DEVELOPMENT AND APLICATION OF A VISION SYSTEM FOR INDUSTRIAL ROBOT

Maurício Velloso Grassi
Universidade Federal do Rio Grande do Sul – Departamento de Engenharia Mecânica
Rua Sarmento Leite 425, CEP 90050-170, Porto Alegre/RS, Brasil
benevitz@brturbo.com.br

Flávio José Lorini
Universidade Federal do Rio Grande do Sul – Departamento de Engenharia Mecânica
lorini@mecanica.ufrgs.br

Maurício Alexandrini
Universidade Federal do Rio Grande do Sul – Departamento de Engenharia Mecânica
malexandrini@hotmail.com

Abstract. *This paper deals with the development of a vision system integrated to an industrial robot for manipulating tasks. For its development, video capture equipment was combined to an industrial robot through a comunication system and a PC, where image analysis is performed. The main piece of the capture and processing system is the program RobVis. Developed using Visual Basic language, its function is to identify the position and orientation of objects to be manipulated by a robot, analysing images captured by a CCD camera. Once the desired variables are defined, these are transmitted to the robot controller, through a IRBCom communication system, developed by the UFRGS Laboratory of Robotics. At the controller, a manipulating program takes the variables to pick up the objects and place them at a pre-established unloading point. The system is qualified as of easy usage and application in robotic manipulating tasks that need information about position and orientation of parts inside a section of the robot workspace that is within the range of the vision system.*

Keywords*: industrial automation, robotics, vision system, image processing*

1. Introduction

In the field of industrial robotics, the application of robots using peripheral sensors to enhance autonomy, flexibility and productivity of automated production cells stands out. In this context, some of the most important peripheral devices are the vision systems, the so-called machine, computer or robotic vision. There are authors like Kragić, D., Petersson, L. and Christensen, H. I. (2002), who describe common manipulating tasks dependent of a vision system, which generates data like position and orientation for the robotic manipulator. Soares, A. M. S. and Gonçalves, V. D. (2004), deal with the development, using C++ language, of a bolts manipulating program using a look-up table. Azuela, J. H. S. (2001) shows the application of the invariant moments as a useful tool in vision systems. Seitz, M. (1999) shows the manipulation of unknown parts using a camera-manipulator device.

To merge industrial robots characteristics, like high production rates, accuracy and no loss of performance during all the whole operational cycle, with the advantages of vision systems, like the capacity of constant evaluation of random conditions in the workspace without the need for human supervision, is the focus of this paper.

2. Technologic Fundamentals

To describe the proposed system, some concepts about industrial robotics and vision systems must be presented.

2.1. Robots

A command unit and a manipulation arm form an industrial robot basically. At the command unit, there are the operational system of the control memory and the software for movement control of the manipulator, executing at the main processor and the axis control processor. Another processor could integrate the control module, as the I/O processor, the in/out signals manager. Also, in the controller, specific programs for different tasks are inserted, observing the manipulator possibilities, written by the users in a language known by the specific robot.

Despite the range of manipulator types, the most versatile is the anthropomorphic, which reproduces the configuration of the human arm. Inside it, there are motors and actuators, the “muscles” of the robotic arm. At its end, or wrist, is attached the end-effector, generic name for a vast range of tools, from weld equipment to devices that pick-up and manipulate material.

2.2. Robotic Vision

Vision systems for robots are considered the most powerful sensorial capability applied to industrial robotics, and can be associated with different kinds of hardware and software. The functions of vision system devices can be summarized in three basic levels for its application (Niku, 2001):

Capture: corresponds to image capture, as well as the equipment, and the most common nowadays is the use of CCD cameras (*Charge-Coupled Device*). They are devices of easy joining with dedicated software, which generate signals easily changed to digital images of common formats, like BMP (*Bit MaP*) (Wilson, 1984).

Processing: defined as a set of routines applied to the captured image, looking to prepare it to a subsequent interpretation. Basic functions are reduction of noise, or clearing the image of interference to make interpretation an easier task, and thresholding, which is to reduce the image, a color or gray levels image, to a black-and-white, or one-and-zero, image. In that process, a pixel-to-pixel analysis is made. Pixels are the basic unit of digital images and to each one a new intensity level is given, black or white, depending on pre-established parameters, looking to point out interest points in the image. The desired details become totally black against the background and other unnecessary details, which become white. To work with only two intensity levels, instead of a bigger range of colors or shades, save time and computational procedures (Shi, 1997).

