



Object Sorting System in Matlab using Robotic Arm

Monal S. Telgote¹, Ganesh S. Sable²

¹PG Student (EC), Dept. of E&TC, Savitribai Phule Women's Engineering College, Aurangabad, Maharashtra, India

²Professor, Dept. of E&TC, Savitribai Phule Women's Engineering College, Aurangabad, Maharashtra, India

(¹10monal@gmail.com, ²sable.eesa@gmail.com)

Abstract- Image processing in today's world grabs massive attentions as it leads to possibilities of Broaden application in many fields of high technology the real challenge is how to improve existing sorting system in the Modular Processing which consists of four integrated stations of distribution, testing, processing and handling with a new image processing feature. Existing sorting method uses a set of inductive, capacitive and optical sensors do differentiate object color. This paper presents a mechatronics color sorting system solution with the application of image processing. Supported by MATLAB code, image processing in which procedure senses the objects in an image captured in real time by a webcam and then extracts color and position information out of it. This information is passed as a signal of sorting commands to the microcontroller and then the depending on this signal will be given to the servomotor that does pick-and-place mechanism. Extensive testing proves that this color based object sorting system works 100% accurate under ideal condition in term of adequate illumination, circular objects' shape and color. The objects tested for sorting are red and black. For non-ideal condition, such as unspecified color the accuracy may reduce.

Keywords- Image processing, MATLAB, object sorting, pick-and-place, industrial robotic

I. INTRODUCTION

A vision based robot has an image acquisition device like a webcam as its eyes. Then we need a processor that can make sense out of those captured images and actuators like dc motors for navigation. One key point to note is that Image Processing has huge computational requirements, and it is not possible to run an image processing code directly on a small microcontroller. Hence, for our purpose, the simplest approach would be to run the code on a computer, which has a webcam connected to it to take the images, and the robot is controlled by the computer via serial or parallel port. The code is written in software that provides the tools for acquiring images, analyzing the content in the images and deriving conclusions. MATLAB is one of the much such software available which provide the platform for performing these tasks. Till now we have been working on images already saved on our computer. But in actual practice, we need to work in real time, i.e., we need to take images continuously from the current environment using a webcam and then process them. Hence, the Image

Acquisition toolbox of MATLAB provides support in this regard.

To start with working in real time, you must have a functional USB webcam connected to your PC and its driver installed. MATLAB has built-in adaptors for accessing these devices. An adaptor is software that MATLAB uses to communicate with an image acquisition device.

II. METHODOLOGY

The project includes a custom made Atmega328 board acting as a main controller. This board will be interfaced to the PC using USB interface. A 5-axis Pick and Place robot using SERVO motors will be build and interfaced to the atmega328 board through PWM lines which will be used to control the pick and place action.

A Matlab based code will interface with a USB web cam on PC/laptop which will sense objects and its colors and according command will be sent to the microcontroller to pick and place object We'll primarily sense three Colors RED GREEN and Blue we'll place red object on first side of conveyer belt, green on other side of conveyer and we'll not touch the blue object so it will fall off from conveyer.

The figure below shows block diagram of a system. The basic theme of this project is object flowing on conveyor are sensed, selected and sorted depending on their color and size. For this, camera used as input sensor, camera is overhead camera which will be mounted on PC, and will be connected to PC by USB. The camera will take a snap and it will feed to PC for color processing. The gripper of robotic arm will pick the object and place it at the pre-defined. This is full automatic process no manual support is needed. The microcontroller used here is with the support of Arduino kit. Thus the real time, continuous object sorting can be done.

