Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 133 (2018) 205–212

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Robotics and Smart Manufacturing (RoSMa2018)

Vision-based Robot Manipulator for Industrial Applications

Md. Hazrat Ali*, Aizat K., Yerkhan K., Zhandos T. and Anuar O.

[a]*School of Engineering, Nazarbayev University, Astana, 010000, Kazakhstan*

Abstract: This paper presents a multi-stage process of the development of a vision-based object sorting robot manipulator for industrial applications. The main aim of this research is to integrate vision system with the existing Scorbot in order to widen the capability of the integrated camera-robot system in industrial applications. Modern industrial robot Scorbot-ER 9 Pro is the focus of this research. Currently, the robot does not have an integrated vision system. Thus; a camera has been integrated to the robot gripper to achieve the target objectives. The main difficulties include establishing a relevant sequence of operations, developing a proper communication between camera and robot as well as the integration of the system components such as Matlab, Visual Basics, and Scorbase.

Keywords: Robot; vision; image processing; programming.

1. Introduction

This research deals with a robot manipulator for real-time, fast and accurate industrial applications. The robotic arm consists of parts that are joined together and can be preprogrammed or controlled manually to perform a mechanical motion. The parts of the robotic arm have functions that are analogous to human arm features and are named as shoulder, elbow, base, etc. In order to control the robot, a specific Advance Control Language (ACL) controller is used. This controller is combined with software on PC through USB connection. So, with the help of this software, a control program can be written which is in turn sent to the controller for further manipulation of the robot. Initially, Scorbot-ER 9 Pro can be controlled only manually but this research and development were done in terms of addition of vision source that leads to automatic operation of the robot. For this purpose, software programs such as Matlab, Visual Basics, and Scorbase are used. There are several steps to be implemented before the robot can move and perform operations. The initial stage is an image processing of the objects which are carried out through Matlab coding with a camera connected to the PC. During this step, digital information about the positions of the objects is obtained. During the next stages, this data is processed through Visual Basics environment and used

* Corresponding author. Tel.: +77172706145
E-mail address: md.ali@nu.edu.kz

10.1016/j.procs.2018.07.025

by Scorbase software connected to the robot. Following all the sequences results in excellent performance of pick and place operation of the robot manipulator by launching a single script.

2. Literature review

Robotic arms have gained a wide range of use in the industrial applications. It could replace human labor in trivial tasks that can be characterized as repetitive or requiring a lot of lift force. Human labor in the production line is subject to fatigue and robotic units replace them having high repeatability and reliability [1]. A number of companies had cut amount of human workers replacing them with automated systems [2].Robot arm kinematics [3] can be described as the transformation of coordinates (x,y,z) in the Cartesian system to the system of joints in the robot with coordinates of angle (θ1, θ2, θ3, θ4). There are two main types: forward kinematics and inverse kinematics. Conversion from Cartesian to the joint system is called forward kinematics, whereas the opposite conversion is called inverse kinematics. The effective algorithm in applying kinematics can increase the speed of the robot by obtaining actions that lead to the shortest path. The Scorbot-ER 9Pro robot arm features 5 DOF and gripper [4]. Multiple axis joints allow the effector to be positioned freely in a large working space.

Robot arms are complex enough, with 5 or more DOFs, to precisely perform human tasks. However, simple systems are limited to hardcoded movements and cannot adapt to any change in the scope of the task. The system specification is dependent on the scope and the application. As the scope increases the system might need corresponding modification in order to fulfill the application needs. Often, this leads to the implementation of specific subsystems into the current project and serves as an alternative to ready-to-go market solutions [5]. The Scorbot-ER 3 robotic arm was modified with additional camera and markers which resulted in advanced capabilities of the system. Since then, the system obtained functions such as autonomous movement and color detection. Moreover, it was possible to expand the functioning further by implementing sensor and utilizing it as a sun tracking robot. Computer vision deals with several problems and object recognition is considered to be one of the highest priority. This project deals with object sorting based on the size and the color. A wide variety of studies is present on the color recognition. For example, a robotic manipulator by Mitsubishi Movemaster RV-M1 with integrated webcam for image processing of round objects is described in [6]. The system proved to be a complete success provided that perfect conditions exist. However, the success rate reduces by an average of 20% if the color is not specified.

