Automated Robot with Object Recognition and Handling Features

Amiraj Dhawan 2, Anuradha Bhat 3, Soumya Sharma 4, Harish Kumar Kaura 1

Department of Computer Engineering
Fr. C. Rodrigues Institute of Technology, Vashi
Email- {1h.k.kaura, 2amirajdhawan, 3bhatanu.09, 4sharma.sony}@gmail.com

Abstract- With the advent of new technologies, every industry is moving towards automation. A large number of jobs in industries, such as Manufacturing, are performed repeatedly. These jobs require a lot of human effort. In such cases, there is a need of an automated robot which can perform the repetitive task more efficiently. This paper is about a robot which has object recognition and handling features. The robot will optically recognize the objects and pick and place them as per the hand gestures given by the user. It will have a camera to capture image of the objects and one arm to perform the pick and place function.

Keywords – Gestures, Inverse Kinematics, OpenCV, Servo Motors, Arduino

I. INTRODUCTION

A robot can be defined as an electro-mechanical machine that is guided by a computer program or electronic circuitry. Robots are being used in every field today. They are replacing humans to perform the same job with more accuracy and at a higher speed. Tasks which require objects to be moved from one place to another can be automated by using a ‘Pick and Place robot’. Such a system can be easily used in areas where it is difficult and/or dangerous for humans to work. Humans need only give commands to the robot and it performs the required operation. There are various input methods available, such as Voice command, Text command and command by gestures.

Of these, hand gestures provide the most interactive way of communication with the robot and also provide an efficient real time interface to the user. Interpretation of the gestures involves the use of Image Processing. The pick and place operation is performed by the end effector/robotic arm.

II. EXISTING SYSTEM

A. Stereo Vision System for A Bin Picking Adept Robot –

In bin picking applications, robots are required to pick up an object from a pile of stacked or scattered objects placed in a bin. For performing such tasks, identification of the objects to be picked using a vision system is used. This system [8] identifies the topmost object from a pile of occluded objects and computes its location. This system consists of two modules namely object segmentation module and object localization module. In the segmentation module, an ‘Acclimatized Top Object Threshold’ [ATOT] algorithm is proposed for segmentation of topmost object and in the localization module, the object’s location is estimated by computing the ‘x’, ‘y’, ‘z’ co-ordinates of the object midpoint using a unified stereo imaging algorithm. Using the object location co-ordinates, the validity of the algorithms is experimentally verified for object pick and place operations.

B. Multiple Object Detection for Pick-and-Place Applications –

This system [7] provides a novel approach for detecting multiple instances of the same object for pick-and-place automation. The system uses SIFT to obtain a set of correspondences between the object model and the current image. In order to segment multiple instances of the object, clustering of the correspondences is done, among the objects, using a voting scheme which determines the best estimate of the object’s center through mean shift. The system has two main modules- ‘Feature extraction and matching’ and ‘Object segmentation’. In Feature extraction, significant features are extracted from the object model and the current image, which are then matched using
similarity measures and the best correspondences are retained. In Object segmentation, a registration transform between the model and the segmented object in the current image is computed.

### III. TECHNOLOGIES

**A. OpenCV**

Image processing on the camera video will be done using OpenCV. OpenCV (Open Source Computer Vision Library) is a library of programming functions mainly aimed at real-time computer vision, developed by Intel. Computer vision is concerned with modeling human vision using hardware and software. It combines knowledge in computer science, electrical science and physics. Using OpenCV we can determine how far an object is, what is the shape and color of the object etc. OpenCV has C, C++, python, java interfaces and supports Windows, Linux, Android and Mac OS platforms.

**B. C++**

C++ is a general-purpose programming language with features found both in very high-level languages and in very low-level languages. Its name is in part a pun on the C increment-by-one operator, reflecting its nature as an evolution of the C programming language. C++ adds object oriented programming capabilities to C: the use of classes, virtual methods and multiple inheritance. It also adds reference types, operator overloading, templates, namespaces, and exception handling. Recent enhancements include minor additions to the language and a good many extensions and improvements to standard C++ library.

### IV. IMPLEMENTATION

**A. Robotic Arm**

The arm/end effector of the robot consists of three motors. [1] There is an electromagnet attached to pick up the objects.

