RESEARCH ARTICLE OPEN ACCESS

Robot Navigation Using MATLAB'S Bounding Box Algorithm

Vishal Thakur [1], Vaibhav Khedkar [2], Ajay Patkar [3], Akshay Chidrawar [4]
Department of Electronics Engineering
Vivekanand Education Society Institute of Technology Mumbai
We School Bengaluru
India

ABSTRACT

This paper illustrates the application of the color detection technique using MATLAB algorithms, to control the state of the output pins of an Arduino (AT mega 328) and also to control the movements of Bot. This gives an example of implementing Bounding Box algorithm and use of the simple color detection technique and other 'REGIONPROPS' parameters we can control the outputs of an Arduino board and also control the Bot. We have implemented a system through which the user can give commands to a robot using colored object. Through this method, the user can control or navigate the robot by using color object, thereby interacting with the robotic system. The command signals are generated using image processing. These signals are then passed to the robot to navigate it in the specified directions through the Arduino.

Keywords:- Matlab, Aurdino, COM

I. INTRODUCTION

In today's age, the robotic industry has been developing many new trends to increase the efficiency, accessibility and accuracy of the systems. Basic tasks could be jobs that are harmful to the human, repetitive jobs that are boring, stressful etc. Though robots can be a replacement to humans, they still need to be controlled by humans itself. Robots can be wired or wireless, both having a controller device. Both have pros and cons associated with them. Many systems exist that are used for controlling the robot through gestures. Some gesture recognition systems involve adaptive color segmentation, hand finding and labelling with blocking, morphological filtering, and then gesture actions are found by template matching and skeletonizing. This does not provide dynamicity for the gesture inputs due to template matching.

Another system uses machine interface device to provide real-time gestures to the robot. Analog flex sensors are used on the hand glove to measure the finger bending, also hand position and orientation are measured by ultrasonic for gesture recognition. This system though is not very cost effective.

Beyond controlling the robotic system through physical devices, recent method of object detection control has become very popular.

The main purpose of using colored fingertip is that it provides a more natural way of controlling and providess a rich and intuitive form of interaction with the robotic system.

II. PROBLEM STATEMENT

Thus, we can summarize our project as follows: the main aim of this project is to design a prototype that can be used in robot vision system. Using the camera to detect the colored object and analyze its movements to drive the robot.

III. PRODUCT DESIGN OVERVIEW

We use colored object movements for robot control. For the simplicity and to make a prototype, we are going to design a small, motorized platform. We will use serial communication to communicate between the camera and the microcontroller. The microcontroller will be placed on the robot which will be connected to the motors, driving the robot in the direction as per the movements of the colored object.

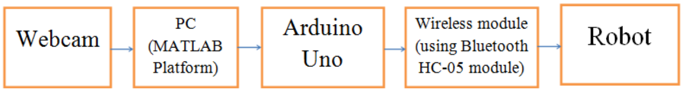


Fig. 1 Block Diagram of Project

So, now we will have a look at the overall code structure of our algorithm and the logic behind the decision making. There are two parts in the code structure. The first part is to detect the object movements and the other part is to drive the motors. The code structure can be explained in the following steps:

IV. INITIALIZATION

Initially we set up the serial communication that will be used later for the interface between MATLAB and the controller, the video capture and the program variables.

V. IMAGE PROCESSING

The web cam is used as image capturing device. Here the device takes snapshots continuously after a predefined very short interval of time. Due to human's perception of vision these snapshots taken within a very short period of time and displayed continuously might look as a continuous image i.e. video. Depending on the processing speed of the computer being used, this interval for taking snapshots is adjusted.

VI. DETECTION

Colored object is detected using Bounding Box algorithm. [1] & [2] Centroid of the detected object is calculated. Now after detecting the object movements, we have to come up with a decision algorithm that will help the controller to drive the motors accordingly.

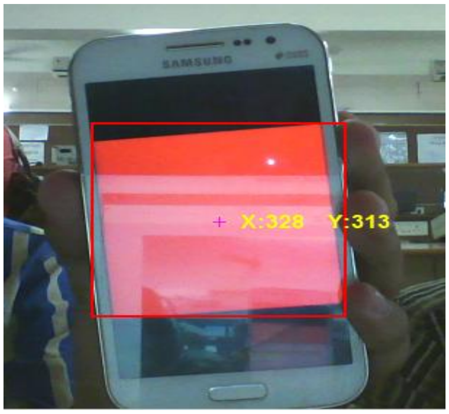


Fig. 2 Object Detection

A. MOTION

Now after detecting the object movements, we have to come up with a decision algorithm that will help the controller to drive the motors accordingly.

