I.J. Image, Graphics and Signal Processing, 2013, 6, 43-48
Published Online May 2013 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijigsp.2013.06.06

Multiple Objects Tracking Using CAMShift Algorithm and Implementation of Trip Wire

Aditi Jog
K.J. Somaiya college of Engineering, University of Mumbai, India
e-mail: jogaditi100@gmail.com

Prof.Shirish Halbe
Department of Electronics, K.J. Somaiya college of Engineering, University of Mumbai, India
e-mail: kjscetpo@gmail.com

Abstract— In this paper we represent Security application which is developed using concepts of Video Analytics. User can draw Trip wire on video stream with help of Mouse Callback events. Using this application user can restrict any area of total video scene. Direction selection for tripping is also a choice of a user. If any undesired moving object cross this drawn trip wire then motion of this moving object is getting detected and also tracked. If object crosses trip wire in the same direction as that of user selected then Alarm Indication will appear on that moving object. OpenCV library functions are used for motion detection and motion tracking. CAMShift algorithm is implemented for tracking. An experimental result shows Motion detection, Motion Tracking and drawn trip wire on video.

Index Terms— CAMshift, Motion detection, OpenCV library, Trip Wire

I. Introduction

Video Analytics technology delivers effective solution for surveillance system. This technology performs variety of task ranging from analysis of video for motion detection, motion tracking and also implementation of trip wire on video stream. Hence, Video Analytics is ideal solution to meet the needs of surveillance system.

Our application is developed in following three stages. First is detection of motion contours in video frames, Second stage is tracking of detected motion contours and third and last stage is drawing of trip wire on video stream. Depending upon tripping direction selected by user, Alarm indication will appear on that moving object and its motion is getting detected and tracked. Trip wire can be drawn in Horizontal Vertical or in inclined manner.

Background building is prime task necessary to perform before achieving detection of motion contours. Here weighted sum of two video frames is calculated by OpenCV function cvRunningAvg(). In order to identify moving parts of video scene and exclude them in course of background, cvAbsDiff(), cvThreshold() functions are used[1]. Further cvFindContour() finds contours from image which is a result of function cvThreshold(). In order to display location of detected motion contours green rectangle box is drawn around each motion contour with help of function cvRectangle() .

Motion tracking is achieved by Continuous Adaptive Mean-Shift algorithm (CAMShift) [1]. For each video frame, raw image is converted into color probability image via histogram model [3].The centre and sizes of color object are found via CAMShift. But in our application we need to track each and every moving object in video scene regardless of its color. Hence, we have added some additional changes in basic scope of tracking algorithm. First addition in scope is use of Binary image. Which is obtained from function cvThreshold(). This binary Image keeps on updating for every quarried frame. This image provides location of object to the CAMShift Algorithm. Secondly Objects to be tracked is formed by using image called Mask image. This image provides exact object which is need to be tracked by tracking algorithm. Mask image is also keeps on updating for each quarried frame. Mask image is formed by using certain combination of OpenCV functions.

After performing Motion detection and motion tracking finally trip wire is drawn on video. Mouse callback events are used for drawing trip wire. This trip wire can be drawn by user in either horizontal, vertical or in inclined manner. Direction of tripping is chosen by user. Bidirectional tripping is also available. Undesired motions are detected tracked and Alarm indication will appear on that moving objects which will cross trip wire in same direction that is of tripping direction selected by user.

In further paper, we started with Background building for quarried video frames [2] and motion contour detection. Further we proceed with mask image formation for extracting exact object which is to be tracked, Motion tracking concept and trip wire implementation on video scene by mousecallback Events. Direction selection idea for tripping is also added by drawing appropriate pictures.

Experimental results screenshots are shown to get clear idea of our actual concept implementation.

II. Background Building and Motion Contour Detection

Background stands for set of motionless pixels that do not belong to any object, but moving in front of camera.

Following steps are performed to achieve background building and motion detection.

Step1- Get the video and perform frame by frame query from obtained video stream. cvQuerryFrame() function performs this task.

