



# Color based Object Tracking Robot



Deshpande R<sup>1</sup>, Nair A<sup>1</sup> and Razban A<sup>2</sup> \*

<sup>1</sup>Department of Electrical and Computer Engineering, Indiana University Purdue University Indianapolis, Indiana

<sup>2</sup>Department of Mechanical and Energy Engineering, Indiana University Purdue University Indianapolis, Indiana

**Submission:** November 07, 2017; **Published:** March 15, 2018

\*Corresponding author: Ali Razban, Department of Mechanical and Energy Engineering, Indiana University Purdue University Indianapolis, Indiana; Tel: 317-274-8458; E-mail: arazban@iupui.edu

## Abstract

This article presents a vision based object tracking robot controlled by a computer software. We developed a robotic object tracking such that it can track objects based on color. Introducing color along with the shape of the object reduces false detection. The unique features of this system are its ability to perform simultaneous real-time image acquisition and range sensing. Furthermore, its flexible design, confers the ability to handle a wide range of tasks. In this case the object of interest is a red circular shape. The system is designed to track the object in its Y-Z plane using robotic manipulator. The program for color based object tracking is written in MATLAB. The camera is used to track the object and decide the motion of the system. The object of interest is detected successfully using two stages, color detection followed by shape detection. The detected object is then tracked by the robotic end effector very closely.

**Keywords:** Object tracking; Robotic manipulator; Color based object

## Introduction

Object detection and tracking are essential elements of many computer vision applications including activity recognition, quality control, parts recognition, autonomous automotive safety and surveillance. It is required for surveillance application, autonomous vehicle guidance, collision avoidance and smart tracking of moving objects. They are used in various fields such as science, engineering and medicine.

Detecting and tracking an object of interest in a known or an unknown environment is a challenging task in areas of image processing and computer vision system. While detection of color or shape irregularities is of interest in image processing, the techniques for faster detection, tracking and extracting object information is more complex in computer vision. Prior knowledge of object to be tracked like its shape, color etc. helps in faster tracking in mobile robots. Moreover, the change in lighting conditions in real time environment offers a challenge in color based tracking systems.

The robotic vision systems are mostly operated in environment which undergo a lot of dynamic changes. Learning models is an important aspect for autonomous robots to be able to navigate in such dynamic environment [1]. The work presented in [2] designed a robotic system that tracks an unknown color spherical object with uses HSV color space and automatic color detection mechanisms. The technique of Kalman based object tracking filter was presented while the

proposal in [3] presents a modelling of robot using MATLAB Sim mechanics. The simple, intuitive and accurate modelling method eliminates the need to compute the forward dynamics which is cumbersome. Researchers have performed this tracking and detection in some wireless mobile robot using the concept of perpetual color space. Vision based systems introduced in such works uses a wireless camera that keeps the desired target at the center of the image plane [4].

Object tracking using computer vision is a crucial component in achieving robotic surveillance. For object tracking we need to locate the object in subsequent frames. Object can be tracked by identifying and tracking some specific features of the object in motion such as color or shape. The trajectories of the object in motion can be traced by using this process over a longer duration of time. In the method proposed in [5] the moving object is separated from the static background. The algorithm assumes that the background is relatively more static than the foreground. When the object moves, some regions of the video frames will differ significantly from the background. Such frames can be the moving object which is our object of interest.

The main aim of the object tracking is to track the object based on the information obtained from video sequences. Various methods have been proposed to encounter this problem [6]. In this research, we are presenting a robotic vision system that track an object based on its color in a known

environment. This includes color detection based on binary image. Once the object of interest is identified, its coordinates are tracked and given as physical input signals to the robotic joints that control its movement. The mobile robot should be able to process the moving targets to detect the object of interest. This can be best achieved by addition of an active camera for real time tracking. This research uses a camera as the eye of the robot. The camera captures the multiple frames video in RGB color scale within a very short time. The robot is designed using solid works software and MATLAB is used for image processing and sending the signals to the robotic joints. MATLAB converts the RGB image frame to binary image for better object tracking. To enhance the tracking operation, the objects are located using color based image segmentation that preserves the object information. The system can locate and detect the color based objects using information obtained from image sensor. We have also used filtering to remove noise from the image to improve the efficiency of the system.

