

Colour Object Tracking on Embedded Platform Using Open CV

Krutika A Veerapur, Ganesh V. Bhat

Abstract: Object tracking in real time is one of the most important topics in the field of computer Vision. Detection and tracking of moving objects in the video scenes is the first relevant step in the information extraction in many computer vision applications. This idea can be used for the surveillance purpose, video annotation, traffic monitoring, human-computer interaction, intelligent transportation, and robotics and also in the field of medical. In this paper, we are discussing color object tracking using OpenCV software on Eclipse platform and the implementation of the tracking system on the Pandaboard ES. CAMShift algorithm is used for object tracking which is based on Meanshift algorithm. The proposed approach is demonstrated for real-time multiple object tracking system.

Index Terms: CAMShift algorithm, Meanshift algorithm, OpenCV, Pandaboard-ES.

I. INTRODUCTION

Object tracking can be defined as the process of segmenting an object of interest from a video scene and keeping track of its motion, orientation, occlusion etc. in order to extract useful information. Real-time object detection and tracking is a critical tasks in many computer vision applications such as surveillance, driver assistance, gesture recognition and man machine interface. Here, colour object tracking is done using OpenCV software on Eclipse platform and the implementation of the tracking system is on the Pandaboard-ES. Research has been carried out for many years in this field. In this sequel, algorithms have been developed and tested on desktop machines [1]. Due to its programming flexibility, easy availability and user friendly, OpenCV has been used by many researchers to implement object tracking. The computational complexity of object tracking algorithms is quite high. The major contribution of this paper is implementation of color object tracking using OpenCV on embedded linux platform with the functions implemented using embedded C++. During the last few years, various researches has been undergoing for the different algorithms used for the tracking the particular segment. Meanshift method, Camshift method [1], Kalman filter [3], Background Subtraction[2] methods are the different methods used for the tracking of the objects in the video scenes. Meanshift method is an iterative process, by computing the meanshift value of the current position of the object and then moves the point to its meanshift value as the new position. The camshift method used for tracking the object is the modification of the meanshift method.

Revised Manuscript Received on 30 July 2013.

* Correspondence Author

PG Scholar Krutika A. Veerapur, M.Tech Electronics Engineering, Visvesvaraiya Technological University, Canara Engineering College, Mangalore, India.

Dr. Ganesh V Bhat, Professor, Department of Electronics and Communication Engineering, Visvesvaraiya Technological University, Canara Engineering College, Mangalore, India.

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](http://creativecommons.org/licenses/by-nc-nd/4.0/) article under the CC-BY-NC-ND license <http://creativecommons.org/licenses/by-nc-nd/4.0/>

The rest paper is organized as follows: section (ii) presents the overview of the complete embedded system used for object tracking. (iii) Presents the object tracking system using CAMshift algorithm, section (iv) presents the experimental analysis and its results, and section (v) presents the conclusion and future scope of the paper.

II. EMBEDDED SYSTEM

The goal of the object tracking is to track the position of one or more objects in real-time. Thus to ensure the progress of the work, we used the Opencv software in Ubuntu 12.04 OS. And the object tracking system implemented on the Pandaboard ES, which is a low cost embedded Linux based board that can be supported the Windows CE, Linux, Symbian OS. The Video camera used is a normal USB webcam which has a 4.0 MP video resolution at 15 fps. Pandaboard ES is specifically designed for the open source community. We can access the memory of the hardware by using a μ SD card. The experimental setup is shown in the fig 1.

