An Approach for Integration of Industrial Robot with Vision System and Simulation Software

Ahmed Sh. Khusheef, Ganesh Kothapalli and Majid Tolouei-Rad

Abstract—Utilization of various sensors has made it possible to extend capabilities of industrial robots. Among these are vision sensors that are used for providing visual information to assist robot controllers. This paper presents a method of integrating a vision system and a simulation program with an industrial robot. The vision system is employed to detect a target object and compute its location in the robot environment. Then, the target object's information is sent to the robot controller via parallel communication port. The robot controller uses the extracted object information and the simulation program to control the robot arm for approaching, grasping and relocating the object. This paper presents technical details of system components and describes the methodology used for this integration. It also provides a case study to prove the validity of the methodology developed.

Keywords—industrial robot, integration, simulation, vision system

I. INTRODUCTION

INDUSTRIAL robots are widely used in production lines of modern manufacturing industries particularly where there is a need for high precisions; or where the processing operations are to be performed in hazardous and/or dynamic environments. Spray painting, welding, and pick and place tasks can be mentioned as typical operations in manufacturing environments where industrial robots are widely used. Without making use of appropriate sensors, industrial robots will not be able to perform most of these tasks since environmental conditions constantly change. Vision sensors are one of the common types of sensors used in dynamical work environments. The availability high speed processers and low cost cameras that use little power, is the main reason for the rapid development of image sensor applications [1]. Visual sensors can also extract a relatively large amount of environmental information from an image [2]. As a result, a vision sensor is widely used to provide visual information and to increase the robot's interaction with its environment. Vision system generally consists of the following three parts [3]:

- Image acquisition: includes selection of visual sensor (camera).
- Image processing tools: Includes image processing techniques, and computer program languages such as Matlab and C++ programming.

- Data Bus tools: covers tools that enable the interface connection between the computer and the robot controller such as USB, parallel and serial ports.

Many researchers have built up a robot control system based on a vision system. Zhao et al. [4] developed a new image recognition method which was improved Hough transform to detect and recognize objects on the robot workspace. In this work, industrial manipulator was used to grasp and move an object detected by a vision system. Muhammedali et al. [5] proposed a vision-based control system. They presented an image processing approach that was used to detect the edges of objects and then to find the centroids in the image. However, the integration of this vision system with an industrial robot controller has not been considered.

This paper describes the method of integrating a vision system with the industrial robot. The vision system is used to detect and recognize the target object and then determine its location. The industrial robot is programmed to approach, grasp and relocate a target object that is detected using the vision system. The next sections explain the integration method developed. First, the system hardware architecture is described. This is followed by explanation of camera calibration method. Then, the software concept is described, and finally the experimental results are presented.

II. SYSTEM HARDWARE ARCHITECTURE

As shown in Figure 1 the hardware architecture of the system consists of an industrial robot equipped with a horizontal conveyor, and a vision system which consists of
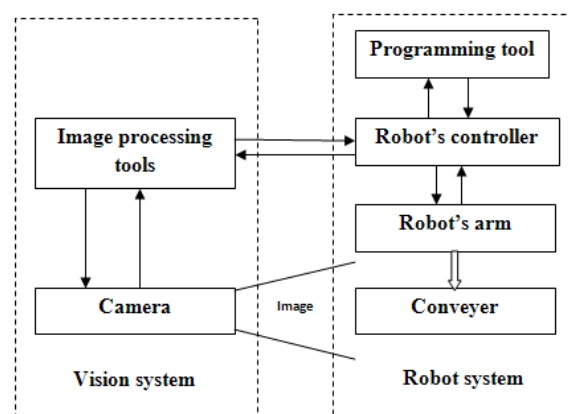
Ahmed Sh. Khusheef is student in Edith Cowan University, Australia (phone: 0402628213; e-mail: ahmedk@ our.ecu.edu.au).

Ganesh Kothapalli is Senior Lecturer in engineering school in Edith Cowan University, Australia (e-mail: g.kothapalli@ecu.edu.au).