**System components**

Vision based object tracking robot comprises of multiple tasks; such as image acquisition, mapping, data analysis and image location identification.

**Image acquisition and mapping**

Real time environment vision applications require a functional USB camera attached to the robot. Objects can be detected easier in high resolution images. To take real time video, video input object is created, and stream is viewed. This can be done by computing the zero moment of the coordinates (x, y) using;

$$M_{00} \sum_x \sum_y I(x, y)$$

We then compute the first moment values of x, y using;

$$M_{10} \sum_x \sum_y xI(x, y)$$

$$M_{01} \sum_x \sum_y yI(x, y)$$

Localization values are obtained using;

$$X_c = M_{10}/M_{00}, = M_{01}/M_{00}$$

**RGB to binary image conversion and filtering**

Tracking of the moving object involves concentrating the luminescence of the non-ideal environment which is achieved using various mathematical models called color spaces. RGB is a convenient color model for computer graphics which is compatible with human vision system and it is used in this research. RGB color space uses three chromaticity of red, blue and green color [4]. Once the image is captured, it is converted from RGB input image to binary image where “one” represent the presence of an image and “zero” represent the absence of an image. Binary image shows the presence of noise as unexpected colors spread all over the image. To remove the scattered color

components, function `bwareaopen()` in MATLAB is used. This function removes all connected components (objects) that have fewer than P pixels from the binary image BW thereby producing another binary image, BW2. The default connectivity is 8 for two dimensions, 26 for three dimensions, and `conndef(ndims(BW), 'maximal')` for higher dimensions. This operation is known as an area opening [5].

Furthermore, the median filter is used to enhance or smoothen the obtained image. This filtering technique basically modifies each value of pixels by the median value of the neighboring pixels. The new value obtained is the median of the sorted pixel values present within the window as given in equation (1) [7].

$$NewPixel(x, y) = Median(sorted\ array\ of\ neighboring\ pixels) \quad (1)$$

The object of interest detection which is a circular red object in this case can also be detected and tracked using the Hough transform followed by Mexican hat filter [8].

**Centre calculation**

The center coordinates or pixel coordinate (Figure 1) is calculated using equations (2) and (3) [4], where ‘x’ is the ratio of Sum of column numbers having “one” to the total number of columns having “one”.

$$x = \frac{\sum_n nx_n}{N - n + 1} \quad (2)$$

Similarly, ‘y’ is obtained by dividing the sum of row numbers having 1 with the total number of rows having “one”.

$$Y = \frac{\sum_m mx_m}{M - m + 1} \quad (3)$$

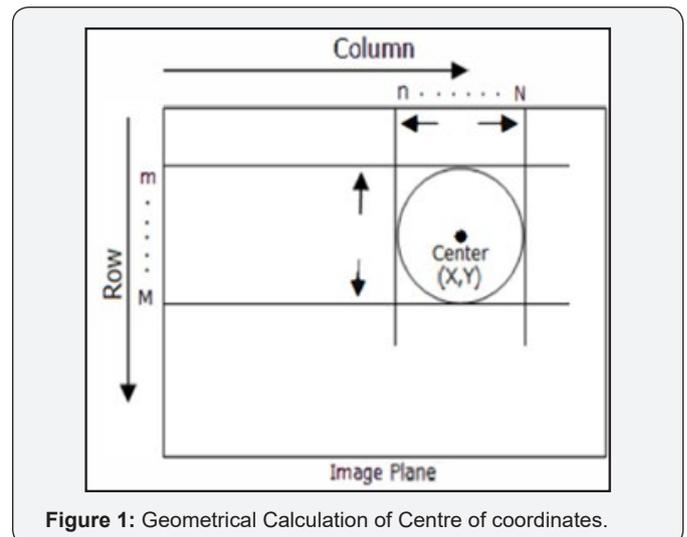


Figure 1: Geometrical Calculation of Centre of coordinates.