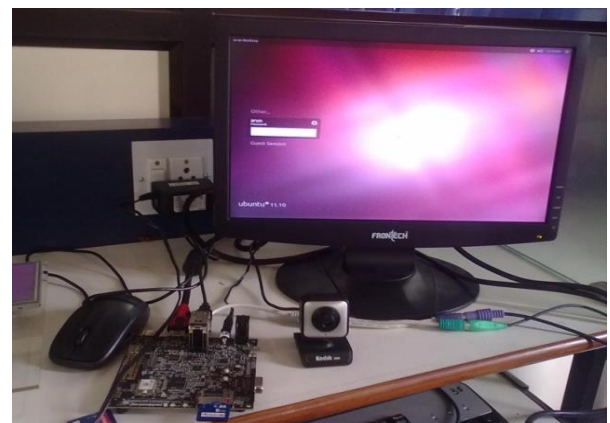


Fig.1. Experimental setup with Pandaboard and Web Camera

OpenCV has so many capabilities that it can seem overwhelming at first. A good understanding of how these methods work is the key to getting good results when using OpenCV. Fortunately, only a select few need to be known beforehand to get started. OpenCV's functionality that will be used for object tracking is contained within several modules [10]. OpenCV (Open Source Computer Vision Library: <http://opencv.org>) is an open-source BSD-licensed library that includes several hundreds of computer vision algorithms. The OpenCV Library is a way of establishing an open source vision community that will make better use of up-to-date opportunities to apply computer vision in the growing PC environment. The software provides a set of image processing functions, as well as image and pattern analysis functions.

Some of the OpenCV functions used in this tracking system are **CamShift**- Finds object center, size, and orientation. `int cvCamShift (IplImage* imgProb, CvRect windowIn, CvTermCriteria criteria, CvConnectedComp* out, CvBox2D* box=0);`

imgProb: 2D object probability distribution.

windowIn: Initial search window.

criteria: Criteria applied to determine when the window search should be finished.

out: Resultant structure that contains converged search window coordinates (rect field) and sum of all pixels inside the window (area field).

box: Circumscribed box for the object. If not NULL, contains object size and orientation.

The function CamShift finds an object center using the Mean Shift algorithm and, after that, calculates the object size and orientation. The function returns number of iterations made within the Mean Shift algorithm.

MeanShift- Iterates to find object center.

`int cvMeanShift (IplImage* imgProb, CvRect windowIn, CvTermCriteria criteria, CvConnectedComp* out);`

The function MeanShift iterates to find the object center given its 2D color probability distribution image. The iterations are made until the search window center moves by less than the given value and/or until the function has done the maximum number of iterations. The function returns the number of iterations made.

III. OBJECT TRACKING USING CAMSHIFT ALGORITHM

CAMshift is called Continuously Adaptive Mean Shift based on the mean shift algorithm.

A. MeanShift Algorithm

Mean-Shift is a robust method on finding mode in a density distribution of data set. This method has multi-functionality since the density is not only for color distribution, but also texture, motion, etc. This is an easy process for continues distributions which merely just hill climbing applied to a density histogram of the data. This method is efficient compare to standard template matching since it eliminates brute force search [1]. Those characteristics made Mean-Shift very well accepted by researchers. That is why we choose this method to review.

Calculation of parameters:

1) Zeroth moment calculation

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

2) First moment calculation

$$\begin{aligned} M_{10} &= \sum_x \sum_y x * I(x, y); \\ M_{01} &= \sum_x \sum_y y * I(x, y) \end{aligned} \quad (2)$$

3) Mean search window calculation

$$x_c = \frac{M_{10}}{M_{00}}; \quad y_c = \frac{M_{01}}{M_{00}} \quad (3)$$

where $I(x, y)$ is the intensity value of point (x, y) in the probability distribution image.

B. CAMShift Algorithm

CAMshift is called Continuously Adaptive Mean Shift based on the mean shift algorithm. CAMshift uses the Hue channel to trace objects since by using the Hue channel based on HSV color model, objects with different colors can be recognized. Based on the color information, CAMshift

tracks objects faster and consumes relatively little CPU resources. Lower computing resource requirement enable CAMshift to become a one of real-time face tracking algorithms. The procedure of CAMshift shown in the Figure 4 includes two important parts which are the histogram and the search of peak probability. The first step of Camshift is to obtain the histogram of tracked object. In the second step, the next frame will be converted into a map of skin color probability based on the histogram. In the third step, the peak probability center will be found based on zero moment and first moment [1] [6].

Finally, CAMshift will check whether it converges. If it is not, it will go to the third step; otherwise, it gets the position of the tracked object in this frame and fetches the next frame to track the object continuously.

