ENHANCING PALLETIZING AND SHAPE DRAWING USING IMAGE PROCESSING ON PARALLEL AND SERIAL LINK MANIPULATORS

Samuel Kariuki[*1], Eric Wanjau[*1], Waweru Njeri[1,2], Joseph Muguro[1,2], Harrison Ngetha[1], Kojiro Matsushita[2], Minoru Sasaki[2]

[1]Department of Electrical & Electronic Engineering, Dedan Kimathi University of Technology, Private Bag - 10143, Dedan Kimathi, Kenya

[2]Department of Mechanical Engineering, Gifu University, 1-1 Yanagido, Gifu City, 1193, Japan

*Corresponding authors: samuelkariuki544@gmail.com, muriithi.wanjau@outlook.com

Abstract The integration of robotics and image processing has led to the realization of robot autonomy in dynamic environments through the provision of visual feedback. This paper presents the application of parallel and open-link robots in palletizing and shape drawing tasks as enhanced by visual feedback from image processing. In determining the set of joint angles that could be used to attain the position and orientation of the end effector, the geometric approach in which the spatial geometry of the robotic arms was decomposed into several plane geometry problems was employed. In enhancing the performance of the robotic manipulators, image processing techniques were explored. In one approach, Color-based segmentation was used to distinguish between different objects in the workspace by using predefined color markers as references in the L*a*b color space. Classification of each pixel in the workspace image was then done by calculating the Euclidean distance between that pixel and a predefined color marker. A second approach employed Edge detection to identify the boundaries of objects within the workspace image by employing the Hough Transform mathematical model to detect the abrupt changes in the image brightness pixel-wise. The pixel locations from Hough were then sorted sequentially to outline the detected object. The integration of image processing with the robotic tasks was expected to improve the precise detection of the position of objects as well as the outline of geometric shapes. The incorporation of visual feedback allowed for dynamic robot manipulation in which prior knowledge of the workspace was not required. This led to improved pick and place as well as shape detection as applied in palletizing and shape drawing actuated by the parallel and serial link manipulators respectively.

Keywords: Color-based segmentation, Canny-Edge detection, Hough Transform.

1. Introduction

The field of robotics has continued to grow tremendously in the last decade as a new research frontier and has demonstrated its relevance and impact in modern applications of industrial automation. These robots came handy in carrying out repetitive chores, in hazardous work environment and in heavy industries to mention just but a few [1].

The recent advances in image acquisition capabilities and processing power provides for excellent tools in designing more complex image processing and pattern recognition tasks[2]. The integration of image processing with robotics has led to the evolution of perceptive robots, what is termed as Robot Vision.

Robotic manipulators are broadly grouped into family of open chain and closed chain configurations. Serial link industrial robots are best examples of open chain configuration characterized by smaller footprint, lower specific pay load capacities, large workspace and better reach. The closed chain manipulators categorized as planar manipulators and parallel robots, offers better stiffness and higher pay load capacity and faster actuation over a confined workspace [3]. The acceptability of a manipulator is based on the comparative performance evaluation which makes estimation

of the performance of manipulators an important factor in determining its application and design. In recent years, there has also been an increasing demand for weight reduction of robot arms in order to realize high speed operation and energy saving [4].

The major performance characteristics considered in this paper were the manipulators' workspace, dexterity and positional accuracy. Dexterity index is a measure of the possibility of a manipulator to achieve different orientations for each point within the workspace [3]. The ability of a serial manipulator to attain multiple orientations for a set of points results in serial manipulators being more dexterous than parallel manipulators. The workspace of parallel manipulators is smaller compared to that of serial link manipulators. This is brought about by the several independent kinematic chains connecting the end effector in parallel to the base. Theoretically, most of the parallel manipulators are presumed to be more accurate than serial link robotic manipulators. The reasons behind this anticipation are the rigidity of the structure in which links are arranged in parallel to share pay loads, making the structure more rigid compared to serial link configurations.

From the performance features discussed above, the parallel link manipulator is most suited for a wide range of assembly, pick and place, and material handling applications with limited workspace which make the palletizing task a great fit [5]. Conversely, the serial link manipulator is suited for industrial applications tasks that require high dexterity and speed such as welding, painting and parts-cutting which make the geometric shape drawing a good focus area.

Palletizing tasks typically involve detecting objects, sorting and stacking them based on particular features. On the other hand, shape drawing involves detecting the boundaries of objects and extracting out the shape outline. As a result, the most appropriate image processing technique was color-based segmentation for the palletizing task while Canny edge detection coupled with Hough transform was deemed as the appropriate technique for the shape drawing task.

In this paper we present the application of parallel and open-link robots in palletizing and shape drawing tasks respectively based on image processing.

2. Manipulator Kinematics

Kinematics is the science of motion that treats the subject without regard to the forces that cause it [6]. The study of the position, velocity, acceleration, and all higher order derivatives of the position variables (with respect to time or any other variable(s)) is done within the science of kinematics. In this regard, manipulator kinematics could either be static, or time-based where the velocities and acceleration are involved. In this research, we considered the position and orientation of the manipulator linkages (forward and inverse kinematics) in static situations.

