

# Object Detection and Displacement Integrated with RoboRealM

Pavan Sathish Ch,<sup>1</sup> D.Lalitha Bhaskari,<sup>1</sup> D N D Harini<sup>2</sup>

<sup>1</sup>Dept. of CS&SE Andhra University, Visakhapatnam, India  
pavansatish\_123@yahoo.co.in, [lalithabhaskari@yahoo.co.in](mailto:lalithabhaskari@yahoo.co.in)

<sup>2</sup>Dept. of CSE, GVPCEW, Visakhapatnam, India  
[hariniauphd@gmail.com](mailto:hariniauphd@gmail.com)

**Abstract:** *This paper illustrates the design and implementation of computer vision algorithms for object detection combined with object displacement. RoboRealM software is used for object detection which is interfaced to the serial communication along with ATmega328 controller and a gripper which helps in controlling the movements of the robot based on the state of output. The detection process involves perception of color, shape and size of the object which facilitates the robot to pick and place the object at a particular location. This paper incorporates supervised Artificial Intelligence techniques for better detection and movement of the object.*

**Keywords:** *Object detection, Supervised learning, RoboRealM, ATmega328 controller, Gripper, Object displacement*

## 1. Introduction

The nature of the mathematical tools and software available makes a large difference in the kind of things that can be done both in theory and in practice in the field of Computer vision and Robotics. Any software which scales down or lessens the complexity or which makes simulation easy may be regarded as useful. In this paper an attempt to integrate computer vision algorithms for object detection [1] along with object displacement is studied and implemented. The work in this paper involves both hardware and software programming. RoboRealM[1] software is used for Object detection and mini conveyer belt, camera and robotic grippers are used as hardware. ATmega328 controller is used to coordinate between the software and the hardware for efficiently detecting and displacing the object.

RoboRealM is a software application for use in computer vision, image analysis and robotic vision systems that provides us the ability to rapidly process the images in order to visually detect field elements. ATmega328 controller [4] is one of the important hardware components which are used to communicate between the software and gripper. Logitech camera is used for viewing the object and Gripper is used to hold and displace the Object [7]. A detailed discussion about the hardware and software components is mentioned in section 2. Section 3 discusses the adopted methodology while Section 4 deals with the experimental phase. Section 5 concludes the work along with future scope followed by References.

## 2. Software and hardware description

### 2.1 Hardware Components:

The different hardware components used in object detection and displacement are shown in Fig 1. A description of each component is given in Fig 1.1 to Fig 1.4. The main hardware components as shown in Fig 1 are 1) Gripper for holding the object 2) Spur and worm gears to drive the gripper 3) Logitech camera for object viewing 4) Relay board for power controlling and a 5) Mini conveyer belt for carrying objects[5].