Search window's new size can be computed as follows:

$$\begin{aligned} l &= \sqrt{\frac{(a+c)+\sqrt{b^2+(a-c)^2}}{2}} \\ w &= \sqrt{\frac{(a+c)-\sqrt{b^2+(a-c)^2}}{2}} \end{aligned} \quad (4)$$

Where l and w are the long and short axes of the search window respectively, and a , b , and c are obtained from Equation (5).

$$\begin{aligned} a &= \frac{M_{20}}{M_{00}} - x_c^2 \\ b &= 2 \left(\frac{M_{11}}{M_{00}} - x_c y_c \right) \\ c &= \frac{M_{02}}{M_{00}} - y_c^2 \end{aligned} \quad (5)$$

And the second order moments are calculated from Equation (6):

$$\begin{cases} M_{20} = \sum_x \sum_y x^2 I(x, y) \\ M_{02} = \sum_x \sum_y y^2 I(x, y) \\ M_{11} = \sum_x \sum_y xy I(x, y) \end{cases} \quad (6)$$

Finally the new size and location of the search window are used to iterate the algorithm from step 2.

The CAMSHIFT algorithm can be summarized with these steps:

- 1) Chose the initial region of interest which contain the object we want to track
- 2) Make a color histogram of that region as the object model
- 3) Make a probability distribution of frame using the color histogram. As a remark, in the implementation, they use histogram back projection method.
- 4) Based on the probability distribution image; find the center mass of the search window using mean-shift method.
- 5) Center the search window to the point taken from step 4 and iterate step 4 until convergence.
- 6) Process the next frame with the search window position from the step 5.

CamShift stands for the ‘‘Continuously Adaptive Mean-SHIFT’’ algorithm. Figure 2 summarizes this algorithm. For each video frame, the raw image is converted to a color probability distribution image via a color histogram model of the color being tracked, e.g., flesh color in the case of face tracking.

The center and size of the color object are found via the CamShift algorithm operating on the color probability image. The current size and location of the tracked object are reported and used to set the size and location of the search window in the next video image. The process is then repeated for continuous tracking. The algorithm is a generalization of the Mean Shift algorithm, highlighted in gray in Figure 2.

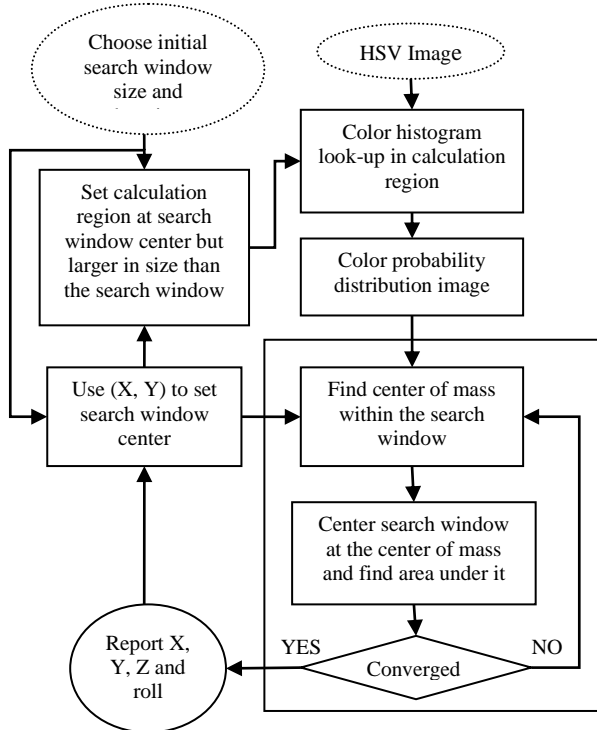


Fig 2: Block diagram of CamShift Algorithm

Camshift is able to handle the dynamic distribution by adjusting the size of the search window for the next frame based on the moment zero of the current distribution of images. In contrast to the algorithm Meanshift which is conceived for the static distributions, Camshift is conceived for a dynamic evolution of the distributions [9]. It adjusts the size of searching window by invariant moments. This allows the algorithm to anticipate the movement of objects to quickly track the object in the next frame. Even during the fast movements of an object, Camshift is still capable of tracking well. It occurs when objects in video sequences are tracked and the object moves such that the size and location of the change in probability distribution changes in time. The initial search window was determined by a detection algorithm or software dedicated to video processing. The CamShift algorithm calls upon the MeanShift one to calculate the target centre in the probability distribution image, but also the orientation of the principal axis and dimensions of the probability distribution.