The robot manipulators used for this research are as shown in Fig. 1 and Fig. 2:

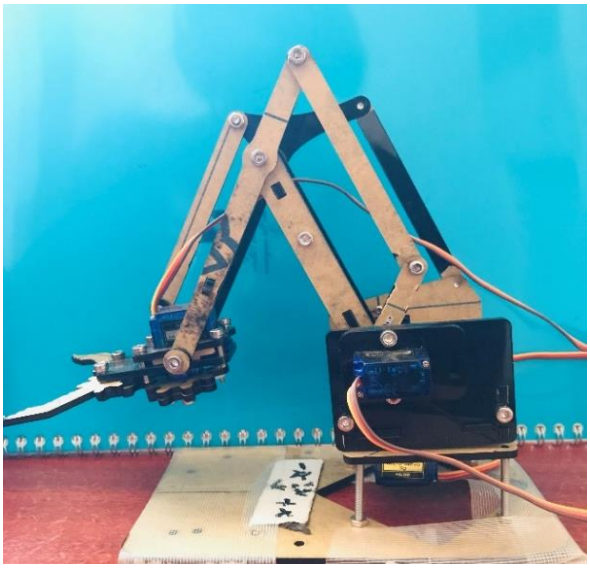

Fig. 1: Parallel link manipulator (meArm)

This is a 3-DoF low cost open-source robot arm that uses four SG90 servomotors including the gripper.

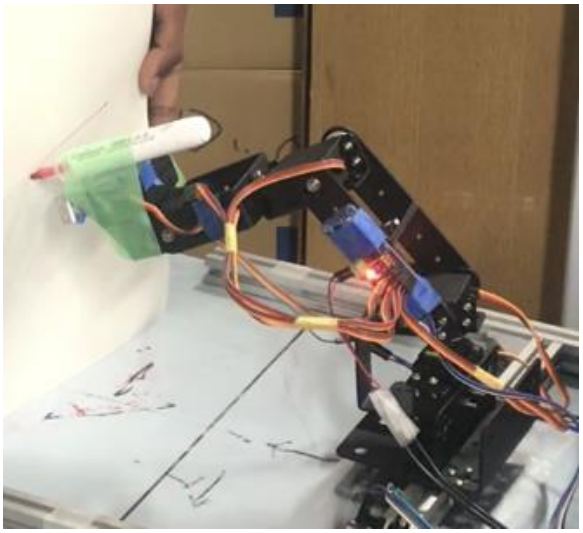

Fig. 2: Open link manipulator (ROT3U)

This is a 6-DoF aluminium robot arm DIY kit that uses 6 servomotors including the gripper. For the shape-drawing application, only 4-DoF was considered as the gripper was not included as part of the setup.

2.1 Forward kinematics

Forward kinematics addresses the problem of computing the position and orientation of the end effector relative to the user's workstation given the joint angles of the manipulator. According to the DH convention, any robot can be described kinematically by giving the values of four quantities, typically known as the DH parameters, for each link. The link length (a) and the link twist (α) describe the link itself and the remaining two, link offset (d) and the joint angle (θ), describe the link's connection to a neighboring link.

Based on the DH-Convention, the robots were decomposed into stationary and moving frames as in Fig. 3 and Fig. 4:

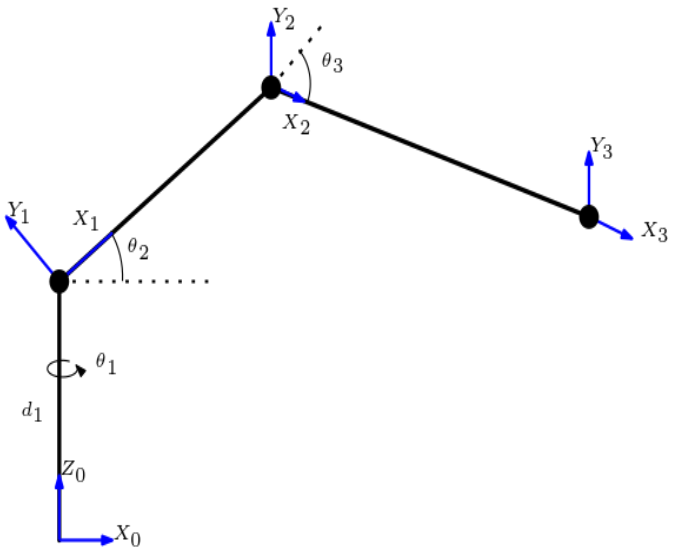

Fig. 3: DH Convention Frames assignment for meArm

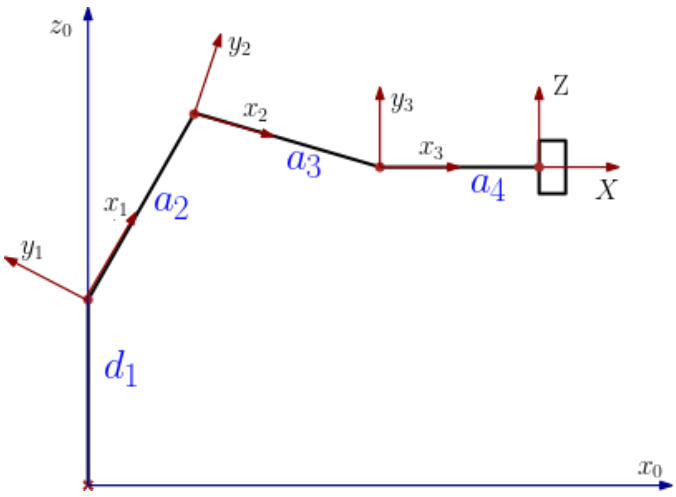

Fig. 4: DH Convention Frames assignment for ROT3U

The DH parameters for the parallel link manipulator (meArm) and the open link manipulator (ROT3U) were obtained as in Table 1 and Table 2 respectively.

