Security increasing in Industrial robotics based on artificial vision

José Lima, José Gonçalves, Paulo Leitão
Instituto Politécnico de Bragança
Quinta de Stª Apolónia 1134, 5301-857
Bragança, Portugal
{jllima,goncalves,pleitao}@ipb.pt

ABSTRACT

This paper describes a vision validation system allowing security increasing in industrial robotics applications using, as case study, a playful application: the chess robot system. This system allows remote users to play chess, using a six axes anthropomorphic robot to move chess pieces in the chessboard on getting commands from the player and from the application chess engine. In this work, the mechanism to detect pieces in the chessboard is based in artificial vision due to its hardware simplicity. The developed vision software implements a positioning calibration allowing a high degree of system flexibility: positioning of the chessboard, and its distance to the camera. The calibration of colours is also possible in order to change the chessboard and pieces colours without interference. In this paper are presented and discussed some image processing algorithms and the future work.

INTRODUCTION

The automation laboratory of the Escola Superior de Tecnologia e Gestão, from the Instituto Politécnico de Bragança, possess a set of industrial automation systems and technologies, aiming the apprehension, learning and training of skills in the automation scientific domain, both included in research activities and in the curricula of Electrical Engineering, Mechanical Engineering, Computer Engineering and Industrial Engineering courses. One of the available automation systems in the laboratory is an industrial anthropomorphic robot ABB IRB 1400. Industrial robots allow executing repetitive operations normally performed by human operators. Aiming to demonstrate the potentialities of the automation systems and technologies, namely robotics and artificial vision, it was initiated an internal project to develop a chess robot system. This project allowed to run an experience in automation education applying the concept of 'learning by doing', involving the integration of multi-disciplinary skills and teams. Similar initiatives are reported in the literature, such as Eric, a Chess playing Robot [2] developed at Department of Mechanical Engineering of University of Coimbra, and ChessRobot.net developed at Institute of Production Engineering of Tampere University of Technology [1]. The chess robot system described in this paper, and partially presented at the Robotics Portuguese Open 2004 - Scientific Meeting [4], allows remote users to play chess (playing with the white chess pieces). The chess robot system uses the six axes anthropomorphic robot ABB IRB 1400, which holds an electro-pneumatic gripper with parallel fingers that pick and place chess pieces, avoiding that the robot moves one chess piece to a not empty position. The robot is able to move chess pieces in the chessboard on getting commands from the player and from the application chess engine. The paper is organized as follows: initially is described the system architecture, following is presented the vision based validation system which allows the security increasing: prevents robot collisions. Finally, the last section rounds up the paper with conclusions and presents the future work.

SYSTEM ARCHITECTURE

The chess robot system comprises the following main modules, as illustrated in the Figure 1: chessboard, robot, software application and artificial vision.

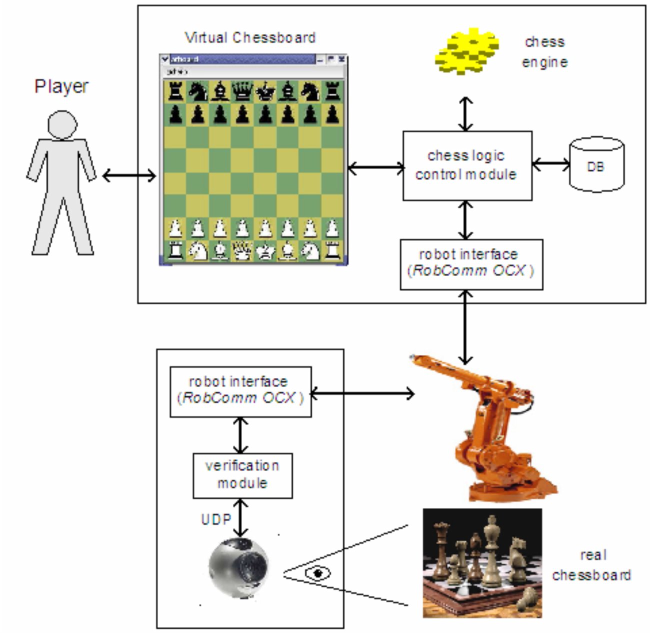

Figure 1 - Modules of the chess robot system

Figure 2 - Robot moving chess pieces

The software application, which acts as the brain of the system, comprises the interface with the player, the chess logic control module, the chess engine, and the robot interface. The virtual chessboard gives an easy interface to the player, by displaying the chessboard of the chess game. On the virtual chessboard, the player drags and drops a white chess piece. The command associated to this movement is delivered to the chess logic control module that forwards it to the chess engine evaluating the move. If the move is valid, the chess logic control module calculates the robot parameters for the move and the chess logic control module sends movement commands to the robot, which reflects the move requested by the player, or the move calculated by the system after the invocation of the chess engine, as illustrated in Figure 2. The communication between the control module and the industrial robot is performed through the robot interface module by ethernet.

