

## Multiple Objects Tracking using CAMshift Algorithm in Open CV

Aditi Jog<sup>1</sup>, Prof. Shirish Halbe<sup>2</sup>

<sup>1</sup>(Student, Pursuing masters in engineering in Electronics, /Department of Electronics, K.J.Somaiya college of Engineering / Mumbai University, India)

<sup>2</sup>(Associate Professor, Department of Electronics, K.J.Somaiya college of Engineering / Mumbai University, India)

---

**Abstract:** These paper reviews tracking of multiple moving objects with CAMshift algorithm in OpenCV. In many other cases CAMshift considered a feature of tracked object which need to track in video stream. Many times CAMshift algorithm used for face tracking. But in our paper we implement approach of CAMshift algorithm which is capable of track multiple moving objects in video stream.

In our paper, we have not considered any feature of object to be tracked. Here first Reference frame is selected by user and further continuously updated Binary foreground pixels are given as an argument to CAMshift, for locating location of desired objects in each frame. Motion contour detection is achieved. By providing overlap threshold criteria in scope of CAMshift algorithm multiple objects are tracked in consecutive video frames and ellipse is getting drawn around each tracked object.

We proved, CAMshift algorithm can be implemented in effective manner for multiple objects tracking in video stream.

**Keywords:** CAMshift, foreground pixels, multiple Object tracking, OpenCV.

---

### I. Introduction

Object Tracking is one of the key tasks in the field of computer vision. This is a useful operation in automated security solutions, traffic monitoring system, etc. Various methods of object tracking are available. But two main methods used of tracking are widely used in real time application [1]

MeanShift algorithm is designed for static distribution. The method tracks targets by finding the most similar distribution pattern in a frame sequences with its sample pattern by iterative searching. It is simple in implementation. But it fails to track the object when it moves away from Camera. [1]

Continuous adaptive mean-shift (CAMShift) was used to overcome this problem. CAMShift[5] is designed for dynamically changing distributions. These occur when objects in video sequences are being tracked and the object moves so that the size and location of the probability distribution changes in time. The CAMShift algorithm adjusts the search Window size in the course of its operation. Initial window size can be set at any reasonable value.

Feature selection plays an important role in object tracking. The most commonly used features are color, edges, [2] and various color spaces other than RGB are used for tracking purposes such as HSV. Because RGB color space does not correspond to the colour differences perceived by humans [3].

In this paper we use CAMshift algorithm for tracking of multiple moving objects but we don't use any feature information of objects to be tracked. First we performed detection of motion contours and tracked these motion contours in subsequent video frames. For motion detection, Image differencing method is used. Tracking of multiple objects is achieved by Continuously Adaptive Mean-SHIFT" algorithm. After calling CAMshift algorithm, the Ellipse shape is getting drawn around tracked objects. Instead of tracking all motion contours we have set overlapping threshold criterion for contour bounding boxes and tracked objects. So the tracked objects which satisfy this overlapping criterion, ellipse is getting drawn around them and these objects are only tracked in subsequent video frames. Small contours formed because of motion of a leaves of tree because of wind, etc are ignored. These small motion contours are not required to track

### II. Background Building

Term "Background" stands for a set of motionless image pixels, pixels that do not belong to any object, but moving in front of the camera [4]. Adoption of background changes such as change in lighting conditions and background scenes is also required.

In order to learn background we execute following steps with help of Open CV library functions and C code.

**Step 1-** Get video and perform a frame by frame query from input video stream. To achieve this cvQueryFrame() function is used.

**Step2-** Now here we need to calculate Running average of motion.CvRunningAvg () is used for this purpose. It calculates weighted sum of two images. I.e. as frames are quarried one after another, average is need to be updated depending upon change in background conditions for example slowly changing lightening. This is nothing but a moving average. This is stored in one of the created image. This image keeps on updating when every frame is quarried

**Step 3-** Now it is required to convert the scale of moving average calculated in step 2.for this CvConvertScale() function is used. This function applies linear transformation to all pixels in output image of step 2 and puts that result of linear transformation into some other destination image.

**Step 4-**Now we need to calculate absolute difference between current quarried frame and linear transformed image formed in step 4. In this step, each element is current quarried frame is getting subtracted from linear transformed image formed in step 3.As a result, function returns absolute difference of subtraction process. This whole process of calculating Absolute difference is called “Motion Segmentation”

**Step 5-** Resulting image obtained in step 4 is color image. It needs to convert in grey image. This conversion is performed by cvConvertScale() function of OpenCV. Result of this step is grey scale image.

