

# Disclosure and Sniffout of Various Moving Entity in Real World

Vidya R

Dept.CSE

T.John Institute of Technology

Bangalore,India.

Rajitha Nair

Asst.Prof,Dept.CSE,

T.John Institute of Technology

Bangalore,India.

**Abstract:** Object detection and tracking is an important task within the field of computer vision due to its promising applications in many areas such as video surveillance, traffic monitoring, vehicle navigation, robotics, 3D reconstruction and content based indexing and retrieval. Motion detection consists of identifying the foreground and background objects into the scene, and tracking is the process of following image objects in their movement through an image sequence. Many methods for object detection and tracking have been proposed each having its own strengths and weaknesses. In-case of motion detection without using any models, the most popular region-based approaches are background subtraction and optical flow. Background subtraction detects moving objects by subtracting estimated background models from images. This method is sensitive to illumination changes and small movement in the background. Optical flow also has a problem caused by illumination changes since its approximate constraint equation basically ignores temporal illumination changes.

The proposed, system is robust in various environments including indoor and outdoor scenes and different types of background scenes. The proposed method is robust because it uses edge-based features and clustering is used which makes it insensitive to illumination changes. The method is also fast because the area to be covered in edge-based features is less than region based features and is not much computationally expensive.

The proposed algorithm for detection and tracking of multiple moving objects in real time for both indoor and outdoor environment was extensively tested to operate in complex, real world, non-plain and changing background. To evaluate the performance of the proposed tracking system quantitatively, the pixel wise distance from the centroid of the tracking window with ground truth that is obtained manually is compared and found to possess remarkable very high accuracy and precision as compared to other previous work.

**Keywords:** Detection , Tracking , K-means,clustering

## I. INTRODUCTION

Motion detection is a well-studied problem in computer vision. There are two types of approaches: the region-based approach and the boundary-based approach. In the case of motion detection without using any models, the most popular region-based approaches are background subtraction and optical flow. Background subtraction [1] detects moving objects by subtracting estimated background models from images. This method is sensitive to illumination changes and small movement in the background. Many techniques have been proposed to

overcome this problem. The mixture of Gaussians is a popular and promising technique to estimate illumination changes and small movement in the background. However, a common problem of background subtraction is that it requires a long time for estimating the background models. It usually takes several seconds for background model estimation because the speed of illumination changes and small movement in the background are very slow. Optical flow also has a problem caused by illumination changes since its approximate constraint equation basically ignores temporal illumination changes. In the case of boundary-based approaches, many of them use edge-based optical flow, level sets, and active contours [2][3]. In detecting moving edges, the zero-crossings at each pixel are calculated from the convolution of the intensity history with the second-order temporal derivative of the Gaussian function[4][5]. However, the results of this method are inaccurate when the image is not smoothed sufficiently because it computes velocity without using spatial information of neighborhood pixels.

Most of the existing edge pixel based approaches suffer from random noise[6][7]. Pixel by pixel matching of edge segments is not suitable for matching as well as tracking due to higher computational cost. Edges that are visible at current frame might not be found at some later frames. In the existing edge pixel based methods it is not possible to apply different amount of transformation for different parts of edge pixels at the same time. As a result it cannot achieve accurate matching of all parts of the object model in subsequent frames. This phenomenon creates problem in object tracking and it makes difficult to deal with complex motion and shape change situation.

## II. MOTIVATION

Although Object detection & tracking has been studied for dozens of years, it remains an open research problem. A potent, rigorous and high attainment approach is still a great challenge today. The difficulty level of this problem highly depends on how the object to be detected and tracked is defined. If only a few imaged appearance, such as a specific color, are used as representation of an object, it is fairly easy to identify all pixels with same color as the object. On the other extremity, the face of a specific person, which full of perceptual details and interfering information such as different poses and illumination, is

very hard to be accurately detected, recognized and tracked. Most challenges arise from the image variability of video because video objects generally are moving objects. As an object moves through the field of view of a camera, the images of the object may change dramatically. This variability comes from three principle sources: variation in target pose or target deformations, variation in illumination, and partial or full occlusion of the target.