In [7], a mechatronics system with image processing for object sorting is discussed. The project is aimed at developing the existing sorting system by introducing image processing technique. Apart from that, [8] discussed the use of two CCD cameras for lemon sorting tasks. Visual Basic 6.0 software was used to assess the RGB color space of the lemons. For three different lemon grades, the system showed a success rate ranging from 86.6% to 100%. The RGB analysis method is also proved to be successful in [9]. In this work, mirrors were used to assess three images of the wheat, thus increasing the number of features analyzed. The results show that increasing the number of features processed by the computer increases the success rate of the system with almost no expenses at computational power. The system showed a 10% greater accuracy than conventional one feature sorters. A sorting system that uses automated visual inspection for contaminants removal from wool is described in [10]. The image processing is approached with RGB and HSV methodologies, which are processed into a single image showing the contaminants location. The approach is reported to achieve 96% accuracy. A system that can classify fruits at high speeds [11] was studied with a camera with the Hue, Saturation and Intensity color system is used. The findings of the paper suggest that colors can be classified accurately with just a single parameter, hue, which provides an order by 3 faster classifications than the conventional RGB system. However, perfect lighting should be present for the system operation. This problem was addressed in [12] where various lightning conditions for raisin sorting were studied. Arduino kit is used as a microcontroller for the project. A genetic algorithm for HSI system GAHSI was tested. The results show that GAHSI approach performs comparably to lightning-controlled approaches. Thus, the system can work under variable conditions with a reasonable error.

One of the essential parts of the work is shape and size identification of the manipulated object for an autonomous machine perception and vision. Image processing can be integrated into ScorBot for 2D and 3D models recognition. The most common technique used for size and shape capture is image thresholding. The main idea of this approach lies in the creation of binary images by replacing certain pixels that above some threshold with white pixels and others with black [13]. For 3D models, there are two methods used in order to identify size and shape: depth reconstruction and knowledge-based vision [14]. For depth reconstruction method, several viewpoints are required in order to capture the object and its depth, as human's eye does, and then transmitting it in 3D model [15]. Authors use 4 cameras in order to capture its size and shape and create 3-D model using prescribed methodology. Later this model is used to generate instructions for ScorBot-ER 9Pro for assembly and manufacturing of the given objects. Another method allows capturing 3D object from its 2D projection using Spatial Correspondence, Evidential Reasoning and Perceptual Organization technique described [14]. For 2D model recognition, there are currently two concepts exploited: comparison of captured object's metric with predetermined values and neural network similarity testing [16]. The former concept is described in [17] which lies in the identification of the boundaries of the object and then enclosing it in the boundary box. By calculating pixels within object's boundary, the size and shape can be deduced. The model is explained in [18]. Author has programmed ScorBot ER-4 Plus for visual classification and simple manipulation of the object. The latter concept was used in another work described in [19]. A robotic arm is used for object sorting application. Using camera they obtained a picture of the objects and compared it with each picture in a pre-constructed database using a neural network.

3. Methodology

The robot is electrically and pneumatically activated. It consists of the electrical and electronic controller unit, pneumatic drive system, and mechanical structural and transmission components.

3.1 Camera setup

The research was aimed to modify the Scorbot by making it automated and a smart robot which has vision source. Therefore, the camera is supposed to be an inherent part of the robot. Since Scorbot has no special place to integrate a camera, its holder was designed by SolidWorks and manufactures in a 3D printer. Vision source has to be located as close as possible to the Scorbot origin which is designed as gripper center. However, it is impossible due to the size of the camera. Therefore, it was planned to mount it near the gripper. As shown in **Fig. 1**, there is a slider space and the bolt-holes that are used as fixtures.

