Real Time Object Detection and Tracking: Histogram Matching and Kalman Filter Approach

Madhur Mehta, Chandni Goyal, M.C. Srivastava, R.C. Jain.
Department of Electronics and Communication, Jaypee Institute of Information Technology
A-10, Sector-62, Noida-201301, Uttar Pradesh, India
{Madhur.eng,chandnigoyal}@gmail.com,{mc.srivastava,rc.jain}@jiit.ac.in

***Abstract*—In this paper we present an approach to develop a real-time object tracking system using a static camera to grab the video frames and track an object. The work presents the concepts of histogram matching and absolute frame subtraction to implement a robust automated object tracking system. Once the object is detected it is tracked using discrete Kalman filter technique. The histogram matching algorithm proposed here helps to identify when the object enters the viewing range of the camera and the absolute frame subtraction gives better results even with low quality videos. Such a tracking system can be used in surveillance applications and proves to be cost effective.**

Keywords- object tracking; absolute frame subtraction; histogram matching.

I. INTRODUCTION

The video surveillance systems have been the subject of intensive research due to their great importance for security reasons. Several authors in the past tried to develop a robust real time tracking system, for static camera video sequence [1-3]. Most of them work on the principle of combining foreground detection with object tracking. Number of algorithms has been proposed in this field which includes segmentation of an image [4], N-cut technique [5], Background Subtraction Method [6] followed by tracking.

Video Surveillance is described as the task of analyzing videos to detect certain unusual activities. It can be classified in two broad categories: (a) semi-autonomous and (b) fully-autonomous. Semi-autonomous video surveillance involves some form of video processing with significant human intervention. Typical examples are systems that perform simple motion detection or tracking of a suspected person, either by drawing a rectangle around an object to be detected [7] or marking some part of it [8]. By a fully-autonomous system, we mean a system which has its input as a video sequence where surveillance is performed with no human intervention and the system does both the tasks of motion detection and tracking.

Classical image segmentation using graphical tools use either texture (color) information, e.g. Magic Wand, or edge (contrast) information, e.g. Intelligent Scissors which are normally user-interactive [9-11] i.e. the operator or user creates a rectangle or marks certain part of the required object that is to be detected and tracked. The drawback of these techniques is that these are operator driven and is very slow.

To overcome the above two problems background subtraction method (BSM) has been considered widely [12]. It makes the processing fast as well as automated. Although this method is fast and is automated but it comes at the expense of addition of excessive noise due to either change in the position of objects in the reference frame, example movement of the leaves or change in position of dustbins or wall paintings etc. or sudden change in the luminosity of the light. The above problem is solved by applying a certain threshold which removes smaller particles of noise and certain morphological operations like erosion to reduce the non connected parts smaller in size. Even though the noise is removed we also lose on the essential information i.e. certain part of the object gets eroded while applying erosion.

Thus we present our approach of absolute Histogram subtraction of consecutive frames to detect the object (if appeared) in the frames to avoid complete processing of each and every frame. If object appears, we use absolute image subtraction to extract the object followed by the Kalman tracking of the object.

The outline of the paper is as follows:

Section A describes the algorithmic approach to extract the object once it appears in the frame. Section B describes the tracking of the object using Kalman Filter. The framework of the proposed approach is shown in Fig. 1.

II. SECTION A

A. Object Detection Using Histogram Matching Technique

Histogram is the representation of the frequency of pixels lying in the range of certain color. Our Histogram function gives the frequency of the pixels within a range of 20 as shown in Table 1. The above obtained histogram is used for the analysis of the appearance of the object by the absolute subtraction of the histograms of consecutive frames. Adding the absolute difference in the frequency of the pixels gives a value (Hist value) which, if is greater than certain threshold (obtained by testing) indicates that the object has appeared in the frame. This threshold helps us in overcoming the problem

TABLE I. COLOR RANGE VS. PIXEL FREQUENCY

Color range	Number of pixels
0-20	300
20-40	200

978-1-4244-5586-7/10/$26.00 ©2010 IEEE

Authorized licensed use limited to: Universidade Federal dos Pampas (UNIPAMPA). Downloaded on November 23,2023 at 00:32:47 UTC from IEEE Xplore. Restrictions apply.

40-60	0
60-80	10
80-100……	25
>240	321

of change in light intensity and any small insignificant changes in the background. We used this technique to save time and memory by avoiding the excessive processing in those frames that restrain a significant object.