Understanding activities of objects in any environment by the use of video is both a challenging scientific problem and a very fertile domain with many promising applications. Thus, it draws attentions of several researchers, institutions and commercial companies. The motivation in studying this problem is to create a visual surveillance system with real-time moving object detection and tracking capabilities.

### III. METHODOLOGY

The proposed method uses algorithm which includes three main stages, the first stage is generation of reference edge lists, detection of moving object and then updating edge lists. The system maintains two background reference edge lists. The first one is initial reference edge list, obtained by accumulating the training set of background edge images and the other one is temporary reference edge list which is constructed by the conditional addition of the moving edge segments. The initial reference edge list is static and requires no update but the temporary reference edge list is updated at every frame. Here, edges are extracted from video frames using the canny edge detector. For the detection of moving objects at current frame, the input edge segments are extracted from current frame forming current edge list. Distance maps for reference edge lists are obtained by a distance transformation for fast computing. The cluster segments are based on the edge segment connectivity which emphasize on relative edge distance and motion direction. Initially, the grouping of clusters is done by using a distance based iterative k- means clustering. To cluster segments each edge segment is enclosed within a rectangular boundary and the midpoint of this boundary is used as the edge representative in the computation of the K-centroids for the K-clusters.

### IV. SYSTEM ARCHITECTURE

System architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures of the system.

The system maintains two background reference edge lists. The first one is initial reference edge list, obtained by accumulating the training set of background edge images and the other one is temporary reference edge list which is constructed by the conditional addition of the moving edge segments. The initial reference edge list is static and requires no update but the temporary reference edge list is updated at every frame. Here, edges images

are extracted from video frames using the canny edge detector. During edge extraction process some of the prominent background edges might not be extracted in a particular illumination. For the detection of moving objects at current image, the input edge segments are extracted from current image forming current edge list. Distance maps for reference edge lists are obtained by a distance transformation for fast computing. Reference update allows detecting moving objects in dynamic environment. The figure 1 shows the system architecture for moving edge detection.

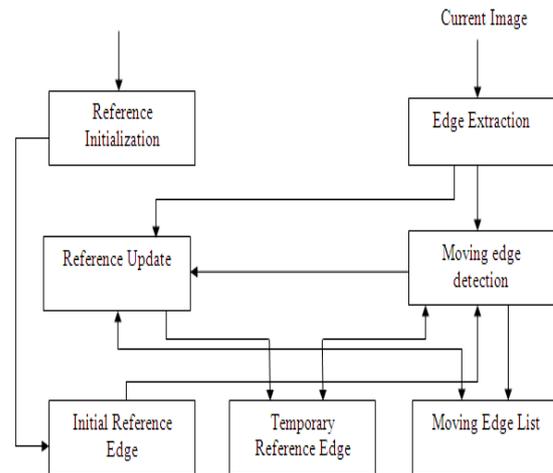


Figure 1: Architecture for Moving Edge Detection

### V. IMPLEMENTATION

Tracking problem is to reliably and accurately recognize the moving regions within frames which correspond to same moving object over time. To do so, first detect moving edge segments in current frame. A short description of the methods is explained below:

#### Moving Edge Segment

The proposed method uses algorithm which includes three main stages generation of reference edge lists, detection of moving object and updating edge lists.

#### Generation of reference edge lists

The system maintains two background reference edge lists. The first one is initial reference edge list, obtained by accumulating the training set of background edge images and the other one is temporary reference edge list which is constructed by the conditional addition of the moving edge segments.

#### Detection of Moving Object

For the detection of moving objects at current image, the input edge segments are extracted from current image forming current edge list. Distance maps for reference edge lists are obtained by a distance transformation for fast computing. Each edge segment on current edge list is searched.

**Updating edge lists**

Reference update allows detecting moving objects in dynamic environment. If a detected moving edge segment is already registered in the moving edge list in the same position, its associated weight value is increased. On the other hand, already registered moving edge segment from the moving edge list will lose its weight value if it is not redetected in the same position. In this process the segments having the weight value greater than  $T_M$  are moved from moving edge list to temporary reference edge list. An edge segment will be dropped from the temporary reference edge list if its weight reaches to zero.