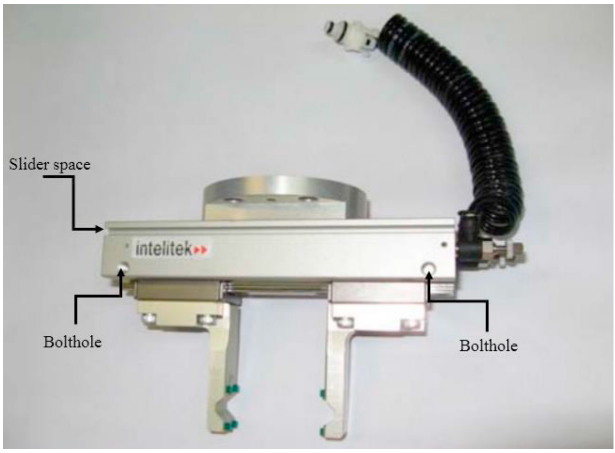


Fig. 1. Gripper structure

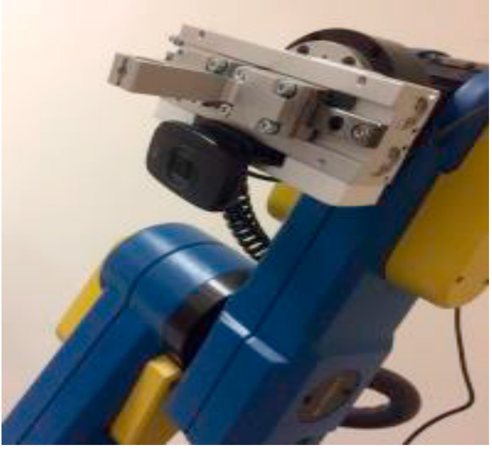


Fig. 2. Scorbot ER9 Pro gripper with camera attachment

All dimensions were set accordingly. Plastic was chosen as a material for it. The reason is that it is soft and its size can be easily changed manually. The position of the camera is on the gripper as illustrated in Fig. 2. The change could be required due to the inaccurate manufacturing. Finally, the holder was manufactured using the 3D printer as highlighted in **Fig. 2**.

3.2 Image processing

Image processing is a method of converting an image into a digital form and implementing certain operations on it in order to get the enhanced image or to obtain some information from the image. Matlab is used as a platform for this operation. As mentioned before, the main objective of this project is to capture the image of the blocks, to identify the areas of the blocks, sort them, get their centroid coordinates and arrange them accordingly. Since the aim is to make the robot be able to sort objects based on vision sensor, integration of the web camera and Matlab was implemented. **Figure 3** shows the schematic of the system which is used for image processing. The output results are in terms of object coordinates, and it is sent to the *Scorbase* through *Visual Basics.*

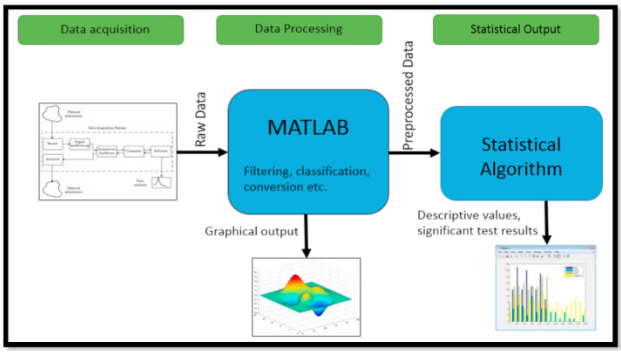


Fig. 3. Schematic of image processing

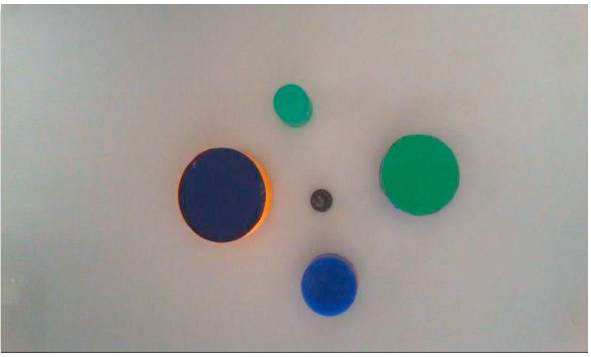
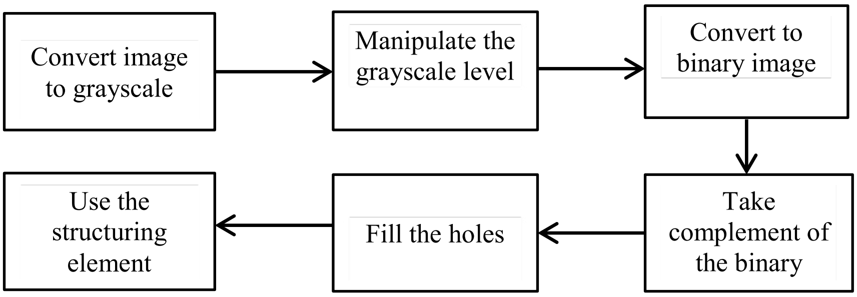


