Computer Vision for Object Recognition and Tracking Based on Raspberry Pi

Ali A. Abed (SMIEEE) [a], Sara A. Rahman[b]
[a] Department of Computer Engineering-University of Basra, Basra, Iraq
ali.ahmed.abed@gmail.com
[b] Department of Electrical Engineering-University of Basra, Basra, Iraq
eng.sarah22489263@gmail.com

Abstract. The tasks of object recognition and tracking are a key component of video surveillance and monitoring systems. This paper presents CamShift (Continuously Adaptive Mean Shift) algorithm and color detection in darkness for tracking a target with video sequences in real time. The system described in this paper contains a camera that is connected to a Raspberry Pi. The Raspberry Pi has an image processing algorithm which detects an object first and then tracks it. Color detection generally is a primary stage in most of the image processing applications, if the application is based on the color information. To monitor object in video, an embedded board is adopted to monitor the activity of the object of interest based on Raspberry Pi with LCD touch screen display TFT monitor. A software method for real time implementation of moving object tracking and recognition is done using Python programming language with OpenCV libraries. The two algorithms are tested and compared to prove the robustness of the proposed color detection algorithm operating in a low light environment. Good practical results for recognition and tracking are obtained.

Keywords: CAM shift, Color detection, Raspberry Pi

1 INTRODUCTION

Vision-based systems become part of everyday life; therefore this work will be in the field of artificial vision based on image processing suitable for many applications such as mobile robots navigation. Computer vision is a type of processing input images producing output that could be a set of characteristics or parameters related to images. Its application in robotics, surveillance, monitoring, and security systems makes it very important and widespread worldwide. The work starts with creating a model to make useful decisions about real physical objects and scenes. Camera mimics and uses real-time digital videos for object recognition and tracking [1]. Object tracking is the main tasks in the field of computer vision. It has many applications in traffic control, human computer interaction, digital forensics, gesture recognition, augmented reality and visual surveillance [2]. An efficient tracking algorithm will lead to the best performance of higher level vision tasks such as automated monitors, human computer interaction and other. Among the various tracking and recognition algorithms, CamShift tracking and color detection algorithms will be adopted in this paper. CamShift is primarily intended to perform efficient head and face tracking in a perceptual user interface. It is based on an adaptation of Mean Shift that gives a probability density image and finds the mean of the distribution by iterating in the direction of maximum increase in probability density [3]. The aim of the color detection is to identify the category pixel color in a given image. Color detection already had acquired a superior attention of researchers because opportunities that lead to have a robust and efficient human body detection.

2 LITERATURE SURVEY

Several researches dealt with the topic of this paper with different methods. In [4] Chuan Zhao proposed a method for tracking objects with their size and shapes which changing with time, on the basis of a group of mean-shift and affine structure. The results which showed the object's tracking capability in the existence of wide change and partial blockage. In [5], Ebrahim submitted an effective color based CamShift algorithm for target tracking and expressed how the style can be resolved by its flaws. In [6], Halis suggests a method for efficient color detection in RGB space in hierarchical structure of neural networks. The results showed that the proposed hierarchical structure of neural networks is the best on traditional neural network classifier in color detection. In [7], Peng used the color co-occurrence histogram (CH) for recognizing objects in images. The results showed that the algorithm works in spite of confusing background clutter and moderate amounts of blockage and object praise. Although the color detection and CamShift algorithms have many advantages, but they does not work in the dark and this makes it difficult to continue tracking during the lack of light. Therefore, this paper introduces an algorithm that makes these algorithms operate normally even in the darkness. A method for recognition and tracking using color detection in low-light environment is proposed and compared with CamShift. The presented algorithms are programmed with Python programming language supported by OpenCV libraries, and executed with a credit-card computer board called Raspberry Pi with attached external camera.