The integration of artificial vision in the chess robot system allows implementing a validation mechanism. Once the robot receives the move to perform, the validation mechanism verifies if the target position of the chess piece is free, asking the vision system. In affirmative case, the verification module gives authorization to the robot to move the chess piece, through the robot interface module.

VISION SYSTEM

The detection of chess pieces inside the chessboard can be done using several approaches, such as electrical capacitance, switches, or artificial vision. The last one is easier to setup not only due to the simplicity of hardware but also because it allows building a robust system. Otherwise, it would be necessary to use 64 sensors as well as a complex electric circuit, drawing on to failures. In this work, the mechanism to detect pieces in the chessboard is based in artificial vision (see the screenshot of Figure 3).The developed system comprises an USB web-cam placed above the chessboard. The developed vision software implements a positioning calibration allowing a high degree of system flexibility in terms of size and positioning of the chessboard, as well as its distance to the camera. The calibration of colours is also possible in order to change the chessboard or pieces colours without interference.

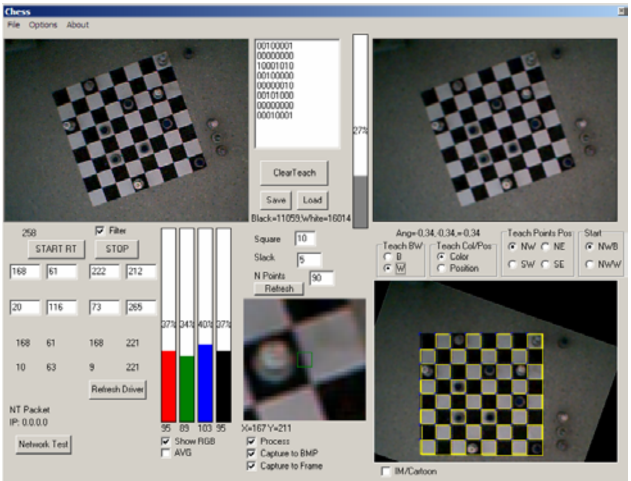

Figure 3 - Vision system Screenshot

In the next sections it will be described the details related to the implementation of the vision system, namely the method of noise filtering supported by a Gaussian filter, the positioning and the colour calibrations.

Noise Filtering

During the image acquisition, some random noise is introduced by the acquisition system into the real image. In order to filter it, a Gaussian filter [3] is implemented using a 2D convolution that can be written as the following finite sum (1), where *Y* is the resulting image and *H* is the convolution kernel.

$$Y(m,n) = a(m,n) \otimes H(m,n) = \sum_{i=0}^{I-1}\sum_{j=0}^{J-1} H(i,j) \cdot a(m-i,n-j) \qquad H = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad (1)$$

Chessboard Direction and Orientation

The chessboard positioning must be calibrated by the vision system in order to recognize exactly the 64 squares included in the chessboard. This positioning calibration could be dynamically found out across an extra image processing but, since the chessboard and the camera are motionless, this must be done just once. So, the chessboard corners must be introduced by the user clicking in the corners image. The knowledge of corners enables the system to calculate all square positions. This is true only if the barrel distortion is despised and if the camera is placed parallel to the chessboard [3]. If not, an image of the world map function must be created. The angle of the chessboard, see Figure 4, is needed for the calculation of the square coordinates.

A way to solve this problem is to rotate the image in order to have the chessboard boarders parallel to the image limits (i.e. $\beta = 0^{\circ}$), as shown in Figure 4. The rotation through the β angle, is applied to the image around A' keeping this point unaltered (A' ≡ A). This rotation is based in the equation (2), where the (X_2,Y_2) values are the new coordinates for the (X_1,Y_1) point [3]. The knowledge of *A*, *B*, *C* and *D* coordinates (the four corners of the chessboard after the rotation) allows finding the edge coordinates for each square, as illustrated in Figure 5, that shows all the 64 squares of the chessboard.

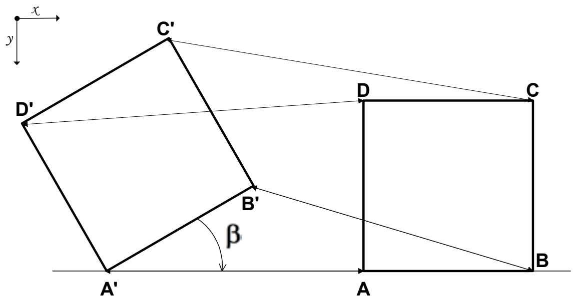

Figure 4 - Square rotation

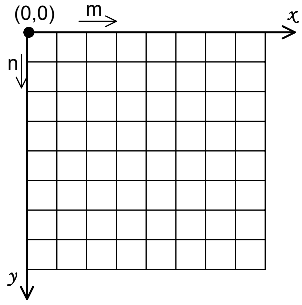

Figure 5 - Chess board axes

$$\begin{pmatrix} X_2 \\ Y_2 \end{pmatrix} = \begin{pmatrix} \cos(\beta) & -\sin(\beta) \\ \sin(\beta) & \cos(\beta) \end{pmatrix} \cdot \begin{pmatrix} X_1 - A_x \\ Y_1 - A_y \end{pmatrix} + \begin{pmatrix} A_x \\ A_y \end{pmatrix} \quad (2)$$