Table 1:DH parameters for the meArm

Link	$\boldsymbol{\theta}$	**a(mm)**	$\boldsymbol{\alpha}$	**d(mm)**
1	θ_1	0	90˚	55
2	θ_2	80	0	0
3	θ_3	120	0	0

Table 2: DH parameters for the ROT3U manipulator

Link	$\boldsymbol{\theta}$	**a(mm)**	$\boldsymbol{\alpha}$	**d(mm)**
1	θ_1	0	90˚	110
2	θ_2	105	0	0
3	θ_3	100	0	0
4	θ_4	70	0	0

Consequently, the total homogeneous transformation that specifies how to compute the position and orientation of the end effector frame with respect to the base frame for both manipulators was obtained as:

a. For the Parallel link manipulator

Total transformation Matrix: $${}^{3}_{0}T = T_0^1 \times T_1^2 \times T_2^3 \quad (1)$$

Where the homogeneous transformations of each link were obtained as:

$$T_0^1 = \begin{pmatrix} \cos(\theta_1) & 0 & \sin(\theta_1) & 0 \\ \sin(\theta_1) & 0 & -\cos(\theta_1) & 0 \\ 0 & 1 & 0 & \frac{-11}{2} \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2)$$

$$T_1^2 = \begin{pmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & 8\cos(\theta_2) \\ \sin(\theta_2) & \cos(\theta_2) & 0 & 8\sin(\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (3)$$

$$T_2^3 = \begin{pmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & 12\cos(\theta_3) \\ \sin(\theta_3) & \cos(\theta_3) & 0 & 12\sin(\theta_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4}$$

Considering the three links of the meArm, the total homogeneous transformation was obtained as the product of the transformations of each individual link and was evaluated as:

$${}_0^3T = \begin{pmatrix} \cos(\theta_2+\theta_3)\cos(\theta_1) & -\sin(\theta_2+\theta_3)\cos(\theta_1) & \sin(\theta_1) & 4\sigma_1\cos(\theta_1) \\ \cos(\theta_2+\theta_3)\sin(\theta_1) & -\sin(\theta_2+\theta_3)\sin(\theta_1) & -\cos(\theta_1) & 4\sigma_1\sin(\theta_1) \\ \sin(\theta_2+\theta_3) & \cos(\theta_2+\theta_3) & 0 & -\frac{11}{2}+12\sin(\theta_2+\theta_3)+8\sin(\theta_2) \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{5}$$

Where the variable

$$\sigma_1 = 3\cos(\theta_2+\theta_3)+2\cos(\theta_2) \tag{6}$$

The coordinates (XYZ) of the final end effector positions are the top right 3x1 matrix of the total homogeneous transformation matrix:

$$\begin{matrix} X \\ Y \\ Z \end{matrix} = \begin{pmatrix} 4\cos(\theta_1)[3\cos(\theta_2+\theta_3)+2\cos(\theta_2)] \\ 4\sin(\theta_1)[3\cos(\theta_2+\theta_3)+2\cos(\theta_2)] \\ -\frac{11}{2}+12\sin(\theta_2+\theta_3)+8\sin(\theta_2) \end{pmatrix} \tag{7}$$

b. For the Open link manipulator

The total homogeneous transformation that specifies how to compute the position and orientation of the end effector frame with respect to the base frame was obtained as:

$$T_0^4 = T_0^1 \times T_1^2 \times T_2^3 \times T_3^4 \tag{8}$$

Where the homogeneous transformations of each link were obtained as:

$$T_0^1 = \begin{pmatrix} \cos(\theta_1) & -\sin(\theta_1)\cos(\alpha) & \sin(\theta_1)\sin(\alpha) & a_1\cos(\theta_1) \\ \sin(\theta_1) & \cos(\theta_1)\cos(\alpha) & -\cos(\theta_1)\sin(\alpha) & a_1\sin(\theta_1) \\ 0 & \sin(\alpha) & \cos(\alpha) & d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{9}$$

$$T_1^2 = \begin{pmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & a_2\cos(\theta_2) \\ \sin(\theta_2) & \cos(\theta_2) & 0 & a_2\sin(\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{10}$$

$$T_2^3 = \begin{pmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & a_3\cos(\theta_3) \\ \sin(\theta_3) & \cos(\theta_3) & 0 & a_3\sin(\theta_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{11}$$

$$T_3^4 = \begin{pmatrix} \cos(\theta_4) & -\sin(\theta_4) & 0 & a_4\cos(\theta_4) \\ \sin(\theta_4) & \cos(\theta_4) & 0 & a_4\sin(\theta_4) \\ 0 & 0 & 1 & d_4 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{12}$$

In a similar manner to the parallel link manipulator above, the coordinates (XYZ) of the final end effector positions were the top right 3x1 matrix of the total homogeneous transformation ${}_0^4A$ determined as:

$$\begin{aligned} X = {} & a_4c_4\left[c_3\left(c_{12}-c_\alpha s_{12}\right)-s_3(c_1s_2+c_{2\alpha}s_1)\right]-a_4s_4\left[c_3\left(c_1s_2+c_{2\alpha}s_1\right)+s_3\left(c_{12}-c_\alpha s_{12}\right)\right]+ \\ & a_3c_3\left[c_{12}-c_\alpha s_{12}\right]-a_3s_3\left[c_1s_2+c_{2\alpha}s_1\right]+a_2c_{12}-a_2c_\alpha s_{12} \end{aligned}$$

$$\begin{aligned} Y = {} & a_4c_4\left[c_3\left(c_2s_1+c_{1\alpha}s_2\right)-s_3(s_{12}-c_{12\alpha})\right]-a_4s_4\left[c_3\left(s_{12}-c_{12\alpha}\right)+s_3\left(c_2s_1-c_{1\alpha}s_2\right)\right]+ \\ & a_3c_3\left[c_2s_1+c_{1\alpha}s_2\right]-a_3s_3\left[s_{12}-c_{12\alpha}\right]+a_2c_2s_1+a_2c_{1\alpha}s_2 \end{aligned} \tag{13}$$

$$Z = a_4\left(c_{2+3}s_{4\alpha}+s_{2+3}s_\alpha c_4\right)+a_3\left(c_2s_{3\alpha}+c_3s_{2\alpha}\right)+a_2s_{2\alpha}+d_1$$