3 OBJECT TRACKING USING CAMSHIFT ALGORITHM

3.1 Mean Shift Algorithm

The mean-shift algorithm is a non-parametric density pheasant resident algorithm. It is basically an iterative expectation maximize clustering algorithm executed within the local find areas. The mean-shift tracker provides accurate resettlement and it is computationally possible. The form that widespread used for target representation is color histograms, due its independence from scaling and for the purposes of rotation and its durability to partial blockage. To keep track the target using the Mean Shift algorithm, it repeats the following steps [7]:

1. Choose a search window size and the initial site of the search window.
2. Account the mean site in the search window.
3. Determine the center of the search window at the mean site computed in Step2.
4. Repeat Steps 2 and 3 until rapprochement (or even the mean location moves less than a predefined threshold).

3.2 CamShift Algorithm

A probability distribution image of the color required in the video relay is created by using CamShift algorithm. CamShift algorithm first establishes a model of the desired hue using a color histogram and uses the HSV color system which is compatible with the projecting standard BGR color space along its director diagonal from white to black. Color distributions derived from video image sequences change with time. Therefore mean shift algorithm must be modified to adapt dynamically to the probability distribution with its relay.
CamShift algorithm is dependent on an adaptation of mean shift algorithm. It is calculated as follows [7]:

1. Select the initial location of the search window.
2. Mean Shift described above (one or many repetitions); storage of the zeroth moment.
3. Adjust the search window size equal to a function of the zeroth moment located in Step 2.
4. Repeat Steps 2 and 3 even rapprochements (average site moves less than a predefined threshold).

3.3 Account search window size

In CamShift, to calculate the search window, follow the steps below [8]:

1. Computing zero-order moment:

$$m_{00} = \sum_x \sum_y I(x, y), \quad (1)$$

2. Calculating a first-order moment:

$$m_{10} = \sum_x \sum_y xI(x, y), \quad (2)$$

$$m_{01} = \sum_x \sum_y yI(x, y), \quad (3)$$

3. Calculating the centroid position of the search window:

$$x_c = \frac{m_{10}}{m_{00}}, \quad (4)$$

$$y_c = \frac{m_{01}}{m_{00}}, \quad (5)$$

4. Setting the initial window size ***s***:

$$s = 2 * \sqrt{\frac{m_{00}}{256}}, \quad (6)$$

Taking into account the symmetry, the calculated results for the nearest odd number by calculating the two-order image moments can track target direction:

$$m_{20} = \sum_x \sum_y x^2 I(x, y), \quad (7)$$

$$m_{02} = \sum_x \sum_y y^2 I(x, y), \quad (8)$$

$$m_{11} = \sum_x \sum_y xyI(x, y), \quad (9)$$

The length and width of search window are as follows:
Window direction:

$$\theta = \frac{\arctan(\frac{b}{a-c})}{2}, \quad (10)$$

Where:

$$a = \frac{m_{20}}{m_{00}} - x_c^2, \quad (11)$$

$$b = 2\left(\frac{m_{11}}{m_{00}} - x_c y_c\right), \quad (12)$$

$$c = \frac{m_{02}}{m_{00}} - y_c^2, \quad (13)$$

4 OBJECT TRACKING USING COLOR DETECTION

Color is used in identifying and isolating objects. The RGB values of every pixel in the frame is read and turned in to the HSL color space. The flowchart of the CamShift algorithm is shown in Figure 1[9]. The HSL color space is selected because that chromatic information is independent from the lighting situations. Hue identifies basic color, saturation decides the intensity of color and lighting depends on lighting condition. Because each color has its own range of H values, program compares the H values of each pixel with a predefined group of H values of the relegation zone. If it within 10%, the pixel is marked out as being a part of relegation zone. With the group of H correctly chosen, the landing spot is identified more accurately. After passing all the pixels for one frame, the centroid of all the marked pixels is calculated.
Conversion BGR color space to HSV color space is performed using the following equations [10]:

$$h = \begin{cases} 0, \ldots\ldots\ldots if \ \max = \min, \\ \left(60^o * \frac{g-b}{\max - \min} + 360^o\right), \ldots\ldots if \ \max = r, \\ 60^o * \frac{b-r}{\max - \min} + 120^o, \ldots\ldots if \ \max = g, \\ 60^o * \frac{r-g}{\max - \min} + 240^o, \ldots\ldots if \ \max = b, \end{cases}, \quad (14)$$

