STUDY OF OBJECT DETECTION IMPLEMENTATION USING MATLAB

L.S.Alandkar[1], S.R.Gengaje[2]

[1]*Student, Department of Electronics Engineering, Walchand Institute of Technology, MH, India*
[2]*Professor, Department of Electronics Engineering, Walchand Institute of Technology, MH, India*

Abstract

Object detection is most prevalent step of video analytics. Performance at higher level is greatly depends on accurate performance of object detection. Various platforms are being used for designing and implementation of object detection algorithm. It includes C programming, MATLAB and Simulink, open cv etc. Among these, MATLAB programming is most popular in students and researchers due to its extensive features. These features include data processing using matrix, set of toolboxes and Simulink blocks covering all technology field, easy programming, and Help topics with numerous examples. This paper presents the implementation of object detection and tracking using MATLAB. It demonstrates the basic block diagram of object detection and explains various predefined functions and object from different toolboxes that can be useful at each level in object detection. Useful toolboxes include image acquisition, image processing, and computer vision. This study helps new researcher in object detection field to design and implement algorithms using MATLAB.

Keywords: *Video Analytics, Object Detection, Tracking, MATLAB.*

----------***----------

1. INTRODUCTION

Video analytics is popular segment of computer vision. It has enormous applications such as traffic monitoring, parking lot management, crowd detection, object recognition, unattended baggage detection, secure area monitoring, etc[1]. Object detection is critical step in video analytics. The performance at this step is important for scene analysis, object matching and tracking, activity recognition[2]. Over the years, research is flowing towards innovating new concept and improving or extending the established research for performance improvement of object detection and tracking.

Various object detection approaches has been developed based on statistic, fuzzy, neural network etc. Most approaches involve complex theory. These approaches can be evolved further by thorough understanding, implementation and experimentation. All these approaches can be learned by reading, reviewing, and taking professor's/ expert guidance. However, implementation and experimentation requires good programmer.

Various platforms are being used for the design and implementation of object detection and tracking algorithm. These platforms include C programming, Open CV, MATLAB etc. The object detection system to be used in real time should satisfy two conditions. First, system code must be short in terms of execution time. Second, it must efficiently use memory. However, programmer must have good programming skill in case of programming in C and OpenCV. Moreover, it is time intensive too for new researcher to develop such efficient code for real time use.Assuming all these facts, the MATLAB is found as better platform to design and implementation of algorithm. It contains more than seventy toolboxes covering all possible fields in technology. All toolboxes are rich with predefined functions, system objects and simulink blocks. This feature helps to write short code and saves time in logic development at various steps in system. MATLAB supports matrix operation which is huge advantage during processing of an image or frame in video sequence. MATLAB coding is simple and easily learned by any new researcher. This paper presents implementation of object detection system using MATLAB and its toolboxes. This study explored various toolboxes and identified useful functions and objects that can be used at various levels in object detection and tracking. Toolboxes mainly include computer vision, image processing, and image acquisition. MATLAB 2012 version is used for this study.

This paper organized in four section second section describe general block diagram of object detection. Third section involves MATLAB functions and objects that are useful in implementation of object detection system. Sample coding is presented for object detection and tracking in section four. Paper is concluded in fifth section.

2. BLOCK DIAGRAM OF OBJECT DETECTION

This section explains general block diagram of object detection and significance of each block in the system. Common object detection mainly includes video input, preprocessing, object segmentation, post processing. It is shown in Fig. 1.

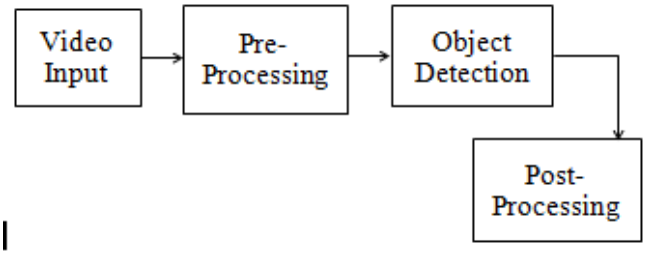


Fig. 1 General framework of Object Detection

The significance of each block is as follows

- Video Input:- It can be stored video or real time video.
- Preprocessing:-It mainly involves temporal and spatial smoothing such as intensity adjustment, removal of noise. For real-time systems, frame-size and frame-rate reduction are commonly used. It highly reduces computational cost and time[1].
- Object detection: It is the process of change detection and extracts appropriate change for further analysis and qualification. Pixels are classified as foreground, if they changed. Otherwise, they are considered as background. This process is called as background subtraction. The degree of "change" is a key factor in segmentation and can vary depending on the application. The result of segmentation is one or more foreground blobs, a blob being a collection of connected pixels [1].
- Post processing: Remove false detection caused due to dynamic condition in background using morphological and speckle noise removal.