Colours Calibration

The RGB 3D cube is defined from (0, 0, 0) to (2^8R; 2^8G; 2^8B) with *RGB* components in each vertex, as shown in the Figure 6.

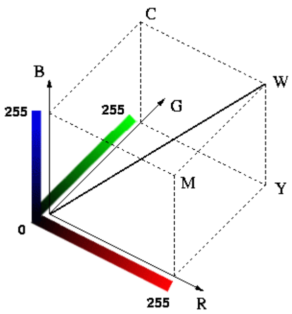

Figure 6 - RGB cube

It is necessary to teach the system of the 2 visible colours in chessboard that will be known as *chessboard colours*. In practice, several cube points [5] are fitted in one chessboard colour. By this way, it is now possible to guess the colour for each square based on its probability. Lightning stability and camera control are very important to this task (image processing). Lens iris must be stilled, otherwise colours calibration is lost.

Decision System

The main objective of the vision system is to detect if any chess piece is inside of a requested square. This can be done applying the probabilities theory. Decision is based on the available information:

(1) The colour arrangement (left upper square colour introduced by user).

(2) The square positions.

(3) The number of matched pixels for each chess colour for each square.

The square areas are scanned and for each one it must be studied the colour of each pixel. A gap of 15% (θ and δ), as shown in the Figure 7, is not considered in order to avoid errors.

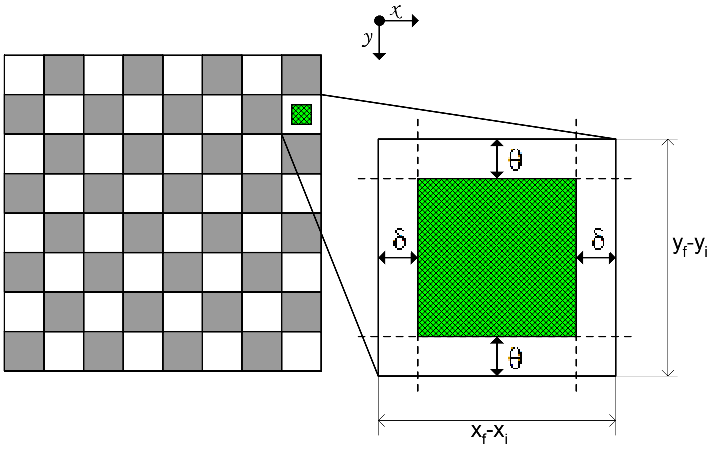

Figure 7 - Square areas scan

The probability to have chess piece inside a square can be found considering the square area and the number of matched pixels of the same colour as the chess colour arrangement. The following expression shows the probability of piece existence.

$$p(ExistingPiece) = 1 - \frac{ColouredMatchedPoints}{area}; \qquad area = \frac{x_f - x_i}{1/(1-2\cdot\delta)} \cdot \frac{y_f - y_i}{1/(1-2\cdot\theta)} \quad (3)$$

CONCLUSIONS AND FUTURE WORK

An important issue associated to the chess robot system is the vision system that allows implementing a validation mechanism, avoiding that the robot moves one chess piece to a not empty position. The described system is implemented in the automation laboratory of the Escola Superior de Tecnologia e Gestão, from the Instituto Politécnico de Bragança and is used to introduce some subjects like industrial robotics and image processing in order to promote student motivation. As future work, we intent develop an auto calibration method in order to reduce operator intervention.

REFERENCES

[1] Lobov, A., E. Lopez, J. Lastra and R. Tuokko (2004). ChessRobot.net: An Interactive Online Experience. In: *Proceedings of the 2nd International Conference on Industrial Informatics*. Berlin, Germany. pp. 64–68.

[2] Pires, N. (1999). Eric, a Chess Playing Robot. http://robotics.dem.uc.pt/norberto/eric/.

[3] Pratt, William K. (1991). *Digital image processing*. John Willey and Sons Inc.

[4] Sousa, F., J. Gonçalves and P. Leitão (2004). Sistema Robotizado de Xadrez. In: *Proceedings of the Scientific Meeting of Robótica 2004*. Porto, Portugal. pp. 163–168.

[5] Jain, Anil K., (1989), *Fundamentals of Digital Image Processing*, Prentice Hall Inc.