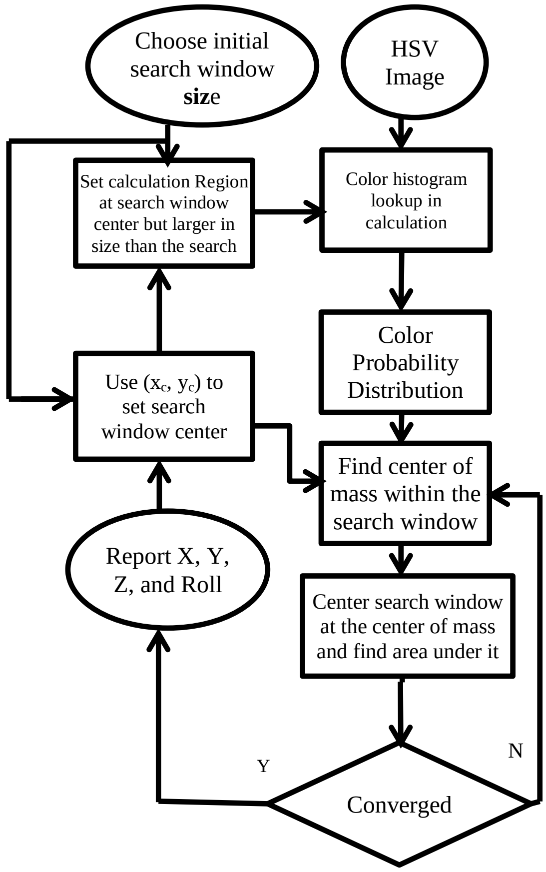


Fig. 1. Flow Chart of CamShift Tracking Algorithm.

Where:

r (red), g (green), b (blue) ∈[0,1] are the coordinates of the BGR color space and max and min compatible with the greatest and least of r, g and b consecutive .The Hue angle h∈[0,360] for HSV color space.

The value of h is normalized to lie between 0 and 180 to fit into an 8 bit gray scale image (0-255), and h= 0 used when max=min, although hue has no geometric meaning for gray. The s value for HSV color space is determined as follows:

$$s = \begin{cases} 0, \ldots\ldots \text{if } \max = 0, \\ \dfrac{\text{max-min}}{\max} = 1 - \dfrac{\min}{\max}, \ldots\ldots \textit{otherwise} \end{cases}, \qquad (15)$$

The v or value channel represents the gray scale portion of the image. The threshold for Hue value of the image is scheduled on the basis of color mounted marker color. Using the threshold value, segmentation between color required and other colors is performed. The resulting image is a binary image with white that indicate to the desired color region and black assumed to be the noisy region.

Visible colors located between 400nm (violet) and 700nm (red) on the electromagnetic spectrum is shown in Figure 2 [11]:

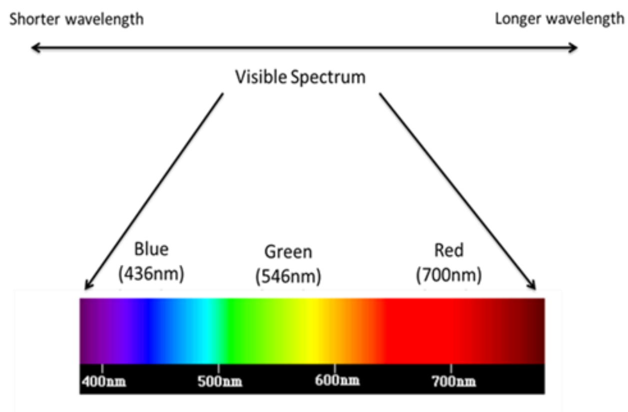


Fig. 2. The Visible Spectrum.