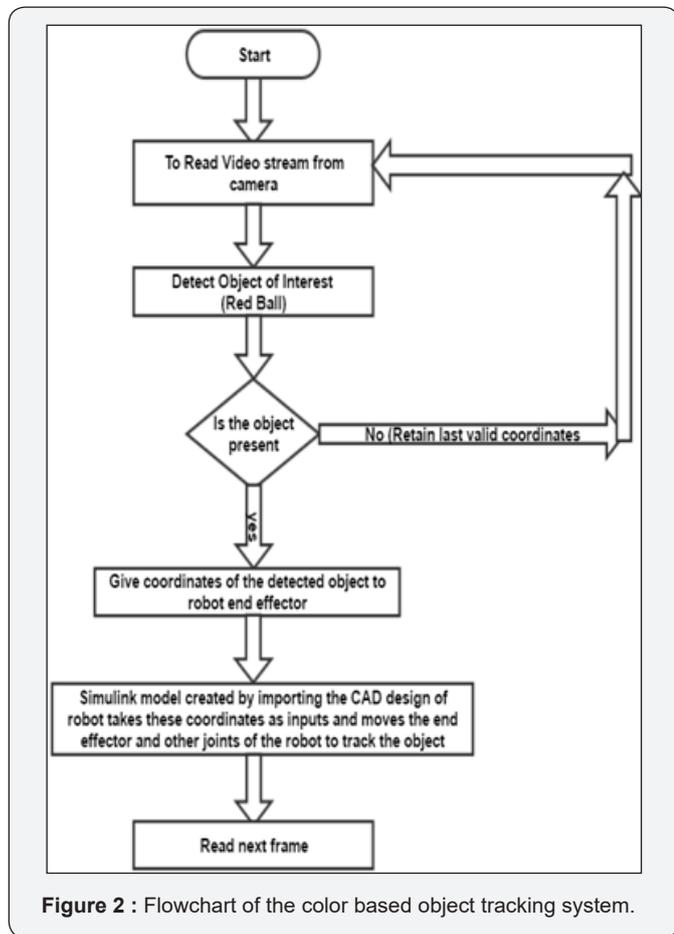
**Robotic arm**

Fanuc 430i which is six axes for flexible automation has been used in this research. In our application, we are using two translational degrees of freedom (heaving and swaying) and

two rotational degrees of freedom (pitching and yawing). We can't use the other two degrees due to the physical constraints of the current system.

**Methodology**

Figure 2 shows simulation flow chart used for detection and tracking of a red colored circular object using robotic manipulator. The integration of color with the circular shape helps to overcome false reading even if the lightening conditions are changed. False detections can be greatly reduced by considering the shape of the object. First stage is to read the video stream of the real-time environment. The MATLAB code written will configure the camera to read real time data, then extract from the video stream. The acquired frame is then processed to identify the object of interest.

