ISSN: 2321-7782 (Online)
Volume 2, Issue 5, May 2014
International Journal of Advance Research in Computer Science and Management Studies
Research Paper
Available online at: www.ijarcsms.com

Multiple Object Tracking and Segmentation in Video Sequences

K. Meenatchi[1]
PG Scholar, Dept. of ECE
CMS College of Engineering
Namakkal, India

P. Subhashini[2]
Assistant Professor, Dept. of ECE
CMS College of Engineering
Namakkal, India

Abstract: Object Tracking is an important and challenging problem in video-based Intelligent Security Systems. A robust and real-time method for tracking objects is presented in this paper. The proposed algorithm includes two stages: object tracking, object Segmentation. Object detection is a key step. The concept of tracking object is built upon the object-segmentation method. Motion segmentation is a key step in many tracking algorithms as it forms the basis of object detection. Improving segmentation results as well as being able to extract additional information such as frame difference, Gaussian of mixture model, background subtraction allows for improved object detection and thus tracking. According to the segmented object shape, a predict method based on Kalman filter is proposed. Kalman filter model is used to tracking and predicting the trace of an object. Image enhancement is the process of adjusting tracked frame images so that the results are more suitable for display. Edge detection technique is used for finding discontinuities in gray level images of tracked frames. Finally segmented frames are converted into video sequence. The model can be used in the real time environment, and can track multiple object targets in a big area. The proposed method has been tested on a number of monocular video sequences and the experimental results show that the algorithm is robust and can meet the real-time requirement.

Keywords: Gaussian of mixture model, background subtraction, Kalman filter

I. INTRODUCTION

This Object detection and tracking are important and challenging tasks in many computer vision applications such as surveillance, vehicle navigation and autonomous robot navigation. Object detection involves locating objects in the frame of a video sequence. Every tracking method requires an object detection mechanism either in every frame or when the object first appears in the video. Object tracking is the process of locating an object or multiple objects over time using a camera. The high powered computers, the availability of high quality and inexpensive video cameras and the increasing need for automated video analysis has generated a great deal of interest in object tracking algorithms. There are three key steps in video analysis, detection interesting moving objects, tracking of such objects from each and every frame to frame, and analysis of object tracks to recognize their behavior. Therefore, the use of object tracking is pertinent in the tasks of, motion based recognition.

Automatic detection, tracking, and counting of a variable number of objects are crucial tasks for a wide range of home, business, and industrial applications such as security, surveillance, management of access points, urban planning, traffic control, etc. However, these applications were not still playing an important part in consumer electronics. The main reason is that they need strong requirements to achieve satisfactory working conditions, specialized and expensive hardware, complex installations and setup procedures, and supervision of qualified workers. Some works have focused on developing automatic detection and tracking algorithms that minimizes the necessity of supervision. They typically use a moving object function that evaluates each hypothetical object configuration with the set of available detections without to explicitly compute their data association. Thus, a considerable saving in computational cost is achieved. In addition, the likelihood function has been designed to account for noisy, false and missing detections.

In moving object detection various background subtraction techniques available in the literature were simulated. Background subtraction involves the absolute difference between the current image and the reference updated background over a period of time. A good background subtraction should be able to overcome the problem of varying illumination condition, background clutter, shadows, camouflage, bootstrapping and at the same time motion segmentation of foreground object should be done at the real time. It's hard to get all these problems solved in one background subtraction technique. So the idea was to simulate and evaluate their performance on various video data taken in complex situations.

Object tracking is a very challenging task in the presence of variability Illumination condition, background motion, complex object shape, partial and full object occlusions. Here in this project, modification is done to overcome the problem of illumination variation and background clutter such as fake motion due to the leaves of the trees, water flowing, or flag waving in the wind. Sometimes object tracking involves tracking of a single interested object and that is done using normalized correlation coefficient and updating the template.

II. Object Tracking and Segmentation

1. Object Detection

Object detection method is requested to automatically segment every object so that there can be a unique tracking associated with the object. It includes five steps: background estimation, background updating, background subtraction, moving cast shadow elimination, and object detection. In this proposal solve several problems as follows:

A. Extract the background image automatically from a sequence of images and update the background

B. Select an adaptive filter to eliminate abnormal moving object in the binary background subtraction image so that the system can be more robust.