B. SERIAL COMMUNICATION

Now according to the detected command, the MATLAB application will transmit L, R, F or B for left, right, forward and backward respectively to the controller which will drive the motors.

VII. IMAGE PROCESSING ALGORITHM

Step 1. Start

Step 2. Get the details of the hardware device interfaced with system for imaging.

Step 3. Declare the Serial COM and start the serial communication and webcam.

Step 4. Start the Video input.

Step 5. Run image grabbing function inside while loop.

Step 6. Take snapshot.

Step 7. From required color index, perform image subtraction.

Step 8. Apply median filter for noise cancelation and convert the image into black and white image.

Step 9. Extract the interested color region and label them.

Step 10. Calculate the "regionprops" [4] of each of the labelled components of image.

Step 11. Calculate the Centroid of the detected region.

Step 12. Track the Centroid path as per the movement of the colored object and generate the corresponding movement signals. e.g. Left, Right, Forward, Backward etc.

Step 13. If count of the snapshots reached, then Stop else go to step 6.

VIII. BOT CONTROL ALGORITHM

A. TRANSMITTER END

Step1. Initialize and Start the Serial Communication.

Step2. Check for if Serial Data is available

Step3. If available follow step 4 else go to step 8

Step4. Declare char in to Read the Serial Data

Step5. Check If serial data is "R" if yes Serially Write 'R'

Step6. If Serial Data is "L", then serially write 'L'

Step7. Or if Serial Data is "F", then serially write 'F'

Step8. if Serial Data is "B", then serially write 'B'

Step9. Repeat

B. RECEIVER END

Step1. Initialize and Start the Serial Communication.

Step2. Declare Pin 7, 8, 12 and 13 as output pins for Bot connection

Step3. Check for if Serial Data is available

Step4. If available follow step 5 else go to step 11

Step5. Declare char d to serially read the data

Step6. If serial data is "F" analog write 1 on pins 7, 8, 12 and 13 to move bot forward.

Step7. If serial data is "B" analog write 0 on pins 7, 8, 12 and 13 to move bot backward.

Step8. If serial data is "L" analog write 0 on pins 12, 13 and write 1 on pins 7 and 8 to move bot Left.

Step9. If serial data is "R" analog write 1 on pins 12, 13 and write 0 on pins 7 and 8 to move Bot to right.

Step10. Repeat.

IX. FLOWCHART

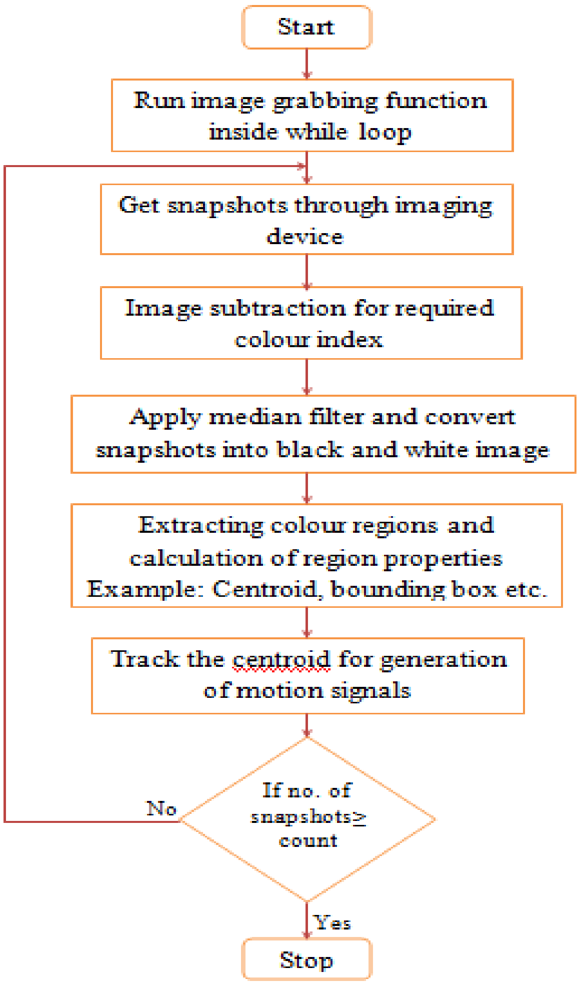

Start
Run image grabbing function inside while loop
Get snapshots through imaging device
Image subtraction for required colour index
Apply median filter and convert snapshots into black and white image
Extracting colour regions and calculation of region properties Example: Centroid, bounding box etc.
Track the centroid for generation of motion signals
If no. of snapshots≥ count
No
Yes
Stop