The block diagram for object tracking by color detection is shown in Figure 3 [12]:

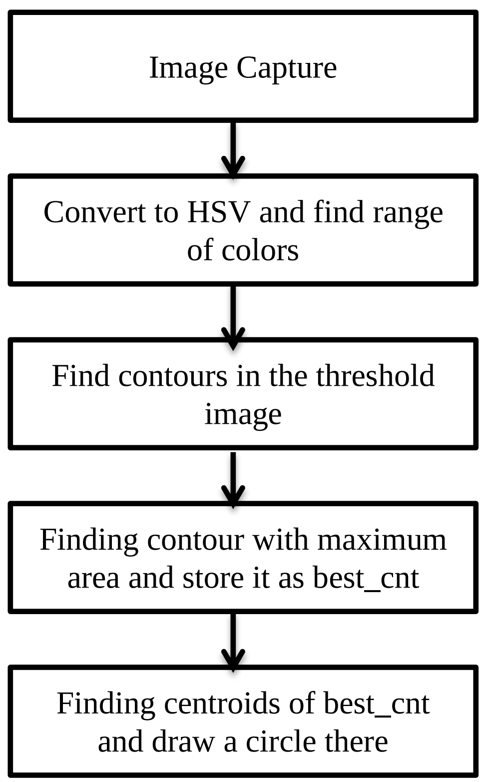


Fig. 3. Flow Chart for Object Tracking by Color Detection.

5 HISTOGRAM EQUALIZATION

Histogram equalization is a nonlinear technique for adjusting the contrast of an image using its histogram. It is the process that remaps the pixels of the image [13]. It increases the brightness of a gray scale image which is different from the mean brightness of the original image. To calculate the histogram to any image follows the following steps:

- The acquired image is converted into HSV.
- The image is then decomposed into two parts by using exposure threshold.
- Equalize them independently over enhancement and controlling using clipping threshold.

For measuring the performance of the enhanced image, entropy and contrast are calculated. Image Enhancement is the process of improving the visual quality of an image so that the results are more suitable than the original.

6 SYSTEM REQUIREMENTS

The hardware system consists of the following parts:

1. The Raspberry Pi Model B to run object recognition and tracking programs on it.
2. An image capturing camera.

6.1 Raspberry Pi

The Raspberry Pi Model B shown in Figure 4 is a low cost, credit-card sized computer that plugs into a computer monitor or TV and uses a standard keyboard and mouse. Raspberry Pi 2 Model B features more than 1GB of RAM memory. The operating system kernel has been upgraded to take full advantage of the latest ARM Cortex-A7 technology and is available with version 1.4 of NOOBS (New Out-Of-Box Software software) [14].

Fig. 4. Raspberry Pi Model B.

6.2 Camera Module

The camera module for Raspberry Pi is shown in Figure 5. It can be used for taking high-video clarity in addition to stills photographs.

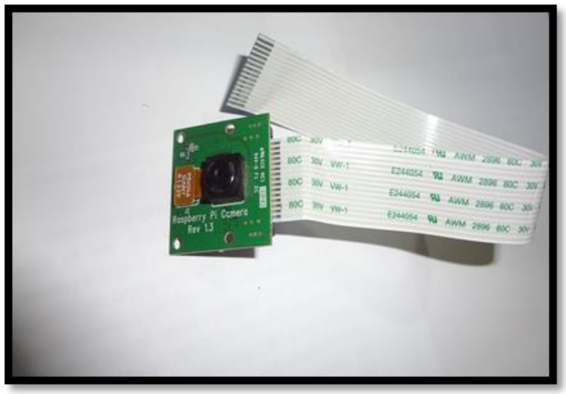


Fig. 5. Raspberry Pi Camera Module.

