Color Vision Robot

Chandra Sekhar Muramalla, B.MuthuKumaran

***Abstract*— The main purpose is to design a Proto-type of robotic pick and place system that utilize visual perception components. The System mainly consists of image acquisition unit, processing unit and a control unit. Used Raspbery-Pi for processing and controlling because of its capability to work as a single board computer at low cost and Python with Open-CV library is used for image processing because of its reliability and open source. The main purpose behind this is to develop a low cost pick and place robot.**

***Index Terms*—Color recognition, Digital image processing, Open-CV, Pick and place robot, Raspberry-Pi, System on Board.**

I. INTRODUCTION

Robotics is the science of designing and building robots suitable for real life applications in automated manufacturing and other non- manufacturing environments.‖ As per International Standards Organization (ISO), it can also be defined as, ―An industrial robot is an automatic, servo-controlled, freely programmable, multipurpose manipulator, with several areas for the handling of work pieces, tools or special devices. Fig. 1 shows an overview of some sensored based robotic system.

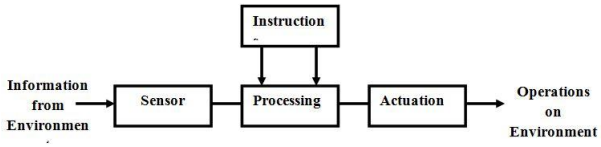

Fig. 1 An Overview of some sensored based robotic Systems.

The potential application areas for vision driven automated systems are numerous. Each brings its own particular problems which must be resolved by a system designer if successful operation is to be achieved. The applications can be categorized broadly according to the processing requirements they impose. Some examples of such applications are

- Object identification
- Picking and placing tasks
- Visual guidance
- Trajectory control
- Visual inspection

The designed robot will detects the colored object, picks and place it in required location.

II. DIGITAL IMAGE

Fig.2 shows the general block diagram of a vision based robotics system. For processing and controlling we used Raspberry Pi .The Raspberry Pi is a series of credit card-sized single-board computers developed in the UK by the Raspberry Pi Foundation with the intention of promoting the teaching of basic computer science in schools. For processing the images we used python with open-cv library and library for accessing GPIO pins.

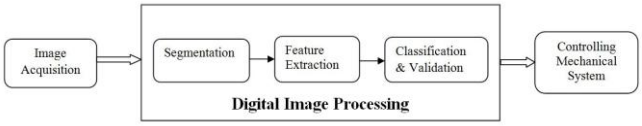

Fig.2 Block Diagram

A digital image is a representation of a two-dimensional image as a finite set of digital values. In image processing, the digitization process includes sampling and quantization of continuous data. The sampling process samples the intensity of the continuous-tone image, such as a monochrome, color or multi-spectrum image, at specific locations on a discrete grid. The grid defines the sampling resolution. The quantization process converts the continuous or analog values of intensity brightness into discrete data, which corresponds to the digital brightness value of each sample, ranging from black, through the grays, to white. A digitized sample is referred to as a picture element, or pixel. The digital image contains a fixed number of rows and columns of pixels. Pixels are like little tiles holding quantized values that represent the brightness at the points of the image. Pixels are parameterized by position, intensity and time. Typically, the pixels are stored in computer memory as a raster image or raster map, a two-dimensional array of small integers. Image is stored in numerical form which can be manipulated by a computer. A numerical image is divided into a matrix of pixels (picture elements).

The different types of Digital images are

- Binary images
- Monochrome images/Gray images/Intensity images
- Color images

We mainly use color images in specific RGB images. An RGB color image records both the brightness and colors of each pixel i.e. for each pixel in an image it records the contents of Red, contents of Green and contents of Blue.

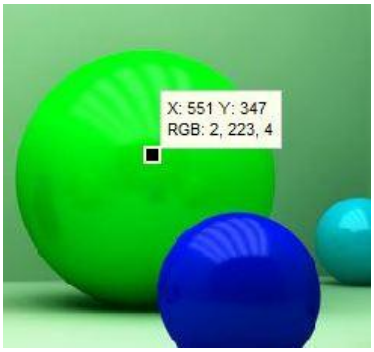

Fig. 3 RGB Color image

ISSN: 2278 – 7798

Fig.3 shows the values of Red, Green and Blue intensities and no two colors will have all the three values same.

III. Color Detection

Color detection is the major and crucial step of this project. The major steps in color detection

Step-1:We used RGB web-camera for capturing the images continuously.

Step-2: We have a RGB Image and we separate the channels i.e. Red channel , Green channel and Blue channel.