**Selection of Feature points**

A common problem in computer vision is the registration of feature point sets. Especially, when a cluster of point samples from one frame of an object is matched with another cluster in another frame. The task of registration is to place the data into a optimum location by estimating the transformations of the object between the two frames. The parameter vector of the object's motion is estimated based on minimizing the sum-of-squared differences between the reference feature points in the reference frame and the observed feature points in the tracking sequence frame. This becomes challenging because correspondences between the point sets are not known beforehand. The selection of invariant features that can be reliably computed is a key component to such registration approaches. Curvatures are local features that are invariant to all affine transformations. Past research on range data has shown that surfaces may be classified by observing the signs of their mean and Gaussian curvatures. Thus if curvatures could be reliably and consistently calculated they would be ideal for feature-based registration. The computation of curvature of a surface requires the estimation of the second derivative of the surface. For a curve given parametrically as  $c(u) = (x(u), y(u))$ , the curvature is

$$K = \frac{|x''y' - x'y''|}{(x'^2 + y'^2)^{3/2}} \text{ -- Equation (1)}$$

where  $x' = dx/du$ ,  $x'' = d^2x/du^2$ ,  $y' = dy/du$  and  $y'' = d^2y/du^2$ .

**Determining edge Transformation parameters**

Affine transformations are well known in computer vision for recognizing objects. It has been shown that a 2-D affine transformation is equivalent to a 3-D rigid motion of the object followed by orthographic projection and scaling. Given the point correspondences between the two views the affine transformation which relates the two views can be computed by solving a system of linear equations using a least-squares approach. For computing edge segment orientation let us assume that each edge is represented by list of curvature points ( $k_1, k_2, \dots, k_n$ ), where point  $k(x,y)$  can move to point  $k'(x',y')$ , then coordinates of  $k'$  can be expressed in terms of the coordinates of  $k$ , through an affine transformation, as follows:

$$k' = \theta x \ k + b \text{ ----- Equation (2)}$$

An Affine transformation that handles translation, rotation, scale and shear in 2D Space can be described by 6

parameters. Rewriting the Equation (2), in terms of the image point coordinates, we have :

$$x' = \theta_{11}x + \theta_{12}y + b_1 \text{ ----- Equation (3)}$$

$$y' = \theta_{21}x + \theta_{22}y + b_2 \text{ ----- Equation (4)}$$

**Prediction filter for edge segment tracking**

The edge orientation parameters and location of the tracked edge segment is obtained by determining edge transformation parameters. Using these parameters as an observation for the edge position we apply Kalman filtering to predict every edge segments location in the next frame. Since a Kalman filter only needs information from the previous state, we can update filter for each frame and predict for the next frame. Here, the idea is to feed the parameters obtained by using the least squares method to the Kalman predictor to predict edge location for the successive frames and take the output of the Kalman predictor as the predicted edge location in the next frame. The advantage of doing this is that feature point mismatch in some frame as well as noise effects will be effectively suppressed. The main reasons which motivated us in using Kalman predictor to solve this problem is that, it is interesting to think of this problem as a learning problem. This is also been exploited in the approach which treat similar problems as learning problems. Now, for every frame, a tracking list for every segment is maintained. If a corresponding segment match is not found we keep this old segment to be tracked in future frames by lowering its weight value. If this weight becomes less than some threshold we then delete this segment from the tracked edge list. This type of activity helps us to track segments even if segments are missing due to noise or due to object's occlusion. In the last stage measurement update is used to refine this already predicted parameters so that new arriving edge segments (from the next frames) can be tracked more effectively. Let,

$s_i^{(t)}$  = State vector, for the edge segment  $i$  from frame  $t$

- $w_i^{(t)}$  = process noise; white, Gaussian, mean=0, covariance matrix  $Q$ ;
- $v_i^{(t)}$  = measurement noise; white, Gaussian, mean=0, covariance matrix  $R$ , uncorrelated;
- $S_i^{(t)}$  = covariance of the innovation, residual;
- $K_i^{(t)}$  = Kalman gain, gain matrix;

$P_i^{(t)}$  =covariance of predication error;  
 $z_i^{(t)}$  =measurement of system state;  
 $loc_i^{(t)}$  = $i^{th}$  edge segment location.