![Robotic Arm](image)

**Figure 1. Robotic Arm**

**1. Arduino**

---

862
Automated Robot with Object Recognition and Handling Features

Arduino is an open-source single board microcontroller powered by an Atmel AVR microcontroller. The hardware consists of an 8 bit Atmel AVR based microcontroller, simple input/output support & UART, I2C for communication. Arduino makes use of shields to interface different modules like GPS, Motor Drivers etc. These shields are circuit boards with all the necessary parts required to constitute a module. The use of shields simplifies the interfacing of various modules with the microcontroller.

The software consists of a standard programming language compiler (IDE based on Processing platform) and a bootloader. The programming language used for programming Arduino is based on Wiring language and is similar to C++. The Arduino IDE is a cross-platform application written in java. It includes a code editor with features such as brace matching, automatic indentation and syntax highlighting and it is also capable of compiling and uploading programs to the board with a single click. The bootloader simplifies the task of uploading programs to the on-chip flash memory eliminating the need of an external programmer.

2. Servo Motors –

Servos refer to an error sensing feedback control which is used to correct the performance of the system. Servo or RC Servo Motor is DC motor equipped with a servo mechanism for precise control of angular position. The rotation limit for RC servo motors, usually, ranges from 90° to 180°. Some servos also have rotation limit of 360° or more. But they do not rotate continually. Their rotation is restricted in between the fixed angles.

Servo motors have 3 pin input wires. First pin is +5V power, second cable is GND and third pin is the control pin. Servos are controlled by sending them a variable width pulse. The control wire is used to send this pulse. The parameters for this pulse are- repetition rate, minimum pulse, and a maximum pulse. The duration of a pulse that is applied to the control wire determines the angle. This is called “Pulse Width Modulation”. The servo expects to see a pulse every 20 ms. The length of the pulse determines how far the motor turns. For example, a pulse of 1.5 ms will make the motor turn to the 90 degree position (neutral position). If the input pulse width is 1.25 ms, the servo motor would move to 45 degree position and so on.
B. Inverse Kinematics –

This module converts the movements in Cartesian space to the movements in Joint space. [4]To give the resultant movement of the arm, the degree by which each motor of the robot should move is required. Inverse Kinematics performs this conversion by the following process:

\[
\cos \Theta = \frac{a^2 + c^2 - b^2}{2ac}
\]

For any given triangle having sides \( a \), \( b \) and \( c \),

Now consider the following figure,
Let A, B and C be the three joints of the robot. Let AB be L₁ (link 1) and BC be L₂ (link 2).

In triangle ADC,

\[ \text{Angle D} = 90^\circ \]

[According to Pythagoras theorem, \( DC^2 + AD^2 = AC^2 \)]

Therefore, \( AC^2 = x^2 + y^2 \)

So, \( AC = \sqrt{x^2 + y^2} \)

Now,

\[ \cos \Theta_3 = AD / AC \quad \text{[Cos = base / hypotenuse]} \]

So, \( \cos \Theta_3 = x / (\sqrt{x^2 + y^2}) \)

Also in triangle ABC,

\[ \cos \Theta_1 = (L_1^2 + L_2^2 - (x^2 + y^2)) / 2 * L_1 * L_2 \]

Also,

\[ \cos \Theta_2 = (L_1^2 + x^2 + y^2 - L_2^2) / 2 * L_1 * (\sqrt{x^2 + y^2}) \]

Thus,

\[ \text{Angle CBE} = 180^\circ - \cos^{-1} \left[ (L_1^2 + L_2^2 - (x^2 + y^2)) / 2 * L_1 * L_2 \right] \]

[Angle CBE + \( \Theta_3 = 180 \)]

Also,

\[ \text{Angle BAP} = 45^\circ + [\cos^{-1} \left( (L_1^2 + x^2 + y^2 - L_2^2) / 2 * L_1 * (\sqrt{x^2 + y^2}) \right)] + (\cos^{-1}(x / (\sqrt{x^2 + y^2}))) \]

[Angle BAP = \( \Theta_2 + \Theta_3 + 45 \)]

C. Object Recognition –

1. Shape detection –

   Objects of shapes like circle, triangle and square can be detected by applying OpenCV functions on the image of the objects.