Majid Tolouei-Rad is a Senior Lecturer in engineering school in Edith Cowan University, Australia (e-mail: m.rad@ecu.edu.au).

Fig. 1 System hardware architecture image processing tools and a video camera

A. Industrial Robot Configuration

As shown in Figure 2 the industrial robot used in this project is an elbow Lab-Volt robot Model 5250 with five rotational axes with servo actuators: base, shoulder, elbow, wrist, and gripper. The base supports the arm mechanism and it houses the motor that provides shoulder rotation. The shoulder houses the motors that move the other sections of the arm. The mechanism that move the forearm, wrist and gripper are carried in the lower end of the upper arm. The upper arm and forearm have up and down motions. The wrist, that can rotate 360-degree clockwise and counterclockwise directions, moves up and down to control the gripper location and direction. The two-fingered parallel jaw gripper is attached to the arm of the robot to enable it to grasp and move materials, parts and tools [6]. The controller has 16 inputs and outputs TTL channels which can be used to integrate the controller with external subsystems as shown in Figure 3.
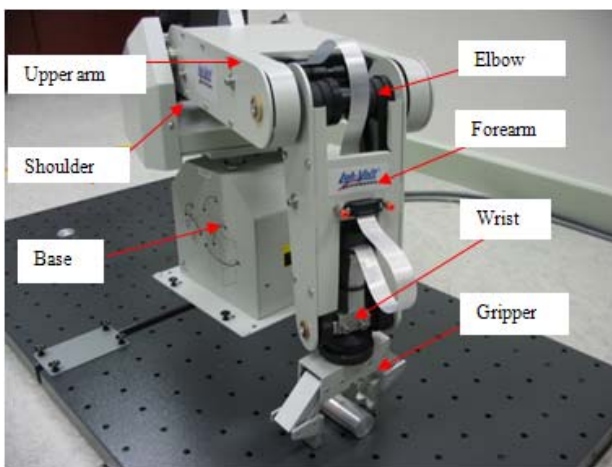
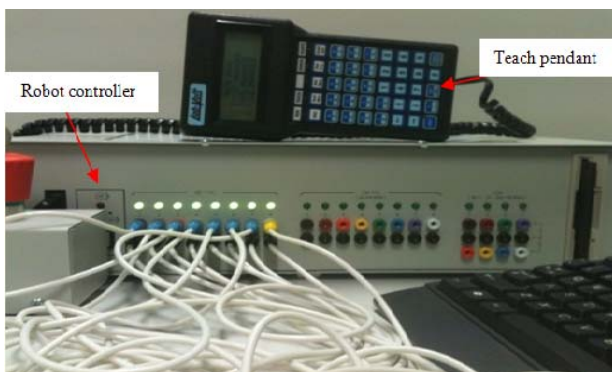

Fig. 2 The elbow Lab-Volt robot Model 5250

Fig. 3 The robot controller and the teach pendant

This robot can be programmed by using either a hand-held terminal (teach pendant) or a RoboCIM simulation software. The teach pendant is used to direct control of the robot articulation. Objects' locations are defined by recording points and using these points to edit a robot program which controls robot's motion. The RoboCIM simulation software involves all controlled commands that help the robot to perform its tasks. It allows creating a program, performing a test in simulation mode, and then performing a test in the control mode with actual robot. Similar to the teach pendant, the software is used to record points and then use them to edit a program. In this case, the robot carries out repetitive operations. Therefore, if the object's location in the robot workspace is changed, the robot will not be able to grasp it unless the recorded points are modified accordingly. Figure 4 shows the robot's software workspace with a number of recorded points.
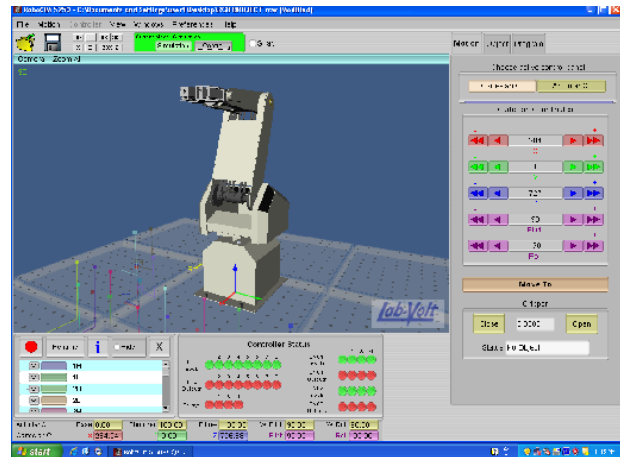