Where considering $a,b=1,2,3$

$$\begin{aligned} & c_\alpha = \cos\left(\theta_\alpha\right), s_\alpha = \sin\left(\theta_\alpha\right) \\ & c_b = \cos\left(\theta_b\right), s_b = \sin\left(\theta_b\right) \\ & c_{a+b} = \cos\left(\theta_a+\theta_b\right), s_{a+b} = \sin\left(\theta_a+\theta_b\right) \\ & c_{abc} = \cos\left(\theta_a\right)\times\cos\left(\theta_b\right)\times\cos\left(\theta_c\right), s_{abc} = \sin\left(\theta_a\right)\times\sin\left(\theta_b\right)\times\sin\left(\theta_c\right) \end{aligned} \tag{14}$$

2.2 Inverse Kinematics

Inverse kinematics addresses the more difficult converse problem of computing the set of joint angles that will place the end effector at a desired position and orientation. It is the computation of the manipulator joint angles given the position and orientation of the end effector. It involved extracting the

Cartesian coordinates of a given position based on the image processing, translating those positions into joint angles and rotating the manipulator servo motor angles to that desired position.

In solving the inverse kinematics problem, the Geometric approach was used to decompose the spatial geometry into several plane-geometry problems based on the sine and the cosine rules.

The determination of the joint angles for both the parallel and open link manipulators was given as:

a. For the parallel link manipulator

The calculation of the joint angles ($\theta_1, \theta_2, \theta_3$) for some known position and orientation of the meArm's end effector was done by considering the trigonometric decomposition of various planes of the manipulator as graphically illustrated below.

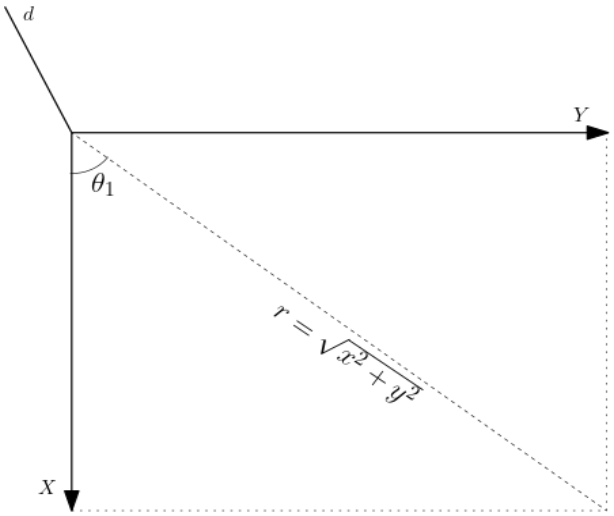

Fig. 5: Representation of θ_1

The angle θ_1 was determined by considering Fig. 5 and calculated as:

$$\theta_1 = \tan^{-1}\left(\frac{y}{x}\right) \tag{15}$$

With the hypotenuse r, connecting x and y obtained using the Pythagoras theorem as:

$$r = \sqrt{x^2 + y^2} \tag{16}$$

The angles θ_2 and θ_3 were obtained by considering the plane formed by the second and third links as illustrated below

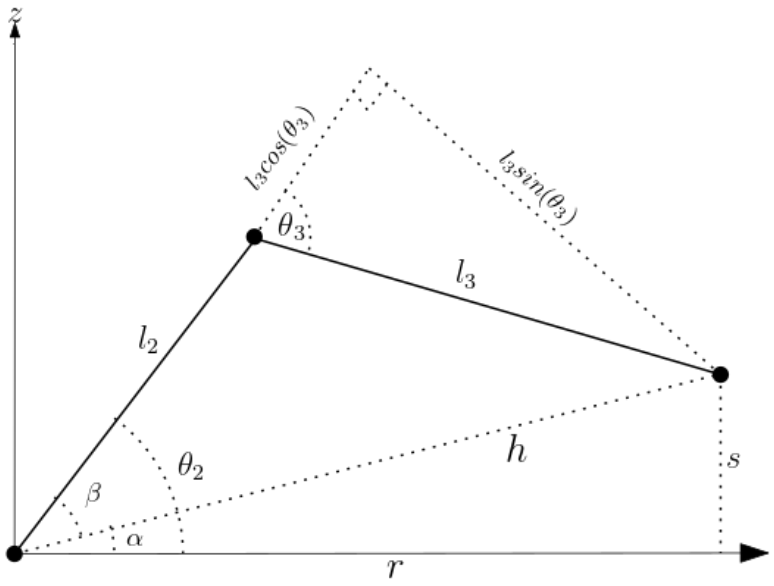

Fig. 6: Representation of θ_2 and θ_3

$$\theta_2 = \tan^{-1}\left(\frac{s}{r}\right) + \tan^{-1}\left(\frac{l_3 \sin(\theta_3)}{l_2 + l_3 \cos(\theta_3)}\right) \tag{17}$$

$$\text{where } \theta_2 = \alpha + \beta \tag{18}$$

$$\theta_3 = \cos^{-1}\left(\frac{x^2 + y^2 + s^2 - l_2^{\,2} - l_3^{\,2}}{2 l_2 l_3}\right) \tag{19}$$

Where s is the difference between the distance of the end effector from the base and the offset:

$$s = z - d \tag{20}$$

b. For the open link manipulator

The calculation of the joint angles ($\theta_1, \theta_2, \theta_3, \theta_4$) for some known position and orientation of the ROT3U's end effector was done by considering the trigonometric decomposition of various planes of the manipulator as graphically illustrated below.