IV. EXPERIMENTAL RESULTS

We implemented the CAMShift algorithm for object tracking using opencv software in Ubuntu 12.04 platform. The whole object tracking system is implemented on the Pandaboard ES. The program has been tested using the objects which have different shapes and size. The objects are detected and tracked successfully in real time and implemented the system on the Pandaboard ES.

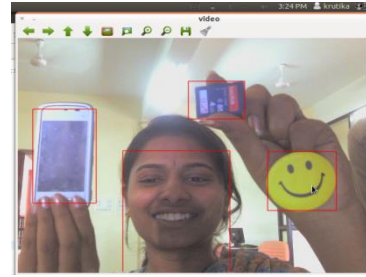


Fig 3: Multiple objects tracked using camshift algorithm

V. CONCLUSION

This paper focuses on the simultaneous tracking of the multiple objects in real time. The tracking of the various objects which have different shapes, sizes and colours is tested and implemented successfully. From this discussion we can conclude that the CAMShift algorithm is superior to other algorithms in order to track the position of the respective objects. The whole object tracking system is easily implemented on the Panda board-ES. In the future, we will implement the object tracking system more efficiently which tracks the objects at higher speed and displays the time taken to track the objects.

REFERENCES

1. Afef Salhi and Ameni Yengui Jammoussi, "Object tracking system using CamShift, Meanshift and Kalman filter", World Academy of Science, Engineering and Technology, 2012.
2. Alok K. Watve ,Indian Institute of Technology, Kharagpur, seminar on "Object tracking in video scenes", 2005.
3. Amir Salarpour and Arezoo Salarpour and Mahmoud Fathi and MirHossein Dezfulian, "Vehicle tracking using kalman filter and features", Signal & Image Processing : An International Journal (SIPIJ) Vol.2, No.2, June 2011.
4. C. Lakshmi Devasena, R. Revathi, " Video surveillance system-A survey", IJCSI International journal of computer science Issues, vol 8, issue 4, no.1, July 2011
5. Flavio B. Vidal and Victor H. Casanova Alcalde (2010). "Object Visual Tracking Using Window-Matching Techniques and Kalman Filtering", Kalman Filter, VedranKordic (Ed.), ISBN: 978- 953-307-094-0.
6. Greice Martins de Freitas, Clésio Luis Tozzi, "Object Tracking by Multiple State Management and Eigen background Segmentation", International Journal of Natural Computing Research, 1(4), 29-36, October-December 2010.
7. Hamidreza Rashidy Kanan and ParastoKarimi, "Visual Object Tracking Using Fuzzy-based Thresholding and Kalman Filter", International Journal of Modeling and Optimization, Vol. 2, No. 3, June 2012.
8. Jiyun Pan, Bo Hu, and JianQiu Zhang, "An Efficient Object Tracking Algorithm with Adaptive Prediction of Initial Searching Point", 2006 IEEE Pacific-Rim Symposium on Image and Video Technology (PSIVT'06), December 2006.
9. Marek Chovanec, "Computer vision vehicle tracking using background subtraction", Journal of Information, Control and Management Systems, Vol. 1,(2005),No.1.

AUTHOR PROFILE



Krutika A Veerapur, PG scholar, pursuing M.Tech in ELECTRONICS, under Department of Electronics and Communication, Canara Engineering College, Mangalore, Visvesvaraiya Technological University, Belgaum, India. She has completed her under graduate course in Electronics and Communication Engineering from Visvesvaraiya Technological University, Belgaum, India. Areas of interest are Image Processing and Embedded Systems.



Colour Object Tracking On Embedded Platform Using Open CV



Dr. Ganesh V Bhat has over 14 years experience in teaching and 2 years in Research and Development. His areas of interest include image processing, pattern recognition and open source embedded systems. He has published over 15 papers at international conference and journals. He is a Member of IEEE, Life member of Indian Society for Technical Education (ISTE). He is presently working as professor under Department of

Electronics and Communication, Canara Engineering College, Mangalore.