Fig. 4 Robot's workspace in the simulation mode

Various types of conveyors are used in automated manufacturing systems for automatic transfer of materials and parts often along fixed paths. In this project a Lab-Volt belt conveyor model 5118 with an electrical power source is used for moving objects. This conveyor has been integrated with robot controller via TTL input and output ports.

B. Vision system configuration

As mentioned before the key requirements of machine vision system are image sensors, image processing tools, and data bus tools. In this project a webcam model C200 has been used as the image sensor. It takes images with a resolution of up to 640 by 480 pixels and the maximum rate of image capture is 30 frames per second. It has been connected to the host computer by means of a high-speed USB port. In robotic environments there are two options to position the camera with respect to the robot. The camera can be either placed at a fixed position and remain stationary while the robot moves; or it can be mounted on the robot arm that moves with it. In this project, the webcam has been placed at a fixed position where it has with a good view of robot's workspace. In this scenario, the relationship between the camera coordinate frame and workspace coordinate frame is fixed and does not change when the robot moves. Therefore, each point in the camera's image plane refers to a fixed position in the real world. MATLAB and C++ program library are installed in the host computer for image processing purposes. MATLAB library for

image processing can be directly used to analyze image contents to obtain object information; while C++ should be used in conjunction with OpenCV software [7] in order to process the image. Image processing techniques that are used in this project include Speeded Up Robust Features (SURF) [8], template matching, and colour segmentation. Data bus tools should be connected to the host computer to be able to communicate with other subsystems. In this project the parallel and serial ports of the host computer have been used to send digital signals to the robot controller. The parallel port of the host computer has 25 pins, as shown in Figures 5(a) and 5(b). Only sixteen pins can be used for sending signals to the robot controller. Pins 2 to 9 (Figures 5(b)) are used as output pins and pins 18 to 25 as ground pins. As shown in Figure 5(a) the serial port has 9 pins which transmit 8 bits as 1 bit at a time. The signals that are sent by these communication ports can be directly controlled by the MATLAB and C++ software.
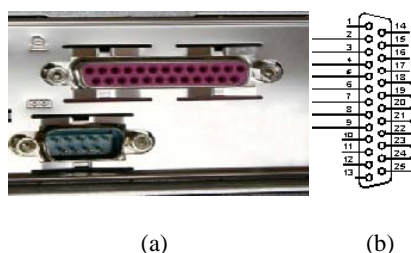

(a)   (b)

Fig. 5 (a) Serial and parallel ports, (b) Parallel port

III. CAMERA CALIBRATION

As mentioned the camera is located in a fix position, and therefore, the relationship between camera coordinate frame and workspace coordinate frame is fixed and does not change when the robot moves. Camera calibration is the process of defining the relationship between coordinates of an object in an image with its actual position in the real world. In this project, the robot controller receives eight digital bits of the binary information and therefore $2^8 = 256$ different values can be achieved. A value of zero ($2^0 = 0$) means no object has been detected in the image and this leaves 255 useable values for defining the position of the object in the robot workspace or on the conveyor. Therefore 255 locations can be pre-defined on the conveyer. Consequently, the robot workspace image is acquired and processed by the MATLAB image processing software in order to determine the exact position of objects in the image.There are two methods for defining pre-determined locations on the conveyor. The first method relies on taking an image of the working environment and processing it MATLAB. In doing so MATLAB's Tool pull-down menu is used and up to 255 arbitrary locations on the conveyor are chosen by simply clicking once on each of these points in the image. The MATLAB does record the coordinates of these points. This is shown in Figure 6. In the second method a number of objects are placed in desired positions and the image of these objects on the conveyor is taken and then processed in MATLAB such that the centre coordinates of the objects are computed and recorded. This is shown in Figure 7. After completion of the calibration process; the relationships