Fig. 4. Original image

3.3 Data acquisition

The web camera was connected to PC via USB port which was used as a visual source for MATLAB. Therefore, it was needed to get information on available image acquisition hardware. To perform this task, ***imaqhwinfo(adaptorname, deviceID)*** function was used. The original image is shown in **Fig. 4** below. The results were obtained through the code. These results gave additional data on web cameras and were useful because when creating an algorithm, adaptor name and device IDs have to be indicated correctly. Next step was to create a video using web camera, and, then at the certain instant it captured the image of the objects placed in front of the robot gripper. The objects were several circular waxes with different sizes. To perform the above-mentioned task, ***video input*** and ***getsnapshot*** functions were used. Then, this image was saved on PC through ***imwrite*** function and was used as an input for the data processing. The *returnedcolorspace* property specifies the color space of the toolbox in order to use when it returns image data to the MATLAB workspace. This is relevant when it is necessary to access the acquired image data with the *getsnapshot* function. *RGB* is the most commonly used color space, where *RGB* stands for red-green-blue. There is also another color space widely used in the digital video which is *YCbCr*. However, it is mainly used when proceeding with colorful images. **Figure 5** shows the centroid location of the images for each object. Image processing steps are shown as follows:


Convert image to grayscale
Manipulate the grayscale level
Convert to binary image
Take complement of the binary
Fill the holes
Use the structuring element

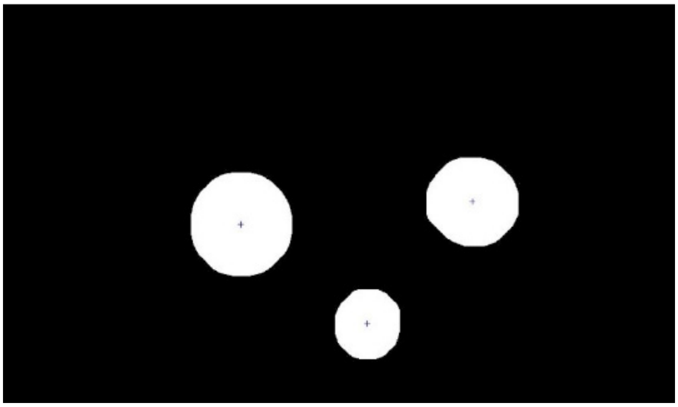
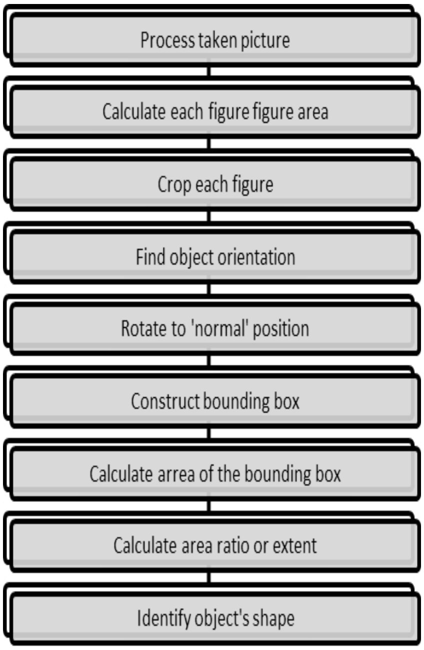


Fig. 5. Centroid locations

3.4 Size and Shape Recognition

All calculations are based on the next block diagram and are performed on MATLAB. Firstly, the image is captured and processed using thresholding technique. The size of each region is identified using regionprops command. The size is obtained by counting every non-black pixel in the closed region. This can easily be done by converting picture to the binary image (black and white image) using command im2bw that creates a matrix with the same dimensions as array uint8 but without color entry. The matrix consists of elements with values 1 for the white region and 0 for black. Each element represents a pixel in the original picture. Thus, the size of any region can be computed in terms of these pixels. For more precise calculations, knowing the area of some object and its pixel size, area metric can be calculated.



