

Object Tracking Robot on Raspberry Pi using Opencv

Pradeep Kumar.G.H

Assistant Professor, Department of CSE, KSIT

Rupa Gurrum

SweathaSuresh.B.

Sneha.B.R.

Sushmitha.R.

Students, Department of CSE, KSIT

ABSTRACT

Autonomous vision based robots are intelligent robots which take visual data, process it and provide appropriate output. These robots are totally independent and don't need any kind of human intervention since they are pre-fed with instructions. A robot is designed on Raspberry Pi using OpenCV, which is used for object detection based on its colour, size and shape. Here only a single object is being detected at a time. The tracking of the object is based on division of the image into virtual grids. The movement of the robot is based on the position of the object in the grid. An arm is incorporated to pick the object once it is stationary. After picking the object, it would be dropped into its respective coloured container which has a predefined position.

Keywords- OpenCV, Raspberry Pi, Virtual Grids, RDP Desk, GPIO, Contours.

1. INTRODUCTION

According to the *Oxford English Dictionary*, the word **robotics** was first used in print by Isaac Asimov, in his science fiction short story "Liar!". Asimov was unaware that he was coining the term and he assumed **robotics** already referred to the science and technology of robots. Stories of artificial helpers, companions and attempts to create them have a long history, but fully autonomous machines only appeared in the 20th century.

Artificial Intelligence (AI) powers the evolution of robotics in the modern era. The core of robotics is AI and computer science is a building block of AI. Highly intelligent robots and robots with AI are fruits of continuous research and years of development. The present project was time and resources bound to build a robot with high level AI; hence, a simple, yet an intelligent robot which could act on real time information was built.

Image Processing has been the talk of e-world for some years. Its varied applications have found its uses in almost every field. In Electrical Engineering and Computer Science, image processing is any form of signal processing for which the input is an image, such as photographs or frames of video; the output of image processing can be either an image or a set of characteristics or parameters related to the image. Most image-processing techniques involve treating the image as a two-dimensional signal and applying standard signal-processing techniques to it. **Robotic Vision** deals with image processing and Computer Vision. The whole

idea behind robotic vision is: viewing an object from the robot's perspective, deciding on object of interest and to act accordingly. It is about giving artificial sight to robots. The implemented robot can be used in various chemical industries, military application, coin separation, pencil industry.

2. HARDWARE DESIGN

2.1 List of Hardware

A chassis is used as a base on which following hardware components are fit:

- Raspberry Pi (rev B) for GPU and CPU computations.
- Wi-Fi dongle to connect to Pi remotely.
- Motor driver which can control two motors
- Batteries to provide power.
- Jumper wires to connect individual components.
- Web camera.
- Servo motor to control the arm of the robot.

2.2 Hardware and Software Description

2.2.1 Raspberry Pi

The Raspberry Pi is a credit card-sized single-board computer. There are currently five Raspberry Pi models in market i.e. the Model B+, the Model A+, the Model B, the Model A, and the Compute Module. All models use the same SoC (System on Chip - combined CPU&GPU), but hardware features differ.

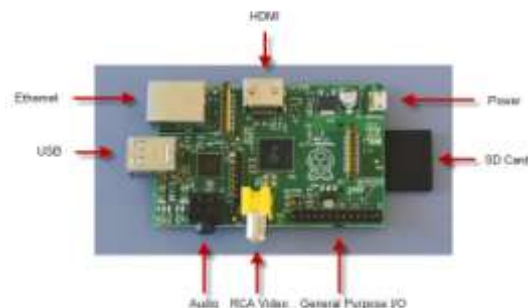


Fig 1: Features offered in Raspberry Pi Model B Rev

2.2.2 Web Camera

A webcam is a video camera that feeds or streams its image in real time to or through a raspberry pi. The

quality and configurability of the camera module is highly superior to a standard USB webcam.

2.2.3 Raspbian OS

Raspbian OS is available for Raspberry Pi, Raspbian comes out on top as being the most user-friendly, best-looking, has the best range of default softwares and optimized for the Raspberry Pi hardware. Raspbian is a free operating system based on Debian (LINUX), which is available for free from the Raspberry Pi website.