**Step6-**Grey level image obtained in step 5 is thresholded to get binary image. Threshold values can be adjusted by user.

**Step7-** closing operation is performed here to eliminate dark patches and small holes from binary image obtained in step6.

In this way background is getting build up slowly. User can select count for frame which is used for background building.



Fig1. Background Building



Fig2. Quarried frame with foreground object

### III. Detection of Multiple Motion Contours in Video Frames

Moving object detection is first step toward Tracking of moving objects. In motion detection following steps are performed.

First step is to grab a frame from camera. For that `cvQueryFrame()` function is used. The function `cvQueryFrame` grabs a frame from a camera or video file, decompresses it and Returns it. This function is just a combination of Grab Frame and Retrieve Frame [3].

In order to get background accumulation information we use basic function such as `cvRunningAvg()`. The function `cvRunningAvg` [7] calculates weighted sum of two images. Once a statistical model is available, there is often a need to update the value slowly to account for slowly changing lighting, etc.

Next step is to find the absolute difference between current frame and calculated running avg. for this `cvAbsDiff()` function is used. Further image obtained from this operation is converted into Grey scale image and then thresholded to get Binary image. In order to remove small holes in binary image closing operation is performed.

Where there is a motion in video, motion contours are formed at that place. Now, our task is to find these contours. To achieve this `cvFindContours()` [7] function is used. The function `cvFindContours` retrieves contours from the binary image and returns the pointer to the first contour. Other contours may be accessed through the `h_next` and `v_next` fields of the returned structure. The function returns total number of retrieved contours.

Once Contours are obtained, bounding rectangles are drawn around them in order to get it visible on screen. This is the way we have detected a multiple contours in each frame of video by incorporating above mentioned steps in appropriate c code sequence.

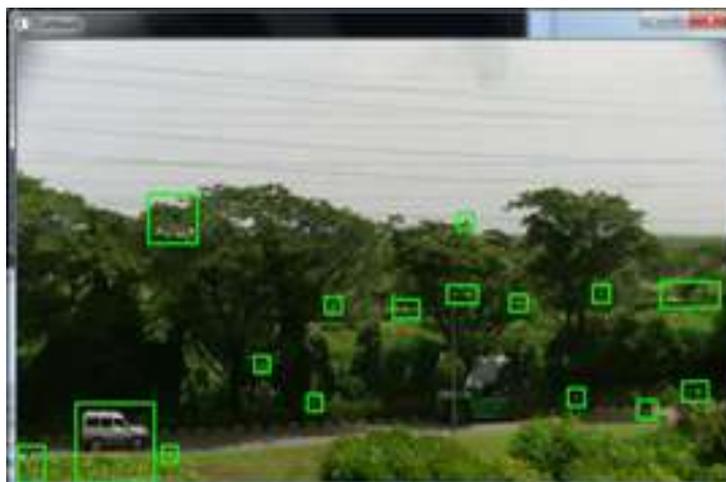


Fig3. Detected Motion contours in video frame.

### IV. Our Approach to implement CAMshift Algorithm to track Multiple Moving objects in video along with application of overlapping Criteria for tracked objects and contours

We have implemented CAMshift algorithm for multiple objects tracking in video frames. Ellipse shape is drawn around tracked objects and this ellipse follows this tracked object in further consecutive video frames. First task we performed is Background Building [8]. Running average of motion frames is calculated by function `cvRunningAvg()`. As the new frame is quarried, Current image frame is subtracted with background image and absolute differences is calculated by using function `cvAbsDiff()`. If difference is observed in current frame and background frame then that is detected as motion. These motion contours are finding by using function `cvFindContours`. These are some initial steps we performed before calling CAMshift Algorithm.

In CAMshift Algorithm, new position of tracked object is predicted. If it matches with its previous position then it is getting tracked in every consecutive video frames. In scope of CAMshift, RGB to HSV colour space conversion is performed by using `cvCvtColor()` and Hue value of object is extracted using function `cvSplit()`. Back projected image is calculated. This back projected image is nothing but histogram image of 'hue' value of object to be tracked. This hue image is getting updated every time when frame changes and so object position. But we don't use any hue information of objects in video frames to track it further. CAMshift function returns one 2D box which is nothing but a position of object in new frame. If this box position and contour bounding box satisfies predefined overlapping criteria written in scope of CAMshift algorithm then ellipse is drawn around that objects only. Ellipse follows these tracked objects in further video frames.

## 5. Experimental Results



Fig 4. Motion contours are detected and rectangle box is drawn around each of them.

**Step 1-** Get video and perform a frame by frame query from input video stream. To achieve this `cvQueryFrame()` function is used.