B. Object Extraction

If the object is detected, we go for further processing i.e. extraction followed by tracking. Thus for extraction we take the absolute difference (subtracting each element in array Y from the corresponding element in array X and return the absolute difference in the corresponding element of the output array Z) of the current image from the ref. image. This gives a better output than background subtraction method. The resultant image is converted into a binary image using global image threshold by Otsu's method [13].

Otsu's method is used to compute a global threshold (level) that can be used to convert gray level image to a binary image. This level is a normalized intensity value that lies in the range [0, 1]. It chooses the threshold to minimize the intraclass variance of the black and white pixels.

We then perform smoothening to connect the small non connected parts. This effectively removes the small particles by connecting them into one and also helps in connecting the disconnected components of the object if any.

The above process of smoothening helps in connectivity. The connectivity can be done in two ways a) 4-connectivity b) 8 connectivity as shown in Fig. 2

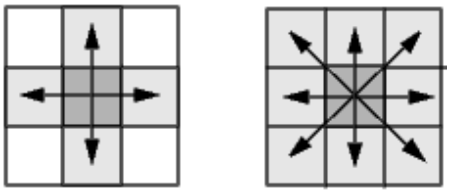

Figure 1.

Thus after connectivity the pixels labeled 0 are the background. The pixels labeled 1 make up one component; the pixels labeled 2 make up a second component, and so on.

The components with max label assigned have max area

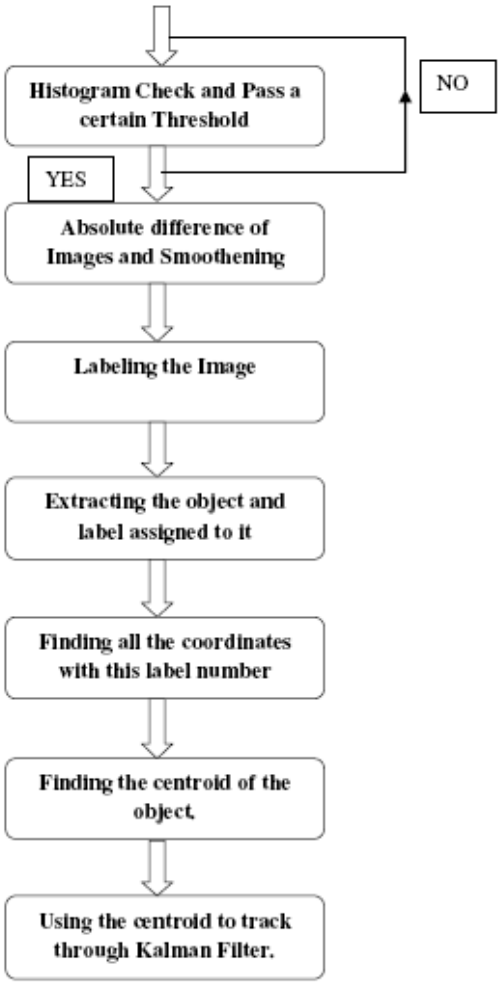

Figure 2. Framework of the proposed approach

which corresponds to the object. This component having max area has some id or label assigned to it. Through that label we could extract the pixels of the object very easily and thus can calculate the centroid of the extracted object.

Similarly, the system can be extended for extracting more than one object as the component with the next maximum area will represent our second object and so on. If two objects are very close to each other they are treated as one and tracked together.

The centroid calculated is used in tracking to calculate the further states by using Discrete Kalman Filter [14].

III. SECTION B

To track the object we use Discrete Kalman filter technique which as discussed by G. Welch and G. Bishop [14] is a recursive method to predict the object using present and past state. Kalman filter is a set of mathematical equations which first predict the state of object before the object makes its next move and then corrects this prediction using Kalman gain. In Discrete Kalman filter we use two sets of equations to determine the object position. One set of equations are

Authorized licensed use limited to: Universidade Federal dos Pampas (UNIPAMPA). Downloaded on November 23,2023 at 00:32:47 UTC from IEEE Xplore. Restrictions apply.

called *prediction equation* or *time update equations* which are used to predict the object on the basis of past state. Another set of equations is called *correction equations* or *measurement update equations* which are used to correct our prediction, this we get from the previous set of equations and give a better estimate to track the object.

In Discrete Kalman filter first we define the process function and measurement function. The process function is defined as

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1}$$

Where A (*n*x*n)* is a state transition matrix, used to relate the current state of process to the previous state. B (*n*x*l*) is a matrix, used to relate the control input to the current state, where $u_{k\text{-}1}$ is a control input.

The measurement function is defined as

$$z_k = Hx_k + v_k$$

Where z_k is our actual measurement, H (*m*x*n*) is a matrix which relates the state *x* at time step *k* to the measurement function.