3. MATLAB IMPLEMENTATION

Different toolboxes have been explored for functions and objects which can be useful atvarious levels in the object detection. All such functions/ objects are described in this section.

3.1 Video Input

Input video has two possible ways Stored Video and real time video. Stored video can be obtained from standard dataset available from internet. Real time video includes camera continuously monitoring specific area producing real time video. These video can be understood by MATLAB after reading.

3.1.1 Stored Video

Some commonly used standard video dataset are as follows

- Wallflower Dataset [4]: It is provided by Toyama et al[]and contains seven canonical sequences with different background situation.
- PETS Dataset: ”Performance Evaluation of Tracking and Surveillance” (PETS) consist of various datasets like PETS 2001, PETS 2003 and PETS 2006. They are more useful for tracking evaluation rather for background.
- ChangeDetection.net Dataset[5]: The CDW29 dataset presents a realistic video dataset consisting of 31 video sequence which are categorized in 6 different challenges. Color and Thermal IR type of video included in dataset.
- BMC 2012 Dataset[6]: This dataset include real and synthetic video. It is mainly used for comparison of different background subtraction techniques.
- Fish4knowledge Dataset[7]: The Fish4knowledge 35 dataset is an underwater benchmark dataset for target detection against complex background.
- Carnegie Mellon Dataset[8]: The sequence of CMU25 by Sheikh and Shah involves a camera mounted on a tall tripod. The wind caused the tripod to sway back and forth causing vibration in the scene. This dataset is useful while studying camera jitter background situation.

Storedvideo need to be read in appropriate format before processing. Various related functions from image processing(IP) and computer vision(CV) toolbox can be used for this purpose. Those are summarized in Table 1

Table 1 Useful function/object for stored video acquisition

Tool -box	Object/ Function	Function Name	Use
IP	Function	imread	Read image from graphics file
IP	Function	iminfo	Information about graphics file
IP	Function	imwrite	Write image to graphics file
IP	Function	imshow	Display image
CV	Object	vision.VideoFileReader	Read video frames and audio samples from video file
CV	Object	vision.VideoFileWriter	Write video frames and audio samples to video file
CV	Object	vision.VideoPlayer	Play video or display image

3.1.2 Real Time Video

Image acquisition is widely used toolbox which allows real time acquisition of video from video acquisition device. Some commonly used function are explained below

- Imaqtool:It launches an interactive GUI and allowsusers to explore, configure, and acquire data from image acquisition devices.
- Videoinput: It can be used to create video input object. This object can further be used to acquire and display the image sequences.
- Propinfo: It captures all the property information about image acquisition object. This information can be useful in further video processing.
- Getsnapshot: It immediately returns one single image frame, from the video input object. This function is useful to capture image at critical moment.
- Trigger: Initiates data logging for the video input object. It can be used to initialize video at appropriate moment and collect a video data.
- Triggerconfig: User can configuretrigger properties of video input object.

3.2 Preprocessing

Preprocessing may include series of operation those are shown

3.2.1 Video Type Conversion

The video is needed to be converting to appropriate data type after reading. Useful objects and functions are listed in Table 2.

Table 2 Useful function/object for video data type conversion

Tool box	Function /object	Name	Use
CV	Object	vision.ImageDataTypeConverter	Converts and scales an input image to a specified output data type. Output Data type may include double, single, int8, uint8, int16, uint16, boolean, Custom
IP	Function	im2doubl, im2single, im2 uint8, im2uint16	These function can be used to convert image to specified form

3.2.2 Video Enhancement

This step may include noise removal, contrast adjustment, image correction. Useful function and object summarized in Table 3

Table3 Useful function/object for video enhancement

Tool box	Function/object	Name	Use
CV	Object	vision.MedianFilter	2D median filtering(to remove salt and pepper noise)
CV	Object	vision.ImageFilter	Perform 2-D FIR filtering of input matrix
CV	Object	vision.ContrastAdjuster	Adjust image contrast by linear scaling
CV	Object	vision.HistogramEqualizer	Enhance contrast of images using histogram equalization
IP	Function	imadjust	Adjust image intensity values or colormap
IP	Function	imcontrast	Adjust Contrast tool
IP	Function	histeq	Enhance contrast using histogram equalization

3.2.3 Feature Extraction

Any object detection system performs segmentation based on one or more feature of the scene. It may include color, corner, edge, shape, gradient, texture, DCT or DFT coefficient. Different functions are available to extract these features as shown in Table 4.