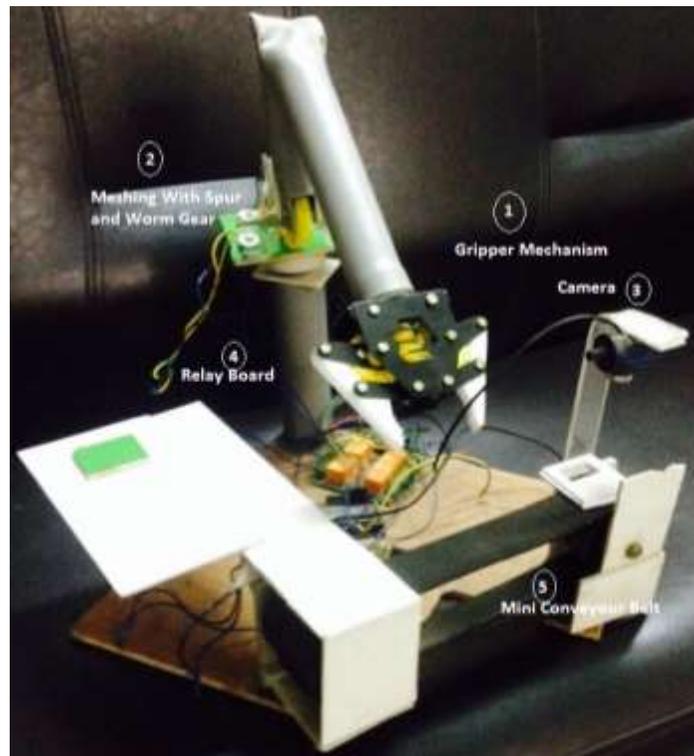


Fig 1: Basic hardware unit developed for this work

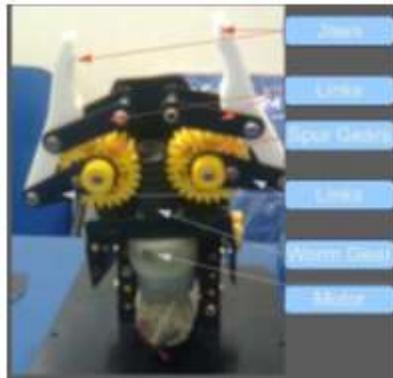


Fig 1.1: Gripper



Fig 1.2: Spur Gear and Worm gear coupled



Fig 1.3: Logitech Ultra Vision Camera

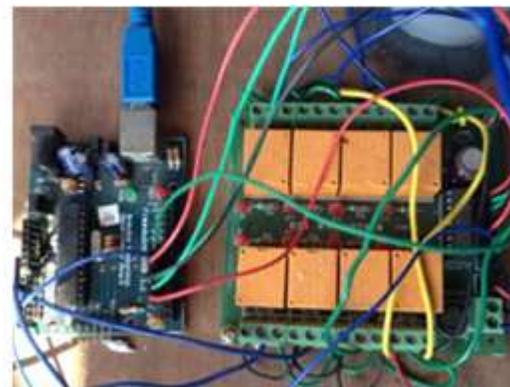


Fig 1.4: ATmega328 controller interfacing with Relay

- The Gripper as shown in Fig 1.1 enables to grip the object comfortably with the help of jaws. The spur and worm gears as shown in the Fig 1.2 have straight teeth and is mounted on parallel shafts to enable circular movement of the arm. Worm gear helps the spur gears to move in opposite directions. In this experiment we have used Logitech Ultra Vision Camera with a resolution of 2 Mega pixels for capturing the images of the object as shown in Fig 1.3
- The main hardware component used in this work is as shown in Fig 1.4 which is Arduino Uno microcontroller board [3] based on the ATmega328 controller. It has 14 digital input/output pins out of which 6 can be used as pulse width modulation(PWM)outputs, 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. We can simply connect it to a computer with a USB cable or power it to get started using an electrically operated relay switch.

## 2.2 RoboRealM Software:

RoboRealM [1] is the software used for object detection in this work by tracking coloured objects. This software processes the acquired images, analyses, identifies and sends the needed signals to our robot's motors. Using Windows based GUI an easy click interface, RoboRealM simplifies computer vision programming. RoboRealM as a part, views the robot's environment, process the acquired image, analyse what needs to be done and sends the needed signals to our robot's motors, servos, etc. The main aim behind using RoboRealM is to translate visual input into actuator commands that can be used to move the robots or trigger actions based on what a machine perceives. The experimental setup of both hardware and computer vision software which is numbered 6 is as shown in Fig 2.