Fig. 3 Flowchart

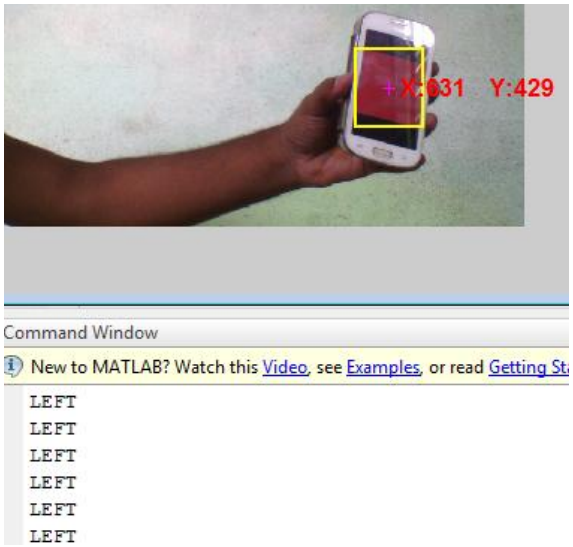
X. SIMULATION

The simulation was carried out on Matlab, the code was executed and the simulation results were as shown as follows:

A. <u>LEFT POSITION</u>

When the object is moving **left**

Result: The command window displays the message "**Left**".


X:631 Y:429
Command Window
New to MATLAB? Watch this Video, see Examples, or read Getting St

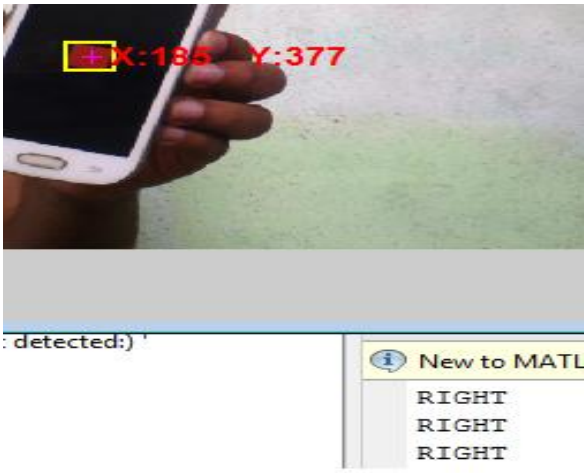
```
LEFT
LEFT
LEFT
LEFT
LEFT
LEFT
```

Fig. 4 Result at Left Position

B. RIGHT POSITION

When the object is moving **right.**

Result: The command window displays the message "**Right**"



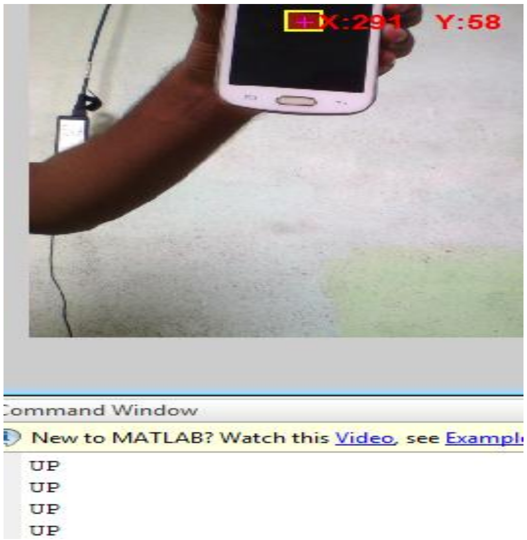
```
RIGHT
RIGHT
RIGHT
```

Fig. 5 Result at Right Position

C. FORWARD POSITION

When the object is moving **up.**

Result: The command window displays the message "UP"