Analysis: corresponds to the image interpretation, where the needed information is extracted. Simplified by the processing, information like the presence or not of objects and its dimensions, position and other data relevant to specific applications could be extracted from the image. As in the processing, mathematics based techniques are applied over the image pixels matrix. As a result of an analysis, vision devices could take decisions, observing pre-established parameters, and coordinate the sending of information to other equipment, such as robots.

3. Developed System

The proposed vision system, intended to equip an ABB IRB1400 robot, is composed as follows: capture equipment and lighting; image processing program; PC/controller interface; manipulating program.

3.1 Capture Equipment and Lighting

The used equipment is a CCD camera, which generates digital signals of easy usage by programs and devices used in a PC. A monochrome Samsung BW-2302EA with a lens set SLA-124C was chosen, a device largely used in vision systems and positioning (LACTEC, 2004). To pick up the signals, a capture board VTV 2004 PCI is used, adequate to the Windows environment, with capacity of 30 fps (frames per second) and up to four cameras.

The lighting of the capture area is of great importance for the performance of the system. In this case, direct lighting is used, with the light source positioned beside the camera and aiming directly at the workspace.

3.2 PC/Robot Controller Interface

A locally developed device, the communication module IRBCom, makes communication between the PC and the ABB S4 controller. It is qualified as a software and hardware interface of easy usage and which allows real time control actions (Bayer, 2004). It connects the PC parallel port and the digital I/O ABB DSQC 223 board, intended originally to control devices integrated to the robot. It uses routines developed in the interface software in DLL format (Dynamic Link Library), employed in communications and control. They allow integration of the system to any program developed for the Win32 platform (Bayer, 2004). Programming in its native language, in this case, ABB Rapid, makes the application of routines in the controller. These routines must be introduced in the source code of the programs intended to use communication, as in the case of this vision system.

3.3 Image Analysis Program

Developed in Visual Basic, it received the name of *RobVis*, from the acronym Robotic Vision. It is divided in capture, preparation and processing modules, whose interfaces are shown in Fig. (1).

3.3.1 Capture Module

Its function is to generate the images for the other modules and activate them, using its menu bar.

3.3.2 Preparation Module

The look-up table is generated for part's orientation determination. Each solid, because of its geometry, has specific moments of inertia for each orientation, which is the parameter used in its determination. A look-up table is made for each solid to be manipulated and is stored in a *.txt file*, which is accessed by the processing module.

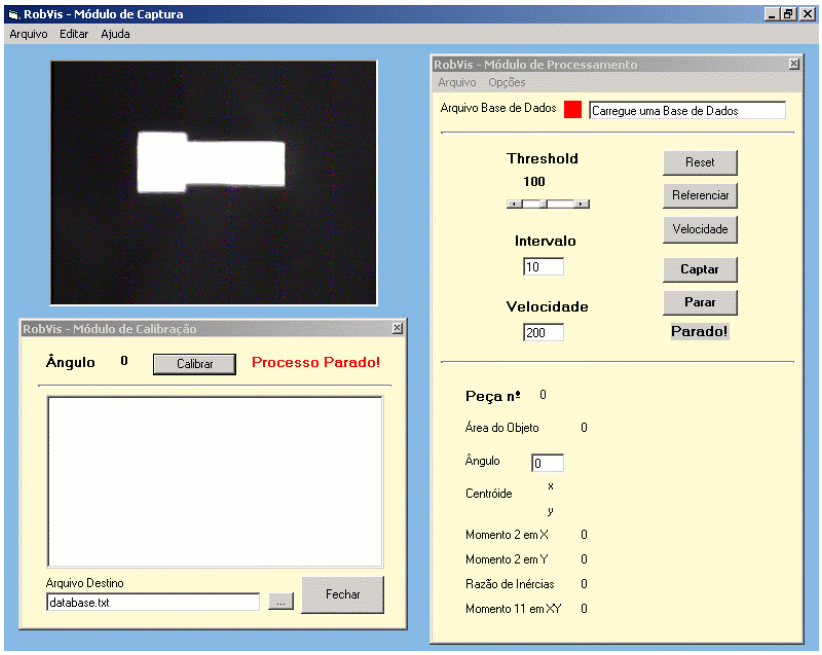

Figure 1. Visual interface of the *RobVis* program

3.3.3 Processing Module

The image processing and analysis re made. The generated results correspond to the position and orientation of the solid relative to the robot. They are sent to the controller when automatic commands generate signals in the IRBCom. Its main features are:

Menu Bar: allows the loading of a database or selection whether the operation will be over a revolution solid, like a cylinder, where orientation is not needed. Also, it allows selection whether the condition is of light background/dark object or vice versa.

