar:8 (Fig. 6), but the highest fuzzy value is obtained using LT HistVar:7. 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.

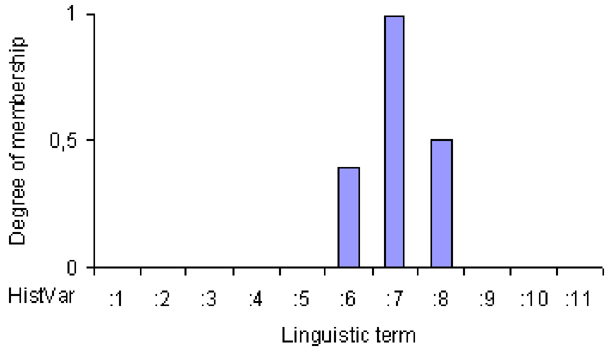


Fig. 6. The highest fuzzy value for LV $FD20$ is obtained using LT HistVar:7.

The last element of the rule indicates that the object has a small size.

If more than one linguistic member gives a result superior to 0.75; we apply fuzzy modifiers like "More than", "Less than" and "Between", to combine the obtained results [17]. Fig. 7 illustrates the procedure for fuzzy modifier "Less than" HistVar:2. The final fuzzy value results from the combination of LT HistVar:2 and LT HistVar:3.Similar procedures apply for fuzzy modifiers "Less than" and "Between".

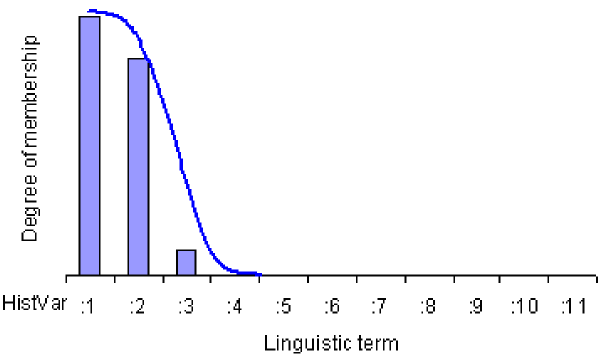
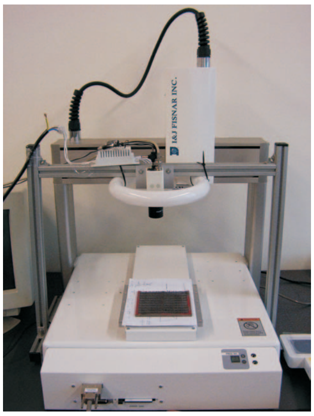


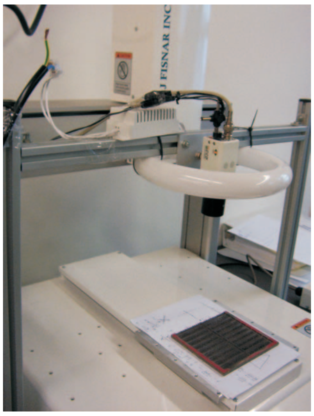
Fig. 7. Linguistic term for the linguistic variable C1 - Fuzzy Modifier "Less than" HistVar:2.

III. Experimental Results

The experimental setup is shown in Fig. 8. The robot is a JR2000 Series Desktop robot from I&J Fisnar Inc [1] with simultaneous control of 3 axis. The robot performs 3D linear and arc interpolation to include compound arcs in a working area of 150x150 mm. 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 and that the illumination is constant over time. We have chosen to apply front lighting. The CCD camera is a TRC TC5173 color 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.



(a) General front view.



(b) Detailed view of the front lighting technique.

Fig. 8. Experimental setup showing the desktop robot with the mounted CCD Camera, the fluorescent lamp and a mould.

In order to increase the processing speed and reduce the development time, we used the commercial software, LabView 6.1 with IMAQ 6.0. The fuzzy grammar was developed in C++. A DLL was created in order to make easier the integration with LabView. Fig. 9 illustrates three different panels of the developed application.

A. A complete cycle

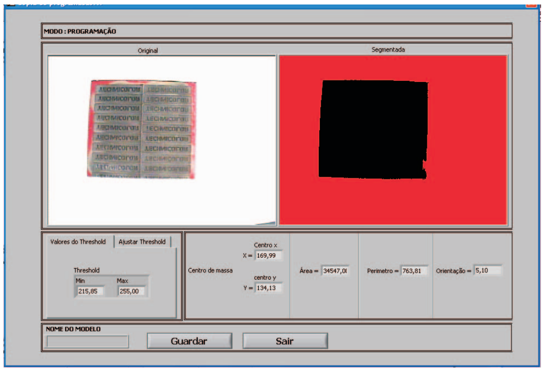
Herein, we present the results obtained for 10 different objects. During the learning phase, the robot's program and the generated fuzzy rule for each object are stored together in the database. On the execution phase, a set of workpieces is presented to the system which identifies each one of them. Rotations, R, alignments and offset values in x, y are calculated, the robot's stored programs are adjusted accordingly and sent to the robot via RS-232C protocol. Finally, the robot executes 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 degrees. Table II 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).