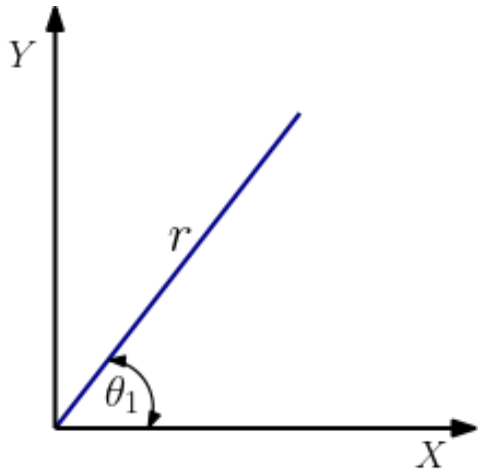

Fig. 7: Representation of θ_1

The angle θ_1 was determined by considering Fig. 7 and calculated as:

$$\theta_1 = \tan^{-1}(\frac{y}{x}) \tag{21}$$

The angles θ_2, θ_3 and θ_4 were obtained by considering the plane formed by the second, third and fourth links as illustrated below

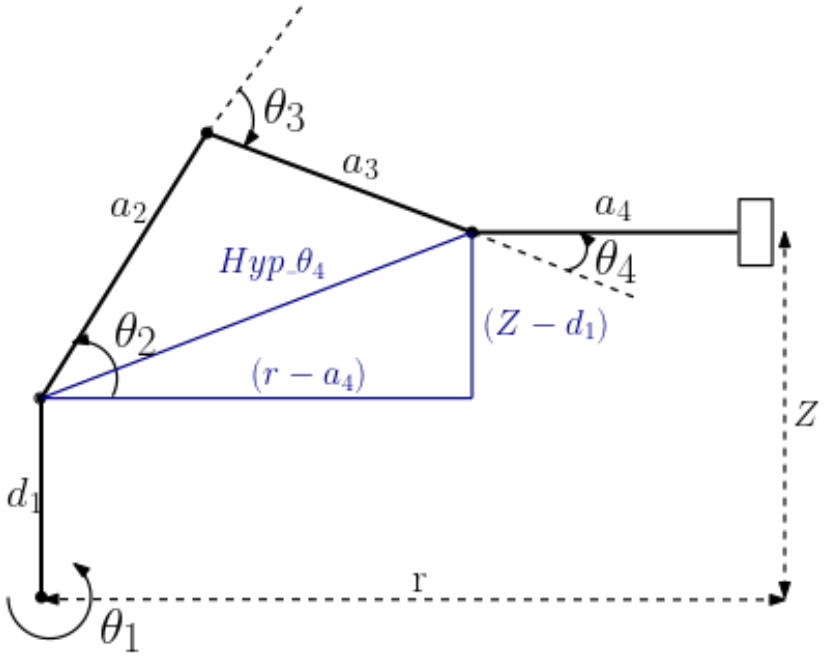

Fig. 8: Representation of θ_2, θ_3 and θ_4

By considering the sine and cosine rules:

$$\theta_2 = \cos^{-1}(\frac{(a_2)^2 + (Hyp\_\theta_4)^2 - (a_3)^2}{2 \times a_2 \times Hyp\_\theta_4}) + \sin^{-1}(\frac{z - d_1}{Hyp\_\theta_4}) \tag{22}$$

$$\theta_3 = -(180 - \cos^{-1}(\frac{(a_2)^2 + (a_3)^2 - (Hyp\_\theta_4)^2}{2 \times a_2 \times a_3})) \quad (23)$$

$$\theta_4 = \cos^{-1}(\frac{(Hyp\_\theta_4)^2 + (a_3)^2 - (a_2)^2}{2 \times Hyp\_\theta_4 \times a_3}) - \sin^{-1}(\frac{z - d_1}{Hyp\_\theta_4}) \quad (24)$$

Where Hyp_θ_4 is the length directly connecting joint 2 to joint 4 and forming the triangle joining link lengths a_2 and a_3 and is determined as follows:

$$Hyp\_\theta_4 = \sqrt{(r - a_4)^2 + (Z - d_1)^2} \quad (25)$$

$$\text{With} \quad r = \sqrt{X^2 + Y^2} \quad (26)$$

3. Image Processing

In this research, image processing was employed in both the palletizing and shape drawing tasks to address the problems of detecting an object and determining its location in the workspace. Sample input images to the palletizing and shape drawing algorithms were as shown in Fig. 9 and Fig. 10 respectively