2.2.4 Python

Python is a widely used general-purpose, high-level programming language. Its syntax allows the programmers to express concepts in fewer lines of code when compared with other languages like C, C++ or java.

2.2.5 RPi.GPIO Python Library

The RPi.GPIO Python library allows you to easily configure and read-write the input/output pins on the Pi's GPIO header within a Python script. This package is not shipped along with Raspbian.

2.2.6 OpenCV

It (Open Source Computer Vision) is a library of programming functions mainly aimed at real-time computer vision.

2.2.7 Numpy

Numpy is a package that defines a multi-dimensional array object and associated fast math functions that operate on it. Numpy is an extension to the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large library of high-level mathematical functions to operate on these arrays.

2.2.8 RDP Desk

Remote Desktop Protocol (RDP) is a proprietary protocol developed by Microsoft, which provides a user with a graphical interface to connect to another computer over a network connection. The user employs RDP client software for this purpose, while the other computer must run RDP server software.

2.2.9 Servo Motor

A **servomotor** is a rotary actuator or linear actuator that allows for precise control of angular or linear position, velocity and acceleration. Servomotors are not a specific class of motor although the term servomotor is often used to refer to a motor suitable for use in a closed-loop control system.

2.2.10 Wi-Fi dongle

The Wi-Fi dongle is attached to the USB port in Raspberry Pi in order to connect to it wirelessly. Since raspberry pi needed its own IP, it needs to be connected to a Wi-Fi router or Hotspot [7]. For the same we need to make some changes in the field specified so as to make raspberry pi recognize the router every time it boots up. Navigate to the file "/etc/network/interfaces" and add following lines to make the PI connect with your router.

```
iface wlan0 inet dhcp
wpa-ssid "Your Network SSID"
wpa-psk "Your Password"
```

3. ARCHITECTURE

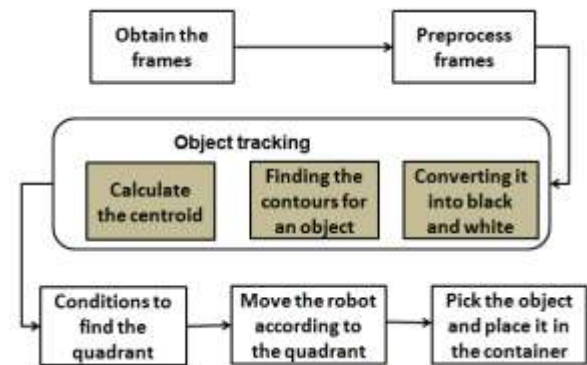


Fig 2: Architecture diagram

A Live video is captured from the webcam. The pre-processed frames are then converted to black and white, object is identified by finding the contours of the object and then, centroid of the object is calculated. Pre-processing of the frames include noise removal and conversion to grey scale. The frames are divided into virtual grids which help the robot to move to the object based on its centroid. Once the robot reaches the object and finds the object is stationary for few frames, it picks the object. The colour of the picked object is analysed and then the object is placed into its respective coloured container.

4. QUADRANT BASED ALGORITHM

Step 1: Capture live video.

Step 2: Pre-process the frames. This step includes converting of the frames to gray scale and also removing the noise from the frames.

Step 3: Further converting the frames to black and white frames.

Step 4: Black and white frames are then virtually divided into grids as shown in the figure 3.

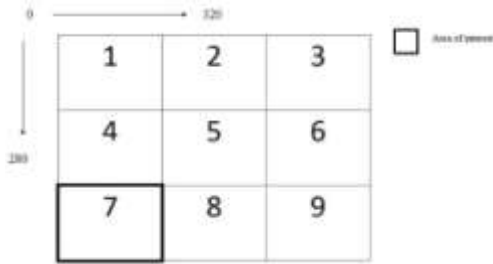


Fig 3 : grid

These grids are virtual and help in identifying the position of the object.

Step 5: Object is identified by finding out its contour[6] as shown in fig 4 which is found by the following.

```
contours1, _=cv2.findContours(im_bw,cv2.RETR_TREE,cv2.CHAIN_APPROX_NONE)
```



Fig 4: contour

Using contours we find the centroid of the object, which is calculated by the formula given below:

```
M=cv2.moments(im_bw)
centroid_x = int(M['m10']/M['m00'])
centroid_y = int(M['m01']/M['m00'])
```

By the use of the centroid we find the actual position of the object(in which quadrant it lies).