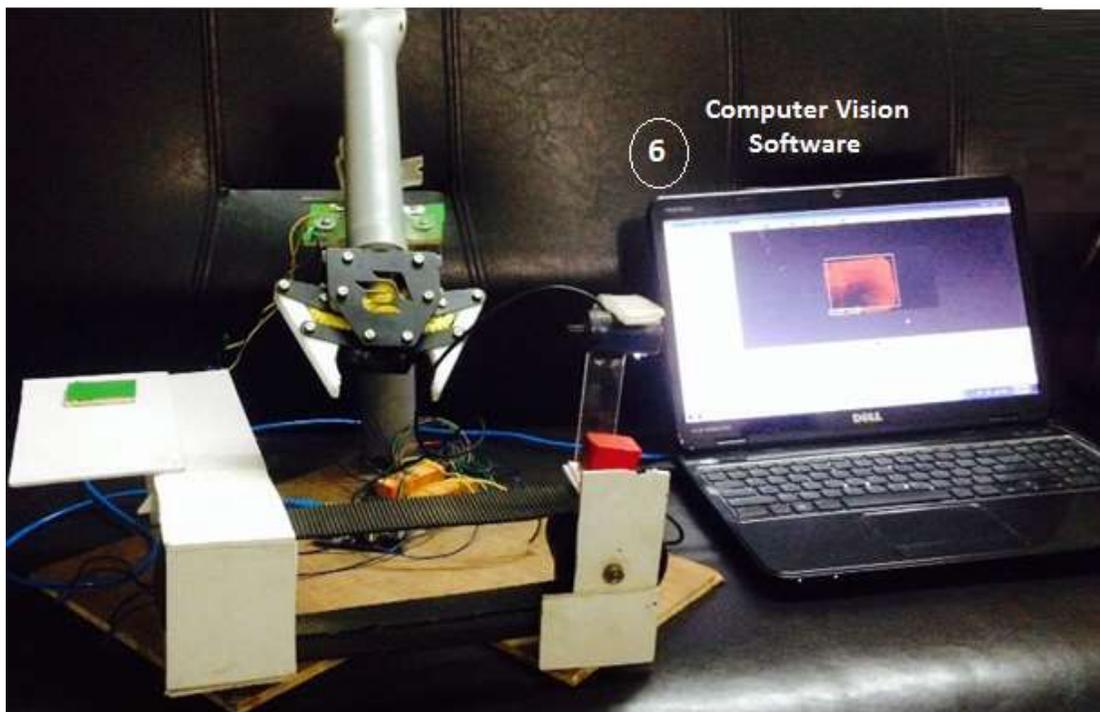


Fig 2: PC interfaced with hardware unit

## 3. Methodology

Initially, the objects shall be placed over the mini conveyor belt one after the other in sequence. A camera which is hooked above to the belt captures the image of the object and delivers it to the computer through serial port communication. With the help of RoboRealM  $x, y$  coordinate values of the object's blob are generated to compute the mean result and then transmit it to the microcontroller (ATmega328p) which arrives at a conclusion to where the Robotic arm gripper should displace the object.

The software processing involves RGB filtrations, Calculating blob size, determining smooth hull, finding centre of gravity which are shown below in Fig 3 and as explained in sections 3.1 to 3.5.

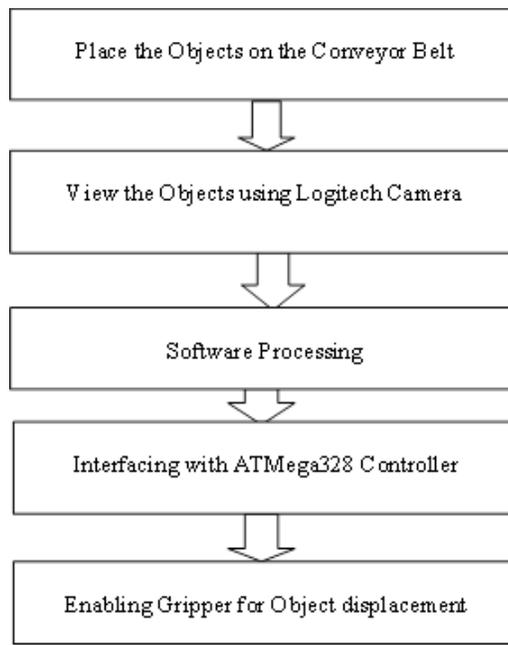


Fig 3: Steps in involved in the Object detection and displacement

### 3.1. RGB filtrations

The primary RGB colours can be focused attention by the RGB Filter [2] using RGB values. So, in the whole computer vision the RGB filter takes part in filtering the objects. Depending upon the colour chosen [8], this filter will impair all the pixels that are not of the colours opted. This function differs from RGB channel in which the white pixels are also attenuated even though they may comprise the colour selected.

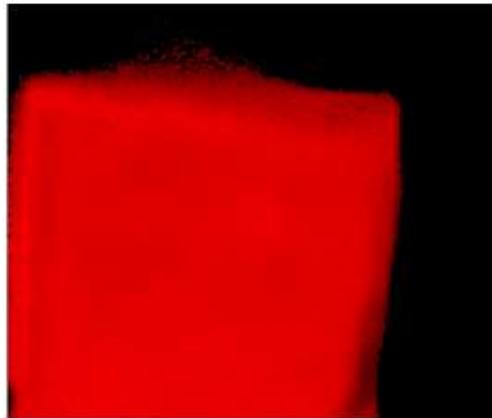


Fig 3.1: RGB Filtration

For example, if Red is chosen:

$$\begin{aligned}
 R &= ((R-B) + (R-G)) & (1) \\
 G &= 0 \\
 B &= 0
 \end{aligned}$$

With respect to maximum red value, R is normalized.