**Step 2-** Now here we need to calculate Running average of motion. `CvRunningAvg()` is used for this purpose. It calculates weighted sum of two images. I.e. as frames are queried one after another, average is need to be updated depending upon change in background conditions for example slowly changing lightening. This is nothing but a moving average. This is stored in one of the created image named as 'movingavg'. This image keeps on updating when every frame is queried

**Step3-** Now it is required to convert the scale of moving average calculated in step 2. for this `CvConvertScale()` function is used. This function applies linear transformation to all pixels in movingavg image and puts that result of linear transformation into some other destination image.

**Step 4-** Now we need to calculate absolute difference between current queried frame and linearly transformed image. Here, each element in current queried frame is getting subtracted from linearly transformed image formed in step3. As a result, function returns absolute difference of subtraction process. This whole process of calculating Absolute difference is called "Motion Segmentation"

**Step 5-** Resulting image obtained in step 4 is color image. It needs to convert in grey image. This conversion is performed by `cvConvertScale()` function of OpenCV. Result of this step is grey scale image.

**Step 6-** Grey level image obtained in step 5 is thresholded to get binary image. Threshold values can be adjusted by user. `cvThreshold()` function is used for thresholding. Closing operation is performed here to eliminate dark patches and small holes from binary image obtained in s

**Step 7-** In order to find motion contours in video frames `cvFindContours()` function is used. It Retrieves contours from the binary image and returns the number of retrieved contours. To make this obtained contour visible we have drawn Rectangle around each contour with help of function `cvRectangle()`. As shown in above Fig.4.



Fig 5. After calling CAMshift Algorithm, Ellipse is getting drawn around only those motion contours which are getting tracked in consecutive frames.

CAMshift algorithm basically returns the position of tracked object in next video frame. If this box position satisfies overlapping threshold criteria mentioned in scope of CAMshift algorithm with bounding box of rectangle then Ellipse is drawn around that object. For drawing ellipse `cvEllipseBox()` function is used. So as shown in above result fig 5, ellipse is drawn around tracked objects. But if box position and bounding box not satisfies overlapping threshold criteria then no ellipse is getting drawn around that object. Just contour bounding box is seen.



Fig 6. After calling CAMshift Algorithm, Ellipse is getting drawn around only those motion contours which are getting tracked in consecutive frames.

## V. Conclusion

We proposed the efficient approach of CAMshift algorithm to track a multiple objects in consecutive video frame. In this approach, feature information of objects to be tracked is not required. We finally evaluated our algorithm's performance in practice, and showed how our approach can stand tracking Multiple moving objects. We have successfully implemented Use of overlap threshold criteria for tracked objects and contours. Very small area contours are not tracked by using this criterion of overlap threshold. Our experimental results proved that multiple moving objects are getting tracked by use of CAMshift algorithm.

## VI. Future Work

After implementing CAMshift algorithm, we observed that it reduces the speed of operation. So our next task is to find out the root cause and work on it to improve speed of operation. Also we are going to work on the occlusion handling. After doing some modification in scope of algorithm our algorithm can track objects which are occluded by any static object.

## References

- [1] J.A.Yilmaz, O.Javed, and M.Shah,"Object Tracking: A survey" *ACM Computing Surveys*, vol. 38, p. 13. 2006.
- [2] Jean-Yves Bouguet (2000): Pyramidal Implementation of the Lucas Kanade Feature Tracker – Description of the algorithm, Intel Corporation Microprocessor Research Labs.
- [3] PASCHOS, G. 2001. *Perceptually uniform color spaces for color texture analysis:an empirical evaluation*. IEEE Trans. Image Process. 10, 932-937.
- [4] Willow Garage,Inc, Willow garage <http://www.willowgarage.com/> (August 2009).
- [5] G. Bradski, and A. Kaehler, "*Learning OpenCV*", (O`Reilly, 2008, pp.337-341)
- [6] G. J. Allen, Y. D. Richard Xu and S. Jin Jesse, "*Object Tracking Using CAMShift Algorithm and Multiple Quantized Feature Spaces*", Inc. Australian Computer Society, vol.36, 2004.
- [7] *OpenCV Reference Manual* v2.1, March 18, 2010
- [8] Ebrahim E, Mahmood Fathy," Object Tracking Using Improved CAMShift Algorithm Combined with Motion Segmentation", *IEEE Trans*, pp. 1-4, Nov 2011.
- [9] Gauthier Lemoine (2010, October) Follow region of image frame by frame [Online].Available: <http://gauth.fr/category/>