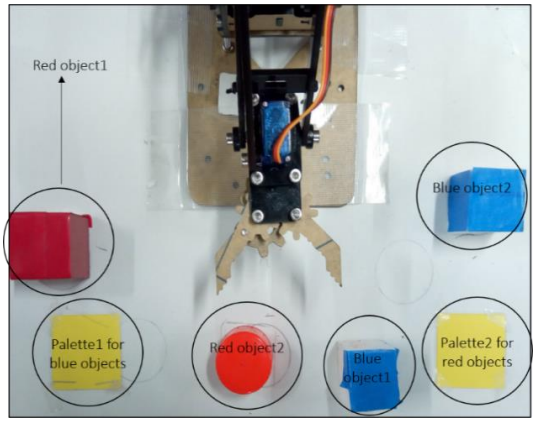

Fig. 9: Input image for the palletizing

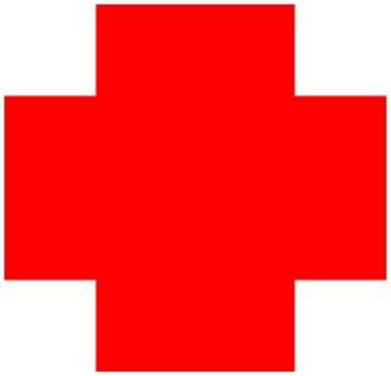

Fig. 10: input image for the shape drawing algorithms

Image pre-processing techniques were used to improve the image data by suppressing unwilling distortions and enhancing some image features important for further processing such as color-based segmentation in palletizing and edge detection in shape drawing. The following pre-processing techniques sufficed for image enrichment improvement for both robotic tasks:

a. RGB to Grayscale conversion - This step involved converting a colored image containing the different color shades (R, G, B) into a grayscale image which only carries intensity information ranging

from black (0) at the weakest intensity to white (255) at the strongest.

b. Binarizing the image - This process involved converting a grayscale image into a binary image based on a luminance threshold such that all pixels with luminance greater than the threshold are white while those below are black.
c. Filling the holes in the image - This process helped in accounting for and minimizing noise in the image.

Specific detection algorithms

The result of the pre-processing steps was a binary image in which the regions of interest i.e. the shape to be drawn or the objects to be detected, were clearly defined. As such, further processing techniques such as image segmentation and edge detection were applied to perform palletizing and shape drawing tasks.

i. Color-based segmentation for palletizing

The palletizing task involved sorting and stacking objects using color-based segmentation. Color-based segmentation is the process of dividing an image into regions of interest based on their color. A simple, memory efficient yet effective L*a*b* color-based segmentation was used to locate objects with similar color. L*a*b* color space is a color-opponent space with dimension L* for perceptual lightness and a* and b* for the four color-opponent dimensions of human vision: red, green, blue, and yellow [7].

The output of the pre-processing steps was used to identify the objects where a logic 1 represented the presence of an object. This information on the location of the objects allowed for subsequent application of color-based segmentation to identify the objects by color on the original RGB image. The L*a*b* color space is designed to approximate human vision and enables one to quantify the visual differences in color [8]. In this particular task, it was used to account for variation in color value of the RGB image across a detected object caused by problems such as camera noise. For each detected object, an average color in a*b* space was calculated, such that each detected object had a single value of ‘a*’ and ‘b*’ for all its pixel.

Since each detected object now had an 'a*' and 'b*' value, it could be classified by calculating the Euclidean distance between its pixels and a predefined color marker. The predefined color markers were a set of 'a*' and 'b*' values for standard Red, Blue, Green and Yellow colors, that were used as reference colors against which an object’s color value was compared. As such the Euclidean distance d between each pixel and a corresponding marker was computed as:

$$d = [\left(a_{pixel} - a_{color\_marker}\right)^2 + \left(b_{pixel} - b_{color\_marker}\right)^2]^{\frac{1}{2}} \quad (27)$$

The smallest distance indicated that a pixel most closely matched a corresponding color marker. Subsequently, by the nearest neighbor rule, if for instance the distance between a pixel and the red color marker is the smallest, then the pixel would be labelled as a red pixel and its corresponding object would be classified as red.

The final step of this algorithm involved determining the centroid locations of the segmented objects in pixels and then mapping them to the real-world manipulator’s workspace using an appropriate mapping function.

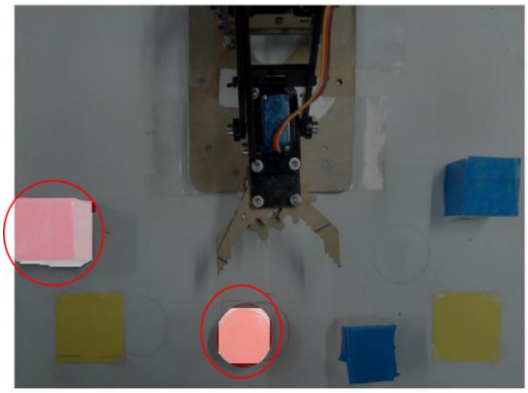

Fig. 11: Image for color-based segmentation

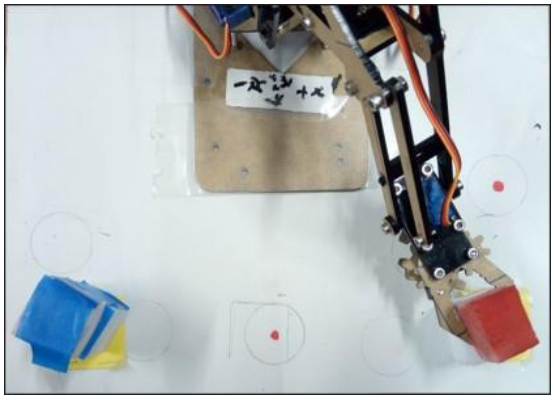

Fig. 12: Image based on palletizing by the meArm

The image on Fig. 11 shows two objects detected as 'red' since their average a*b* values closely matched those of a predefined red color marker. Fig. 12 shows the result of detecting the red and blue objects, obtaining their real-world coordinates in the workspace and finally applying inverse kinematics to sort and stack them on their respective pallets.

ii. Canny edge and Hough transform for shape drawing

In the implementation of the geometric shape drawing, the input image which was either taken using a camera for a hand-drawn image or digitally drawn by the user was supplied to the robot. The image was in the form of an array of pixels, while the drawing manipulator required a set of points in the Cartesian space, convertible into a joint space [9]. As a result, the edge detection method was employed in the determination of the shape outline.