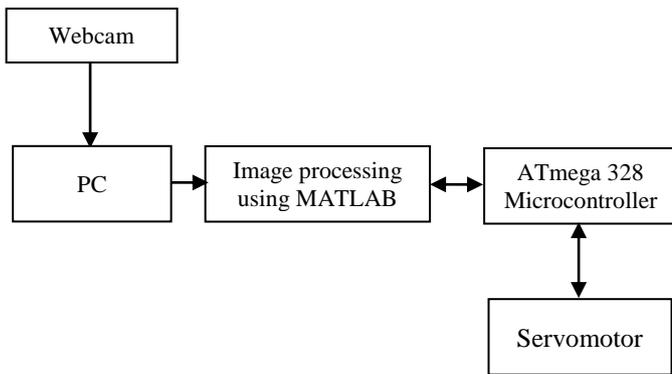


Figure 1. Block Diagram



Figure 3. Color Image

A. Image Processing :

How to Take up Images ?

Important terms and types of Images

Pixel: Pixels are the building blocks of an image. In other words, a pixel is the smallest possible image that can be depicted on your screen.

Binary Image: An image that consists of only black and white pixels.

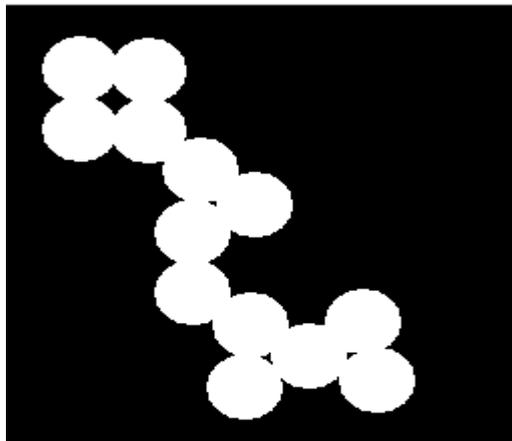


Figure 2. Binary Image

Grayscale Image: It contains intensity values ranging from a minimum (depicting absolute black) to a maximum (depicting absolute white) and in between varying shades of gray. Typically, this range is between 0 and 255.

Color Image: We all have seen this! Such an image is composed of the three primary colors, Red, Green and Blue, hence also called an RGB image.

RGB value: All colors which we see around us can be made by adding red, blue and green components in varying proportions. Hence, any color of the world can uniquely be described by its RGB value, which stands for Red, Blue and Green values. This triplet has each value ranging from 0 to 255, with 0 obviously meaning no component of that particular color and 255 meaning full component. For example, pure red color has RGB value [255 0 0], pure white has [255 255 255], pure black has [0 0 0] and has RGB value [55 162 170].

Representation of an Image in MATLAB: An image in MATLAB is stored as a 2D matrix (of size $m \times n$) where each element of the matrix represents the intensity of light/color of that particular pixel. Hence, for a binary image, the value of each element of the matrix is either 0 or 1 and for a grayscale image each value lies between 0 and 255. A color image is stored as an $m \times n \times 3$ matrix where each element is the RGB value of that particular pixel (hence it's a 3D matrix). You can consider it as three 2D matrices for red, green and blue intensities.

B. Industrial robot and robotic arm

It is defined by ISO as an *automatically controlled, reprogrammable, multipurpose manipulator programmable in three or more axes*. The field of robotics may be more practically defined as the study, design and use of robot systems for manufacturing (a top-level definition relying on the prior definition of *robot*).

1) Defining parameters :

- **Numbers of axes** – two axes are required to reach any point in a plane; three axes are required to reach any point in space. To fully control the orientation of the end of the arm (i.e. the *wrist*) three more axes (yaw, pitch, and roll) are required. Some designs (e.g. the SCARA robot) trade limitations in motion possibilities for cost, speed, and accuracy.
- **Degrees of freedom** which is usually the same as the number of axes.
- **Working envelope** – the region of space a robot can reach.
- **Kinematics** – the actual arrangement of rigid members and joints in the robot, which determines the robot's possible

motions. Classes of robot kinematics include articulated, cartesian, parallel and SCARA.