In the above equations, *w* and *v* are the process and measurement noise respectively. These are white noises, independent of each other with normal probability distribution as $p(w) \sim N(0, Q)$ and $p(v) \sim N(0, R)$, Q is called the *process noise covariance* while R is called the *measurement noise covariance.*

As we cannot measure the values of process noise w and measurement noise v at each time step, we first estimate our state function and measurement function without them as

$$\hat{\mathrm{x}}_{\mathrm{k}}^{-} = A\hat{\mathrm{x}}_{\mathrm{k\text{-}1}}^{-} + Bu_{k-1}$$

$$\hat{z}_k = H\hat{x}_k^-$$

In the above equation $\hat{x}_k^-$ is *a priory state estimate* at step *k* which gives us the knowledge of the process prior to step *k* and $\hat{x}_k^-$ is *a posteriori state estimate* at step *k* which gives us the knowledge of the process given measurement z_k. $\hat{z}_k$ is our predicted measurement.

First we calculate our prediction for the moving object using Discrete Kalman filter time update equations or prediction equations

$$\hat{\mathrm{x}}_{\mathrm{k}}^{-} = A\hat{\mathrm{x}}_{\mathrm{k\text{-}1}}^{-} + Bu_{k-1} \quad (1)$$

$$P_k^- = AP_{k-1}A^T + Q \quad (2)$$

The Equation (1) projects the state ahead and (2) projects the state covariance ahead. They are responsible for the calculation of priory estimates for the next time step before the time step actually occurs.

The *a priory estimate error* and *a posteriori estimate error* denoted by e_k^- and e_k respectively are defined as

$$e_k^- \equiv x_k - \hat{x}_k^- \quad (3)$$

$$e_k \equiv x_k - \hat{x}_k \quad (4)$$

Similarly in (2) *a priory error covariance* is $P_k^- = E[e_k^- e_k^{-T}]$ and a *posteriori error covariance* is $P_k = E[e_k e_k^T]$.

From above equations it's clear that to calculate *a priory* estimates for state k we must know the $(k-1)^{\text{th}}$ stage. For state at time step 2 and ahead the equations have no problem but to predict the state at time step $k=1$ we have no knowledge of process at step $k=0$ and to remove this difficulty for first prediction we assume that state at time step $k=0$ is a null matrix. Now we predict our next state using (1). Similarly we predict the state covariance assuming that at step $k=0$, state covariance has zero matrix.

The Discrete Kalman filter measurement equations are used to calculate and incorporate a new measurement in previously obtained *a priory* estimate to obtain an improved *a posteriori* estimate.

$$G_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \quad (5)$$

$$\hat{\mathrm{x}}_{\mathrm{k}}^{-} = \hat{\mathrm{x}}_{\mathrm{k}}^{-} + G_k(Z_K - H\hat{\mathrm{x}}_{\mathrm{k}}^{-}) \quad (6)$$

$$P_k = (I - G_k H)P_k^- \quad (7)$$

Equation (5) computes the Kalman Gain; (6) updates estimate with measurement z_k and (7) updates the error covariance.

The difference $(z_k - H\hat{x}_k^-)$ is the *residual,* used to find the disparity between our actual measurement (z_k) and the estimated measurement ($H\hat{x}_k^-$).

Here we use values of centroid location which we have calculated in the first part of algorithm as our actual measurement. After a gap of 10-15 frames we get a new value of centroid location. We calculate a new prediction using above equations.

From the Equation (5) we can see a relationship between Kalman gain value and residual. As the value of $R \to 0$, the Kalman gain gives more weightage to actual measurement while less weightage to predicted measurement. Similarly as the value of $P_k^- \to 0$ the Kalman gain gives less weightage to the actual measurement and more to the predicted measurement.

We can also see that as the number of times the steps increases, in other words as the object moves more distance without deviating from its path we get a better prediction which continuously increases. This is because we calculate *a priori state* estimate in which we incorporate the residual of previous time step to get a better *a posteriori* state estimate, using the difference between the actual measurement and predicted measurement of the previous time step.

Authorized licensed use limited to: Universidade Federal dos Pampas (UNIPAMPA). Downloaded on November 23,2023 at 00:32:47 UTC from IEEE Xplore. Restrictions apply.

In the whole process we use different matrices which change at each time step but for our convenience we assume that each matrix is constant during the whole tracking process.