C. Suppress moving cast shadows to avoid the overlapping between adjacent objects.

D. Detect object from the binary background subtraction image.

This paper discusses a target tracking method based on Kalman filter. The method makes a full use of the prediction function of Kalman filter to predict the region where the next frame possibly appears, then carries on the correlation match operation in the smaller forecast region, finds the best correlation match spot and makes the target tracking more initiatively

2. Background Subtraction

The background subtraction is the most popular and common approach for motion detection. The idea is to subtract the current image from a reference background image, which is updated during a period of time. It works well only in the presence of stationary cameras. The subtraction leaves only non-stationary or new objects, which include entire silhouette region of an object. This approach is simple and computationally affordable for real-time systems, but is extremely sensitive to dynamic scene changes from lightning and extraneous event etc. Therefore it is highly dependent on a good background maintenance model. Fig 4.2 shows the original video frame. The background subtraction of the original video frame is shown in fig 4.3.

Fig. 1 Original Video Frames

Fig. 2 Output after Background Subtraction

Background subtraction detects moving regions in an image by taking the difference between the current image and the reference background image captured from a static background during a period of time. The subtraction leaves only non-stationary or new objects, which include entire silhouette region of an object. Detection method is requested to automatically segment every object so that there can be a unique tracking associated with the

In simple background subtraction a absolute difference is taken between every current image $I_t(x,y)$ and the reference background image B(x,y) to find out the motion detection mask D(x,y). The reference background image is generally the first frame of a video, without containing foreground object.

$$D(x,y) = \begin{cases} 1, & if|I_t(x,y) - B(x,y)| \geq \tau \\ 0, & otherwise \end{cases}$$

where r is a threshold, which decides whether the pixel is foreground or background. If the absolute difference is greater than or equal to r, the pixel is classified as foreground; otherwise the pixel is classified as background.

3. Object Tracking

After the object detection is achieved, the problem of establishing a correspondence between object masks in consecutive frames should arise. Obtaining the correct track information is crucial for subsequent actions, such as object identification and activity recognition. For this situation, Kalman filtering technique is used.The Kalman filter is a recursive two-stage filter. At each iteration, it performs a *predict* step and an *update* step.

The predict step predicts the current location of the moving object based on previous observations. For instance, if an object is moving with constant acceleration, we can predict its current location, $\hat{\mathbf{x}}_t$, based on its previous location, $\hat{\mathbf{x}}_{t-1}$, using the equations of motion.

The update step takes the measurement of the object's current location (if available), $\mathbf{z}_t$, and combines this with the predicted current location, $\hat{\mathbf{x}}_t$, to obtain an *a posteriori* estimated current location of the object, $\mathbf{x}_t$.

The equations that govern the Kalman filter are given below

1. ***Predict stage:***

   A. Predicted (a priori) state: $\hat{\mathbf{x}}_{t|t-1} = \mathbf{F}_t\hat{\mathbf{x}}_{t-1|t-1} + \mathbf{B}_t\mathbf{u}_t$

   B. Predicted (a priori) estimate covariance: $\mathbf{P}_{t|t-1} = \mathbf{F}_t\mathbf{P}_{t-1|t-1}\mathbf{F}_t^T + \mathbf{Q}_t$

2. ***Update stage:***

   A. Innovation or measurement residual: $\tilde{\mathbf{y}}_t = \mathbf{z}_t - \mathbf{H}_t\hat{\mathbf{x}}_{t|t-1}$

   B. Innovation (or residual) covariance: $\mathbf{S}_t = \mathbf{H}_t\mathbf{P}_{t|t-1}\mathbf{H}_t^T + \mathbf{R}_t$

   C. Optimal Kalman gain: $\mathbf{K}_t = \mathbf{P}_{t|t-1}\mathbf{H}_t^T\mathbf{S}_t^{-1}$

D. Updated (a posteriori) state estimate: $\hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t\tilde{\mathbf{y}}_t$

E. Updated (a posteriori) estimate covariance: $\mathbf{P}_{t|t} = (I - \mathbf{K}_t\mathbf{H}_t)\mathbf{P}_{t|t-1}$

They can be difficult to understand at first, so let's first take a look at what each of these variables are used for:

- $\mathbf{x}_t$ is the current state vector, as estimated by the Kalman filter, at time t.
- $\mathbf{z}_t$ is the measurement vector taken at time t.
- $\mathbf{P}_t$ measures the estimated accuracy of $\mathbf{x}_t$ at time t.
- $\mathbf{F}$ describes how the system moves (ideally) from one state to the next, i.e. how one state vector is projected to the next, assuming no noise (e.g. no acceleration)
- $\mathbf{H}$ defines the mapping from the state vector, $\mathbf{x}_t$, to the measurement vector, $\mathbf{z}_t$.
- $\mathbf{Q}$ and $\mathbf{R}$ define the Gaussian process and measurement noise, respectively, and characterise the variance of the system.
- $\mathbf{B}$ and $\mathbf{u}$ are control-input parameters are only used in systems that have an input; these can be ignored in the case of an object tracker.