The advantage of this approach is that a high percentage of type of objects (greater than 90 %), when submitted to rules of objects of other types, gives 0 as result. The system creates disjoints rules and assures a good classification.

IV. 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 overall system worked in two phases: a *learning* and an *execution* phase. During the *learning* phase, the robot is programmed off-line but at the same time builds a rule database using a fuzzy grammar. These rules classify each object in an unique fashion which has been shown to be very adequate for classification. This fuzzy rule database is accessed during the *execution* phase, and thus the set of workpieces positioned on the robot's working area are identified. The presented solution is robust enough to deal with common errors such as missing workpieces; and flexible such that a *learning* phase is autonomously

(a) Main panel.



(b) *Learning* phase front panel.

(c) *Execution* phase front panel.

Fig. 9. Different panels of the developed application. A selection must be done among: *learning*, *execution* phase and a statistical option for showing statistical data is also available.

TABLE II
CLASSIFICATIONS OF OBJECTS (IN %)

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

initiated in case an unknown workpiece is detected.

The presented results illustrate both 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.

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

We intend to further improve the classification procedure 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 gaging, verification of the presence of components, hole location and number of holes, detection of surface flaws and defects.

ACKNOWLEDGMENT

We wish to thanks to the students who worked with us in this project at the Industrial Electronics Department in the University of Minho. They are Hugo Oliveira, Sílvio Granjo, Duarte Silva e Fernando Gouveia.

REFERENCES

[1] JR 2000, Series Desktop Robot, Janome, 2004.
[2] A. Malaviya, "On-line handwriting recognition with a fuzzy feature description language," *PHD Thesis*, Technische Universitat Berlin, Berlin, 1996.
[3] Peter L. Stanchev, David Green Jr. "Formal languages for image primitives description", *International journal information theories and applications*, vol. 9, n. 2 pp. 51-60, 2000.
[4] Andrew W. Appel, *Modern compiler implementation in C*: Cambridge University Press, 1998.
[5] Parser Generator, Bumble-Bee Software, Available in WWW: URL:http://www.bumblebeesoftware.com, 1999.
[6] John C. Russ, *The image processing handbook*, 2ed: CRC Press, 1995.
[7] Manuel J. O. Ferreira, "Aplicação de técnicas de processamento de imagem ao controlo da qualidade na linha de produção em indústrias tradicionais," *Tese de Mestrado, Departamento de Electrónica Industrial*, Universidade do Minho, Braga, 1996.
[8] Dana H. Ballard, Christopher M. Brown, *Computer vision*, New Jersey: Prentice Hall, 1982.
[9] Rafael C. Gonzalez, Richard E. Woods, *Digital Image Processing*: Prentice Hall, 1992.
[10] Evangelia Micheli-Tzanakou, "Supervised and Unsupervised Pattern recognition, feature extraction and computational," *in Industrial Electronics Series*, J. David Irwin, Ed.: CRC Press, 2000.
[11] Luciano da F. Costa, Roberto M. Cesar Jr., "Shape Analysis and Classification," *in Image Processing Series*, Philip A. Laplante, Ed.: CRC Press, 2001.
[12] Matrox Imaging Library, *User guide, Manual No. 10513-MN-0500*, Matrox Electronic System Ltd, Canada, 1998.
[13] King-Sun Fu, Taylor L. Booth, "Grammatical inference: introduction and survey," Part I, IEEE Transactions on pattern analysis and machine intelligence, vol. PAMI -8,n. 3, pp. 343-359, May 1986.
[14] King-Sun Fu, Taylor L. Booth, "Grammatical inference: introduction and survey," Part II, IEEE Transactions on pattern analysis and machine intelligence, vol. PAMI 8, n. 3, pp. 360-375, May 1986.
[15] James C. Bezdek, Sankar K. Pal, *Fuzzy Models for pattern recognition:* IEEE Press, 1992.
[16] E.T. Lee, L.A. Zadeh, "Note on fuzzy languages," *Information Sciences-1*, pp. 421-434, 1969.
[17] S. K. Pal, D.K.D. Majumber, *Fuzzy mathematical approach to pattern recognition:* a Halsted Press Book, John Wiley & Sons, New Delhi, 1986.