Database: shows the loaded table or condition of revolution solid. This field blocks the operation until a solid is specified.

Threshold: roll bar with values from 0 to 255 to specify the threshold. Depends of the operational conditions and must be adjusted as well.

Interval: time between captures, in seconds.

Speed: establishes the robot speed, in mm/s, adjusted following operational conditions. The command is sent to the manipulator activating the corresponding button.

Reference Button: frees the manipulator to be referenced by the operator, using its joystick.

Buttons Capture and Stop: activate and stop the sequence of capture-processing-analysis.

Results Section: shows information such as the coordinates and orientation of the last manipulated part.

3.3.4 Mathematical Methods

Some of the existing methods used in robotic vision were applied in the routines of the preparation and processing modules:

Thresholding: reduces complex images to a set of 0's and 1's. Each image pixel has its intensity analyzed, which is between 0 and 255, and is given a new value, 0 or 1, in accordance with a logical function. In the case of light background and dark object, the function is showed in Eq. (1) (Gonzalez and Woods, 2003):

$$g(x,y) = \begin{cases} 1 & if\ f(x,y) < T \\ 0 & if\ f(x,y) \geq T \end{cases} \quad (1)$$

Where $g(x,y)$ is the resulting pixel intensity value, $f(x,y)$ is the original intensity and T is the threshold. As a result, with a threshold T like 100, dark pixels, near 0, will be labeled 1 (black, "on" or "object") and above the threshold, as 0 (white, "off" or "background"), isolating the "on" pixels to be analyzed. If the task will be executed with dark background and light object, the signals must be inverted.

Position Determination: the number of pixels over the total area of the image and the object must be counted. Then, the position determination is made using the following equations (Gonzalez and Wints, 1977):

$$\bar{y} = \frac{\sum y}{área} = \frac{M_{0,1}}{M_{0,0}} \tag{2}$$

$$\bar{x} = \frac{\sum x}{área} = \frac{M_{1,0}}{M_{0,0}} \tag{3}$$

Where the moment $M_{0,0}$ is the sum of as much 1's as "on" pixels in the image, supplying the total number of black pixels, or the object area in pixels. $M_{0,1}$ and $M_{1,0}$ represent the sum of black pixels multiplied by its distances from the image x and y axes, respectively.

Orientation Determination: orientation is the angle between the part's medium axis and the y-axis of the robot base, parallel to the superior edge of the capture area. For its determination a look up table of the moments, which vary with the change in the object's orientation, is made. In this case, the moments $M_{0,1}$ e $M_{1,0}$ are invariant, but this doesn't apply to moments of higher level, as $M_{0,2}$ and $M_{2,0}$, which are the sum of black pixels multiplied by its distances, squared, from the axes x and y, respectively (Niku, 2001). Note that the moments are dependent on the orientation and the position, a problem solved by calculating them in relation to auxiliary axes with origin in the center of the object. Once calculated these values, for known orientations, a look up table with the orientations ant the respective moment values, for a specific object is generated. To simplify the process, instead of comparing two moment values, its ratio R is used, as follows:

$$R = \frac{M_{0,2}}{M_{2,0}} \tag{4}$$

Both these moments and its ratio show symmetric results for 0^o to 90^o and 90^o to 180^o. A factor to differentiate them is used. The simplest is a negative sign. The moments $M_{0,3}$ and $M_{3,0}$ supply this sign but are very sensitive to noise, because the distances are cubed, with errors in excess. As a solution the moment $M_{1,1}$ is used, which is the sum of the distance of the object's pixels from the x-axis multiplied by its distance from the y-axis, with origin in its center. It gives the needed sign to differentiate when the object is oriented between 0^o and 90^o (as between 180^o and 270^o) or between 90^o and 180^o (as between 270^o and 360^o).

Once obtained these results by the preparation module, they are stored in a .txt file, being loaded by the user in the processing module when an identical object to the one that was used to generate the table is manipulated. The results are inserted in logical functions, as follows:

$$se\ a < R \leq b\ e\ sinal = c,\ ang = d \tag{5}$$

Where R is the moments ratio, *sinal* is the positive or negative sign from the result of $M_{1,1}$ (both calculated during the image analysis operation), a and b are the moments ratios of known angles, c is the positive or negative sign from the result of $M_{1,1}$ (both calculated during the preparation). To obtain the resultant orientation, the value of d, which is the known angle for the range $a - b$ and the sign c, is given to *ang*, and then considered the object orientation.