Step 6: The main idea is to ensure that the object lies in the 7th quadrant. When the object moves away from 7th quadrant, suitable signals are sent to the robot to take action accordingly. Robot movement is given according to the direction shown in the figure 5. The arrows depict the movement of the robot based on the position of the object.

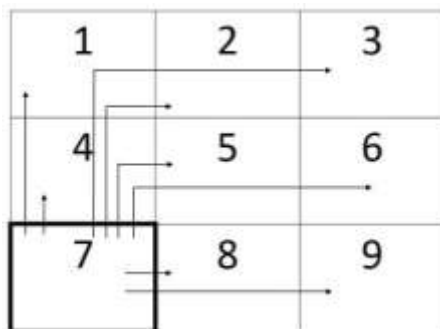


Fig 5: Schematic representation of robot movement

Step 7: Once the Robot finds that the object is stationary for few frames, it picks the object and analyses its colour.

Step 8: Robot places the object into its respective coloured container.

5. MODES OF OPERATION

5.1 AUTOMATIC MODE

The complete work of tracking and placing the object into its respective coloured container is done by robot.

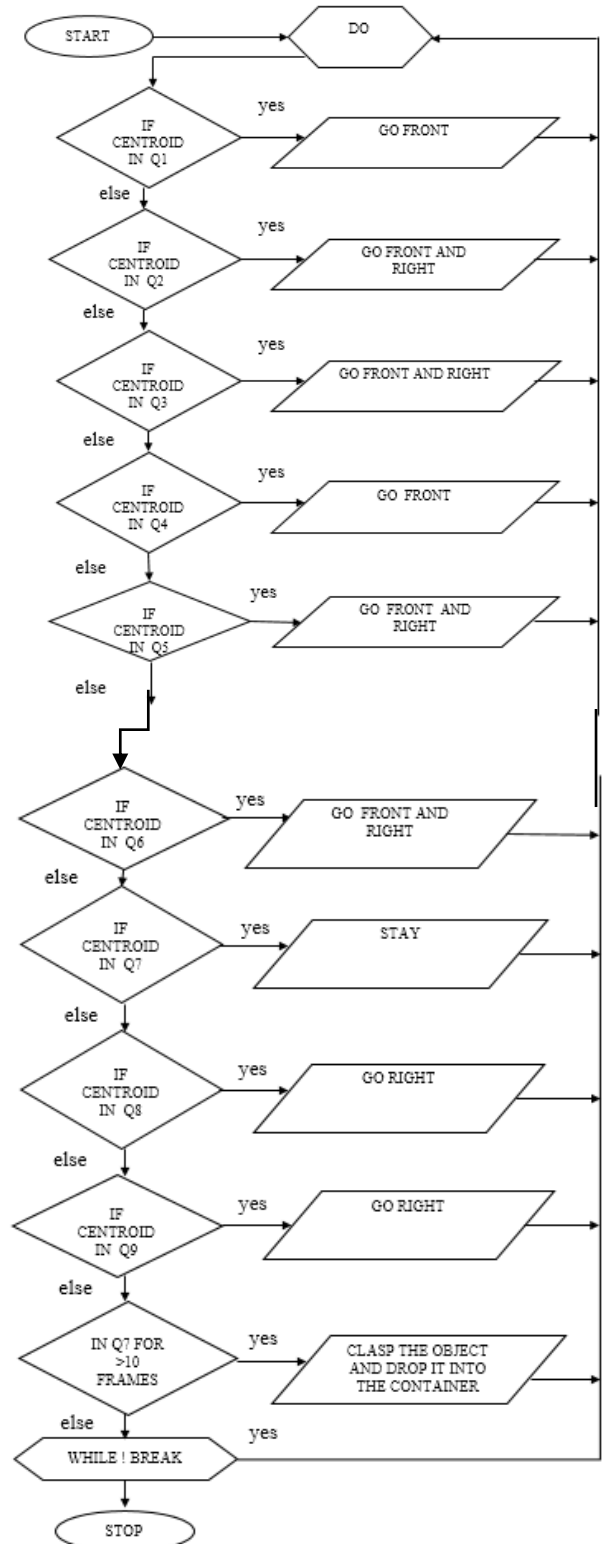


Fig 6: Flowchart

Quadrantbased algorithm is used to achieve the entire task of picking and placing the object into its respective coloured container. Figure 6 shows the conditions for the robot movement.