- Carrying capacity or payload – how much weight a robot can lift.
- Speed – how fast the robot can position the end of its arm. This may be defined in terms of the angular or linear speed of each axis or as a compound speed i.e. the speed of the end of the arm when all axes are moving.
- Acceleration - how quickly an axis can accelerate. Since this is a limiting factor a robot may not be able to reach its specified maximum speed for movements over a short distance or a complex path requiring frequent changes of direction.
- Accuracy – how closely a robot can reach a commanded position. When the absolute position of the robot is measured and compared to the commanded position the error is a measure of accuracy. Accuracy can be improved with external sensing for example a vision system or Infra-Red. See robot calibration. Accuracy can vary with speed and position within the working envelope and with payload (see compliance).
- Repeatability - how well the robot will return to a programmed position. This is not the same as accuracy. It may be that when told to go to a certain X-Y-Z position that it gets only to within 1 mm of that position. This would be its accuracy which may be improved by calibration. But if that position is taught into controller memory and each time it is sent there it returns to within 0.1mm of the taught position then the repeatability will be within 0.1mm.

A robotic arm: It is a type of mechanical arm, usually programmable, with similar functions to a human arm; the arm may be the sum total of the mechanism or may be part of a more complex robot. The links of such a manipulator are connected by joints allowing either rotational motion (such as in an articulated robot) or translational (linear) displacement. The links of the manipulator can be considered to form a kinematic chain. The terminus of the kinematic chain of the manipulator is called the end effector and it is analogous to the human hand

Robotic hand: The end effector, or robotic hand, can be designed to perform any desired task such as welding, gripping, spinning etc., depending on the application. For example robot arms in automotive assembly lines perform a variety of tasks such as welding and parts rotation and placement during assembly. In some circumstances, close emulation of the human hand is desired, as in robots designed to conduct bomb disarmament and disposal.[1]

The 5ARA is a complete self-contained five axis vertically articulated robot arm system designed for bench top automation. The hand terminates in a mounting plate for grippers, and comes standard with electric gripper fitted. The 5ARA uses light weight, high efficiency digital motors driving through steel reinforced polyurethane timing belts and the compact micro-stepping drives providing both speed and

precise control. The 5ARA includes the simple to use controller with fast CPU and DSP processors an incremental optical encoders and an intuitive teach pad. The 5ARA is, fast, accurate and reliable and has a 500mm reach. It is easy to program and is capable of the most complex tasks. The Software makes getting started easy with the programming of the robot, from very simple tasks to the most complex motions, being quickly and easily programmed. Interfaces and peripherals may be programmed, assisted by project manager software in one Windows screen. The Arm comprises two limbs 250mm each plus 360 degrees base rotation, therefore, the workspace is a sphere 1000mm diameter.

The encoders on the 5ARA are miniature optical incremental encoders fitted to each axis and act as watchdogs. The loop is closed after completion of each individual motion which is not the same as servo control. Without encoders the 5ARA will run accurately and indefinitely open loop without error because of the micro-stepping drives and accurate transmission. If the robot has a collision, the controller will not know and will continue with the program, and the robot could be out of position. Such collisions usually only occur accidentally due to programming error, and it would then be necessary to run the calibration routine after such a collision. Software solution is provided for quick calibration using the encoder to accurately read back the position of the robot when de-energized. The robot can be positioned by hand and its position updated by pressing a key on the keypad. The kinematics software then computes the exact X-Y-Z position of the robot with the hand pitch and roll in degrees. [2]



Figure 4. 5 Axis Robotic Arm

C. Servomotor

Servos are DC motors with built in gearing and feedback control loop circuitry. And no motor drivers required. A servomotor is a rotary actuator that allows for precise control of angular position.

What is inside the servo ?: To fully understand how the servo works, you need to take a look under the hood. Inside there is a pretty simple set-up: a small DC motor, potentiometer, and a control circuit. The motor is attached by

gears to the control wheel. As the motor rotates, the potentiometer's resistance changes, so the control circuit can precisely regulate how much movement there is and in which direction.

When the shaft of the motor is at the desired position, power supplied to the motor is stopped. If not, the motor is turned in the appropriate direction. The desired position is sent via electrical pulses through the signal wire. The motor's speed is proportional to the difference between its actual position and desired position. So if the motor is near the desired position, it will turn slowly, otherwise it will turn fast. This is called proportional control. This means the motor will only run as hard as necessary to accomplish the task at hand.