   a. Circle –

      After the image is loaded, it is converted to grayscale. Then a Gaussian blur is applied to the image to reduce the noise and avoid false circle detection. [3] Then HoughCircles transform is applied using “HoughCircles” function which is inbuilt in OpenCV.

   b. Rectangles –
pyrDown function is used which blurs an image and downsamples it. Then pyrUp is used. It upsamples the image and blurs it. The image is then converted from RGB color space image to gray image using cvtColor function which is used to convert the image from one color space to another. Every bit of the image is then inverted using binary_not function. Then a fixed level threshold is applied to each pixel. Then findContour function is used to find the contours of the object. Contours are basically the sequences of points defining a line/curve in an image. DrawContours is used to draw contours outlines or filled contours. approxPolyDP function is used then to approximate a polygonal curve(s) with the specified precision.

For detecting rectangle, cosine of all the angles is calculated. The maximum value is then found. It should ideally be 0 which is not possible. If its value is less than 0.3 then the object is detected as a rectangle.

For detecting triangle, sum of angles of all the corners of the figure (in this case 3) is calculated which is supposed to be 3.1416. If the sum of angles is between the ranges 2.8-3.4 then the object is detected as a triangular object.

These detected shapes are displayed on a screen so that user is able to see what all objects are present in the field.

2. Color detection –
   a. Random pixel Generation –

   Random points can be generated by using rand() function but it is to be ensured that these points lie within the boundary of the figure. For that following procedure is applied.

   i. Random pixel generation in circle –

   In case of circle, the input will be random test points, number of test points, center and radius. As in the case with rectangle and triangle all bounds and ranges are found but the equations used to find those are different. Now to check whether randomly generated points lie within the circle, the equation of the circle is used. If point $(x, y)$ satisfies the equation $(x-a)^2 + (y-b)^2 < r^2$ where $a$ and $b$ are coordinates of the center and $r$ is the radius.
Automated Robot with Object Recognition and Handling Features

The random pixels selected from the object’s image are checked whether they lie in the range of the required color. If 90% of the pixels lie in the required range then the specified color is detected otherwise an error message will be generated “Color Not Found!!”.

ii. Random pixel generation in rectangle and triangle –

Random test points, image and number of random points are input to the function. All bounds (right, left, upper and lower) are initialized. If the test point is greater than the initial right bound (0) then this point will become the new right bound. Thus the pixel with maximum x coordinate value will be chosen as the right bound. Similarly, other bounds are found. Then x-range and y-range is found. Now the ranges and bounds are known, so the randomly generated point can be checked whether it is present within the polygon or not using the function pointPolygonTest.

Figure 7. Color Detection

D. User Command –

There are two ways in which command can be given to the Robot.

1. Finger Counting –

In Finger Counting method, user gives hand gesture in front of the camera. [2] Depending on the number of fingers detected, user’s command is interpreted. There is internal mapping between the number of fingers and color(shape). ‘One’ finger is interpreted as the separator between color and shape. Suppose initially the user gives hand gesture and two fingers are detected, the command is interpreted to mean ‘Blue color’.(Three fingers mean ‘green’, four fingers mean ‘red’) Then the user gives gesture using one finger. Now when the user shows three fingers it is interpreted as ‘Circle’(Two fingers mean ‘square’ and four fingers mean ‘triangle’) Thus the target object is ‘blue circle’.

[3] Blurring is done to the image of the user’s hand to reduce the noise which is followed by Binary Thresholding. The contour of the hand’s image is found. Contour is, basically, not so precise boundary of the object detected. [2] A convex hull will be drawn around the generated contour points. Convexity defect will be used to generate the list of defects in the convex hull. This is in the form of vectors that will provide all the defect parameters which includes defect point in the contour, depth of the defect, start point and end point of the line in the convex hull for which defect will occur.

The convexity defect’s list provides depth parameters for every defect. Defects with the highest depth can be easily extracted from the list. These defects are basically gaps between two consecutive fingers.
Finger count will be one more than the number of defects determined. Like for one defect, finger count will be two. That is, No of fingers= no of defects+1

The figure illustrates the concept of convexity defect. [3]The convex hull is pictured as a dark line around the hand, and the regions labeled A through H are each defects relative to that hull.