between x-y coordinates of 255 locations in the real world with u-v coordinates of 255 points on the image plane have been established. The coordinates of the recorded points in the image have been used to develop a program that produces signals to be sent to robot controller. It should be noted that that u-v image plane coordinates of objects are mapped into x-y coordinates of object locations in the real world that will be used by robot controller. Table 1 represents x-y-z coordinates of eight sample points on the conveyor and their corresponding u-v coordinates on the image plane. It is noteworthy that x and z coordinates of all the points are identical as all the objects are positioned on the same height
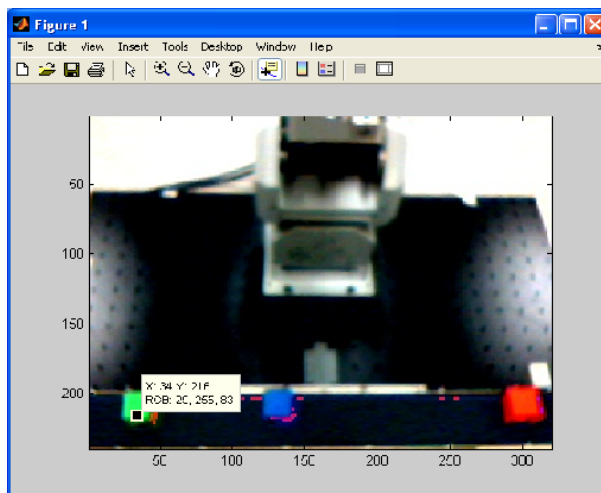

Fig. 6 First method of calibration on the image taken by webcam.

(on the z-axis) on and aligned along y-axis on the conveyor. Accordingly the u-coordinates of these points that correspond to x-values in the real world coordinates are also identical.

TABLE I
CALIBRATION PROCESS FOR EIGHT LOCATIONS

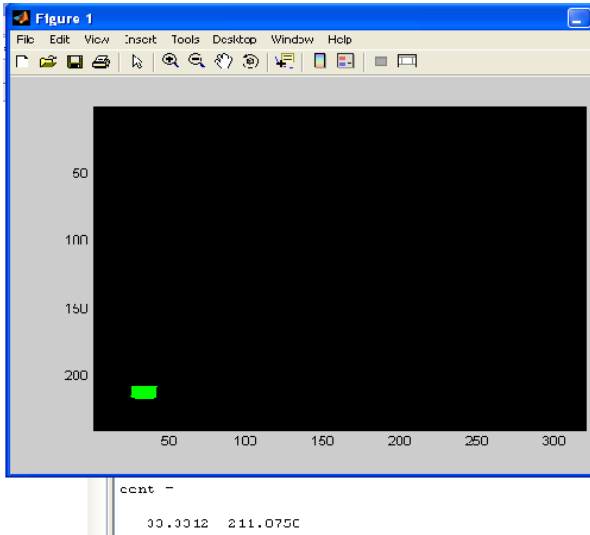| Location	Vision u	Vision v	World x	World y	World z
1	214	302	416	369	115
2	214	254	416	231	115
3	214	211	416	127	115
4	214	171	416	24	115
5	214	131	416	-84	115
6	214	90	416	-200	115
7	214	61	416	-275	115
8	214	32	416	-348	115