Now the important question is how to assign these matrices. For an object whose position continuously changes at each time step but velocity of the object remains constant during the whole process, we can define state matrix x_k as

$$x_k = \begin{bmatrix} x \\ y \end{bmatrix} \tag{8}$$

Here x and y are the x and y coordinates of the object's position. State transition matrix A , can be defined as

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{9}$$

For this process we define the measurement matrix H as

$$H = \begin{bmatrix} H_x & 0 \\ 0 & H_y \end{bmatrix} \tag{10}$$

But in the real time system, tracking a moving object is different because the velocity of the object changes at each time step with the position of the object so we define new matrices for this purpose as x_k.

$$x_k = \begin{bmatrix} x \\ y \\ \frac{dx}{dt} \\ \frac{dy}{dt} \end{bmatrix} \tag{11}$$

In above matrix x and y are the coordinates of the object's position while dx/dt and dy/dt are velocity components in x and y directions.

State transition matrix A can be defined as

$$A = \begin{bmatrix} 1 & 0 & dt & 0 \\ 0 & 1 & 0 & dt \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{12}$$

Here dt is the time difference between two time frames. For this process, the measurement matrix H can be defined as

$$H = \begin{bmatrix} H_x & 0 & 1 & 0 \\ 0 & H_y & 0 & 1 \end{bmatrix} \tag{13}$$

After using the Kalman filter the object was successfully tracked as shown in the Fig. 8(a)-8(e).

IV. RESULTS

Fig. 3(a) - 3(e) shows 5 selected frames from a real time video. Fig. 4(a) - 4(e) shows the outputs when we use normal background subtraction while the next set of images from Fig. 5(a)-5(e) show the use of Absolute Background Subtraction method. The image of the 6th and 7th set i.e. Fig. 6(a) - 7(e) shows the object extracted using the normal background subtraction and absolute subtraction respectively. It's clearly visible from the figures of 4th set that the object is indistinguishable from the background if we use normal background subtraction method while applying absolute background subtraction method to our tracking algorithm gives us an accurate extraction which can be seen in Fig. 7(a)-7(e). The 8th set of Fig. shows the final output. The centroid of the object is marked by red Asteric sign. The red rectangle is the estimated position of the object obtained using Kalman Filter while the green rectangle shows the actual position of the object.

V. CONCLUSION

After applying the above proposed algorithm and testing it with a practical video we come to the conclusion that to save computational time and processing steps we could use the absolute frame subtraction instead of normal frame subtraction and histogram matching technique, to detect the appearance of the object. This method of extracting an object reduces the computation time by a factor of 10 w.r.t. using other methods of segmentation like N-cut [5]. Also the processing time is saved since our algorithm starts working only when the object is detected and comes in the vicinity of the camera. From fig 8(a)-8(e) its clearly visible that our approximation precisely matches the actual position of the object, giving us the authentication of the algorithm used.

Authorized licensed use limited to: Universidade Federal dos Pampas (UNIPAMPA). Downloaded on November 23,2023 at 00:32:47 UTC from IEEE Xplore. Restrictions apply.

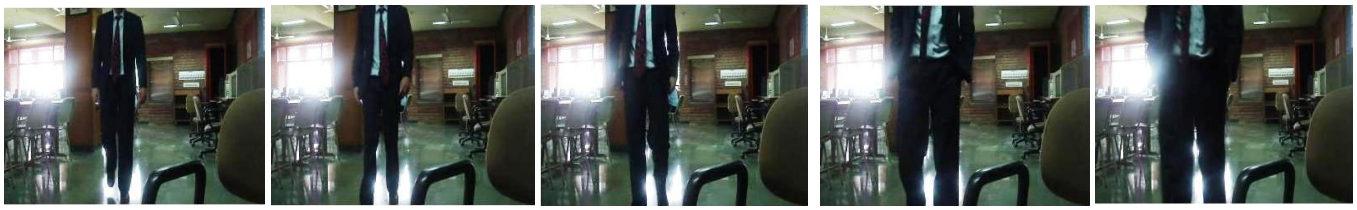

Figure 3. (a) (b) (c) (d) (e)

Figure 4. (a) (b) (c) (d) (e)

Figure 5. (a) (b) (c) (d) (e)

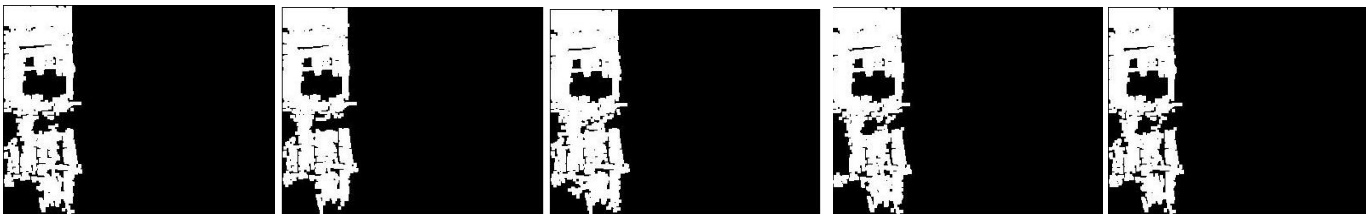

Figure 6. (a) (b) (c) (d) (e)