The predication phase is modeled as:

$$s_i^{(t)} = As_i^{(t-1)} + w_i^{(t-1)}$$

$$z_i^{(t-1)} = loc_i^{(t-1)} + v_i^{(t-1)} \text{-----Equation (5)}$$

The model update equations are:

$$S_i^{(t-1)} = P_i^{(t-1)} + R$$

$$K_i^{(t-1)} = AP_i^{(t-1)}S_i^{(t-1)-1}$$

$$P_i^{(t)} = AP_i^{(t-1)}A^T + Q - AP_i^{(t-1)}P_i^{(t-1)}A^T$$

$$S_i^{(t)} = As_i^{(t-1)} + K_i^{(t-1)}(z_i^{(t)} - As_i^{(t-1)}) \text{-----Equation (6)}$$

The estimated motion information of Kalman predictor helps in limiting search space which increase accuracy as well as speed.

**Edge matching by prediction and curvature**

Given a sequence of  $N_T$  Image Frame  $I^t, t=1,2, \dots, N_T$ , here  $t$  is a time index, we estimated the motion of all curvature points from one edge List  $EL^{(t-1)}$  to another list  $EL^{(t)}$ . Here  $EL^{(t)}$  is the list of all Edge segments extracted from Image Frame  $I^t$ . Now, among this  $n$  candidate curvature points we select one curvature point  $k_{p,q}^c$  that is closest to the predicated corresponding curvature locations.

$$k_{p,q}^c = \underset{\text{location}(k_{(i,j)}^1)}{\text{argmin}} | \text{location}(k_{(i,j)}^c) - \text{location}(k_{(p,q)}^1) | \text{-----Equation (7)}$$

**Algorithm 1: Moving edge tracking for first two frames**

```

Compute  $EL^{(t)}$  from  $I^{(t)}$ 
Compute  $EL^{(t-1)}$  from  $I^{(t-1)}$ 
 $w_k = NUZZ$ 
for  $p := 1$  to  $N^{(t-1)}$  do
  for  $q := 1$  to  $N^{(t-1),p}$  do
    if  $N^{(t-1),p} > 3$  then
      if  $\exists k_{ij} \in E L^{(t)}$ : inside predicated MASK & |
 $k_{ij} - k_{p,q}^1 | \leq T_k$  then
         $k_{ij}^c = k_{(i,j)}$ 
      end if
    end for
  if count ( $k_{ij}^c$ ) > 1 then
     $k_{p,q}^c = \underset{\text{location}(k_{(i,j)}^c)}{\text{argmin}} | \text{location}(k_{(i,j)}^c) - \text{location}(k_{(p,q)}^1) |$ 
  end if
end for
if  $\forall q: k_{p,q}^c : \text{count}(\Phi_{(i)}) > 3$  then
if  $N_w^{(t)} = 0$  or  $\underset{\text{distance}(\Phi_{(i)}, C_k)}{\text{argmin}} > T_{(distance)}$ 
then

```

```

Add  $\Phi_{(i)}$  to new cluster  $w_k$  where:  $\underset{\text{distance}(\Phi_{(i)}, C_k)}{\text{argmin}}$ 
end if
else
  Reduce Weight ( $\Phi_{(p)}^1$ ) and Add  $\Phi_{(p)}^1$  to
 $w_k$  where  $\text{Weight}(\Phi_{(p)}^1) \leq T_{edge}$ 
end if
end for

```

**Edge Segment Clustering**

Cluster segments based on edge segment connectivity while giving emphasis on relative edge distance and motion direction. Initially, we group clusters using a distance based iterative k- means clustering algorithm for the first two frames. To cluster segments each edge segment is enclosed within a rectangular boundary and the midpoint of this boundary is used as the edge representative in the computation of the K-centroids for the K-clusters. The algorithm aims to minimize the following objective function in Equation (8)

$$J = \sum_{k=1}^K \sum_{i=1}^{N^{(t)}} |P_i - C_k|^2 \text{-----Equation (7)}$$

Where  $|p_i - C_k|^2$  is the distance between the midpoint  $p_i$  of the  $i^{th}$  rectangular edge segment boundary to the  $k^{th}$  cluster center ( $C_k$ ) and  $N^{(t)}$  is the total number of moving egde segments present in frame  $t$ .