Note that in a simple system, the current state $\mathbf{x}_t$ and the measurement $\mathbf{z}_t$ will contain the same set of state variables (only $\mathbf{x}_t$ will be a filtered version of $\mathbf{z}_t$) and $\mathbf{H}$ will be an identity matrix, but many real-world systems will include latent variables that are not directly measured. For example, if we are tracking the location of a car, we may be able to directly measure its location from a GPS device and its velocity from the speedometer, but not its acceleration.

The two stages of the filter correspond to the state-space model typically used to model linear dynamical systems. The first stage solves the process equation:

$$\mathbf{x}_{t+1} = \mathbf{F}\mathbf{x}_k + \mathbf{w}_k \qquad (4.4)$$

The process noise $\mathbf{w}_k$ is additive Gaussian white noise (AWGN) with zero mean and covariance defined by:

$$E\left[\mathbf{w}_t\mathbf{w}_t^T\right] = \mathbf{Q} \qquad (4.5)$$

The second one is the measurement equation:

$$\mathbf{z}_t = \mathbf{H}\mathbf{x}_t + \mathbf{v}_t \qquad (4.6)$$

The measurement noise $\mathbf{v}_t$ is also AGWN with zero mean and covariance defined by:

$$E\left[\mathbf{v}_t\mathbf{v}_t^T\right] = \mathbf{R} \qquad (4.7)$$

In order to implement a Kalman filter, we have to define several variables that model the system. We have to choose the variables contained by $\mathbf{x}_t$ and $\mathbf{z}_t$, and also choose suitable values for $\mathbf{F}$, $\mathbf{H}$, $\mathbf{Q}$ and $\mathbf{R}$, as well as an initial value for $\mathbf{P}_t$.

We will define our measurement vector as:

$\mathbf{z}_t = [\ x_{1,t} \ \ y_{1,t} \ \ x_{2,t} \ \ y_{2,t}\ ]^T$ where $(x_{1,t}, y_{1,t})$ and $(x_{2,t}, y_{2,t})$ are the upper-left and lower-right corners of the bounding box around the detected face, respectively. This is simply the output from the Viola and Jones face detector.

The transition matrix $\mathbf{F}$ defines the equations used to transition from one state vector $\mathbf{x}_t$ to the next vector $\mathbf{x}_{t+1}$ (without taking into account any measurements, $\mathbf{z}_t$). It is plugged in to the process equation:

$$\mathbf{x}_{t+1} = \mathbf{F}\mathbf{x}_k + \mathbf{w}_k$$

The process noise matrix Q measures the variability of the input signal away from the "ideal" transitions defined in the transition matrix. Larger values in this matrix mean that the input signal has greater variance and the filter needs to be more adaptable. Smaller values result in a smoother output.

4. Object Segmentation

The basic idea of segmentation algorithm is change detection. The moving object region is separated from other part of the scene by motion information. This method constructs and maintains up-to-date background information from the video sequence and compares each frame with the background. Any pixel that is significantly different from the background is assumed to be in object region. In other words, this method is not trying to get the object shape information from the changing region of the scene because the characteristics of the changing part are very unpredictable. It depends on object motion, texture, and contrast information which cannot be obtained in advance. Our focus is on the stationary part. This information is more reliable, not very sensitive to object characteristics, and easier to obtain.

An obvious assumption of this approach is stationary background. Since, in many video conferencing and remote surveillance applications, the camera is fixed. This algorithm assumed that input sequence has been properly compensated and the background region is stationary.

The proposed algorithm is divided into five major steps. The first step is to calculate the frame difference mask by threshold the difference between two consecutive input frames. Then, according to the frame difference mask of past several frames, pixels which are not moving for a long time are considered as reliable background in the background registration step. This step maintains an up-to-ate background buffer as well as a background registration mask indicating whether the background information of a pixel is available or not.

By the third step, the background difference mask is generated by comparing the current input image and the background image stored in the background buffer. This background difference mask is our primary information for object shape generation.