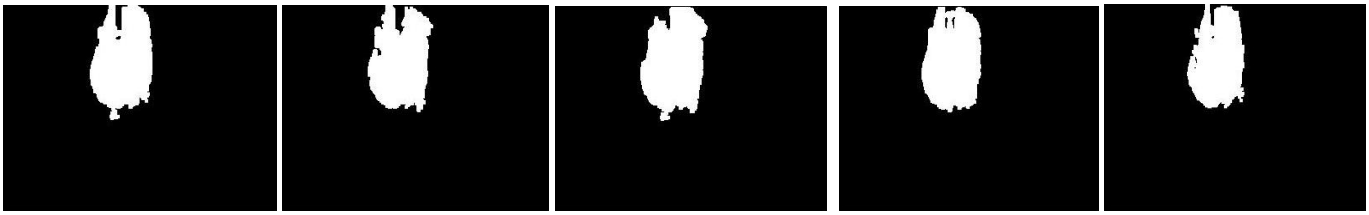

Figure 7. (a) (b) (c) (d) (e)

Figure 8. (a) (b) (c) (d) (e)

Authorized licensed use limited to: Universidade Federal dos Pampas (UNIPAMPA). Downloaded on November 23,2023 at 00:32:47 UTC from IEEE Xplore. Restrictions apply.

REFERENCES

[1] TP Chen and Haussecker et al., “Computer vision workload analysis: case study of video surveillance systems,” Intel Technology Journal, vol. 9, no. 02, 2005.

[2] F Porikli and O Tuzel, “Human body tracking by adaptive background models and mean-shift analysis,” IEEE Int. W. on Performance Evaluation of Tracking and Surveillance, 2003.

[5] Jianbo Shi and Jitendra Malik, "Normalized Cuts and Image Segmentation," IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 2000

[6] A. Elgammal, D. Harwood, and L. Davis. Non-parametric model for background subtraction. In ECCV, pages 751–757, 2000

[7] Carsten Rother, Vladimir Kolmogorov, Andrew Blake. “GrabCut” -Interactive Foreground Extraction using Iterated Graph Cuts, In proceedings SIGGRAPH 2004.

[8] Kar-Han Tan, Narendra Ahuja, “Selecting Objects With Freehand Sketches,” In Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2001.

[9] Eric N. Mortensen1, William A. Barrett2, “Intelligent Scissors for Image Composition.” Proceedings of the 22nd annual conference on Computer graphics and interactive techniques, pages: 191 - 198 ,1995

[3] P.F Gabriel, J.G Verly, J.H Piater, and A Genon, “The state of the art in multiple object tracking under occlusion in video sequences,” Advanced Concepts for Intelligent Vision Systems,pp. 166–173, 2003.

[4] Jaime Gallego, Montse Pardas, Jose-Luis Landabaso, “Segmentation and tracking of static and moving objects in video surveillance scenarios,” In Proc. IEEE International Conference on Image Processing (ICIP), San Diego (California, USA), October 2008.

[10] E. N. Mortensen and W. A. Barrett, "Interactive Segmentation with Intelligent Scissors," Graphical Models and Image Processing, Vol. 60, No. 5, pp. 349-384, Sept. 1998.

[11] E. N. Mortensen and W. A. Barrett, "Toboggan-Based Intelligent Scissors with a Four Parameter Edge Model," in Proc. IEEE: Computer Vision and Pattern Recognition (CVPR'99), Vol. II, pp. 452-458, 1999.

[12] A. Monnet, A. Mittal, N. Paragios, and V. Ramesh, “Background modeling and subtraction of dynamic scenes”. In CVPR, 2003.

[13] Otsu, N., "A Threshold Selection Method from Gray-Level Histograms," IEEE Transactions on Systems, Man, and Cybernetics, Vol. 9, No. 1, 1979, pp. 62-66.

[14] Greg Welch and Gary Bishop, “An Introduction to the Kalman Filter” presented at ACM SIGGRAPH 2001.

Authorized licensed use limited to: Universidade Federal dos Pampas (UNIPAMPA). Downloaded on November 23,2023 at 00:32:47 UTC from IEEE Xplore. Restrictions apply.