The preparation module was designed to take values every 15^o between 0^o and 165^o, which is enough because the results for the other quadrants are symmetrical. This preparation allows the robot to pick objects oriented in any angle, bringing its orientation closer to angles spaced 15^o. The operator must deposit the object in the specified angles and in the order showed on the screen of the preparation module. For each angle, the module takes 10 image captures, to minimize errors from accidental noisy images, at 2-second intervals, and subsequently analyses them. After the analysis for each one of the captures, the program takes the average result, for the moment ratio and $M_{1,1}$ sign, which are stored in the look up table as the values for that angle.

3.4 Manipulating Program

Developed in *ABB Rapid*, the native language of the IRB1400 robot, the manipulating program recognizes and takes the information provided by the processing program and inserts them into its parts manipulating routines.

The program, named as *RobVisR.prg* and executed at the robot controller, has as its main component a looping that reads the command, identifies it and calls the necessary routine, standing responsible to receive its parameters and execute the programmed task. When concluded, the program returns to the looping and calls *ReceiveByte* to receive the next command and, if the PC doesn't send any signal, the controller stays in stand by. In the same way, in the time interval when a task is being executed, the controller stays blocked to other commands. In this situation, the PC verifies that the controller is busy and waits to send a new command.

There are four main routines in its code: *Place* – determines the unloading point; *DoHome* – refers the capture area related to the robot; *SetSpeed* – adjusts the speed, receiving a value from the PC; *Move* – manipulates the parts, based on the parameters sent by the PC and on the established unloading point.

4. Experimental Evaluation

To conclude the development of the vision system, experimental laboratory tests were made, both to evaluate the integration of the various equipments as its efficiency. Among the various tools used, there is a pneumatic gripper, activated by electric signals sent by the controller, one to open and another to close. Blocks of different shapes were used, to evaluate the performance of the look up table working with different parts, and the tests were made with light background/dark object and vice versa.

The first action to use the system is to generate the look up table, as described, and when opening the processing module, to load the *.txt file*, as shown in Fig. (2)

Figure 2. Generating and loading the look up table

The next step is to refer the vision field, to establish a relative position condition between the image capture area and the manipulator. The calculated position for the object in the image is changed to the robot's world coordinates, and the *z* coordinate assumed in the reference establishment is used as the *z* coordinate for picking and placing the object. The reference is executed positioning the center of the object as nearest as possible over the left-superior vertex o the capture area. Using the corresponding button in the processing module, a signal is sent to the robot to free the arm to be moved by the operator via joystick. So, the user must move the end-effector to a position over the object, a pick-up-like position, as shown in Fig. (3).