Step2- Calculate the weighted sum of frames which are quarried one after another. This can be achieved by function cvRunningAvg().

Step3- Apply linear transformation to all pixels in output image obtained from step2. For this transformation cvConvertScale() function is used.

Step4- Calculate absolute difference between current quarried frame and linear transformed image obtained in step3 [3].

Step5- Output image in step4 is color image. It need to convert into grey level image this conversion is performed by cvCvtColor().

Step6- Obtained grey level image is thresholded to get binary image.

Step7- Closing operation is performed to remove small holes and dark patches from binary image obtained in step6.

Now where there is a motion in the video scene, Motion contours are formed at that place. Our task is to find these motion contours [4]. To achieve this cvFindContour() function is used. This function retrieves contours from binary image and returns pointer to first contour. Other contours are accessed through h_next and v_next fields of returned structure [5]. The function returns total number of retrieved contours.

Once contours are detected a green color rectangle box is drawn around each motion contour in order to display its presence in video frame. This is the way multiple motion contours are getting detected.

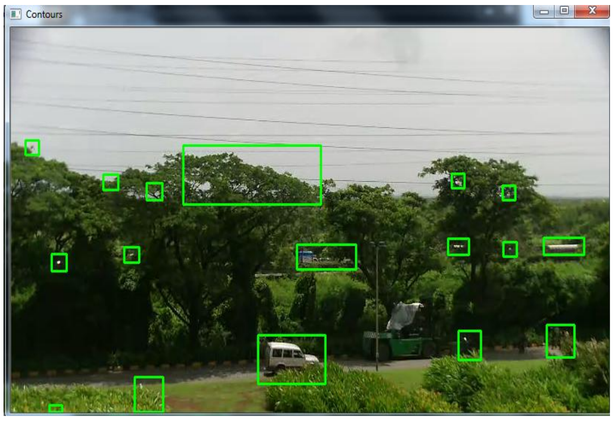

Fig.1 Motion contour detection in video frame

III. Binary Foreground Image and Formation Of Mask Image

Binary foreground image is used to provide location of object to be tracked to the CAMShift algorithm. This image is kept on updating as the frames are quarried and motion contours are formed. This binary foreground image is provided as a one of argument to tracking algorithm.

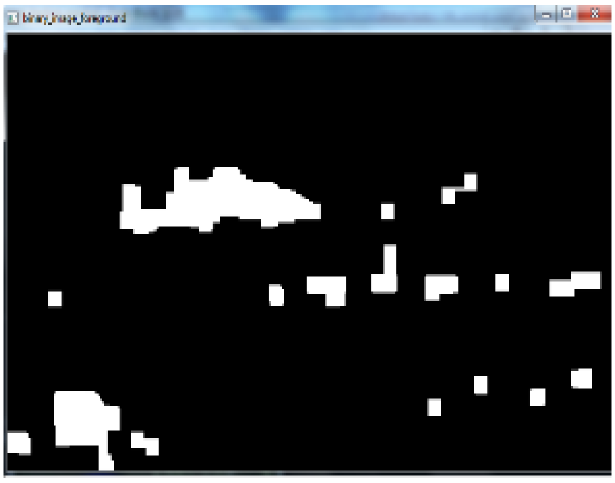

Fig.2 Binary foreground image per quarried video frame

Mask Image is used to create an object which is to be tracked by algorithm. This image is given as a one of argument to cvCAMShift() function. Following steps are performed to form a Mask image.

1. Create a Black image
2. Draw a filled rectangle on that black image. Its dimensions are same as that of bounding box drawn on required motion contour.
3. Perform AND operation of image with filled rectangle, Binary foreground image shown in fig 2.

