

IMPLEMENTATION OF OBJECT TRACKING SYSTEM USING REGION FILTERING ALGORITHM BASED ON SIMULINK BLOCKSETS

DR.P.SUBASHINI

Associate Professor

Department of Computer Science

Avinashilingam Deemed University for Women, Coimbatore

India

mail.p.subashini@gmail.com

MS.M.KRISHNAVENI

Research Assistant

Department of Computer Science

Avinashilingam Deemed University for Women, Coimbatore

India

krishnaveni.rd@gmail.com

MR. VIJAY SINGH

Deputy Director,

Naval Research Board-DRDO

New Delhi

India

Abstract

Video tracking is the process of locating a moving object in time that is visualized by camera and are widely used in surveillance, animation and robotics Tracking describes the process of recording movement and translating that movement onto a digital model. The set of constraints that produce the most accurate tracking is the one that describes better the action performed. The key difficulty in video tracking is to associate target locations in consecutive video frames, especially when the objects are moving fast relative to the frame rate. Here, a video tracking system is been employed in which motion model describes how the image of the target might change for different possible motions of the object to track. The role of the tracking algorithm adopted for this system is to analyze the video frames to estimate the motion parameters. These parameters characterize the location of the target. In this research, three features are been extracted from each moving objects such as centroid, area, average luminance. Finally the similarity function is applied to tracking and the attempt proves that the chosen method has good performance under dynamic circumstances for real time tracking. Simulink is integrated with MATLAB to build a model for object tracking and data transfer is easily handled between the programs. The Simulink based customizable framework is designed for rapid simulation, implementation, and verification of video and image processing algorithms and systems.

Keywords-Tracking; Optical flow; Thresholding; Simulink; Region filtering; motion parameter

I. Introduction

Object tracking is used to describe the process of recording movement and translating that movement onto a digital model. Simulink with Video and Image processing blockset enable to run fast simulations for real-time embedded video, vision, and imaging systems[4]. It can create executable specifications for communicating the system to downstream design teams and to provide a golden reference for verification throughout the design process. The amount of work does not vary with the complexity or length of the performance to the same degree as when using traditional techniques. The aim of this paper work is to implement an efficient methodology to track the moving object present inside the moving videos. Using blockset, the framework is carried out using the

image processing steps such as video processing, frame display, background subtraction, edge Detection, segmentation and tracking [1]. First, the videos are separated as frames and preprocessing method is used for the color conversion to subtract the foreground objects from the background, and background subtraction is used to find the total or sudden change in intensity in the video. Edge detection is performed as the middle work and to extract the boundary of the object from the background segmentation module is executed. Finally tracking will be carried out. The main objective of the object tracking is to detect and track the moving object through video sequence. Most of the image processing techniques covered here are tracking related experimental results. The paper is organized as follows: Section 2 deals with the construction of the framework for tracking. Section 3 deals with the preprocessing for color conversion, frame rate display and background subtraction. Section 4 converses the proposed algorithm for the segmentation which includes edge detection and Color based region filtering algorithm. Section 5 discusses tracking module and Section 6 explains the experimental results and findings of the tracking objects. The paper concludes with the need and the future work of the system.