It is designed to contact to Camera Serial Interface (CSI) of the Raspberry Pi. The Camera module fit with all models and versions of Raspberry Pi. The camera module is very common in homeland security applications and in wildlife and other locations. The camera module can work in programs by executing the following instructions [15]:

```
from time import sleep

from picamera import PiCamera

camera = PiCamera()
```

7 RESULTS

The proposed algorithms are programmed as a python code with OpenCV library. They are used to track the various objects on the basis of color in very low light living situation. The same results, for natural lighting and darkness, are obtained and compared with CamShift traditional algorithm. The blue objects shown in Figure 6 are used for the tracking process.

Fig. 6. Examples of Objects that Used to Track.

In the case of color detection in natural lighting, the results are shown in Figure 7 by keeping track the blue objects, in spite of the emergence of on-screen objects with different colors.

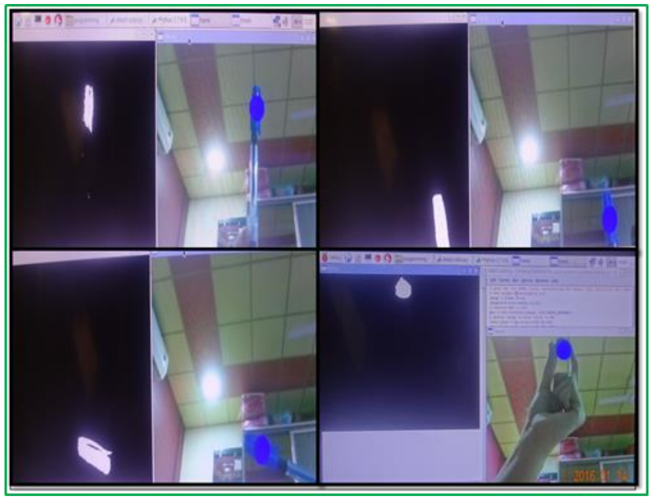


Fig. 7. Color Detection in Natural Lighting.

The same algorithm is used to give the same results in different intensity lighting as shown in Figure 8.

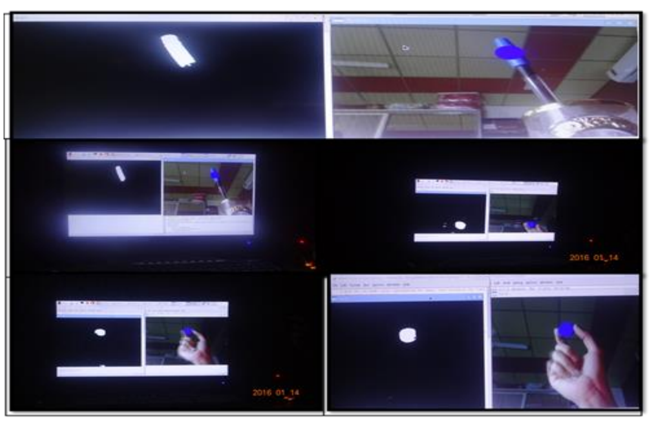


Fig. 8. Color Detection in Very Low-Light.

In addition, the traditional CamShift algorithm is also used for the sake of comparison with the proposed one. Comparing the two, one can clearly notice that the proposed color detection algorithm has more robustness compared to CamShift.

Fig. 9. CamShift Algorithm in Natural Lighting.

Also, in case of CamShift algorithm, it is noticed that the results are vary depending on the intensity of light as shown in Figure 10, in addition to the loss of the tracking process in the case of the dark and this makes this algorithm is inefficient. Therefore, it is found that the tracking process on the basis of color detection is more suitable for tracking.