Process taken picture
Calculate each figure figure area
Crop each figure
Find object orientation
Rotate to 'normal' position
Construct bounding box
Calculate arrea of the bounding box
Calculate area ratio or extent
Identify object's shape

Fig. 6. Shape identification block diagram

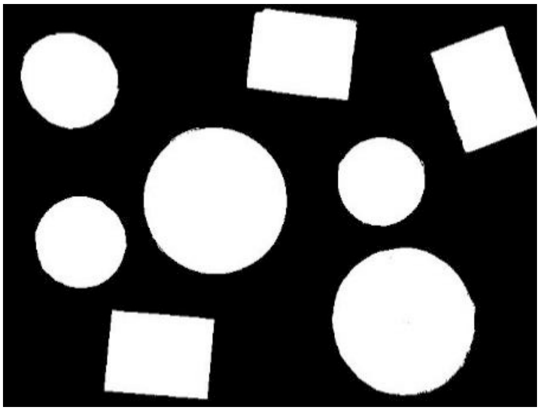


Fig. 7. Binary image

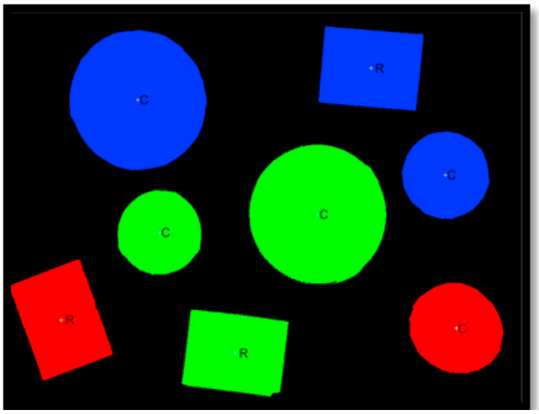


Fig. 8. The real shape generation

Figure 6 shows the shape identification block diagram whereas **Fig. 7** shows the results of binary image detection. **Figure 8** highlights the real shapes of the objects to be detected. Empirically it was that if the extent is equal to 0.78, the shape enclosed in bounding box is a circle. Similarly, if the extent is more than 0.25 and less than 0.5, the figure is a triangle. For a rectangle, the ratio should be around 1. Due to the imperfectness of captured image and picture processing, these values deviate slightly and require some calibration. For a visual representation, the program writes shape letter on each figure, as it can be seen in the next picture. **Figure 9** sums up all the ratios of the areas by which the given shapes can be identified.

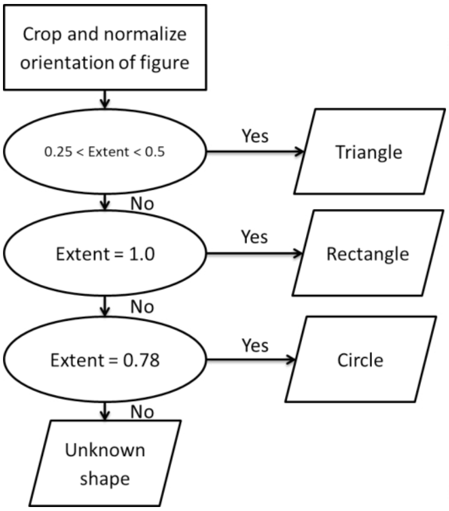
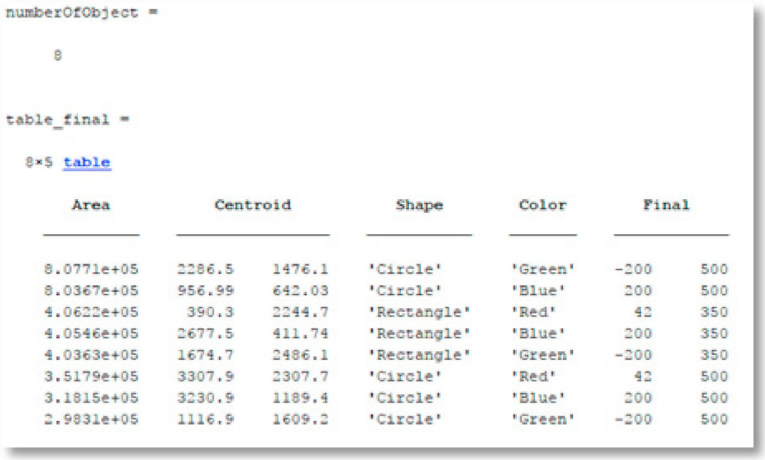