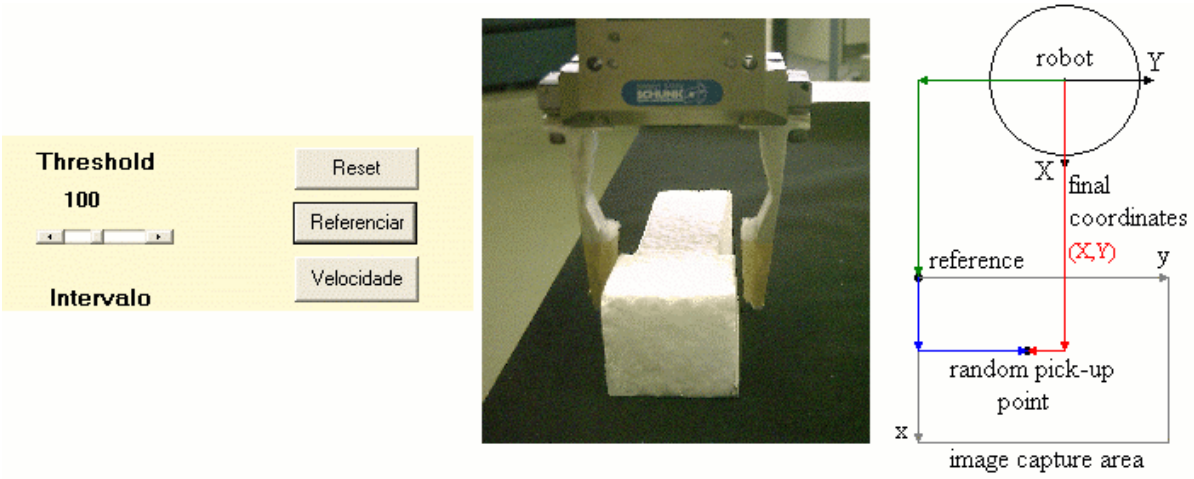

Figure 3. Capture area reference procedure

It is necessary to choose the parameters in the processing module, already described in section 3.3.3, as threshold and time interval, considering the operational conditions. After the preliminary steps, the *RobVis* system is ready to be used. The images are taken considering the time interval followed by a fast processing-analysis procedure and its results are sent to the controller, where the *RobVisR* waits in looping. Once the variables are identified, the manipulator picks up the part, passes through a pre-established trajectory and places it at the active unloading point. An example of operation is shown in Fig. (4).

Figure 4. Capture screen, processing module and manipulator picking an object

5. Concluding Remarks

The results were satisfactory. The system showed capability of operating with different test objects, including revolution objects. The performance for dark object/light background and light object/dark background were identical and satisfactory. The capture, processing and analysis procedures are as fast as the generation, transmission and reception of the signals. The *RobVisR.prg* program at the controller performs the command reception efficiently, allowing a precise pick-and-place operation. Two weak points are identified: some positioning errors were seen if the reference is poorly done by the operator; in the same way, orientation errors occur when the image capture is excessively noisy, changing the results of the analysis and causing a bad positioning of the end-effector, 15^{o} over or under the correct position, which occurred at a rate of one per ten picking operations. In future, these problems can be repaired using a more accurate method for referencing, not so dependent of the operator's precision, and another method, as direct methods like measuring the angle between an object's edge and an image capture area axis, to obtain the orientation. Based in the shown results and the proposed objectives, we can consider that:

- The vision system is satisfactory in pick-and-place operations where information about the part's orientation is needed and in different object/background conditions;
- It is easy to setup, as it only needs a CCD camera and a capture board, equipment that can be easily installed in a common PC, where the *RobVis* program for Windows is operated;
- To send the signals, it uses the simple and versatile *IRBCom* device, for parallel ports and I/O boards;
- The manipulating program *RoBVisR* showed a satisfactory performance, but it is customized for ABB robots, with *Rapid* language. It must be rewritten to other equipments that use another language.

6. References

Azuela, J. H. S., 2001, "Notes for the lecture: Invariant Moments Recognition", Instituto Politecnico Nacional, Mexico.

Bayer, F. M., 2004, "Interface de Comunicação Paralela de Baixo Custo para Robô Industrial", Dissertação de Mestrado, Programa de Pós-Graduação em Engenharia Mecânica, UFRGS.

Gonzalez, R. C., Wintz, P., 1977. "Digital Image Processing", Addison-Wesley, Massachusetts.

Gonzalez, R. C., Woods, R. E., 2003. "Processamento de Imagens Digitais", Editora Edgard Blücher, São Paulo.

Kragić, D., Petersson, L., Christensen, H. I., 2002, "Visually Guided Manipulation Tasks", Robotics and Autonomous Systems, Vol. 40, pp. 193-203.

LACTEC, 2004, "LACTEC – Instituto de Tecnologia para o Desenvolvimento", www.lactec.org.br

Soares, A. M. S., Gonçalves, V. D., 2004, "Controle de um Manipulador Robótico em uma Tarefa de *Pick and Place* Auxiliado por um Sistema de Visão", Anais do III Congresso Nacional de Engenharia Mecânica, Belém-PA, Brasil.

Niku, S. B., 2001, **"**Introduction to Robotics: Analysis, Systems, Applications**"**, Prentice Hall, Upper Saddle River.

Seitz, M., 1999. "Toward Autonomous Robotic Servicing: Using an Integrated Hand-Arm-Eye system for Manipulating Unknown Objects", Robotics and Autonomous Systems, vol. 26, pp. 23-42.

Shi, H., 1997, "Two-Image Template operations for Binary Image Processing", Journal of Mathematical Imaging and Vision, Vol. 3, pp. 269-274.

Wilson, A., 1984, "Solid-State Camera Design and Aplication", Machine Design, April 1984, pp. 38-46.

7. Responsibility notice

The authors are the only responsible for the printed material included in this paper.