2. Proposed framework for tracking System

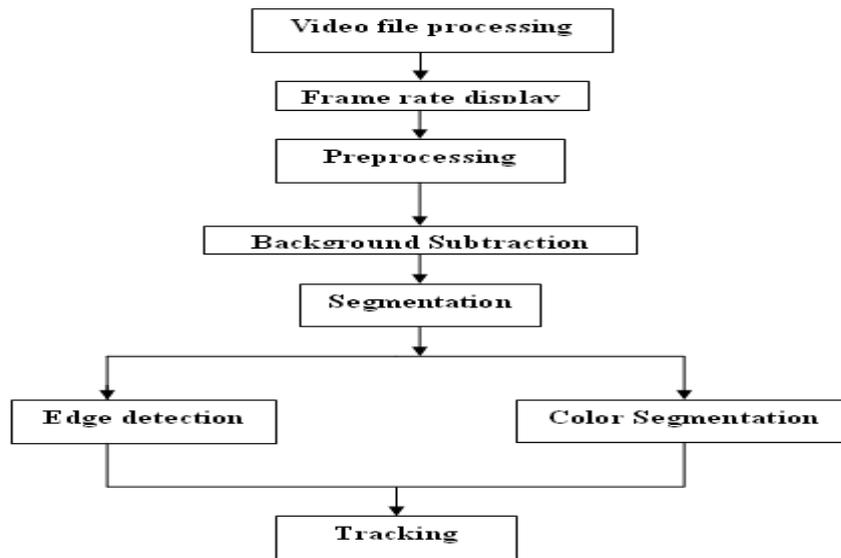


Figure 1: Framework of the proposed system

This proposed framework combines the existing and recent techniques of video processing techniques for tracking system. The output of the system delivers a Matlab-based application development platform intended for tracking system[7]. It allows the user to investigate, design, and evaluate algorithms and applications using object based videos. It offers standardized, which do not require detailed knowledge of the target hardware[6]. The proposed figure is given if figure 1.

3. Preprocessing

Preprocessing is mainly used to enhance the contrast of the image, removal of noise and isolating objects of interest in the image. Preprocessing is any form of signal processing for which the output is an image or video, the output can be either an image or a set of characteristics or parameters related to image or videos to improve or change some quality of the input [5]. Preprocessing helps to improve the video or image such that it increases the chance for success of other process.

3.1 Frame Rate Display

Frame rate or frame frequency is the frequency at which an imaging device produces unique consecutive images, frames that applies equally well to compute graphics, video cameras, film cameras and motion capture system. Frame rate is most often expressed in frames per second and is also uttered in progressive scan monitors as Hertz (Hz). The frame rate display block calculates and displays the average update rate of the input signals. If

the block displays 30 the model is updating the input signal 30 times every second. This block is used to check the video frame rate of the simulation to control the specified number of video frames.

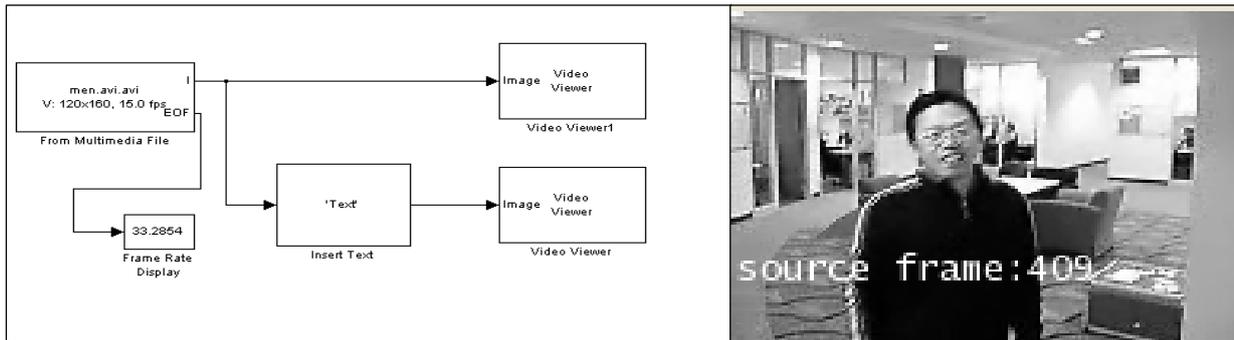


Figure 2: Frame rate display block and result

When the input is a frame-based M-by-N matrix, the block treats each of the N columns as an independent channel, and delays each channel as specified by the delay parameter. When the input is frame based, the delay parameter can be a scalar integer by which to equally delay all channels or a vector whose length is equal to the number of channels. There are four different choices for initial conditions. The initial conditions can be the same or different for each channel. They can also be constant or varying along each channel.

3.2. Color conversion

Color space conversion is the representation of a color from one basis to another. The goal is to make the translated image look as similar as possible to the original color conversions. A color can be represented by its hue, saturation and value components. It provides a set of routines for converting between RGB and other color spaces. The image processing functions themselves assumes all color data in RGB and process an image that uses a different color space by first converting it to RGB, and then converting the processed image back to the original color space. The Color space conversion block enables to convert color information from the R'G'B' color space to the Y'CbCr color space and from the Y'CbCr color space to the R'G'B' color space. This block can also be used to convert from the R'G'B' color space to intensity.