Fig. 9. Shape identification block diagram



```
numberOfObject =

     8


table_final =

  8×5 table

       Area           Centroid            Shape          Color         Final
    __________    ________________    ___________    _______    ______________

    8.0771e+05    2286.5    1476.1    'Circle'       'Green'    -200       500
    8.0367e+05    956.99    642.03    'Circle'       'Blue'      200       500
    4.0622e+05     390.3    2244.7    'Rectangle'    'Red'        42       350
    4.0546e+05    2677.5    411.74    'Rectangle'    'Blue'      200       350
    4.0363e+05    1674.7    2486.1    'Rectangle'    'Green'    -200       350
    3.5179e+05    3307.9    2307.7    'Circle'       'Red'        42       500
    3.1815e+05    3230.9    1189.4    'Circle'       'Blue'      200       500
    2.9831e+05    1116.9    1609.2    'Circle'       'Green'    -200       500
```

Fig. 10. MATLAB table for Scorbot ER9Pro

In the end, a table is constructed as shown in **Fig. 10** which contains the following values for each figure: area (in pixels), coordinates of centroids (which are used for navigation of robot), color, shape, and coordinates of the final position (to which the figure should be transported). These values are required for operation of Scorbot ER9Pro.

4. Modeling, hardware integration and simulation

4.1 Axis Gear Ratios

The overall gear ratio of the output shaft which moves the axis is therefore expressed as:

$N_T \ x \ N_{HD} = N_{Axis}$
Where:
N_T is the belt drive: $\frac{PulleyB}{PulleyA}$
N_{HD} is the Harmonic drive ratio.
N_{Axis} is the overall gear ratio of the axis.

Table 1. Gear Ratios of Scorbot ER 9Pro [20]

	N_T	N_{HD}	N_{AXIS}
Axis 1	1.33 : 1	160 : 1	213.33 : 1
Axis 2	1.52 : 1	160 : 1	243.8 : 1
Axis 3	1.33 : 1	160 : 1	213.33 : 1
Axis 4	1.8 : 1	100 : 1	180 : 1
Axis 5		100 : 1	100 : 1

4.2 Modeling and simulation

The simulation starts with the creation and placement of robot links and joints. Length of all joints and links were assigned in DH tables. These tables contain data on links and joints as well as the position of link origins, which are used for movement development. Next, visualization of assigned links and joints was prepared as shown in **Fig. 11**.