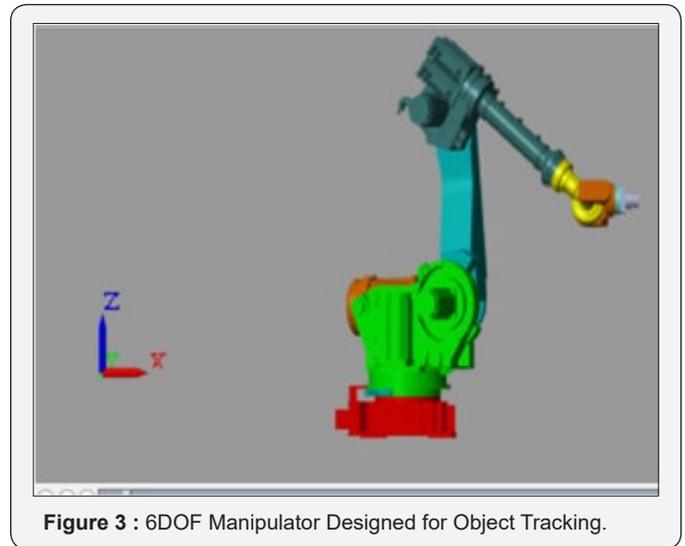


Next stage involves extracting the red color components from the frames. Circular shape is detected using Hough Transform 2 stage function which is an inbuilt function in MATLAB. This is done by converting the image to gray scale image and then extracting the red colored components. The two stage Hough transform function finds circles in an image with radii in the range specified. It also estimates the radii corresponding to each circle center. The detected centers and radii are stored in the Matrix form. The image is then filtered

to remove noise and converted to binary. This image is again filtered to remove pixels less than 300 which further enhance the accuracy and reduce false detection. A circle is then drawn around the detected object. The center of this object is calculated using the MATLAB in built function. This center is then passed to the robot end-effector.

**Design of robot**

Robotic manipulator (Fanuc 430i robot) is designed using Solid Works software. Six degree freedom manipulator is used in this system to give greater flexibility of movement in real world applications. We use the "eye in hand" configuration for the sensor, i.e., the camera is attached to the last link instead of an end effector. Figure 3 shows the snap shot of the robotic arm designed with camera attached to the end position.



The manipulator is built using solid works is exported to MATLAB Simulink. This is then linked with the solid works using some specific MATLAB Commands [9]. The export procedure generates one XML file and a set of geometry files to be imported into Simscape Multibody to generate a new model.

**Track detection & control**

The robot should follow the object of interest (Red Ball). The desired end effector position is given by the coordinates of the Object tracked. MATLAB will provide this coordinates to the sim mechanics model [10]. An Inverse kinematics model is developed and added to the sim scape blocks as shown in Figure 4. This inverse kinematic model will take the Object coordinates as the final position and calculate the joint parameters like joint angle and joint displacement to reach that desired position. The inverse kinematics model consists of 6 Revolute blocks corresponding to the 6 revolute joints of the robot to ensure the movement of robot in 6 DOF. It also consists cartesian coordinate blocks to ensure that the robot can move linearly [11-12].

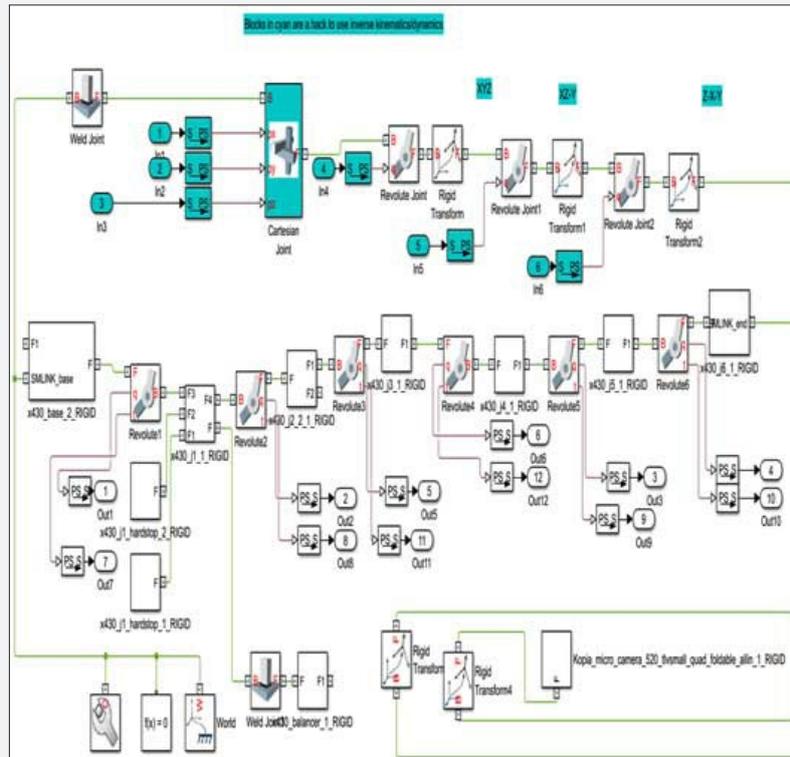


Figure 4 : Simulink Blocks after adding inverse kinematics loop.