Figure 3: Color Space conversion (a) Original image (b). Intensity image

3.3. Background subtraction

Background subtraction as the name suggest is the process of separating out foreground objects from the background in a sequence of video frames [8]. Many methods exist for background subtraction, each with different strength and weakness in terms of performance and computational requirements. Background subtraction is implemented using varying complexity.

- Low complexity-using frame difference method.
- Medium complexity using approximate median method
- High complexity using mixture of Gaussian method.

Background subtraction involves comparing observed image with an estimate of the image, if it contained no object of interest. In this background subtraction process, simulink model is used to specify the edit parameter block to estimate the background[3]. The adopted techniques in the module are

Estimating median overtime: This algorithm updates the median value of the time series data based upon the new data sample. The algorithm increments or decrements the median by an amount that is related to the running standard deviation and the size of the time series data. It also applies a correction to a median value if it detects a local ramp in the time series data.

Computing median over time: This method computes the median of the values at each pixel location over a time window of 30 frames.

Estimating moving objects: This algorithm identifies the moving object in the first few image frames and labels the corresponding pixels as foreground pixels. Next the algorithm identifies the incomplete background as the pixels that do not belong to the foreground pixels. As the foreground object move, the algorithm estimates more and more of the background pixels.

Once the model estimates the background, it subtracts the background from each video frame to produce foreground images. By thresholding and performing morphological closing on each foreground image, the model produces binary feature images. The model locates the object in each binary feature image using blob analysis block. Then it uses shape black draw a green rectangle around the object that passes beneath the white line. The counter in the upper left corner of the results window tracks the number of object in the region of interest.



Figure 4: Back ground subtraction based on frame difference method

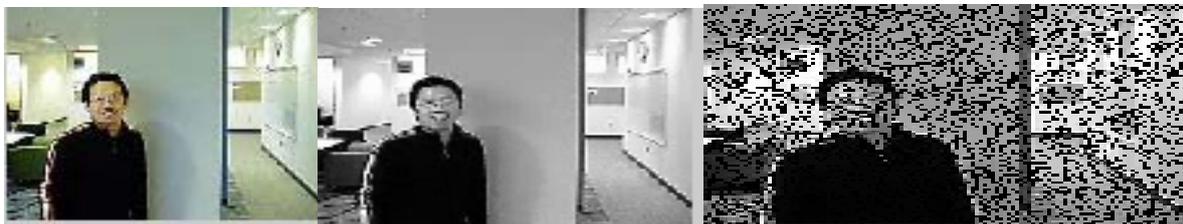


Figure 5: Back ground subtraction based on Mixture of Gaussian Method



Figure 6: Back ground subtraction based on Approximate median method

4. Segmentation

The processing of subdividing the image into its constituent parts or object is called image segmentation.

4.1. Segmentation based on Edge Detection

Edge detecting is a collection of very important local image pre-processing method used to locate changes in the intensity function. An edge is a property attached to an individual pixel and is calculated from the image function behavior in a neighborhood of that pixel. Edge takes a grayscale image as its input, and returns a binary image of the same size as the input image, with 1's where the function finds edges and 0's elsewhere.

- ✓ Sobel operator
- ✓ Canny operator
- ✓ Roberts operators

The Sobel method finds edges using the Sobel approximation to the derivative. It returns edges at those points where the gradient of image is maximum. The Canny method finds edges by looking for local maxima of the gradient of image. The gradient is calculated using the derivative of a Gaussian filter. The Roberts method finds edges using the Roberts approximation to the derivative. It returns edges at those points where the gradient of the image is maximum. The Edge detection block outputs a binary image with the edges shown in white. This output is displayed in the Edges window. The compositing block accepts the original video frames, shown in the original window, and the output of the edge detection block as inputs at its Image1 and Mask ports, respectively. The input to the Mask port tells the compositing block which pixels to highlight. As a result, the model displays a composite image in the overlay window, where the original pixel values are overwritten by the white edge values.