Edge detection is an image processing technique for finding the boundaries of objects within images which works by detecting discontinuities in color and brightness. Considering an image, an edge is a curve that follows a path of rapid change in image intensity and is often associated with the boundaries of objects in a scene. The Canny edge method which is a powerful edge detection method that is relatively simple and more likely to detect true weak edges was used in determining these boundaries[10]. This was achieved by selecting an appropriate threshold based on the color intensity of the preprocessed input image and on whether the image was hand drawn or printed.

On detecting the shape boundaries, the Standard Hough Transform, which is a feature extraction technique was used to identify the Hough peaks which are the potential lines in the image. The Hough lines function then finds the endpoints of the line segments corresponding to peaks in the Hough transform and fills in small gaps in the line segments[11]. The identified endpoints give the start and end points of line and thus a full identification of the shape based on the detected boundaries.

Once the lines on the shape had been identified, the reconstruction of the geometric shape was done by sorting the obtained endpoints for a particular line and its connected neighboring line. As a final step to the image processing, the sorted endpoints pixel locations on the image were then mapped on to the manipulator workspace by using the appropriate mapping function. Using these points, the shape could then be drawn out by the robotic manipulator.

Fig. 13: Edge detection output image

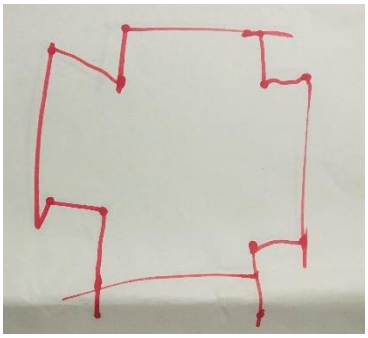

Fig. 14: image drawn by the ROT3U manipulator

For the input image shown in Fig. 10, the application of Canny edge detection technique using Hough transform in the determination of the endpoints in the detected lines resulted in the image shown in Fig. 13 above. In this image, all the lines constituting the shape were identified and an outline of the shape obtained. As the final step of the geometric shape algorithm, the image in Fig. 14 was outlined by the open link manipulator (ROT3U) after the mapping and inverse kinematics of the identified endpoints.

4. Gyroscope for path tracking

The MPU 6050 Six Axis Sensor gyroscope raw data was used to get the Cartesian coordinates of the end effector for motion characterization. The gyroscope was mounted at the tip of the end effector as an external motion sensor for trajectory tracking. The raw data obtained from the gyroscope motion was processed for the axes along which the robot moved such that reconstruction gave the path traced by the manipulator or the shape drawn.

The plot in Fig. 15 below shows the path moved by the meArm in in picking "Blue Object 1" and placing it on "Palette1" shown in Fig. 11 above. This plot made from the gyroscope data closely approximated the trajectory traced by the meArm in performing the given palletizing task.

For the geometric shape drawing task, the rectangular input image A2-B2-C2-D2 in Fig. 16 below was used as the manipulator input shape with the start point for the robot motion being at A2 and moving counterclockwise. The reconstructed shape obtained by considering the gyroscope data then resulted in the figure A1-B1-C1-D1. This was an approximate outline of the shape drawn by the manipulator which showed a decent resemblance of the input image.

