AUTOMATED BOLT PICKING AND SORTING SYSTEM (ABPSS)

Ahmed Elsaharti
Department of Mechanical Engineering, College of Engineering and Mines
University of North Dakota
Grand Forks, ND, 58202, USA.
ahmed.elsaharti@und.edu

Ramtin Kardan
Department of Mechanical Engineering, College of Engineering and Mines
University of North Dakota
Grand Forks, ND, 58202, USA.
ramtin.kardan@und.edu

***Abstract*— The hand-operated bolt size recognition, sort, and arranging is a labor-intensive and tedious task. Consequently, automation of item segregation can enhance efficiency and save cost for any businesses. This paper proposed automatic robot segregation and arrangement methodology that reduces the costs for manual separation of bolts. The hardware of the aimed system comprises a lightweight robot manipulator, a low-cost commercial camera or webcam and a custom-built robotic gripper. The software to be used with the proposed system is MathWorks' MATLAB for image processing and SIMULINK.**

Keywords—Robot Control, Image Analysis, Automation

I. INTRODUCTION

In recent years, the employment of robots in the industry been changing the processing in almost every field. As industries grow, they tend to move towards automation of processes in order to cut down expenses in the form of paid labor. The balance between cutting expenses and maintaining consistent/repeatable quality is usually provided by the implementation of robots which provide high repeatability with remarkable precision. Robot executes different task repeatedly with high precision, thereby many functions like collecting information and studies about the hazardous sites where itis too risky to send human inside [1]. One of the most famous examples of the robots used in the industry is sorting robots that help sort objects or place them in production/packaging lines. These robots are usually coupled with image processing units to automate the entire process which effectively results in the reduction of the labor cost. This, in turn, ends up helping any industrial process grow in scale.

Techniques from statistical pattern recognition have, since the growth of neural networks, become popular in digital image PROCESSING [2].

In this project, different methods of image processing been studied and only the most suitable method discussed explicitly.

This paper is structured as follows: Section II describes the learning environment, Section III the materials and methods used, Section IV the discussion and results, Section V the final conclusions and VI is the References.

II. LEARNING ENVIRONMENT

This paper describes a project-based learning case example developed at the University of North Dakota, USA. This project was addressed to graduate students who involved in course 439, introduction to Robotics and combines knowledge from computer vision and robotic control to complete an automation project task. The learning target was to develop a computer vision system that must detect objects which placed randomly on a target surface and control a robotic arm to pick them up and moves them one-by-one to a predefined destination and sort them based on the object area and predefined orientation.

A. MATLAB Programming

The MATLAB software by Mathworks is widely operated at academic environments and universities, [3] supporting data analysis, the development of algorithms and models, and the development of applications by using a high-level programming language. The objective was to develop a complex but compact set of codes where could be run and process the image in the shortest period of time.

B. Image Processing

To apply standard signal processing techniques in images [4]. These techniques cover image processing algorithms such as binarizing, thresholding, filling, etc., and also image analysis procedures applied after segmentation and morphological filtering, such as size, position, orientation, camera distortion and distance between the image and real-world magnitudes.

III. MATERIALS AND EXPERIMENTAL SETUP

Figure 1 shows the automation cell setup for the development of the proposed topic. The most important elements are a webcam, a robotic arm, a target area or defined workspace, different bolt in term of size as the target. The camera was connected to the Windows 10 with MATLAB 2019 and the robotic arm had access with MATLAB 2008 through a Windows XP computer. The most important methods are the MATLAB functions required to obtain the image acquired with the webcam, forward the information from MATLAB 2019 to MATLAB 2008 and to control the robotic arm.

Figure 1 CRS Robot and Workspace

A. CRS CataLyst-5 Articulated Robotic Arm

The robotic arm used in this paper is the CRS- Catalyst 5 (see Fig. 2) whose technical specifications and operational data are available in [5].