**Algorithm 2 : Moving object cluster tracking**

```

if  $N_w^{(t)} = 0$  then
  Call Algorithm 1.
else
  Compute  $EL^{(t)}$  from  $I^{(t)}$ 
  for  $k := 1$  to  $N_w^{(t)}$  do
     $\forall i \Phi_{k,i}$  predict cluster motion from edge
    ( $\Phi_{k,i}$ ) motion parameters
    for  $i := 1$  to  $N_w^{(t)}$  do
      Register curvatures from  $\Phi_{k,i}$  and  $\Phi_i$  and use
      KMASK as search window
      if  $\Phi_{k,i}$  matched with  $\Phi_i$  then
        Remove  $\Phi_{k,i}$  from  $w_k$  & Add  $\Phi_i$  to  $w_k$ 
      else
        if  $\text{Weight}(\Phi_{k,i}) > T_{edge}$  then
          Reduce Weight ( $\Phi_{k,i}$ )
        else
          Remove  $\Phi_{k,i}$ 
        end if
      if  $N_{w,k}^{(t)} = 0$  then
        Delete cluster  $w_k$ 
      end if
    end if
  end for
end if

```

```

if  $\exists w'_k$ : Boundary( $w_k$ ) Boundary( $w_{k'}$ )  $\neq \Phi$  &
motion( $w_k$ ) $\neq$ motion( $w_{k'}$ ) then
Merge( $w_k, w_{k'}$ )
end if
Update  $C_k$ 
end for
end for
end if

```

## VI. EXPERIMENTAL ANALYSIS & RESULTS

Different size of dataset is used for experimentation. The proposed method has been tested with several video sequences including both indoor and outdoor environment and several other sequences from PETS database. All videos were of size 320x240 with changes in illumination and the system was able to track almost all of the multiple moving objects.

The first feature is declaration of the area for each moving object which is determined based on the result after morphological filtering process. This in turn will be based on the coordination of maximum and minimum on X-axis and Y-axis for the moving object. It will produce the bounding box which is generated by drawing the line from each of the corner obtained from the maximum and minimum coordination of rows and columns from the frame. Then a limitation of area of the bounding box is made. If the area Threshold is less than 800 pixels it will not be considered as a moving object. The second feature-centroid which is the center of the moving object with reference to the bounding box is considered for tracking. Center of moving object is characterized by the rows (x-axis) and columns (y-axis). The third feature-The difference of the average RGB pixel values between current image frame and the previous image frame is the key-point in identifying and differentiating each of the moving objects inside the image frame. This is more accurate due to average value of 5x5 RGB pixels which yields better results than taking one pixel value at the center of the bounding box. Hence, each moving object can be differentiated and this will make the detection & tracking and of the moving object in different environment to have a better accuracy which can be calculated by using equation (8).

$$\frac{\text{Correct Number of Bounding box assigned to moving objects}}{\text{Total number of moving objects}} \times 100 \text{ ----Equation (8)}$$

The proposed algorithm for detection and tracking of multiple moving objects in real time for both indoor and outdoor environment was extensively tested to operate in complex, real world, non-plain and changing background was found to possess remarkable accuracy and precision of 99%.