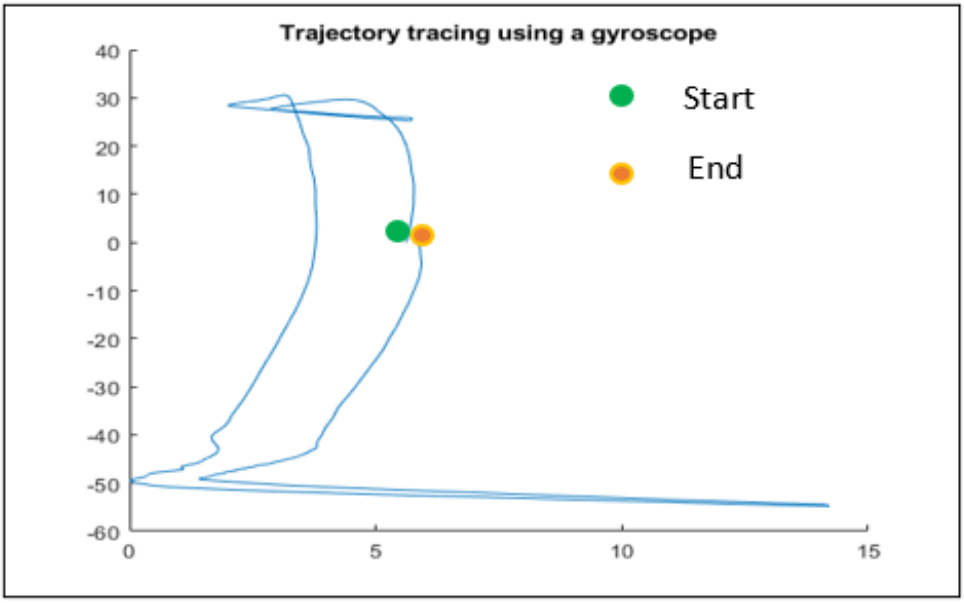

Fig. 15: Trajectory tracking using gyroscope for a sample palletizing task

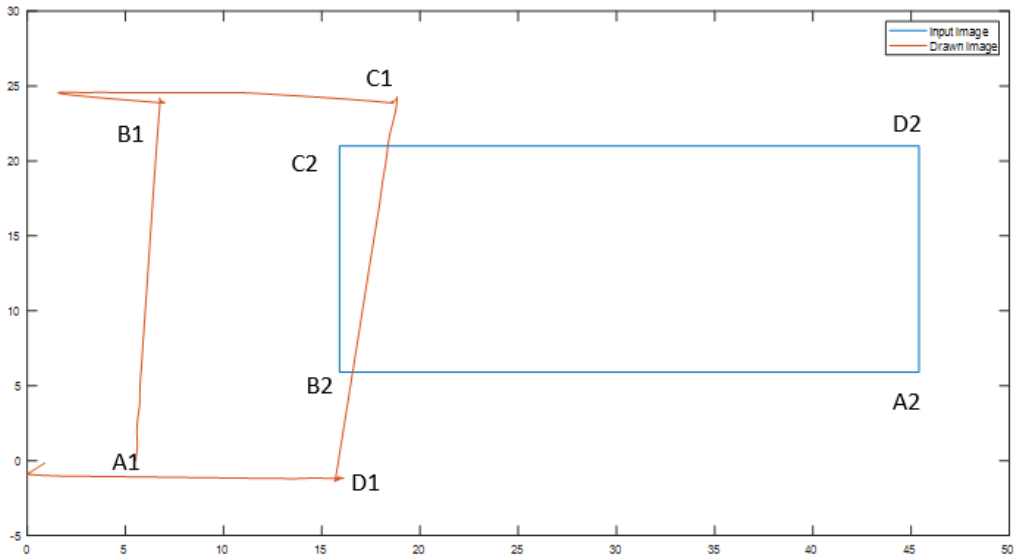

Fig. 16: Trajectory tracking using gyroscope for a sample shape drawing task

5. Summary

This research presented the application of parallel and open-link robots in palletizing and shape drawing tasks based on image processing techniques respectively. For the parallel manipulator, the meArm, a 3-DoF robotic arm was considered while a 4-DoF serial link manipulator (ROT3U) was used for the open-link manipulator task. Previous research in the performance features such as workspace, dexterity and positional accuracy for the two manipulators was also presented. In realizing the palletizing and shape drawing tasks, the forward and inverse kinematics models of the two manipulators were developed and implemented. In both manipulators, the DH-convention was used to solve for the forward kinematics while the Geometric approach was used to solve for the inverse kinematics. Based on the chosen field of application, color-based segmentation was used to distinguish objects in the workspace for the palletizing task while edge detection was used to identify the boundaries of objects within the workspace image for the shape drawing task. This work presented the possibility of using easily accessible and inexpensive manipulators in prototyping standard industrial robots and testing new control algorithms before they are used in real life applications. This can serve as a testbed for how these robots can be used to perform and automate some of the routine, repetitive industrial and art drawing works. The inclusion of feedback control, trajectory generation and speed control in future works can significantly reduce the errors and improve performance in both manipulators.

6. Acknowledgement

The authors would like to thank Prof. Minoru Sasaki, Dr. Harrison Ngetha, Dr. Waweru Njeri and Dr. Joseph Muguro for their invaluable assistance during the research and preparation of this paper. The authors would also like to thank the Japanese Student Services Organization, JASSO for awarding the short-term research scholarship.

References

[1] W. Njeri, M. Sasaki, and K. Matsushita, “Two degree-of-freedom vibration control of a 3D, 2 link flexible manipulator,” *Adv. Sci. Technol. Eng. Syst.*, 2018, doi: 10.25046/aj030649.

[2] A. Radu and S. Mihaela, “Industrial Applications of Image Processing,” *Acta Uiversitatis Cibiniensis - Tech. Ser.*, vol. LXIV, 2014.

[3] H. P. Jawale and H. T. Thorat, “Comparison of open chain and closed chain planar two degree of freedom manipulator for positional error,” *J. Mech. Robot.*, 2014, doi: 10.1115/1.4026329.

[4] M. Sasaki, N. Honda, W. Njeri, K. Matsushita, and H. Ngetha, “Gain Tuning using Neural Network for contact force control of flexible arm,” *J. Sustain. Res. Eng. Vol 5 No 3 J. Sustain. Res. Eng.*, 2020, [Online]. Available: http://sri.jkuat.ac.ke/ojs/index.php/sri/article/view?path=.

[5] Y. G. Zhao, Y. F. Xiao, and T. Chen, “Kinematics analysis for a 4-DOF palletizing robot manipulator,” 2013, doi: 10.4028/www.scientific.net/AMM.313-314.937.

[6] J. J. Craig, *Introduction to Robotics: Mechanics and Control, 3rd Edition*. 2004.

[7] S. kaisJameel and R. R. Manza, “Color Image Segmentation using Automated K-Means Clustering with RGB and HSV Color Spaces,” *Int. J. Appl. Inf. Syst.*, 2012.

[8] M. W.S. and Tatol M., “Perceptual difference in L*a*b* color space as the base for object identification,” 2009, doi: 10.13140/2.1.1160.2241.

[9] A. Kumar and R. Kala, “Geometric shape drawing using a 3 link planar manipulator,” 2015, doi: 10.1109/IC3.2015.7346715.

[10] B. Crnokić and S. Rezić, “Edge Detection for Mobile Robot using Canny method,” 2016.

[11] G. Damaryam, “A Method to Determine End-Points of Straight Lines Detected Using the Hough Transform,” *Int. J. Eng. Res. Appl.*, vol. 6, pp. 67–75, 2016.