**Results**

Simulation was performed on MATLAB that continuously access the image frame from video stream. The code designed to color, and shape detection will detect the red ball which is the object of interest. Figure 5 shows the snap shot of the video frame where the red ball is detected and marked by a yellow circle from a group of different colored and different shaped object. Figure 6 shows Snap shot of Robotic Arm Positions as it Tracks the Object.



Figure 6 : Snap shot of Robotic Arm Positions as it Tracks the Object.

Then, the code calculates the coordinates of the object detected and send it to the Simulink blocks. The inverse kinematic link takes these as input and calculate the joint

variables like displacement and joint angles and feed it to end effector which will then move accordingly.

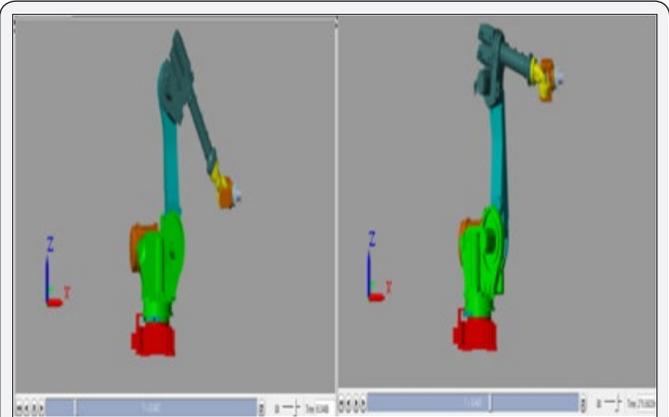


Figure 6 : Snap shot of Robotic Arm Positions as it Tracks the Object.

Figure 7 compares the path of the red ball which is the object of interest and the path of the robot end effector. The robot closely follows the red ball. The use of color along with the shape has helped to minimize the false detection to some extent. The x axis in the graph for the path of Red Ball detected represent the pixel values, whereas the x axis of the path of the end effector represents the movement of the robotic end effector in meters.

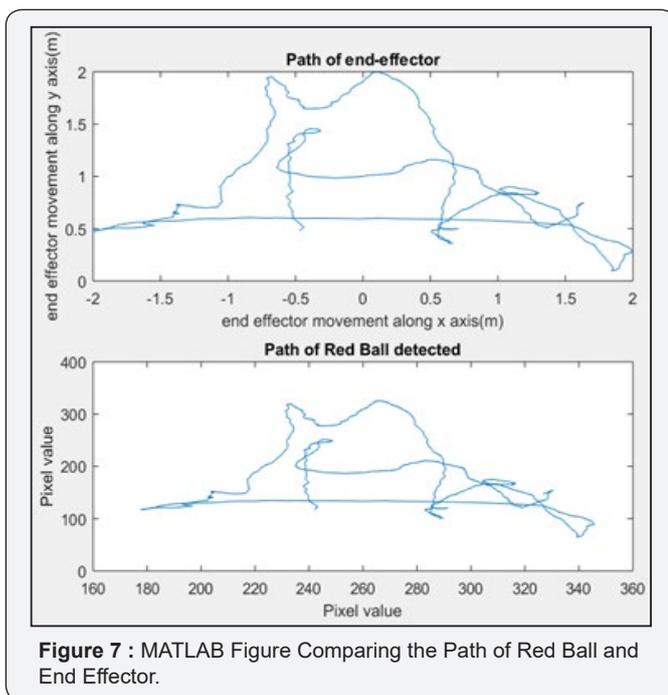


Figure 7 : MATLAB Figure Comparing the Path of Red Ball and End Effector.