Fig. 7 Second method of calibration on the image taken by webcam

IV. SOFTWARE DEVELOPMENT

Figure 8 shows the flowchart that describes the process of robot control based on the vision system. In this work, image processing is performed using Matlab and C++ program with OpenCV. However, it should be noted that the robot is programmed using the RoboCIM software.
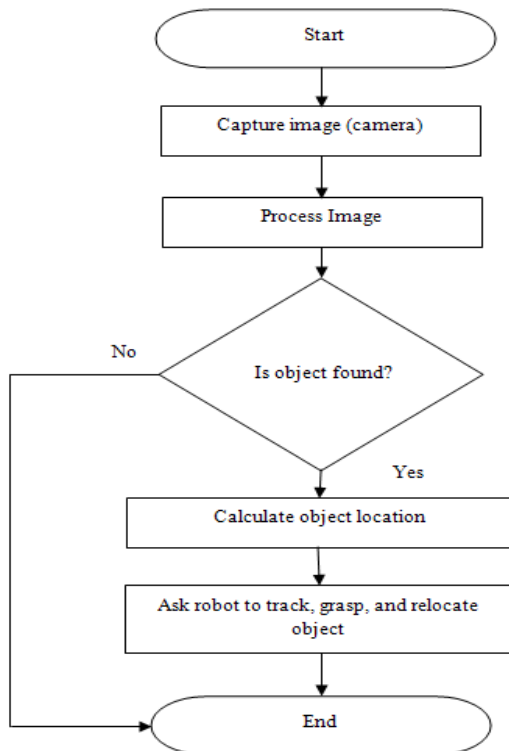

Fig. 8 Software flowchart

A. Image Processing and Object Detection

The vision system is used to acquire and analyze images to detect objects and compute their location information. In this work different image processing techniques have been tested in order to achieve optimal results. These include template matching, SURF, and colour segmentation where the main objective has been to detect and recognize the object, and to compute its centre coordinates.

Template matching [9] is a technique in digital image processing for finding small parts of an image which match a template image. It should be noted that template matching will be affected by object's distance which changes the size of the object in the image. This problem has been overcome by using different scales for object templates. Figure 9 shows template matching results.
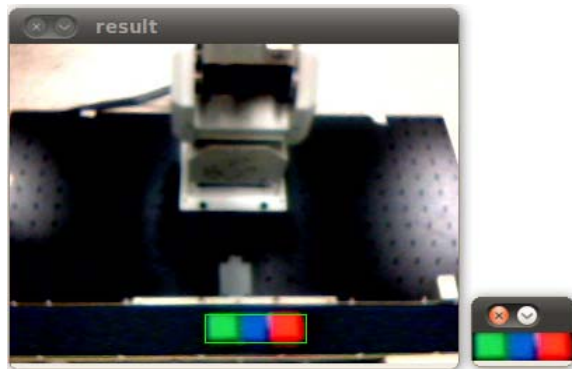

Fig. 9 Object detection by template matching

It has been noticed that template matching method was extremely slow in real-time image processing. Consequently, the SURF technique has also been tested. It is a robust image detection and description method used in computer vision systems for object recognition. This method has proved to be quicker than the template matching. Colour segmentation technique has also been used for detecting objects by making use of its colour. In this process RGB (red, green and blue colour space) and HSI (Hue, Saturation and Intensity colour space) techniques have been revised and used. It was concluded that HSI technique is less sensitive to changes in light intensities, and also it is easier to find the object by its colour. However, the output of camera is in RGB colour space. As a result, the colour space transformation between RGB and HSI affects real-time behaviour. Figure 10 shows colour segmentation results by using RGB colour space. After colour segmentation process and detection of the object its area and centre coordinates are determined by using the technique iproposed by Qingjie et al. [10] as explained below:

$$M_{p,q} = \int \int x^p \ y^q \ f(x,y) \ dxdy \qquad (1)$$

where the $M_{p,q}$ represents the moments with the rank (p, q); f(x,y) represent a continuous 2D image (in binary image it is 1 when pixel follows the detected object otherwise it is 0); the area of object in the image A is represented by $M_{0,0}$; and the centre coordinates of object in the image are represented by $X=M_{1,0}/M_{0,0}$ and $Y=M_{0,1}/M_{0,0}$.