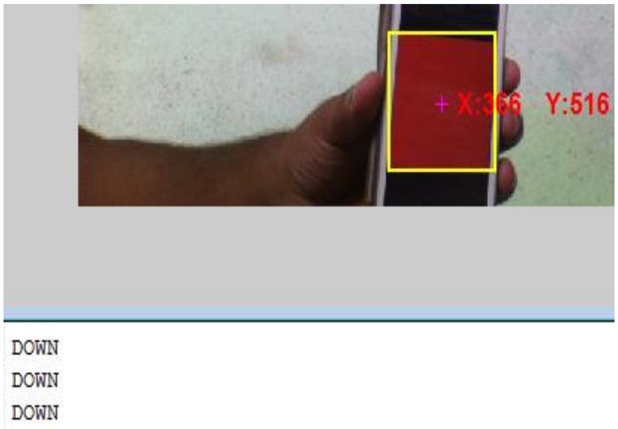
```
UP
UP
UP
UP
```

Fig.6 Result at Up Position

D. BACKWARD POSITION

When the object is moving **down.**

Result: The command window displays the message "Backward"

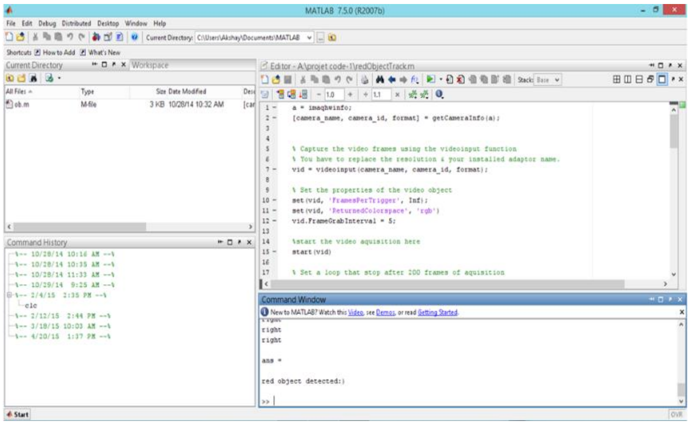


```
DOWN
DOWN
DOWN
```

Fig. 7 Result at Down Position

WHY MATLAB?

- A very large (and growing) database of built-in algorithms for image processing and computer vision applications
- MATLAB allows you to test algorithms immediately without recompilation. You can type something at the command line or execute a section in the editor and immediately see the results, greatly facilitating algorithm development.
- The MATLAB Desktop environment, which allows you to work interactively with your data, helps you to keep track of files and variables, and simplifies common programming/debugging tasks.
- The ability to read in a wide variety of both common and domain-specific image formats.
- Clearly written documentation with many examples, as well as online resources such as web seminars ("webinars").
- Bi-annual updates with new algorithms, features, and performance enhancements
- If you are already using MATLAB for other purposes, such as simulation, optimation, statistics, or data analysis, then there is a very quick learning curve for using it in image processing.
- The ability to process both still images and video.
- Technical support from a well-staffed, professional organization (assuming your maintenance is up-to-date)
- A large user community with lots of free code and knowledge sharing
- The ability to auto-generate C code, using MATLAB Coder, for a large (and growing) subset of image processing and mathematical functions, which you could then use in other environments, such as embedded systems or as a component in other software.


MATLAB 7.5.0 (R2007b)
File Edit Debug Distributed Desktop Window Help
Current Directory: C:\Users\Akshay\Documents\MATLAB
Shortcuts How to Add What's New
Current Directory
Workspace
Editor - A:\projet code-1\redObjectTrack.m
Command History
Command Window
Start
OVR

All Files	Type	Size	Date Modified
ob.m	M-file	3 KB	10/28/14 10:32 AM

```
a = imaqhwinfo;
[camera_name, camera_id, format] = getCameraInfo(a);


% Capture the video frames using the videoinput function
% You have to replace the resolution & your installed adaptor name.
vid = videoinput(camera_name, camera_id, format);

% Set the properties of the video object
set(vid, 'FramesPerTrigger', Inf);
set(vid, 'ReturnedColorspace', 'rgb')
vid.FrameGrabInterval = 5;

%start the video aquisition here
start(vid)

% Set a loop that stop after 200 frames of aquisition
```

```
%-- 10/28/14 10:16 AM --%
%-- 10/28/14 10:35 AM --%
%-- 10/28/14 11:33 AM --%
%-- 10/29/14  9:25 AM --%
%-- 2/4/15  2:35 PM --%
   clc
%-- 2/12/15  2:44 PM --%
%-- 3/18/15 10:03 AM --%
%-- 4/20/15  1:37 PM --%
```