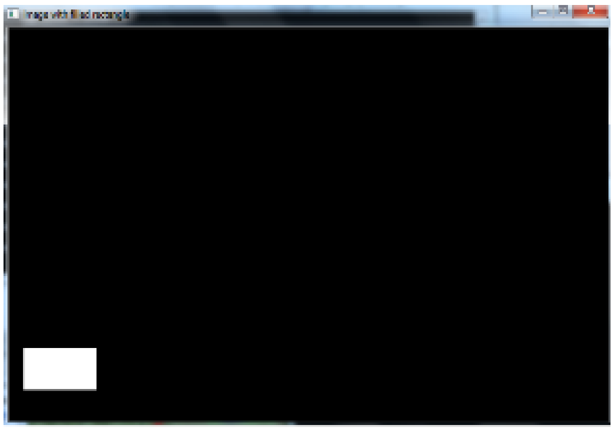

Fig.3 Black Image with filled white rectangle

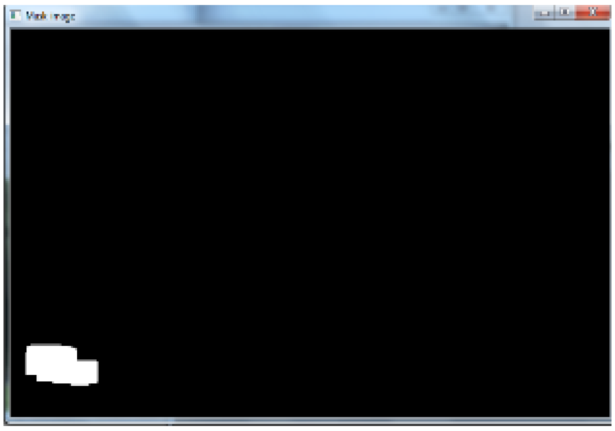

Fig.4 AND operation Result (Figure2 AND Figure3)

In fig. 4 objects to be tracked is created. This image along with quarried video frame is fed to tracking algorithm. Hence CAMShift gets idea about object to be tracked.

IV. Motion Tracking

These created tracked objects along with quarried frames are fed as an argument to CAMshift function. In CAMshift, new position of tracked object is predicted [7]. 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 [8]. 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 [9]. 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.

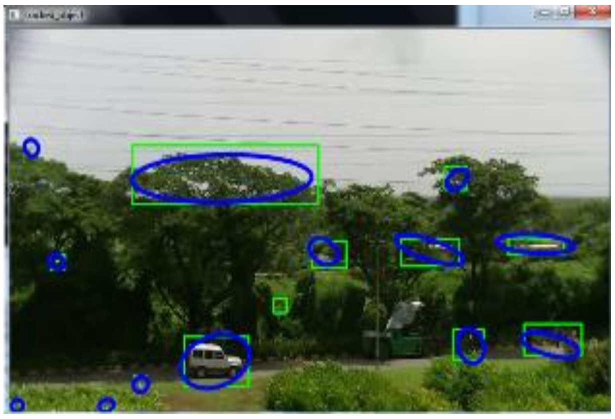

Fig.5 Motion tracking (with No contour area threshold)

As shown in fig. 5, all the detected motion contours are getting tracked. But because of this unwanted small area motion contours are also tracked by tracking algorithm which is undesirable. Hence some contour area threshold is applied to remove these small area motion contours.

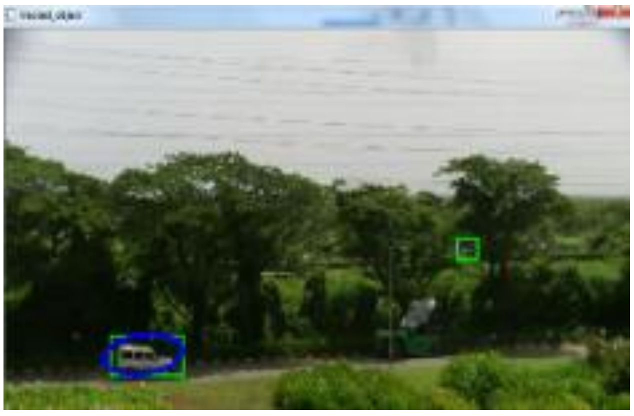

Fig.6 Motion tracking (with contour area threshold)

As shown in Fig. 6, only required object is tracked. Other small area contours are not tracked because of application of contour area threshold.