After the area and centre coordinates of objects in image are calculated, resulting values are used to send signals to robot controller.
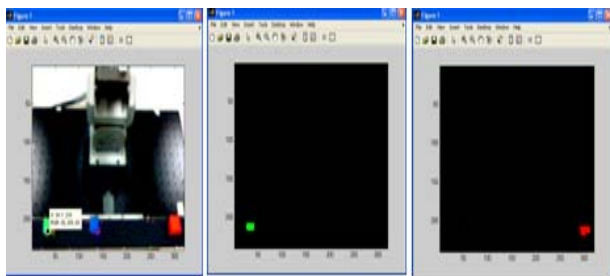

Fig. 10 Object detection by the colour segmentation method

B. Communication Program

Object coordinates computed in the previous step are used to develop a program to generate and send digital signals to robot controller via the parallel port. The values of digital signal represent object's centre coordinates. There are 255 values corresponding to 255 pre-defined positions in the robot environment. Table 2 shows the pre-defined locations numbers and the corresponding digital signal values.

TABLE II
PRE-DEFINED LOCATIONS NUMBERS AND CORRESPONDING DIGITAL SIGNAL VALUES

Object location	Signal values
No object	0000 0000
1	0000 0001
2	0000 0010
3	0000 0011
4	0000 0100
252	1111 1100
253	1111 1101
254	1111 1110
255	1111 1111

C. Robot Program

The robot simulation software program enables the robot to perform repetitive tasks. When equipped with a vision system it becomes possible for the controller to adopt the activities of the robot with the requirements of its working environment. As mentioned, robot controller receives eight digital signals via TTL input channels. As a result 255 different locations on the conveyor can be defined in the robot program. Since RoboCIM simulation software can interact with its environment via TTL input and output channels, the locations of objects will be introduced to the software by use of these TTL input channels. This would enable the robot to approach, grasp and relocate detected objects.

V. CASE STUDY

Three objects in different colours were used to test the system developed. They were in green, red and blue. These objects were positioned randomly on three of the many pre-defined locations on the conveyor as shown in Figure 11(a). The process was started with taking an image from the working environment and sending it to the host computer for image processing. Then the operator entered the desired colour of the object, green in this case, and the image processing function was executed and the object in desired colour was recognized in the image and its center coordinates were computed therefore, the green object's location in the image was determined as shown in Figure 11(b). Then, the appropriate digital signals, in this case 00000001, were sent to the robot controller. Consequently, the robot performed approaching, grasping and relocating of the green object as shown in Figures 11(c) and 11(d). The process was then repeated, this time for the red object, and then for the blue as shown in Figure 11(e). All the steps were executed successfully proving effectiveness of the methodology used.
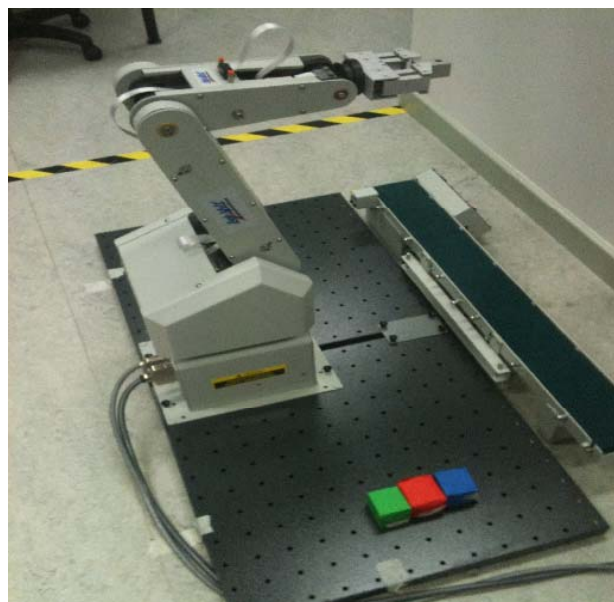

(a)

(b)
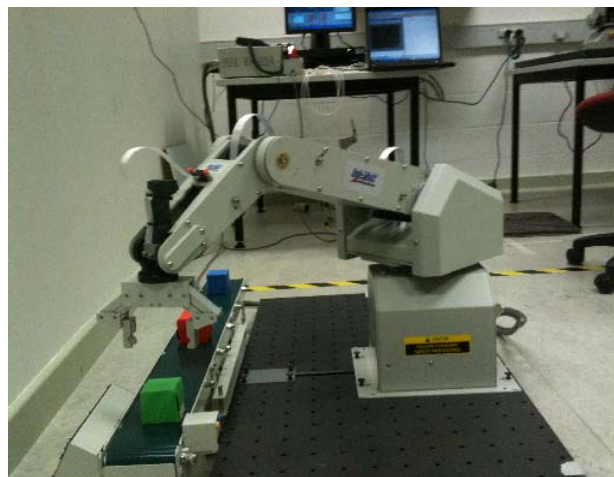
(c)