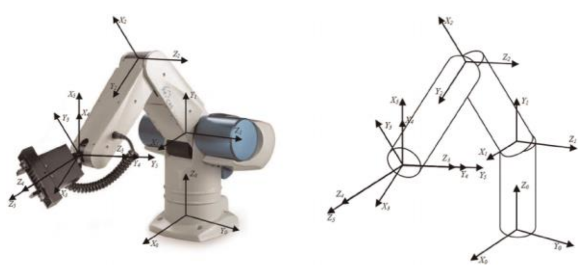

Figure 2 The CRS CataLyst-5 manipulator its schematic and link coordinate frames and D-H

The Control System used is the CRS Catalyst 5 controller packaged with real CataLyst-5 manipulators. The controller can be used to configure parameters for each joint, including control gains, encoder counts, and the physical characteristics of each motor. Using either the software provided by the manufacturer or by communicating with the controller directly through TCP/IP from the User Interface PC it is possible bring the robot on and off-line, home the joints, command positions and trajectories, and perform various other tasks, all as if working with the real robot. The controller communicates directly with the drive boards for each motor and synchronizes commands and motion during operation.

B. HD Webcam C310

The HD Webcam C310 used in the automation cell to obtain images from the target area is a low-cost Logitech WEBCAM (see Fig. 3). This camera has an original resolution of 1280 x 720 pixels, one color per pixel and automatic focusing, HD record video size and a standard USB 2.0 port connection. The default image format and size of this camera is RGB, 24-bit color depth.

Figure 3 HD Webcam C310 Logitech

C. Experimental setup

A CRS CataLyst-5 Articulated Robotic Arm, HD Webcam C310, and two desktop computers used in this project. The experimental setup is shown in fig 4.

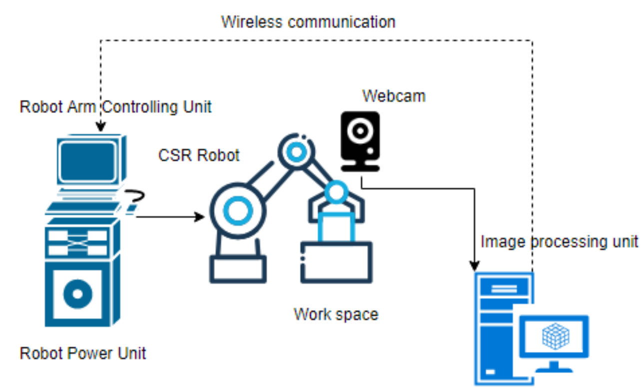

Figure 4 Hardware Description and Experimental Setup

IV. DISCUSSION AND RESULTS

A. Camera Calibration

MATLAB's USB webcam toolbox was used in parallel with the image acquisition toolbox to extract a picture of the workspace. This was taken using a webcam attached to the robot in its permanently set home position. Camera calibration is the process of estimating the parameters of the lens and the image sensor. These parameters are needed to measure objects captured by the camera. This example shows how to calibrate a camera programmatically. Alternatively, you can calibrate a camera using the cameraCalibrator app. To calibrate the

camera, we first need to take multiple images of a calibration pattern from different angles. A typical calibration pattern is an asymmetric checkerboard, where one side contains an even number of squares, both black and white, and the other contains an odd number of squares. The pattern must be affixed to a flat surface, and it should be at approximately the same distance from the camera as the objects you want to measure. The size of a square must be measured in world units, for example, millimeters, as precisely as possible. In this example, we use 9 images of the pattern, but in practice, it is recommended to use 10 to 20 images for accurate calibration.
Steps:

- Prepare Calibration Images
- Estimate the Camera Parameters
- Read the Image of Objects to Be Measured
- Undistort the Image

B. *Image processing*

The purpose of processing image in this project was to segment the image and recognize the parameters of objects on the workspace.
A MATLAB function was built around Rashi Agrawal's youtube tutorial on image segmentation[6] to segment the image, extract objects, label them and obtain their centroids in image coordinates. The code is attached to the following link: (https://drive.google.com/open?id=1zIZMdwbdnHknrsvgS56aPIuA3K9SGYho).
Hu moments are being used to determine the orientations of the objects and their areas for size-based sorting. In order to determine where the 'heavy' section of the object is and obtain the objects' orientation using image moments.