V. Drawing Trip Wire on Video

OpenCV supports mouse events. With help of mouse callback events, we can detect the specific mouse event and (x,y) coordinates of the mouse pointer. Mouse events are handled by a more typical "callback mechanism" [10]. This means that, to enable response to mouse clicks, we must first write a callback routine that OpenCV can call whenever a mouse event occurs. Once we have done that, we must register the callback with OpenCV, thereby informing OpenCV that this is the correct function to use whenever the user does something with the mouse over a particular window. For performing callback, OpenCV uses function cvSetMouseCallback( ).

This tripwire is drawn on video. With help of this user can restrict any area of total video scene [11]. Direction for tripping is depending upon user requirement. Trip wire can be drawn either vertical, horizontal or in inclined manner. If object crosses trip wire in same direction as that of tripping direction selected by user then Alarm indication will appear on that moving object.

Slope of trip line is calculated. Position of this line is needed to be fixed before executing tracking algorithm. If slope of tracked object is lesser than slope of trip wire then object is still behind trip wire. That means object has not crossed trip wire yet. When this slope of tracked object becomes greater than trip wire slope then we can say it has crossed trip wire.

A. Direction Selection of Trip wire.

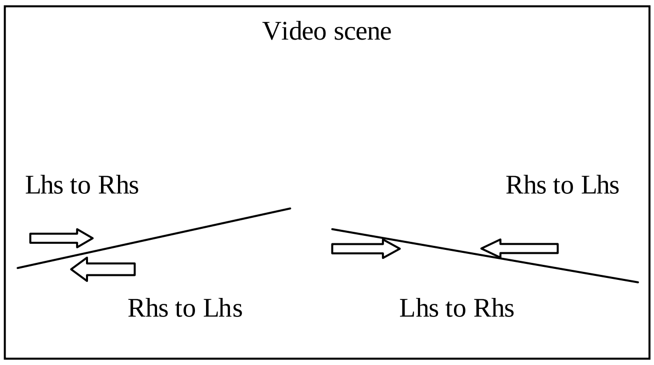

Fig.7 Horizontal trip wires concept

In above fig. 7, Horizontal trip wire is shown. Alarm indication will appear on moving objects which crosses this horizontal trip wire either from Lhs to Rhs direction or from Rhs to Lhs direction and also For Bidirectional. Left hand side trip wire has negative slope and right hand side trip wire has positive slope. Motion detection and tracking is also implemented for objects which crosses trip wire. Trip wire having slope less than 30 degrees is considered to be horizontal trip wire.

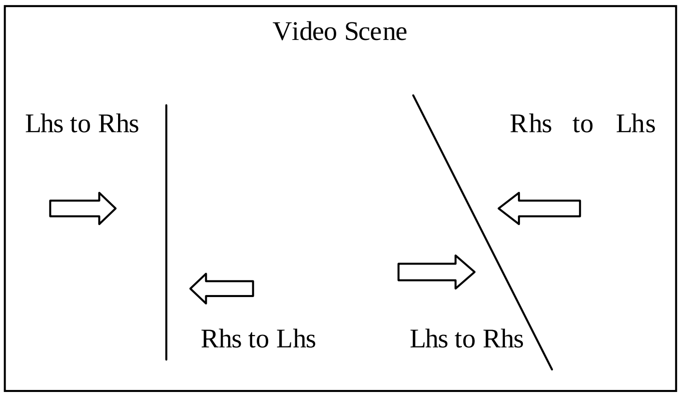

Fig.8 Vertical and inclined vertical trip wires concept

Fig. 8 shows Concept for vertical trip wire and inclined trip wire. Alarm Indication will appear on object which crosses trip wire. Here not only vertical (slope of line = infinity) trip wire but also inclined vertical trip wire (for positive and negative slope) is drawn. User will select any of one trip wire at a time to be drawn on video scene. Trip wire having slope greater than 30 degrees is consider to be Vertical trip wire.