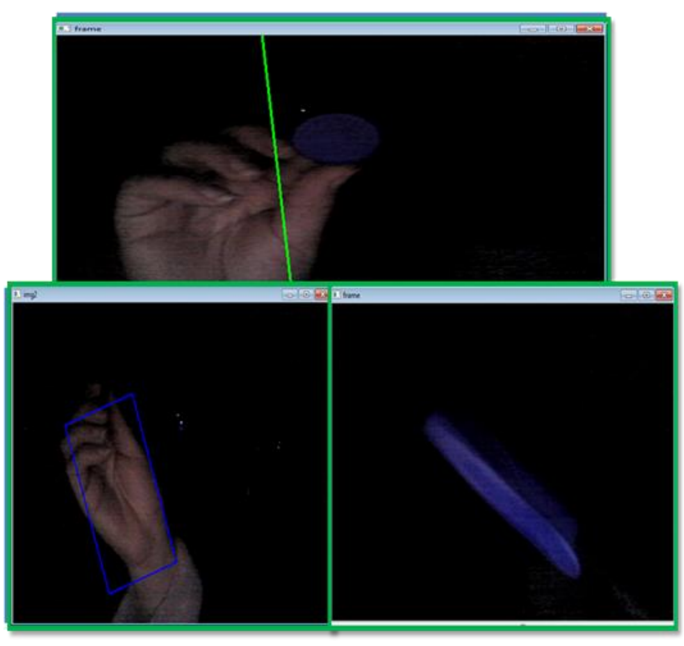
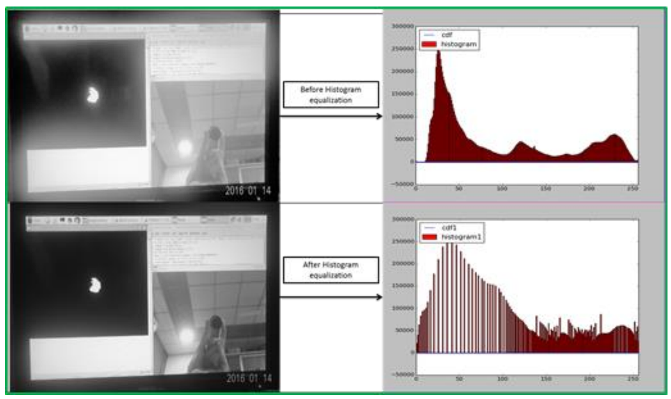


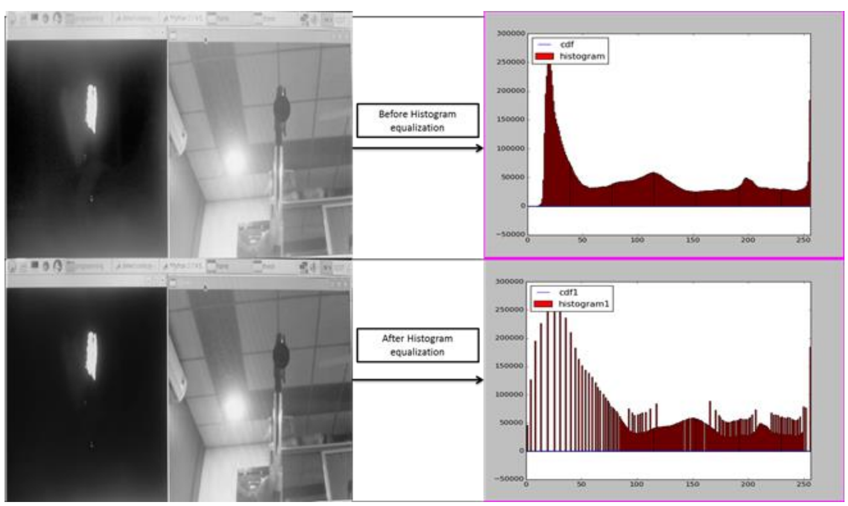
Fig.10. CamShift Algorithm in Very Low-Light Environment

With the results of the adopted algorithms, it is noticed that some of the images may be less obvious due to some external factors, e.g. the intensity of external lighting that negatively affect the tracking process.

Therefore, these results are passed on the histogram equalization techniques. The results proved a more visibility and this of course ensures the success of the tracking process as shown in Figure 11 (a,b).



(a)



(b)

Fig.1. Histogram Equalization.