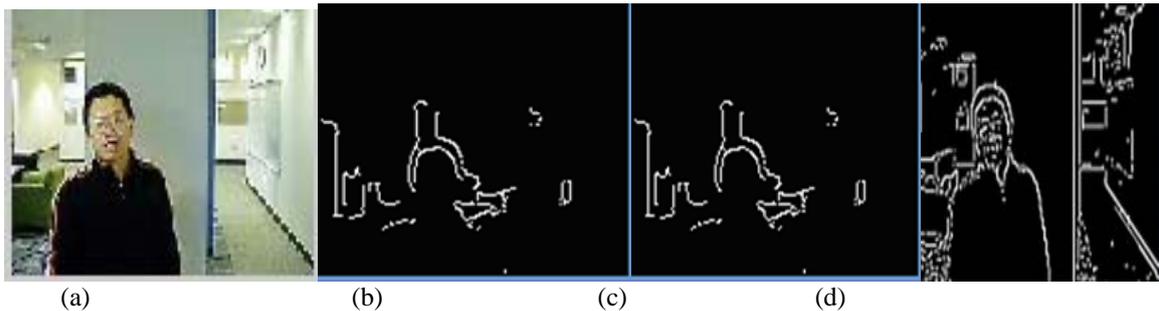


Figure 5: Segmentation based on edge detection (a) Original image (b) Roberts operator (c) Sobel Operator (d) Canny Operator

3.2 Color Segmentation based on Region filtering

To extract feature from the object, the image has to be subdivided into constituent parts or objects. The process of sub dividing the images into its constituent parts or objects is called image segmentation. Segmentation algorithm can be divided into two categories based on the two properties such as discontinuity and similarity [9]. The segmentation algorithm considered here is based on the similarity property. Thersholding technique is used to segment the object from the image. Thersholding is an operation in which a reference gray level is selected using histogram such that the object and the background can be separated.

To create an accurate color model for the demo, many images containing skin color samples were processed to compute the mean (m) and covariance (C) of the Cb and Cr color channels. Using this color model, the Color Segmentation/Color Classifier subsystem classifies each pixel as either skin or nonskin by computing the square of the Mahalanobis distance and comparing it to a threshold. The equation (1) for the Mahalanobis distance is shown below

$$SquaredDistance(Cb, Cr) = (x - m)' * inv(C) * (x - m), \text{ where } x = [Cb; Cr] \dots\dots(1)$$

The result of this process is binary image, where pixel values equal to 1 indicate potential skin color locations. The color segmentation/Filtering subsystem filters and performs morphological operations on each binary image, which creates the refined binary images shown in the Skin region window. The Color segmentation/Region filtering subsystem uses the Blob analysis block and the extract face and hand subsystem to determine the location of the person's face and hand in each binary image. The Display results/Mark image subsystem uses this location information to draw bounding boxes around these regions.

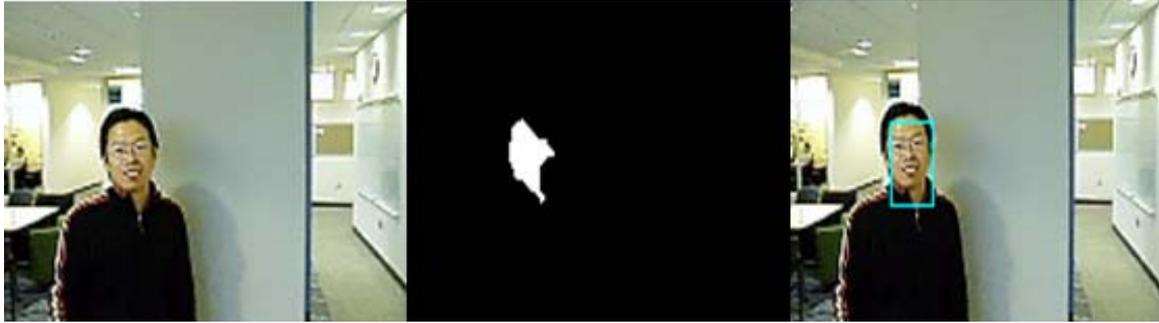


Figure 6: Color Segmentation Based On Region Filtering

5. Object tracking

This paper mainly aims at detecting and tracking the moving object using image processing techniques in MATLAB using simulink. Here, three features of each moving objects such as centroid, area and average luminance are been extracted. At last the similarity function is applied to tracking and the result provides good performance under dynamic circumstances for real time tracking [2]. This model uses an optical flow estimation technique to estimate the motion vectors in each frame of the video sequence. By thresholding and performing morphological closing operation on the motion vectors, the model produces binary feature images. The model locates the objects in each binary feature image using the Blob analysis block, then it uses the draw Shapes block to draw a green rectangle around the objects that pass beneath the white line. The counter in the upper left corner of the results window tracks the number of objects in the region of interest.