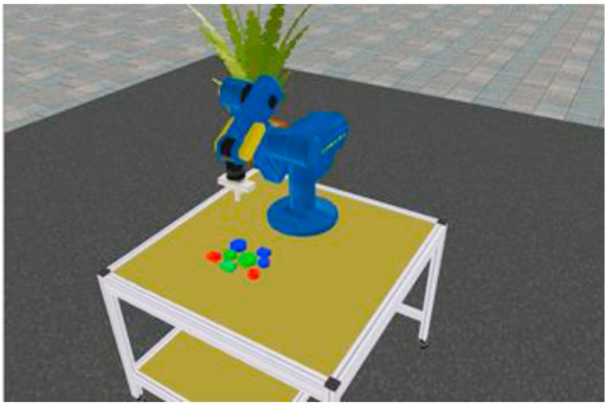


Fig. 11. Visualization of Scorbot-ER 9 Pro links and joints

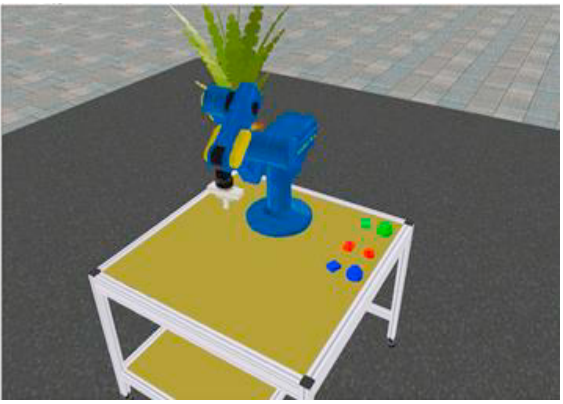


Fig. 12. Scorbot position while gripping the object

Dashed lines represent links and joints. Origin of each link is placed at the end of previous one. Joints used for link connection; however, they do not move. Robot movements and direction of motion are determined according to the origin of each link. Links, joints, and origins are used for the placement of 3D parts of Scorbot-ER 9 Pro and their movement is necessary to consider in forwarding Kinematics Methodology for Scorbot -ER 9 Pro. **Figure 12** shows that the Robot gripper is opened and placed before gripping the object by setting coordinates of the desired point. Using *ScorXYZPR2BSEPR* and *ScorSimSetGripper* functions robot's movement such as picking and place can easily be simulated. **Figure 13** shows the successful object placement with the robot gripper utilizing vision system for the fast and accurate industrial applications.

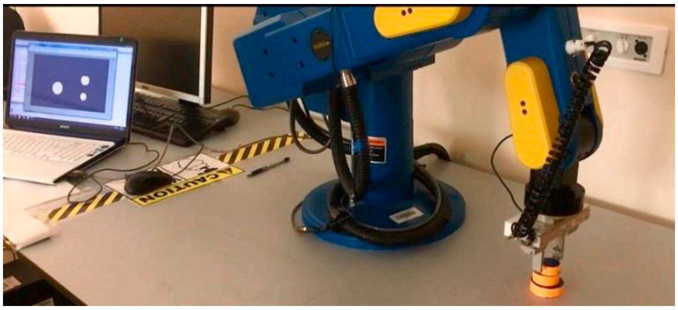


Fig. 13. The robot places the objects in an order

5. Conclusions

This paper presents a vision-based autonomous object sorting system with a robotic arm. Industrial robot Scorbot-ER 9Pro was used throughout this research and a USB camera was installed on camera holder which is manufactured by a 3D printer. The camera holder was mounted on the gripper. The objectives of the project were:

- To determine object specification using image processing techniques
- To establish an interface between Matlab and ACL of the Scrobot
- To pick and place objects based on the size, color, and shape

The following operations were performed. Firstly, pictures of objects were captured using image processing tool in Matlab. Secondly, image processing tool was used to determine the size, color, and shape of an object. Thirdly, after the main bodies were found, coordinates of its centroids were defined. Finally, all the sorted objects based on size, shape, and color are stored as a table in a file. The robot's position control was the key challenge in this research as it was integrated with several software systems. The robot's software could not "Home" the robot due to the error generated by the encoder and it was impossible to program the robot. In the end, the possible causes of this malfunction were determined. Additionally, a simulation was conducted in order to present the performance of the system. Since the Scorbot software was not meant to be autonomous, additionally, the software connection between the RoboCell and Matlab had to be established. Therefore, Visual Basic was introduced to serve as a connection between Matlab and the RoboCell. The digital data from Matlab was processed by Visual Basic and RoboCell, and the robot was able to successfully perform the pick-and-place operation based on an object's size, shape, and color.

References

[1] Basu, Rahul, and Swathi Padage. "Development of 5 DOF Robot Arm-Gripper for sorting and investigating RTM Concepts." Materials Today: Proceedings 4, no. 2 (2017): 1634-1643.

[2] Dvorsky, G. (2017). Cite a Website - Cite This For Me. [online] Gizmodo.com. Available at: https://gizmodo.com/robots-are-already-replacing-human-workers-at-an-alarmi-1793718198 [Accessed 21 Sep. 2017].