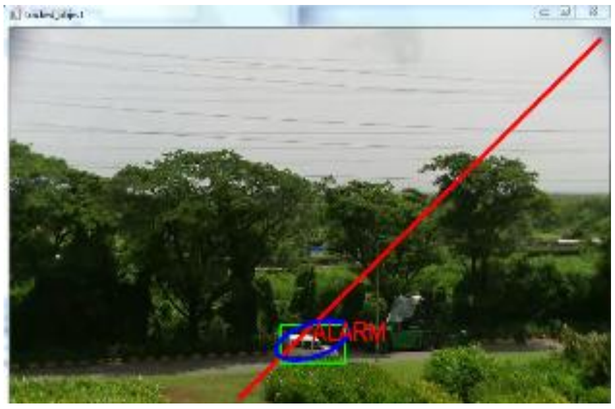

Fig.9 Vertical inclined trip wire (negative slope)

As shown in fig. 9,

Type of trip wire- Negative slope Trip wire

Direction selection- Lhs to Rhs

Action Taken- Alarm indication will appear on moving objects which has crossed trip wire from Lhs side toward Rhs side.

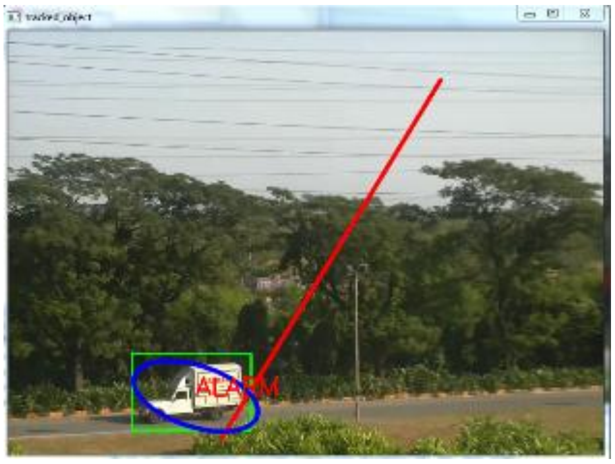

Fig.10 Vertical inclined trip wire (negative slope)

As shown in fig.10,

Type of trip wire- Negative slope Trip wire

Direction selection- Rhs to Lhs

Action Taken- Alarm indication will appear on moving objects which has crossed trip wire from Rhs side toward Lhs side.

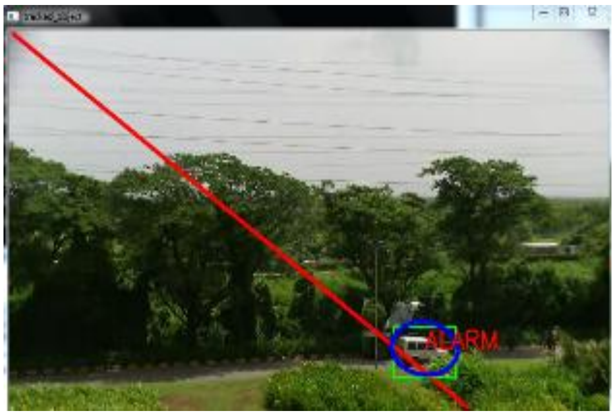

Fig.11 Vertical inclined trip wire (Positive slope)

As shown in fig.11,

Type of trip wire- Positive slope Trip wire

Direction selection- Lhs to Rhs

Action Taken- Alarm indication will appear on moving objects which has crossed trip wire from Lhs side toward Rhs side.