How is the servo controlled?: Servos are controlled by sending an electrical pulse of variable width, or pulse width modulation (PWM), through the control wire. There is a minimum pulse, a maximum pulse, and a repetition rate. A servo motor can usually only turn 90 degrees in either direction for a total of 180 degree movement. The motor's neutral position is defined as the position where the servo has the same amount of potential rotation in the both the clockwise or counter-clockwise direction. The PWM sent to the motor determines position of the shaft, and based on the duration of the pulse sent via the control wire; the rotor will turn to the desired position. The servo motor expects to see a pulse every 20 milliseconds (ms) and the length of the pulse will determine how far the motor turns. For example, a 1.5ms pulse will make the motor turn to the 90-degree position. Shorter than 1.5ms moves it to 0 degrees, and any longer than 1.5ms will turn the servo to 180 degrees, as diagramed below

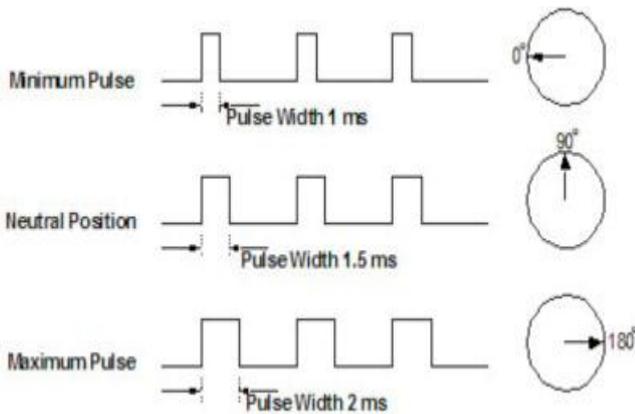


Figure 5. PWM

When these servos are commanded to move, they will move to the position and hold that position. If an external force pushes against the servo while the servo is holding a position, the servo will resist from moving out of that position. The maximum amount of force the servo can exert is called the torque rating of the servo. Servos will not hold their position forever though; the position pulse must be repeated to instruct the servo to stay in position.

Servo Motor Control: The servo motor can be moved to a desired angular position by sending PWM (pulse width modulated) signals on the control wire. The servo understands

the language of pulse position modulation. A pulse of width varying from 1 millisecond to 2 milliseconds in a repeated time frame is sent to the servo for around 50 times in a second. The width of the pulse determines the angular position.

For example, a pulse of 1 millisecond moves the servo towards 0°, while a 2 milliseconds wide pulse would take it to 180°. The pulse width for in between angular positions can be interpolated accordingly. Thus a pulse of width 1.5 milliseconds will shift the servo to 90°.

Power supply for Servo: The servo requires a DC supply of 4.8 V to 6 V. For a specific servo, its voltage rating is given as one of its specification by the manufacturer. The DC supply can be given through a battery or a regulator. The battery voltage must be closer to the operating voltage of the servo. This will reduce the wastage of power as thermal radiation. A switched regulator can be used as the supply for better power efficiency. Learn more about working of a servo motor through exclusive images at the Insight about servo motor.

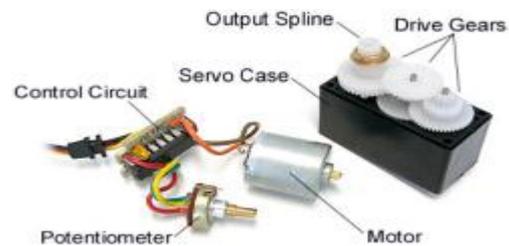


Figure 6. Servo motor

D. Camera

The camera used in this case will be overhead camera, it will take the snapshot of the object. The image captured by the camera will be processed by image processing using Matlab.