[3] Juang, Jih-Gau, Yi-Ju Tsai, and Yang-Wu Fan. "Visual recognition and its application to robot arm control." Applied Sciences 5, no. 4 (2015): 851-880.

[4] Intelitek ®, ―Scorbot-ER 9Pro, User Manual‖, Copyright ©2011 Intelitek ® Inc.Catalog #200034 Rev.B. March 2011.

[5] Szabo, Roland, and Aurel Gontean. "SCORBOT-ER III robotic arm detection and control with color recognition with the possibility for using in sun tracker applications." In Environment and Electrical Engineering and 2017 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe), 2017 IEEE International Conference on, pp. 1-6. IEEE, 2017.

[6] A. Djajadi, F. Laoda, R. Rusyadi, T. Prajogo and M. Sinaga, "A MODEL VISION OF SORTING SYSTEM APPLICATION USING ROBOTIC MANIPULATOR", Journal.uad.ac.id, 2017. [Online]. Available: http://journal.uad.ac.id/index.php/TELKOMNIKA/article/view/615/424. [Accessed: 21- Sep- 2017].

[7] "Object Sorting System Using Robotic Arm", Journaldatabase.info, 2013. [Online]. Available: http://journaldatabase.info/articles/object_sorting_system_using_robotic.html. [Accessed: 21- Sep- 2017].

[8] M. Khojastehnazhand, "Development of a lemon sorting system based on color and size", 2010. [Online]. Available: http://www.academicjournals.org/article/article1380110185_Khojastehnazhand%20et%20al.pdf. [Accessed: 21- Sep- 2017].

[9] T. Pearson, D. Brabec, and S. Haley, "Color image based sorter for separating red and white wheat", Sensing and Instrumentation for Food Quality and Safety, vol. 2, no. 4, pp. 280-288, 2008.

[10] L. Zhang, A. Dehghani, Z. Su, T. King, B. Greenwood and M. Levesley, "Development of a Mechatronic Sorting System for Removing Contaminants From Wool", IEEE/ASME Transactions on Mechatronics, vol. 10, no. 3, pp. 297-304, 2005.

[11] G. Kay and G. de Jager, "A versatile color system capable of fruit sorting and accurate object classification", Proceedings of the 1992 South African Symposium on Communications and Signal Processing.

[12] M. Abbasgholipour, M. Omid, A. Keyhani and S. Mohtasebi, "Color image segmentation with genetic algorithm in a raisin sorting system based on machine vision in variable conditions", Expert Systems with Applications, vol. 38, no. 4, pp. 3671-3678, 2011.

[13] Shapiro, Linda, and George C. Stockman. "Computer vision. 2001." ed: Prentice Hall (2001) p. 83.

[14] Lowe, David G. "Three-dimensional object recognition from single two-dimensional images." Artificial intelligence 31.3 (1987): 355-395.

[15] Bitzidou, Malamati, Dimitrios Chrysostomou, and Antonios Gasteratos. "Multi-camera 3D Object Reconstruction for Industrial Automation." IFIP International Conference on Advances in Production Management Systems. Springer, Berlin, Heidelberg, 2012.

[16] Rege, Sanket, et al. "2D geometric shape and color recognition using digital image processing." International journal of advanced research in electrical, electronics and instrumentation engineering 2.6 (2013): 2479-2487.

[17] Chhaya, Shambhavi Vijay, Sachin Khera, and Pradeep Kumar. "Basic geometric shape and primary color detection using image processing on Matlab." IJRET: International Journal of Research in Engineering and Technology 4 (2015): 2319-1163.

[18] Ali, Md. Hazrat and N. Mir-Nasiri. "Vision-Based Object Classification with Scorbot-ER 4U". International Journal of Mining, Metallurgy & Mechanical Engineering 4.1 (2016): 2320–4060

[19] Kumar, Rahul, et al. "Object detection and recognition for a pick and place Robot." Computer Science and Engineering (APWC on CSE), 2014 Asia-Pacific World Congress on. IEEE, 2014.

[20] ftp://ftp.robotec.co.il/Techsup/er9pro/Books/200034%20Rev%20A%20Scorbot%20ER%209Pro.pdf, accessed on April 26, 2018.