### Conclusion

This paper presents a color and shape based object tracking robot. We have incorporated the color detection along with the shape detection to reduce false tracking. The robot used in this paper utilizes camera for images acquisition and instructions from computer to perform motion trajectory to track the desired object. The end effector could successfully track the red balls path with the help of tracking algorithm developed in Matlab and the Inverse kinematics loop which has been developed in Simulink consisting of revolute and Cartesian coordinate joints to control the movement of the end effector.

The performance of the system can be further enhanced by using trained classifiers that minimizes the false detections which may occur due to varying light conditions. These classifiers use a statistical learning algorithm to produce a generic model of the object to be tracked. Addition of sensors like sonar and infrared helps to extract the image properties like distance, depth etc. more accurately. Multi cameras can

be introduced to enable stereo vision based applications in 3D object reconstruction. Further researches can be done to incorporate fuzzy and neural networks to deal with multiple object tracking methods.

### References

1. Araujo R, Gouveia G, Santos N (2003) Learning self organizing maps for navigation in dynamic worlds. Proc IEEE International Conference on Robotics and Automation (ICRA), pp. 1312-1317.
2. Troppan A, Guerreiro E, Celiberti F, Santos G, Ahmad A, et al. (2013) Unknown-color spherical object detection and tracking. 13th International Conference on Autonomous Robot Systems, Lisbon, p. 1-4.
3. Shaoqiang Y, Zhong L, Xingshan L (2008) Modeling and simulation of robot based on Matlab/SimMechanics. 27th Chinese Control Conference, Kunming, pp. 161-165.
4. <https://www.mathworks.com/help/images/ref/bwareaopen.html>
5. Joy A, Somaraj A, Joe A, Shafi M, Nidheesh T (2014) Ball Tracking Robot Using Image Processing and Range Detection. International Journal of Innovative Research in Computer and Communication Engineering 2(3).
6. Manigandan M, Malathi G, Madhavi N (2010) Wireless vision based moving object tracking robot through perceptual color space. INTERACT-2010, Chennai, p. 20-25.
7. Hasan KM, Al-Nahid A, Al-Mamun A (2012) Implementation of vision based object tracking robot. International Conference on Informatics, Electronics & Vision (ICIEV), Dhaka, pp. 860-864.
8. Çelik Y, Altun M, Güneş M (2017) Color based moving object tracking with an active camera using motion information. International Artificial Intelligence and Data Processing Symposium (IDAP), Malatya, Turkey, p. 1-4.
9. Evans J, Tughrul A (2002) Enhanced image detection on an ARM based embedded system. Design automation for embedded systems 6(4) : 477-487.
10. Lestriandoko NH, Sadikin R (2016) Circle detection based on hough transform and Mexican Hat filter. International Conference on Computer, Control, Informatics and its Applications (IC3INA), Tangerang, pp. 153-157.
11. The MathWorks Inc. (2017) Simscape™ Multibody™ User's Guide, Natick, MA: Mathworks.
12. Qu B, Xu B (2011) A control algorithm of general 6R mechanic arm based on inverse kinematics. IEEE International Conference on Computer Science and Automation Engineering, Shanghai, pp. 327-330.



This work is licensed under Creative Commons Attribution 4.0 License  
DOI: [10.19080/RAEJ.2018.02.555588](https://doi.org/10.19080/RAEJ.2018.02.555588)

**Your next submission with Juniper Publishers  
will reach you the below assets**

- Quality Editorial service
- Swift Peer Review
- Reprints availability
- E-prints Service
- Manuscript Podcast for convenient understanding
- Global attainment for your research
- Manuscript accessibility in different formats  
( Pdf, E-pub, Full Text, Audio)
- Unceasing customer service

Track the below URL for one-step submission  
<https://juniperpublishers.com/online-submission.php>