In the fourth step, an initial object mask is constructed from the background difference mask and the frame difference mask. If the background registration mask indicates that the background information of an pixel is available, the background difference mask is used as the initial object mask. Otherwise, the value in the frame difference mask is copied to the object mask.

The initial object mask generated in the fourth step has some noise regions because of irregular object motion and camera noise. Also, the boundary region may not be very smooth. In the last step, these noise regions are removed and the initial object mask is filtered to obtain the final object mask.

III. IMPLEMENTATION

Object detection method is requested to automatically segment every object so that there can be a unique tracking associated with the object. It includes five steps: background estimation, background updating, background subtraction, moving cast shadow elimination, and object detection.

A. Create System Objects

Create System objects used for reading the video frames, detecting foreground objects, and displaying results. Create objects for reading a video from a file, drawing the tracked objects in each frame, and playing the video.

Create a video file reader and create two video players, one to display the video, and one to display the foreground mask.

Create system objects for foreground detection and blob analysis The foreground detector is used to segment moving objects from the background. It outputs a binary mask, where the pixel value of 1 corresponds to the foreground and the value of 0 corresponds to the background.

Connected groups of foreground pixels are likely to correspond to moving objects. The blob analysis system object is used to find such group (called 'blobs' or 'connected components'), and compute their characteristics, such as area, centroid, and the bounding box.

B. Initialize Tracks

The initialize tracks function creates an array of tracks, where each track is a structure representing a moving object in the video. The purpose of the structure is to maintain the state of a tracked object. The state consists of information used for detection to track assignment, track termination, and display.

Noisy detections tend to result in short-lived tracks. For this reason, the example only displays an object after it was tracked for some number of frames. This happens when total visible count exceeds a specified threshold.

When no detections are associated with a track for several consecutive frames, the example assumes that the object has left the field of view and deletes the track. This happens when consecutive invisible count exceeds a specified threshold. A track may also get deleted as noise if it was tracked for a short time, and marked invisible for most of the frames.

C. Read a Video Frame and Detect Objects

Read the next video frame from the video file and the detect objects function returns the centroids and the bounding boxes of the detected objects. It also returns the binary mask, which has the same size as the input frame. Pixels with a value of 1 correspond to the foreground, and pixels with a value of 0 correspond to the background.

The function performs motion segmentation using the foreground detector. It then performs morphological operations on the resulting binary mask to remove noisy pixels and to fill the holes in the remaining blobs.

D. Predict New Locations of Existing Tracks

Use the Kalman filter to predict the centroid of each track in the current frame, and update its bounding box accordingly.

Predict the current location of the track and Shift the bounding box so that its center is at the predicted location.ass

Fig. 3 Predict New Locations of Existing Tracks

E. Assign Detections to Tracks

Assigning object detections in the current frame to existing tracks is done by minimizing cost. The cost is defined as the negative log-likelihood of a detection corresponding to a track.

The algorithm involves two steps:

Step 1: Compute the cost of assigning every detection to each track using the distance method of the vision. Kalman Filter System objects. The cost takes into account the Euclidean distance between the predicted centroid of the track and the centroid of the detection. It also includes the confidence of the prediction, which is maintained by the Kalman filter. The results are stored in an MxN matrix, where M is the number of tracks, and N is the number of detections.

Step 2: Solve the assignment problem represented by the cost matrix using the assign detections to tracks function. The function takes the cost matrix and the cost of not assigning any detection to a track.

The value for the cost of not assigning detection to a track depends on the range of values returned by the distance method of the vision Kalman Filter. This value must be tuned experimentally. Setting it too low increases the likelihood of creating a new track, and may result in track fragmentation. Setting it too high may result in a single track corresponding to a series of separate moving objects.

The assign detections to tracks function use the Munkres' version of the Hungarian algorithm to compute an assignment which minimizes the total cost. It returns an M x 2 matrix containing the corresponding indices of assigned tracks and detections in its two columns. It also returns the indices of tracks and detections that remained unassigned.

F. Update Assigned Unassigned Tracks

The update assigned tracks function updates each assigned track with the corresponding detection. It calls the correct method of vision. Kalman Filter to correct the location estimate. Next, it stores the new bounding box, and increases the age of the track and the total visible count by 1. Finally, the function sets the invisible count to 0. Mark each unassigned track as invisible, and increase its age by 1.