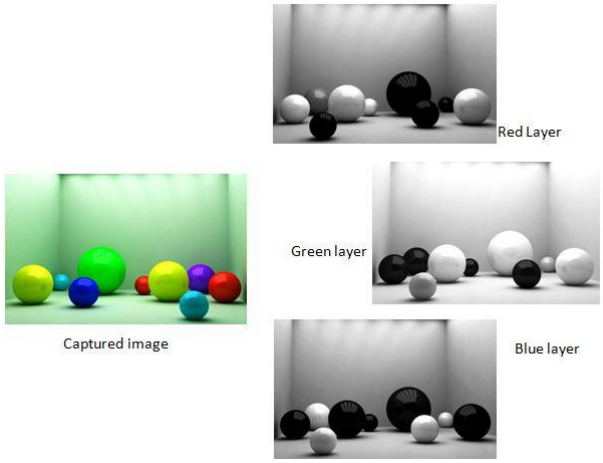

Fig. 4 Channel Separation

Step-3(Thresholding): This is the crucial step because the selection of the intensity will decide the reliability .The major difficulty in color detection is the intensity of the color will depends up on surrounding light i.e. the pixel values are different for the same color captured outdoor and indoor.

Thresholding is the simplest method of image segmentation. From a grayscale image, thresholding can be used to create binary images. Color images can also be thresholded. One approach is to designate a separate threshold for each of the RGB components of the image and then combine them with an AND operation. This reflects the way the camera works and how the data is stored in the computer, but it does not correspond to the way that people recognize colour

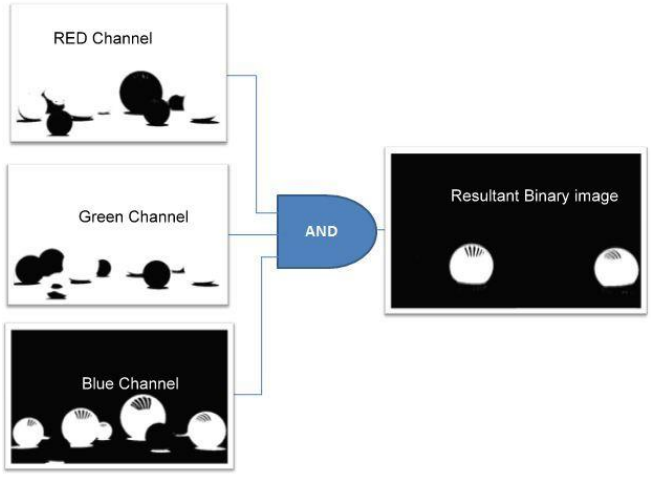

Fig. 4 Multiband thresholding

Fig.4 shows the Multiband thresholding . The simplest thresholding methods replace each pixel in an image with a black pixel if the image intensity I_{i,j} is less than some fixed constant T (that is, I_{i,j}<T), or a white pixel if the image intensity is greater than that constant but for our application we used two thresholds i.e. two constants T1 and T2 if the image intensity is between T1 and T2 then pixel is replaced with white else black pixel Eq. (1) were R{i,j} is the binary image.

$$R\{i,j\} = 1 \text{ if } T1 < I\{i,j\} < T2$$
$$= 0 \text{ else} \qquad (1)$$

Step-4: By performing AND operation between the captured image and resultant image which is shown in fig. 5.

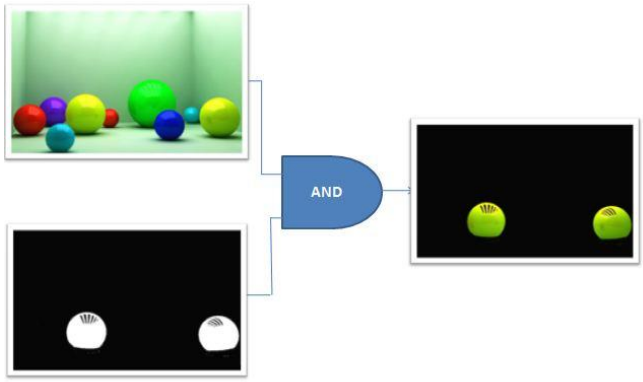

Fig. 5 Resultant Image

IV. Object Tracking

After the detection of the colored object the robot should reach that object , pick and place that object at required location. This will done based up on the flow chart shown bellow.

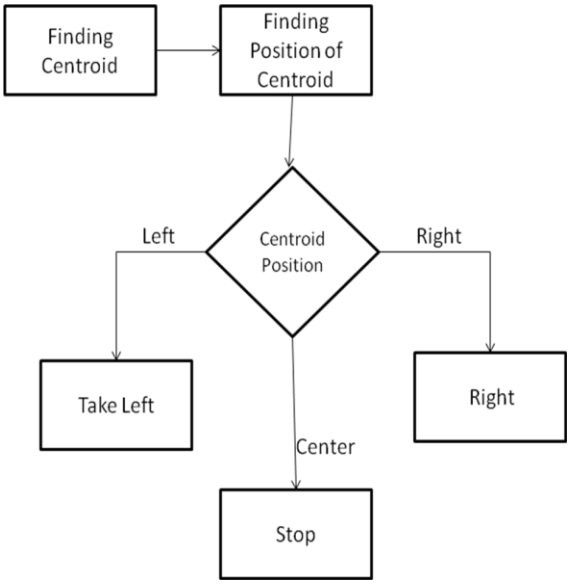

Fig. 5 Steps for object tracking

We the humans know what is left and what is right but the system do not know. So, based up the co-ordinates of the centroid the position will be decided. We took a 640x480 image and divided it into nine regions which is shown in fig.6.