8 CONCLUSIONS

In this paper, color detection algorithm in very low-light conditions technique is proposed for the enhancement of the target tracking. It is improved the traditional color detection performance and showed how our style can be solved. The appropriate color space and the statistical analysis to the various objects of this color space are adopted. The ways used to have the ability to adapt in the intensity of different lighting. So, the results that obtained in the case of daylight are the same in the darkness of night, and this is an important and necessary feature of this proposed algorithm the matter that makes them better than the rest of algorithms used in the tracking process.

Acknowledgments

We would like to express our sincere gratitude and thanks to Professor Dr. Ibrahim El-Emary, Computer Science and Systems, University of king Abdul Aziz, KSA for his continuous support, cooperation, and assistance.

References

[1] Yahya, M. F., and M. R. Arshad "Tracking of Multiple Light Sources Using Computer Vision for Underwater Docking", Procedia Computer Science, 76, pp.192-197, 2015.

[2] Firmanda, Dion, and Dadet Pramadihant "Computer Vision Based Analysis for Cursor Control Using Object Tracking and Color Detection", In Computational Intelligence and Design (ISCID), 2014 Seventh International Symposium on, IEEE, vol. 2, pp. 525-528.

[3] Allen, John G., Richard YD Xu, and Jesse S. Jin. "Object tracking using Camshift algorithm and multiple quantized feature spaces, Proceedings of the Pan-Sydney area workshop on Visual information processing, Australian Computer Society, Vol. 36, Inc., pp. 3-7, 2014.

[4] Zhao, Chuan, Andrew Knight, and Ian Reid "Target tracking using mean-shift and affine structure", 2008, ICPR 2008, 19th IEEE International Conference on Pattern Recognition, pp. 1-5.

[5] Emami, Ebrahim, and Mahmood Fathy, "Object tracking using improved CAMShift algorithm combined with motion segmentation", 2011 7th Iranian Conference on Machine Vision and Image Processing, IEEE, pp. 1-4.

[6] Altun, Halis, Recai Sinekli, and Ugur Tekbas, "An efficient color detection in RGB space using hierarchical neural network structure", Innovations in Intelligent Systems and Applications (INISTA), 2011 International Symposium on, IEEE, pp. 154-158.

[7] Zhang, Chunrong, Yuansong Qiao, Enda Fallon, and Chiangqiao Xu, "An improved Camshift algorithm for target tracking in video surveillance", 9th. IT&T Conference, Dublin Institute of Technology, Dublin, Ireland, 22nd.-23rd. pp.1-9, 2009.

[8] Wang, Aixia, Jingjiao Li, and Zhenlin Lu. "Improved Camshift with adaptive searching window", International Journal of Soft Computing and Software Engineering (JSCSE), Vol.2, No.3, pp.24-36, 2012.

[9] Li, Dongyu, "An Efficient Moving Target Tracking Strategy Based on OpenCV and CAMShift Theory", 3rd International Conference on Multimedia Technology (ICMT-13), Atlantis Press, Grant No. 2010QN026, pp.40-47, 2013.

[10] Vaitheeswaran, S. M. "Leader-Follower Formation Control of Ground Vehicles Using Camshift Based Guidance", The International Journal of Multimedia & Its Applications (IJMA), Vol.6, No.6, pp.21-33, 2014.

[11] Kumar, Nikhil, and Deepak Choudhary, "Real-Time Object Tracking and Color Detection Using Color Feature", International Journal of Computing and Technology (IJCAT), Volume 2, Issue 02, pp.27-30, 2015.

[12] Garofalaki, Zacharenia I., John T. Amorginos, and John N. Ellinas, "Object Motion Tracking Based On Color Detection for Android Devices", World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering, Vol.9, No:4, pp.893-896, 2015.

[13] Kanika Kapoor and Shaveta Arora, "Color Image Enhancement Based on Histogram Equalization", Electrical & Computer Engineering: An International Journal (ECIJ), Volume 4, Number 3, pp.73-82, 2015.

[14] Upton, Eben, and Gareth Halfacree "Raspberry Pi user guide", 2014.

[15] Dave Jones, "Picamera Documentation", 2016.