2. Finger Tracking –
A second method to give hand gestures as command to the robot is Finger tracking. This method is more interacting and user-friendly as compared to Finger counting. A camera placed on the robot’s head
captures images of all the objects present in its field of view. This image is displayed to the user on the screen. Thus the user has a real time view of the objects available to him/her. There is a second camera present, in front of which the user will give hand gestures. Movement of the user’s fingers is tracked. By moving his fingers in front of the camera, the user is able to select the desired object which he wants to pick. [2]

For detecting the finger, a method known as Otsu Thresholding is used. [3] In this method a histogram with two peaks is created. One peak is for the black background and the second peak is for the user’s hand. This histogram is then thresholded at the center to give two contrasting black and white images. The contour of the hand is created, which is an approximate outline. Then the convex hull is formed. Convex hull is a polygon created around the hand such that all the pixels lie within this region. The distance between any point on the contour and hand gives the convexity defect. Convexity defect gives the start point, end point, depth point and the depth. For a user to select any object, his finger should be stable on that object for a minimum time of 20 frames. Successful object selection is shown on the screen to the user.
E. Feedback System –

A Feedback System is used by the robot to reach the required destination. The distance between the robot’s original position and the target position is given by ‘number of pixels in x and y direction’. Depending on which of the two is more the system instructs the robot to move by that much distance in the required (x and y) direction. This is done continuously till the robot reaches its target. For this, Signal is sent from the computer system to the robot.
This signal can be +1, 0 or -1 for positive error, no error and negative error respectively. This one step (+1/-1) is equivalent to 4 mm of distance. The signal is sent for both x-direction and y-direction. Suppose the robot’s current position (Cartesian co-ordinate system) is (x1, y1). The end effectors have to reach destination (x2, y2) and ((x1<x2), (y1>y2)). This indicates a positive error in the x-direction and negative error in y-direction. The system senses this error and send a signal of +1(x-direction) and -1(y-direction). On receiving this signal, the robot’s end effector move by +4mm in x-direction and -4mm in y-direction. Now the end effector’s new position is sent back to the system and new error values are calculated. There is a user defined area around the target object. Once the end effector reaches inside this area, the system senses ‘no error’ and sends a signal of (0, 0) indicating that no movement is required now and the end effector has reached its target position.
F. Pick and Place Operation

On receiving the input from the Feedback System, the microcontroller Arduino sends a signal to the robot and the electromagnet gets switched on. The arm moves to the target object which gets attached to the magnet. The arm then moves to the destination spot and on receiving the signal from microcontroller again, the magnet is switched off and the object is detached. The arm finally moves back to its original position.

Figure 15. Robotic Arm Reaching the Object

Figure 16. The Metallic object is Picked Up by the Robotic Arm
Automated Robot with Object Recognition and Handling Features

V. APPLICATIONS

A. Manufacturing Industry –
   This automated pick and place capability will be of great use in manufacturing industry. There are certain jobs done manually and repeatedly. In such case there is a need of an automated robot with object handling features which will reduce human effort.

B. Nuclear Industry –
   In nuclear industry it can be used to perform hazardous tasks which humans cannot do. The poisonous and dangerous environment found in such industries makes it difficult for humans to perform work. Hence such robots can be easily sent to perform the required work.

C. Packaging Industry –
   In packaging industries, where objects have to be packed in different packaging containers based on their shape, the automated robot can be easily used to perform this repetitive task.

VI. CONCLUSION

The automated robot with object recognition capability can be efficiently used in various fields to reduce cost and increase the throughput. Gesture commands provide the user an interactive and easy way to control the robot. We have implemented an efficient system for the above design.

VII. REFERENCES

[1] ‘Design And App of A 3 Dof Bionic Robot Arm’ Sebastian Klug, Bernhard Möhl, Oskar Stryk and Oliver Barth
[3] ‘Edge Detection Analysis’ Tzu-Heng Henry Lee, National Taiwan University
[6] ‘Computing Inverse Kinematics with Linear Programming’
[7] ‘Multiple Object Detection for Pick-and-Place Applications’ Paolo Piccinini, Andrea Prati, Rita Cucchiara