Figure 7. USB Webcam

Specifications: USB Webcam/PC webcam/Mini Camera

Features &Spec: Solution: 300K, 350K, 480K, 1.3M pixel available;

Supports MSN messenger, Yahoo Messenger, Skype ETC;

Able to rotate 180 degree, ensure capture from any angle;

Video format: 24-bit RGB;

Interface: USB 2.0 port and down compatible USB 1.1 port;

Transmission rate: 640*480/ 30 fps; 1280*960/ 10~15 fps;

VFW interface: support the seizure of both still and moving image;

Automatic white balance;

Automatic gain control;

Automatic color compensated;

Support Wind 98/ME/2000/XP/Vista etc;

CD-ROM drive;

USB port.

III. APPLICATIONS, ADVANTAGES AND DISADVANTAGES

There are many applications in this sorting system. Mainly this finds the important application in agriculture field where it can be used to sort the different agriculture products like grains, lemons, almonds, grapes, and many more. For human beings it becomes a cumbersome task to sort out the objects with high quality also the possibility of accuracy is less. In industry it can be used for sorting of various objects, tools, with high degree of accuracy and quality with an automation. By this way the proposed project can be used. It finds application in enormous way in agriculture, industry.

Some of the advantages are

1. High efficiency: the sorting speed can reach tons/hour.

2. High precision: the margin of error can be reduced to great extent.

3. Widely working range for round and oval objects or vegetables. Also suit to select pears, orange and other fruits of this kind.

4. High degree of intelligence if used with PLC control. The machine with a high degree of intelligent can control it. It has weighing, logic operations, and statistics function.

5. Good quality and low failure rate with long life.

6. Reliable operation, and maintenance.

IV. CONCLUSION

The project can work successfully and can sort the object depending on their color. Hence for color sorting. Is done in Matlab based code will interface with a USB web cam on PC/laptop which will sense objects and its colors and according command will be sent to the microcontroller to pick and place object. We'll primarily sense three Colors RED GREEN and Blue we'll place red object on first side of conveyer belt, green on other side of conveyer and we'll not touch the blue object so it will fall off from conveyer. There are two main steps in sensing part, objects coding and particular algorithm written for that particular object. The system can successfully perform handling station task, namely pick and place mechanism with help of 5 axis robotic arm.

REFERENCES

- [1] <http://en.wikipedia.org/w/index.php?search=PICK+AND+PLACE+ROBOT&title=Special%3ASearch>
- [2] www.bytronic.net/datasheets/Automation/5ARA.
- [3] Students.iitk.ac.in/.../tutorials/Elementary%20Introduction%20to%20Ima.
- [4] www.electroschematics.com/8064/diy-arduino-making-your-own-arduino
- [5] [www.google.com/wisegEEK.com/whatisoptical sensors.htm](http://www.google.com/wisegEEK.com/whatisoptical%20sensors.htm)
- [6] [www.google.com/osa.org/ sensors](http://www.google.com/osa.org/sensors)
- [7] [www.pdfgenicom.com/compacsort.com /sorters](http://www.pdfgenicom.com/compacsort.com/sorters)
- [8] [www.pdfgenicom.com/indiamart.com/ sorting machine](http://www.pdfgenicom.com/indiamart.com/sorting%20machine)
- [9] [www.google.com/ATMEGA 48A/PA/328/P/DATASHEET SUMMARY](http://www.google.com/ATMEGA48A/PA/328/P/DATASHEETSUMMARY)
- [10] www.google.com/zimbio.com/clip-basecam
- [11] www.google.com/pdfs.com/smps
- [12] www.google.com/pdfgeni.com/vermnlstweb.nl
- [13] www.google.com/arduino.cc/main/software
- [14] www.google.com/eyantra.org/home/project
- [15] Project on— Pick n Place Robot, made by Bharat Jain and Dinesh Rajput, under guidance of Prof. KaviArya, IIT Mumbai, 2010.
- [16] www.pdfgeni.com/servo-tutorial
- [17] [www.pdfgeni.com/matlab- mathematical lab/pdf](http://www.pdfgeni.com/matlab-mathematical%20lab/pdf)