## VII. CONCLUSION & FUTURE WORK

A new algorithm allowing a robust and accurate detection of multiple moving objects for a small cost in memory consumption and computational complexity has been proposed. The strength of the proposed approach lies

in the ability to track segments in successive frames. As we know the shape and size of an edge segment changes slowly from frame to frame. Also in some especial case some segments might hide for one or more frames then again shows itself. The proposed system successfully handles edge instability problem by using weighted edge segments. Introducing a partial edge segment-based matching approach in object tracking area helps building inter-frame association within edge segments more accurately. The proposed prediction model and inter-frame association accumulates the shape changing knowledge of edges inside every cluster. Hence, it helps to provide accurate cluster location in successive frames. Finally, in the proposed method while clustering edge segments, clusters were merged when the cluster boundary overlapped having similar motion direction and magnitude. The

The clustering algorithm which has been used in the proposed system is especially suitable for very large databases; the algorithm can be used to produce a large number of clusters; and the algorithm has very low time complexity and space complexity.

The proposed method has some limitations. The application will not detect if objects move suddenly at a high speed or change the motion direction instantly with very high. The video frame rate considered is 24 frames/second. It cannot handle video other than .avi format.

## FUTURE ENHANCEMENT

The proposed method may fail to build correspondence between segments. One way to solve the problem is to choose the size of the search window based on the confidence of prediction. Since tracking algorithm cannot handle full occlusion of moving objects in future work a model to refine the clusters can be used as well as to extract important movements from the scene and counting the number of objects in the scene can be incorporated. Indexing for the objects can be incorporated in future work.

## REFERENCES

- [1] Antoine Manzanera, Julien C. Richefeu, " A survey on motion detection algorithm based on background estimation", Pattern Recognition Letters, Vol. 25,2007,Elsevier,pp. 234-242.
- [2] Chadia Khraief , Sami Bourouis , Kamel Hamrouni, " Unsupervised video objects detection and tracking using region based level-set", IEEE,2011, pp. 631-635.
- [3] Antoine Manzanera, Julien C. Richefeu, " A new motion detection algorithm based on  $\Sigma$ - $\Delta$  background estimation", Pattern Recognition Letters, Vol. 28,2011,Elsevier,pp. 320-328.
- [4] Murat Erisoglu, Nazif Calis, Sadullah Sakallioğlu, " A new algorithm for initial cluster centers in k-means algorithm", Pattern Recognition Letters, Vol. 32,2011,Elsevier,pp. 1701-1705
- [5] P. Assheton, A. Hunter, " A shape-based voting algorithm for pedestrian detection and tracking", In: Pattern Recognition, , Vol. 44,2011, Elsevier ,pp. 1106-1120
- [6] Konrad Schindler, Andreas Ess, Bastian Leibe, Luc Van Gool, " Automatic detection and tracking of pedestrians from a moving stereo rig", ISPRS Journal of Photogrammetry and Remote Sensing , Vol. 65,2010, Elsevier .pp 523-537

- [7] Mohamed A. El-Sayed, Tarek Abd-El Hafeez, "New Edge Detection Technique based on the Shannon Entropy in Gray Level Images", IJSCE, Vol.3 No.6, June 2011, pp.2224-2232
- [8] Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas, "Tracking-Learning-Detection", IEEE Transactions On Pattern Analysis And Machine Intelligence, Vol. 6, No. 1, January 2010, pp.1-14
- [9] D.P. Huttenlocher, G.A. Klanderman, W.A. Rucklidge, "Comparing Images Using the Hausdorff Distance," IEEE Transactions on Pattern Analysis and Machine Intelligence ,vol. 15, no. 9, pp. 850-863, 2003.
- [10] D.G. Lowe, "Object recognition from local scale-invariant features," IEEE Conference on Computer Vision and Pattern Recognition, pp. 1150-1157, 1999.
- [11] D. Crandall, P. Felzenszwalb and D. Huttenlocher, "Spatial priors for part-based recognition using statistical models," IEEE Conference on Computer Vision and Pattern Recognition, pp. 10-17, 2005.
- [12] David Marr, *Computer Vision*. W.H. Freeman and Company, 2<sup>nd</sup> Edition, 2002, ISBN 0-7167-1284-9.
- [13] Azriel Rosenfeld and Avinash Kak, *Digital Picture Processing*. Academic Press, 4<sup>th</sup> Edition, 2008, ISBN 0-12-597301-2.