ISSN: 2278 – 7798

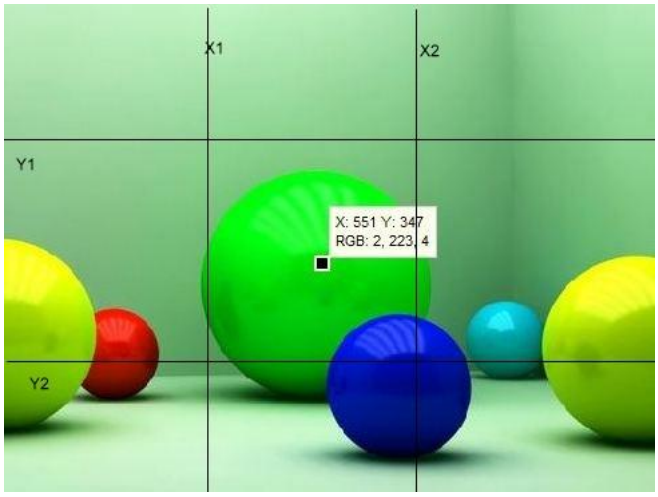

Fig. 6

Were if the centroid X co-ordinate is between 0 and X1 robot will take left , if it is between X2 and 640 it will take right, if the X co-ordinate of centroid is between X1, X2 and Y co-ordinate is between 0, Y1 the robot will move forward and if the position of the centroid is in the center region it will stop and start picking the colored object and the same process will be followed for placing it i.e. from color detection to tracking.

V. RESULTS

Fig. 7 shows the color vision robot were it has Raspberry-Pi board and a camera for capturing the images. The Robot picked the ball and placed it correctly 6 times out of 10 attempts.

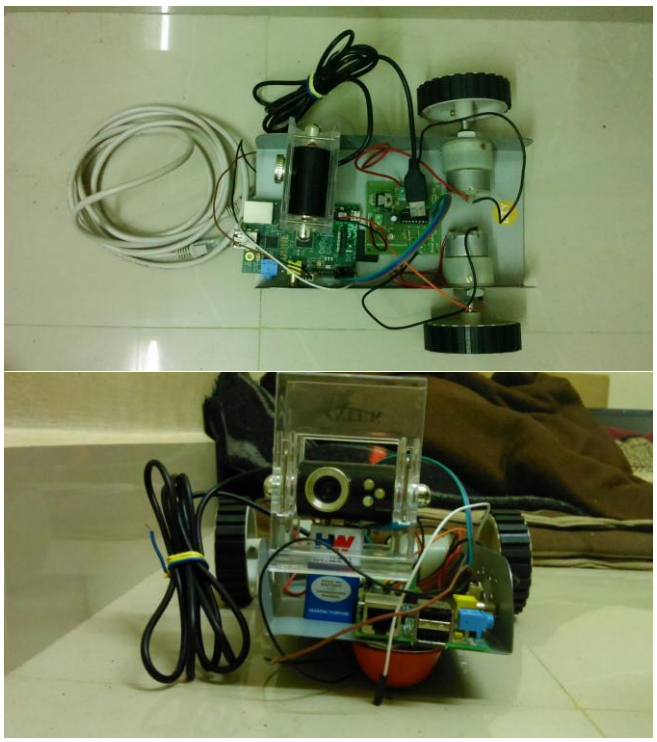

Fig. 7 Color Vision Robot

VI. CONCLUSION

The Robot which we designed satisfy the primary use but if we need it for a real time application the Raspberry-Pi should be replaced with a high end system on board in order to increase the reliability. But at this cost Raspberry-Pi is the best.

For better tracking we can go for stereo vision i.e. using more than one camera.

For increasing the speed of the system we can go C++ programming instead of Python.

The Color recognition is better if we use HSV images instead of RGB because RGB not correspond to the way that people recognize colour.

REFERENCES

[1] G. Bradski and A. Kaehler, Learning OpenCV, O'Reilly Media Inc.2008.
[2] M. Svedman, L. Goncalves, N. Karlsson, M. Munich and P. Pirjanian Structure from stereo vision using unsynchronized cameras for simultaneous localization and mapping, Intelligent Robots and Systems, 2005.
[3] Jayneil Dalal and Sohil Patel, OpenCV Starter, 2013
[4] Hartley, R; Zisserman , Multiple View Geometry in Computer Vision, A. Cambridge University Press, 2003.
[5] Jan Erik Solem, Programming Computer Vision with Python, O'Reilly Media Inc.2013.
[6] Multiple View Geometry in Computer Vision. - Hartley, R; Zisserman, A. Cambridge University Press, 2003.
[7] opencv-python-tutroals.readthedocs.org

Chandra Sekhar Muramalla, Pursuing M.Tech.-Embedded System Technology, Dept. of Electronics and Communications, SRM University, Kattankulathur, Chennai.
B.Muthu Kumaran, Asst.Professor, Dept. of Electronics and Communications, SRM University, Kattankulathur, Chennai.

ISSN: 2278 – 7798