### 3.2. Blob Size

In order to vanish with the objects below certain size, Blob size filter[6] can be applied. The Threshold histogram visualized in the Blob Size interface is used to threshold the picture into object/no object values. Once objects have been determined we use the 'size histogram' slider to withdraw objects below a certain pixel size and we have to notice that the pixel size is the objects pixel area.

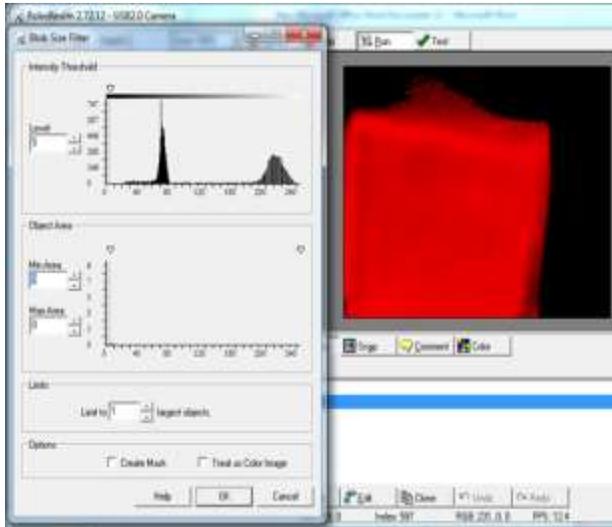


Fig 3.2: Smoothing the Object

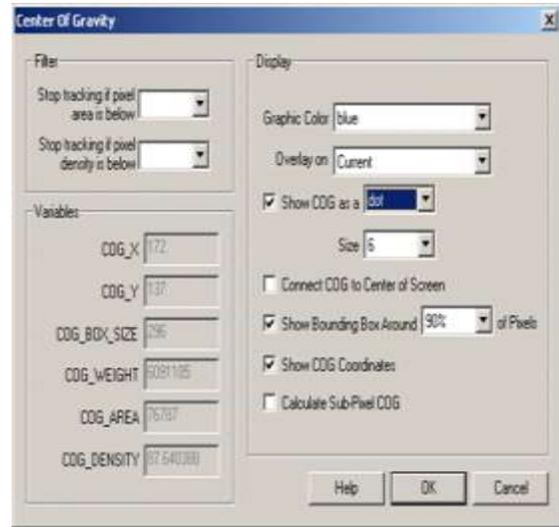


Fig 3.3: Locating the Object boundary with blue colour

### 3.3. Smooth Hull

The smooth hull module aids at smoothing the blobs shape. The perimeter of the blobs is averaged within the defined window. The final outline drawing is then weighted against the new averaged outline and the original outline. A 100% weight substitutes the original outline with the averaged one. The outcome of the module is to smooth the perimeter of the blob without depletion of edges.



Fig 3.4: Smoothing the Hull.



Fig 3.5: Removing the additional pixels and smoothing

### 3.4. Center of Gravity

The Center of Gravity or Center of Mass statistics computes where the COG of the image exists. The COG is calculated by:

$$COG\_X = COG\_X + (I*x) \tag{2}$$

$$COG\_Y = COG\_Y + (I*y)$$

$$Total = Total + I \tag{3}$$

For each pixel where  $I = (R+G+B)/3$  and  $x, y$  is the location of the current pixel. The produced COG value is then divided by the Total value:

$$COG\_X = COG\_X/Total$$

$$COG\_Y = COG\_Y/Total \tag{4}$$

to result in the final x, y location of the COG. Based on the way this COG is calculated we have to observe that brighter pixels will exert more displacement on the final COG location than darker pixels. The COG module interface also provides for diverse graphical overlays.

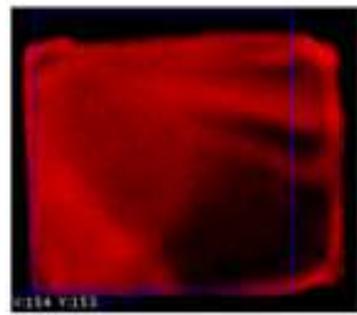
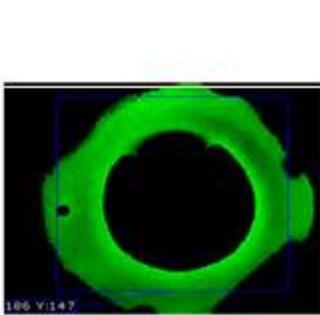