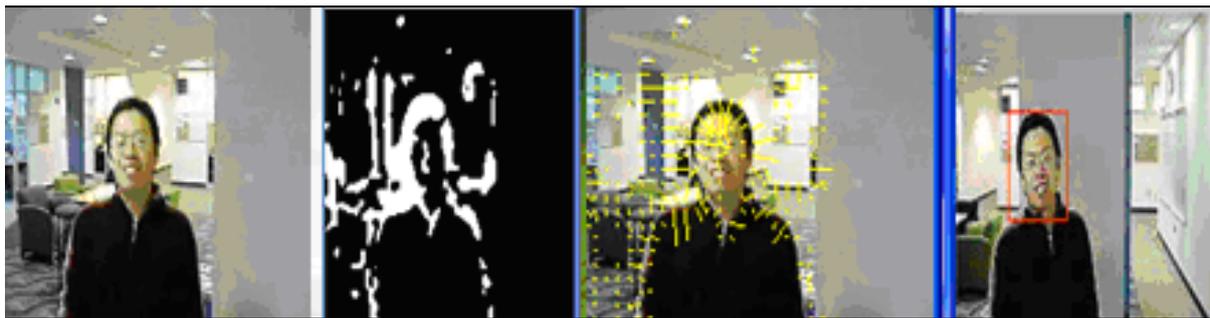


Figure 7: Tracking using optical flow

6. Conclusion

The objective of tracking is to associate target objects in consecutive video frames. It is more useful for military entertainment, sports, and medical applications and for validation of computer vision and robotics. Videos and images from open source can be accomplished and so it can be easily implemented and its usage is wider. Implementation of this methodology using Simulink blockset is more useful for security and video surveillance. This system also provides additional services such as information about the location and identity of objects at different points in time is the basis for detecting unusual object movements. Tracking the objects true position is done by tracking its state using region filtering and this uses information from the current blob and the previous object state to create an estimate of the objects new state. This work emphasis a system in which there are much scope for future investigation and analysis based on the performance on tracking the objects.

References

- [1] Zhou, Q.; Aggarwal, J. K.; "Tracking and classifying moving objects from video", Proc of 2nd IEEE Intl Workshop on Performance Evaluation of Tracking and Surveillance (PETS'2001), Kauai, Hawaii, USA (December 2001).
- [2] Stauffer C.; "Estimating tracking sources and sinks", Proc of 2nd IEEE Workshop on Event Mining (in conjunction with CVPR'2003), 4, Madison, Wisconsin (June 2003).
- [3] Zhenggao Xu. Applications Engineer Implementing "Video Tracking. Systems on DM6437 EVM Using. MATLAB and Simulink". A Model-Based Design Approaches.
- [4] G. Karlsson and M. Vetterli, "Three-dimensional subband coding of video," in Proc. Int. Conf. Acoustics, Speech, Signal Processing, vol. 2, Apr. 1988
- [5] Collins, R. T.;Lipton, A. J.; Kanade, T.; Fujiyoshi, H.; Duggins, D.; Tsin Y.Tolliver,D.; Enomoto, N. and Hasegawa, O.; "A system for video surveillance and monitoring", Technical Report CMU-RI-TR -00-12, CMU, 2000
- [6] Chen, W.P. and J.C. Hou, 2004. Dynamic clustering for acoustic target tracking in wireless sensor networks. IEEE Trans. Mobile Comput., 3: 258-271. DOI: 10.1109/TMC.2004.22.

- [7] Yu, Z., W. Jianming and H. Liu, 2009. An energy efficient target tracking framework in wireless sensor networks. *EUROSIP J. Adv. Sign. Process.*, 2009: 14. DOI: 10.1155/2009/524145
- [8] C. Ridder, O. Munkelt, and H. Kirchner, "Adaptive background estimation and foreground detection using kalman-filtering". In *ICRAM*, 1995.
- [9] F. Calderero and F. Marques, "General region merging approaches based on information theory statistical measures.," in *Proc. of ICIP*, San Diego (CA), October 2008.