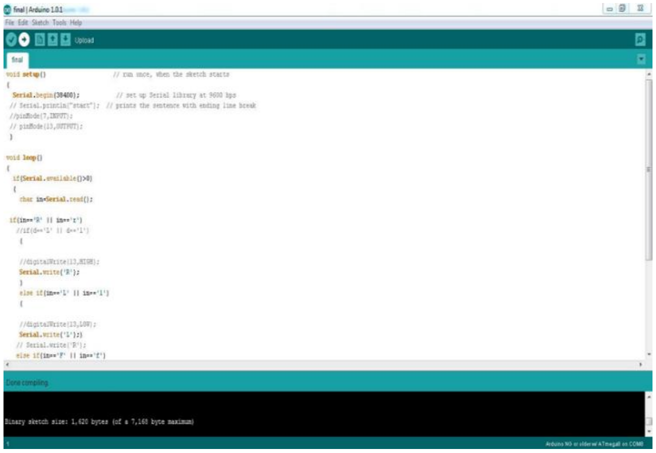
New to MATLAB? Watch this Video, see Demos, or read Getting Started.

```
right
right

ans =

red object detected:)

>>
```

Fig. 8 Matlab Interface

WHY ARDUINO?

- It is an **open-source project**, software/hardware is extremely **accessible** and very flexible to be customized and extended.
- It is **flexible**, offers a variety of digital and analog inputs, *SPI* and serial interface and digital and *PWM* outputs.
- It is **easy to use**, connects to computer via USB and communicates using standard serial protocol, runs in standalone mode and as interface connected to PC/Macintosh computers.


final | Arduino 1.0.1
File Edit Sketch Tools Help
Upload
final
Done compiling.
1
Arduino NG or older w/ ATmega8 on COM8

```
void setup()              // run once, when the sketch starts
{
  Serial.begin(38400);      // set up Serial library at 9600 bps
 // Serial.println("start");  // prints the sentence with ending line break
 //pinMode(7,INPUT);
 // pinMode(13,OUTPUT);
 }

void loop()
{
  if(Serial.available()>0)
  {
   char in=Serial.read();

 if(in=='R' || in=='r')
   //if(d=='L' || d=='l')
    {

    //digitalWrite(13,HIGH);
    Serial.write('R');
    }
    else if(in=='L' || in=='l')
    {

    //digitalWrite(13,LOW);
    Serial.write('L');})
   // Serial.write('R');
   else if(in=='F' || in=='f')
```

```
Binary sketch size: 1,620 bytes (of a 7,168 byte maximum)
```

Fig. 9 Arduino IDE Interface

ADVANTAGES

- The project requires only ***one-time investment*** of components.

- Cost of the project is also very ***cheap and affordable***.
- ***Maintenance*** cost is very ***low***.
- Light weight and can be made ***portable***.
- Any color object can be used for generation of ***movement signals***.

APPLICATIONS

- To control the ***movements of robotic arm*** in unmanned regions.
- Object Detection algorithm can be used in ***robot vision***.
- In ***pick and place robot*** for particular color objects.

XI. FUTURE SCOPE

- Further the object detection can be improved by eliminating unwanted detection of color while execution of Bounding Box algorithm except the target object, multiple objects detection.
- The ***wireless communication*** can be made from the camera to the main controller board.
- The entire project can be burned into a Raspberry Pi so that the need of always carrying a laptop or Personal Computer can be avoided, and project becomes easy to carry.
- The robot can move left, right, front and backwards, but more detailed studies is to be done to rotate it in intermediate angles using the object movements more accurately.
- The idea can be implemented to ***control objects at remote places***.

XII. CONCLUSION

The main aim of the project was to detect the colored object's movement and how that can be applied to control the movement of the Robot. It also gives us an opportunity to control objects in Remote places. The project is successfully running in real-time and further scope is added to improve the accuracy of the object detection system.

REFERENCES

[1] Raquib Buksh, Soumyajit Routh, Parthib Mitra, Subhajit Banik, Abhishek Mallik, Sauvik Das Gupta," Implementation of MATLAB based object detection technique on Arduino board and iROBOT CREATE", International Journal of Scientific and Research Publications, Volume 4, Issue 1, January 2014.

[2] Shubham Srivastava, Pratibha Singh," Real Time Object Tracking Using Color Feature", International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-3, Issue-8, January 2014

[3] Aji Joy, Ajith P Somaraj, Amal Joe, Muhammed Shafi, Nidheesh T M," Ball Tracking Robot Using Image Processing and Range Detection", International Journal of Innovative Research in Computer and Communication Engineering Vol. 2, Issue 3, March 2014.

[4]Region Properties: http:// www.mathworks.in/help/images /ref/ regionprops.html