G. Delete Lost Tracks

The delete lost tracks function deletes tracks that have been invisible for too many consecutive frames. It also deletes recently created tracks that have been invisible for too many frames overall.

H. Create New Tracks

Create new tracks from unassigned detections. Assume that any unassigned detection is a start of a new track. In practice, you can use other cues to eliminate noisy detections, such as size, location, or appearance.

I. Display Tracking Results

The display tracking results function draws a bounding box and label ID for each track on the video frame and the foreground mask. It then displays the frame and the mask in their respective video players.

This proposed system for detecting and tracking multiple moving objects. The likelihood of tracking errors can be reduced by using a more complex motion model, such as constant acceleration, or by using multiple Kalman filters for every object. Also, you can incorporate other cues for associating detections over time, such as size, shape, and color. Fig. 4 shows displaying tracking of multiple objects. The display tracking results function draws a bounding box and label ID for each track on the video frame and the foreground mask. for detecting and tracking multiple moving objects

Fig. 4 Displaying Tracking of Multiple Objects

Any one of the tracked images segmented and the segmented frames are converted into video sequence as shown in fig. 5

Fig. 5 Segmented object from a video sequences

IV. Conclusion

Object detection and tracking methods are being surveyed. This project has examined methods to improve the performance of object detection and tracking algorithms and Block matching technique for object tracking applications and examined methods for multi-modal fusion in an object tracking system.

Motion segmentation is a key step in many tracking algorithms as it forms the basis of object detection. Improving segmentation results as well as being able to extract additional information such as frame difference, Gaussian of mixture model, background subtraction allows for improved object detection and thus tracking.

However strength of kalman filter is their ability to track object in adverse situation. Integrating a kalman filter within a standard tracking system allows the kalman filter is to use progressively updated features and aids in main training identity of the tracked object, and provides tracking system with an effective means.

Acknowledgment

At this moment take the opportunity to convey the deepest thanks to principal, Dr. C. Nelson Kennedy Babu, M.Tech., Ph.D and the head of the department Associate Professor Mr. S. Elango, M.E., (PhD), for their technical guidance to make this achievement and also gratitude to all faculty and supporting staffs of the Department of Electronics and Communication Engineering for their encouragement and timely help. Finally, thank all our Friends and Parents for their blessing to complete this work successfully.

References

1. F.-C. Cheng, S.-C. Huang, and S.-J. Ruan, "Advanced background subtraction approach using Laplacian distribution model," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2010, pp. 754–759.
2. J. Zhong and S. Sclaroff, "Segmenting foreground objects from a dynamic textured background via a robust Kalman filter," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, vol. 1. Oct. 2003, pp. 44–50.
3. J. Sun, W. Zhang, X. Tang, and H.-Y. Shum, "Background cut," in *Proc. 9th Eur. Conf. Comput. Vis.*, 2006, pp. 628–641.
4. F. Liu and M. Gleicher, "Learning color and locality cues for moving object detection and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 320–327.
5. K.C. Lien and Y.-C. F. Wang, "Automatic object extraction in single concept videos," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2011,pp. 1–6.
6. M. Leordeanu and R. Collins, "Unsupervised learning of object features from video sequences," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, pp. 1142–1149.
7. L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1254–1259, Nov. 1998.
8. P. Harding and N. M. Robertson, "Visual saliency from image features with application to compression," *Cognit. Comput.*, vol. 5, no. 1, pp. 76–98, 2012.

9. X. Hou and L. Zhang, "Saliency detection: A spectral residual approach,"in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.

10. C. Guo, Q. Ma, and L. Zhang, "Spatio-temporal saliency detection using phase spectrum of quaternion Fourier transform," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

11. C. Rother, V. Kolmogorov, and A. Blake, "'GrabCut': Interactive foreground extraction using iterated graph cuts," *ACM Trans. Graph.*,vol. 23, no. 3, pp. 309–314, 2004.

12. M. Gong and L. Cheng, "Foreground segmentation of live videos using locally competing 1SVMs," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2011, pp. 2105–2112.

AUTHOR(S) PROFILE

K. Meenatchi received the B.E. degree in Electronics and Communication Engineering from Anna University and pursuing final semester M.E. Embedded Systems in Anna University.

P. Subhashini received the B.E. degree in Electronics and Communication Engineering and M.E. degree in Applied Electronics from Anna University. Presently she is working as Assistant Professor, Department of Electronics and Communication Engineering in CMS College of Engineering.