Fig 3.6: Center of Gravity for Green coloured object

Fig 3.7: Center of Gravity for Red coloured cube

### 3.5. Interfacing with ATmega328 Microcontroller

The serial module is used to perform communication from RoboRealM to serial based microcontrollers. Before commencing on using the serial module, ensure that the particular device is not already supported by another RoboRealM module as that will be easier than understanding and carrying out a specific device's protocol. The serial module allows us to configure the serial port for various communication parameters like baud rate, stop bits, etc. and specify the communication protocol for initializing, transmitting and acquiring data. The text boxes permits the PC to input the modulated command numbers for each respective sequence type such as read or write operations.

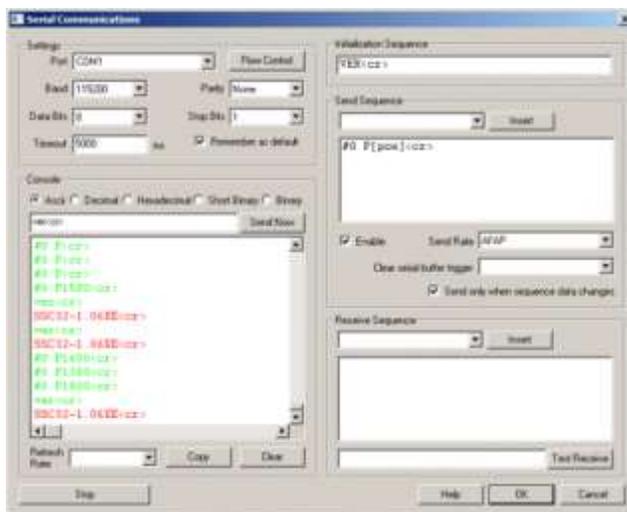


Fig 3.8: Serial Processing the data via UART.

The crux of the entire work lies in the decision making process ATmega328 microcontroller which suggests the motor to drive depending upon the object features.

## 4. Experimental Results

In this work we have taken 10 objects of different sizes, shape and colors for which the screen shots are given in table 1.

TABLE I: Step by step arrangement of entire images of object displacement.

Sno.	Image	Remarks
1		Closing the gripper to hold the object
2		Gripper holding the object and lifting up
3		Gripper placing the object at a particular location
4		Gripper releasing the object at a particular location
5		Gripper moving up
6		Gripper moving back to its original position

## 5. Conclusion and future Work

In the present work we have utilized DC motors since it is my initial research in the robotics field. During the experimentation phase it is found that calibration problems have occurred by using DC motors and hence we would like to develop it by means of Servos and Stepper motor to get more accurate calibrations. This work is further extended to identify the objects in a flying quadcopter.

## 6. References

- [1] Detecting object "RoboRealM" robotics and artificial intelligence and  
[http://www.RoboRealM.com/tutorial/color\\_object\\_tracking\\_2/slide010.php](http://www.RoboRealM.com/tutorial/color_object_tracking_2/slide010.php)
- [2] Mat lab Object recognition in filtration technique  
<http://in.mathworks.com/help/vision/examples/tracking-cars-using-gaussian-mixture-models.html>
- [3] Real Time Embedded systems for robotics using "Open source technology"  
<http://playground.arduino.cc/Projects/Ideas>
- [4] M. A. mazidi , "Embedded systems and 8051 controller",pp. 10-34,1994 .
- [5] J. Greenhalgh, and M. Mirmehdi, "Recognizing Text-Based Traffic Signs," Transactions on Intelligent Transportation Systems, vol.pp, no.99, pp.1-10, 2014.
- [6] D.M. Bulanon, T. Kataoka, S. Zhang, Y. Ota, T. Hiroma, "Optimal thresholding for the automatic recognition of apple fruits", ASAE Paper, No.01-3133, 2001.
- [7] A. Plebe, G. Grasso, "Localization of spherical fruits for robotic harvesting", Machine Vision and Applications, vol.13, no.2, pp. 70-79, 2001.  
<http://dx.doi.org/10.1007/PL00013271>
- [8] D.M. Bulanon, T. Kataoka, Y. Ota, T. Hiroma, "A color model for recognition of apples by a robotic harvesting systems", Journal of JSAM, vol. 64, no.5, pp.123-133, 2002.