(d)

(e)

Fig. 11 The industrial robot experiment (continued)

VI. CONCLUSION

The experiments conducted prove the validity of developed methodology for the integration of the simulation software and the vision system used. It has also been concluded that when an object in a single colour is used then optimal results are achieved by using the colour segmentation method. However, the template matching and SURF methods achieve better results when multi-coloured objects are used. Pick and place operations, packaging, processing, and quality control tasks can be mentioned as a number of potential applications areas for the system developed. As a future work extension of this project, it is intended to improve the vision system and the system to deal with moving objects in the workspace. In this case, the robot motion speed relative to object speed needs to be taken into consideration.

REFERENCES

[1] W. Dae Hee, L. Young Jae, S. Sangkyung, and K. Taesam, "Integration of vision based SLAM and nonlinear filter for simple mobile robot navigation," in Aerospace and Electronics Conference. 2008 NAECON IEEE National, pp. 373-378.

[2] M. Marron, M. A. Sotelo, J. C. Garcia, D. Fernandez, and I. Parra, "3D-visual detection of multiple objects and structural features in complex and dynamic indoor environments," in IEEE Industrial Electronics. 32nd Annual Conference on 2006 IECON, pp. 3373-3378.

[3] J. Wang, T. I. Niekerk, D. G. Hattingh, and T. Hua, "Knowledge-based robot vision system for automated part handling," South African (2008) Journal of Industrial Engineering, 19(1), 119-119-130.

[4] S. Zhao, C. Chen, C. Liu, and M. Liu, "Algorithm of location of chess-robot system based on computer vision," Control and Decision Conference. 2008 CCDC.

[5] B. Muhammedali, M. Z. Abdullah, and M. N. M. Azemi, "Food handling and packaging using computer vision and robot." Computer Graphics, Imaging and Visualization, 2004,. Proceedings. International Conference.

[6] Lab-Volt Ltd., Introduction to robotics manual, 1st edition ISBN 2-89289-721-1, Canada, 2004, pp. 1-2–1-3.

[7] Intel Inc., the OpenCV computer vision library. Available in: http://www.intel.com/research/mrl/research/opencv

[8] OpenSURF. Chris Evans development. Available in: http://www.chrisevansdev.com/computer-vision-opensurf.html

[9] R. Brunelli, Template Matching Techniques in Computer Vision: Theory and Practice, Wiley, ISBN 978-0-470-51706-2, 2009.

[10] Z. Qingjie, D. Hongbin, Z. Wenyao, and M. Aixia, "Selection of image features for robot vision system," in Automation and Logistics. 2007 IEEE International Conference, pp. 2622-2626.

Ahmed Sh. Khusheef is a postgraduate student at the School of Engineering, Edith Cowan University (ECU) in Perth, Australia. He received his BSc in Mechatronics Engineering from the University of Baghdad, Iraq in 2001.

Ganesh Kothapalli is a Senior Lecturer at the School of Engineering, Edith Cowan University (ECU) in Perth, Australia. He received his Masters degree in Electrical Engineering from the University of Alberta (Canada) and a PhD from the University of New South Wales Australia. He held academic positions at Monash University, the University of Ulm and ECU with significant contributions to teaching, research and research supervision. His research interests include computer-aided design, control and renewable energy systems.

Majid Tolouei-Rad is a Senior Lecturer at the School of Engineering, Edith Cowan University (ECU) in Perth, Australia. Since obtaining a PhD in Mechanical Engineering from the University of South Australia he has been extensively involved in tertiary teaching and research and supervising research students. Prior to joining ECU he held academic positions at reputable universities in Iran and Canada. His research interests include computer-integrated manufacturing, robotics and automation, and materials and manufacturing.