5.2 MANUAL

The commands are given to the robot through an application to perform the desired task.

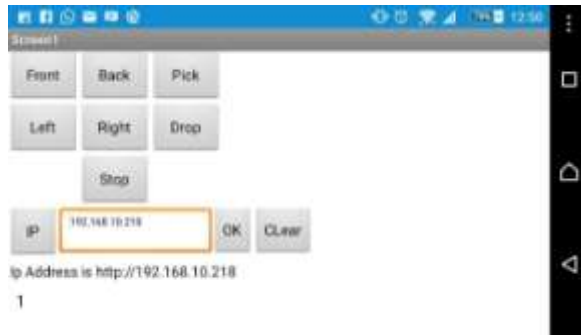


Fig 7: Frontend of App

Controlling the movement of the robot can be done using some kind of remote control interface. It may be web interface or a mobile interface. Since setting up python and SSH in raspberry pi is very easy without any cons, it is recommended to use a mobile based user interface for controlling the robot motion. One of the most common approach which is used to send command to the server is AppInventor. Frontend of the App which is shown consists of a slot to enter the IP address of the sever to connect to the raspberry pi. Various controlling buttons are used to send the signals to the server and that will execute the corresponding controlling action.

6. RESULT

At the end of the automatic mode, robot tracks, analyses the colour of the picked object and drops the object into the respective coloured container.

At the end of manual mode, robot moves and does the task as desired by the user according to the commands given through the application(app).



Fig 8: Robot picking the object

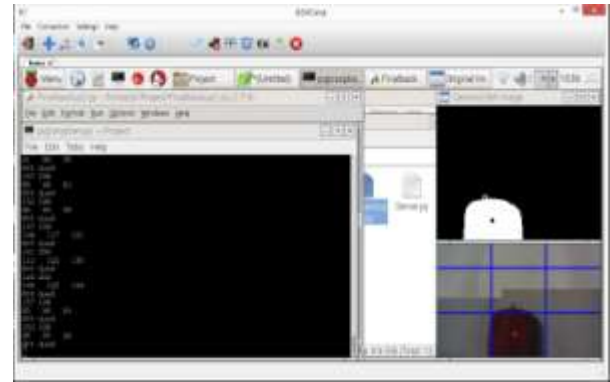


Fig 9: Snapshot of image processing

7. FUTURE ENHANCEMENT

- Tracking of multiple objects.
- Position of container can be autonomous.
- Area of coverage can be expanded.

8. ACKNOWLEDGEMENT

We would like to thank VGST (Vision Group on Science and Technology), Government of Karnataka, India for providing infrastructure facilities through the K-FIST Level II project at KSIT, CSE R&D Department, Bengaluru

9. REFERENCES

- [1] DhruvPande, Chhavi Sharma, VikasUpadhyaya, "Object Detection and Path Finding Using Monocular Vision", Inc. 2014 IEEE.
- [2] Hyung-Bok Kim and Kwee-Bo Sim, "A Particular Object Tracking In An Environment Of Multiple Moving Objects", International Conference on Control, Automation and Systems 2010.
- [3] MaimunHujaHusin, FauzilianoOsma, "Development of shape pattern recognition for FPGA-based object tracking system" Inc 2014.
- [4] Ana Maria Ocana, Franciscocalderon, "preliminary studies on the taxonomy of objects tracking algorithm in video sequences" Inc. 2013.
- [5] https://en.Wikipedia.org/wiki/Green%27s_theorem
- [6] Jaya P and GeethuBalakrishnan, Contour Based Object Tracking, IJCSIT 2014.
- [7] Narayan Pandharinath Pawar & Minakshee M. Patil, Driver Assistance System based on Raspberry Pi, International Journal of Computer Applications (0975 – 8887) Volume 95– No.16, June 2014, pp. 36-39.