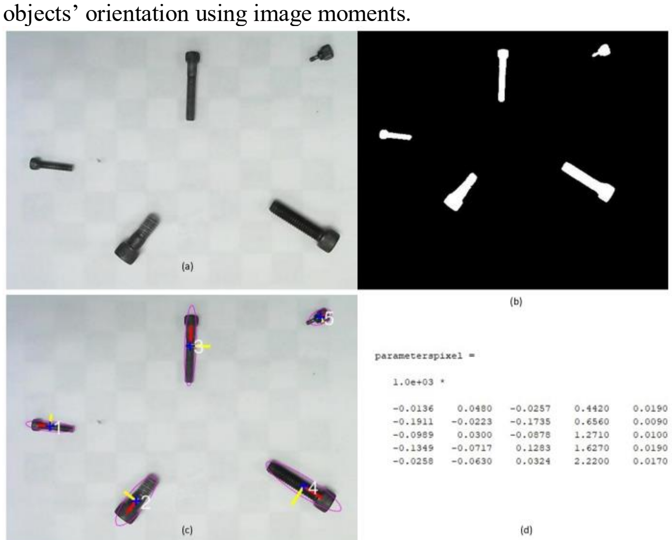

Figure 5 image processing steps: (a) Original Image (b)Bolts are recognized using black and white techniques. (c) Objects are recolonized and orientation were added on the original image (d) The object parameters exported from computer 1 to be sent to computer 2

C. *MATlab Comminucation*

As discussed earlier the location convention of objects from image coordinates to real-world coordinates necessitated being sent to the robot controlling PC. MATLAB's computer vision toolbox is being employed to try calibrating the used webcam using checker box prints. The ready code modified in order to accomplish what we demanded from it. Please refer to: (https://drive.google.com/open?id=1zIZMdwbdnHknrsvgS56aPIuA3K9SGYho).

D. *Simulink*

Robot support package had predefined orders to control the arm. The link between MATLAB and the robot was established using the Quanser SIMULINK library which was found on the console PC. Controlling the robot arm using MATLAB was one of the biggest problems in this project. However, some slight mathematic based modification in predefined command provides us with desired motion.

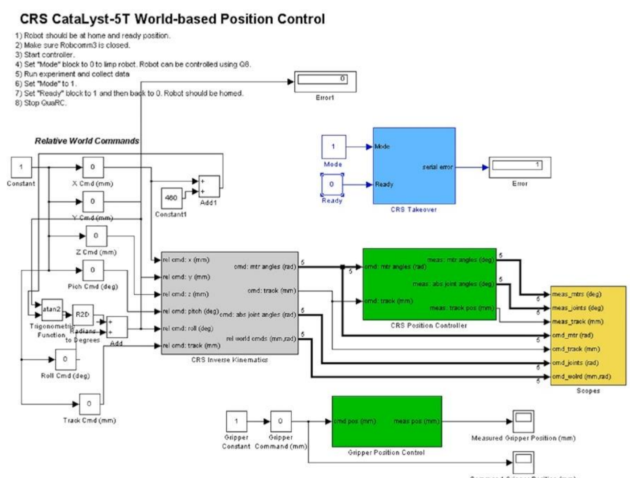

Figure 6 Proposed Simulink model to achieve the desired motion

E. *Results*

There were many challenges prior to code and run the experiment. Learning about the CRS robot, controlling the robot arm and set the communication link between both computers.
The experiment itself had different sections which were camera calibration, image recognition, image moments and data conversion from the processed image into the form that robot arm can follow the process the orders.

V. CONCLUSIONS

Regardless of the size and number of bolts, they were recognized, picked and sorted based on their dimensions and orientation. There were two modes defined in this project which considered the dark objects on the bright background or otherwise. The efficiency and persistence of the purposed algorithm are being tested. The video on the functionality of the project is in the following links: https://youtu.be/9zQwnlUxm6E
or https://youtu.be/iJnEU_IVeAI.

VI. REFERENCES

[1] S. Shyam and R. Nair, "Design of a Voice Controlled Robotic Arm for Picking and Placing an," *IOSR Journal of Engineering,* vol. 2, pp. 670-673, 2012.

[2] J. Cornfield, "Statistical classification methods," in *Proceedings of the Second Conference on the Diagnostic Process, Computer Diagnosis and Diagnostic Methods, Chicago*, 1972, pp. 108-130.

[3] L. R. Soares and V. H. C. Alcalde, "An educational robotic workstation based on the rhino XR4 robot," in *Proceedings. Frontiers in Education. 36th Annual Conference*, 2006, pp. 7-12.

[4] C. Solomon and T. Breckon, *Fundamentals of Digital Image Processing: A practical approach with examples in Matlab*: John Wiley & Sons, 2011.

[5] T. E. Corp. (2010). *CRS CataLyst-5 Articulated Robot[Online]*. Available: http://www.robotsdotcom.com/images/cat.pdf

[6] R. Agrawal. (2015). *Labeling of objects in an image using segmentation in Matlab*. Available: https://www.youtube.com/watch?v=XrC1r80-p-k