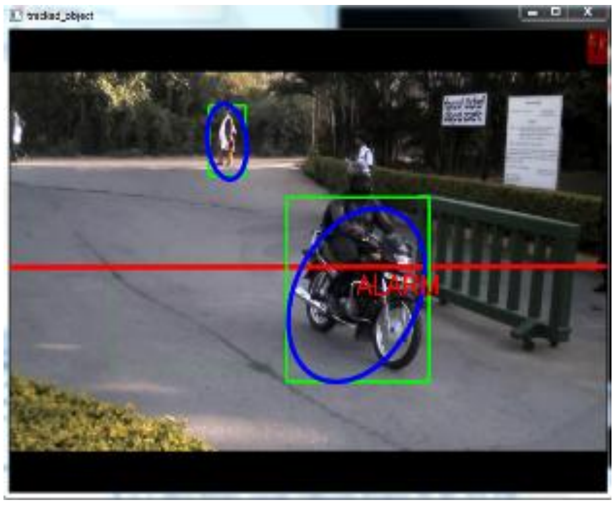

Fig.12 Horizontal trip wire (slope is equal to zero)

As shown in figure12,

Type of trip wire- Horizontal trip wire with slope equals to zero

Direction selection- Lhs to Rhs

Action Taken- Alarm indication will appear on moving objects which has crossed trip wire from Lhs side toward Rhs side

VI. Conclusion

This is the conclusion. The major contributions of this article arise from the formulation of a new approach to implement CAMshift algorithm for moving object tracking even though features of object to be tracked is not known to user. It's a security application. High security is achieved by implementing trip wire along with tracking of moving objects. One can track undesired motion taking place in restricted areas just by sitting in control room. Alarm indication provides alert for user. This type of security application is useful for indoor as well as outdoor scenes. Maintaining Proper Illumination is necessary.

Acknowledgment

This paper is fully supported by Electronics department of the K. J. Somaiya college of Engineering, Mumbai, India.

References

[1] Ebrahim E, Mahmood Fathy," Object Tracking Using Improved CAMShift Algorithm Combined with Motion Segmentation", *IEEE Trans*, pp. 1-4, Nov 2011.

[2] L. Zappella, X. Lladó, and J. Salvi. (2008) "Motion Segmentation: A Review". 11th International Conference of the Catalan Association for Artificial Intelligence (CCIA'08), 2008.

[3] Y. Cheng "Mean shift, mode seeking, and clustering," *IEEE Trans,Pattern Anal.*,vol.17,pp. 790-799, 1995.

[4] G. Bradski, and A. Kaehler, *Learning OpenCV.* United States of America: O`Reilly, 2008.

[5] Niel Joubert [Online] Background Modelling and Subtraction (2010). Available:http://dip.sun.ac.za/

[6] Birgi Tamersoy[Online] Background Subtraction (September 2009). Available: http://www.cs.utexas.edu/

[7] Gary R. Bradski,R.T. Collins, Y. Liu, and M. Leordeanu "Computer Vision Face Tracking For Use in a Perceptual User Interface",Intel Technology J., Q2, 1998.

[8] Fahad E.G,Pierre-YvesBayle, "People tracking by modified CAMShift algorithm"Distributed computing and Algorithm for business Portal (DCABES),2009

[9] 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.

[10] Gary R. Bradski,"Computer vision face tracking for use in a perceptual user interface", Intel Technology Journal Q2, 1998

[11] Zulikha Kadim, Lian KimMeng and Norshuhada Samudin, [Online] Video Analytics Algorithm for detecting Objects crossing lines in specific direction using BlobBasedAnalysis (2012). Available: https://www.iiwas.org/ACM/p58-Samudin

Aditi Jog, female, is received her Bachelor's of Engineering(BE-Electronics) degree from Mumbai University in Year 2009. Currently she is pursuing her Masters of engineering (ME-Electronics) from K.J.Somaiya College of engineering at University of Mumbai. Her areas of Intrests are Image Processing, Data communication Networks.

Prof. Shirish Halbe, male, is currently Assistant Professor in the K.J.Somaiya College of engineering affiliated by university of Mumbai. He received BE, ME degree in 1982, 1995 from Pune and Mumbai university respectively.

He is a life member Indian Society of Technical Education, New Delhi, India. His research areas are Image processing, Robotics and Medical Electronics.