Table 4 Useful function/object for feature extraction

Toolbox	Function/ Object	Name	Use
IP	Function	rgb2gray	It can be used to convert RGB image to grayscale image
IP	Function	rgb2ycbcr	This function can convert RGB image to YCbCr color space
IP	Function	ycbcr2rgb	It convert YCbCr color space of image to RGB image
IP	Function	corner	Find corner points in image
IP	Function	edge	Find edges in grayscale image
IP	Function	imgradient	Gradient magnitude and direction of an image Directional gradients of an image
IP	Function	entropyfilt	Local entropy of grayscale image
IP	Function	rangefilt	Local range of image
IP	Function	stdfilt	Local standard deviation of image
CV	Object	vision.ColorSpaceConverter	Convert color information between color spaces such as RGB,rgb, grayscale,LAB,HSI, YCbCR.
CV	Object	vision.DCT	Compute 2-D discrete cosine transform
CV	object	vision.FFT	Two-dimensional discrete Fourier transform
CV	Object	vision.EdgeDetector	Find object edge. Algorithms can be selected sobel, prewitt, robert, canny
CV	Object	vision.CornerDetector	Detect corner features

3.3 Object Detection

Various Object Detection methods being used to detect object. These methods are classified based on Template, motion, classifier, feature. Computer vision toolbox includes some predefined objects which can be useful to implement these object detection methods as shown in Table 5.

Table5 Useful function/object for object detection

Toolbox	Function/object	Name	Use
CV	Object	vision.CascadeObjectDetector	This function extract various feature of face like nose , eye, mouth, upper body using Viola-Jones algorithm.
CV	Object	vision.OpticalFlow	This function can be used for the estimation of object velocities using the Horn-Schunck or the Lucas-Kanade method.
CV	Object	vision.PeopleDetector	It detects un-occludedpeople inupright positionusing the Histogram of Oriented Gradient (HOG) features and a trained Support Vector Machine (SVM) classifier.
CV	Object	vision.TemplateMatcher	It is used to perform object detection based on template match.

3.4 Post Processing

Post processing is required to remove unwanted portion in the foreground mask. It may arise due to false detection caused by dynamic background condition. This False detection may include speckle noise, small holes in the scene etc. Along with this task, detected object can be annotated for proper display. Useful function and object are listed in Table 6.

Table 6 Useful function/object for post processing

Toolbox	Function/Object	Name	Use
IP	Function	imclose	This function performs morphological closing on the binary or grayscale image. It is very important for removing small holes/ pepper noise in an image.
IP	Function	imopen	This function performs morphological opening on the grayscale or binary image IM with specified structuring element. Salt noise can be removed by this function.
IP	Function	imdilate	It performs dilation of grayscale or binary, image.
IP	Function	imerode	It performs erosion of grayscale or binary image.
IP	Function	imfill	It displays the binary image on the screen and allow user to define the region to fill in interactive way by selection of point.
CV	Object	vision.MorphologicalOpen	This object works same as imopen but uses only flat structuring elements.
CV	Object	vision.MorphologicalClose	This object works same as imclose but uses only flat structuring elements.
CV	Object	vision.MorphologicalDilate	This object dilates an intensity or binary image.
CV	Object	vision.MorphologicalErode	This object erode an intensity or binary image.
CV	Object	vision.BlobAnalysis	This object computes statistics for connected regions in a binary image. It may involve centroid, minor axis, major axis, eccentricity, orientation, perimeter etc.
CV	Object	vision.ConnectedComponentLabeler	This object labels and counts the connected regions in a binary image. In this labeling, 0 represents background other numbered pixels represents corresponding numbered object.
CV	Object	vision.MarkerInserter	Draw markers on output image
CV	Object	vision.ShapeInserter	Draw rectangles, lines, polygons, or circles on an image
CV	Object	vision.TextInserter	Draw text on image or video stream
CV	Function	insertObjectAnnotation	Annotate RGB color or grayscale image or video stream

4. SAMPLE MATLAB CODE FOR OBJECT DETECTION

Simple example is presented in this section about object detection and object tracking.

4.1 Object Detection using MATLAB

Object detection is carried out on canonical sequence Time of Day in Wallflower Dataset

- Video input: stored video
- Preprocessing: RGB to Gray conversion and Gaussian noise removal using median filter
- Object Detection: Adaptive GMM based Background Subtraction method with number of component=3, Learning Rate=0.05,thresold=0.7

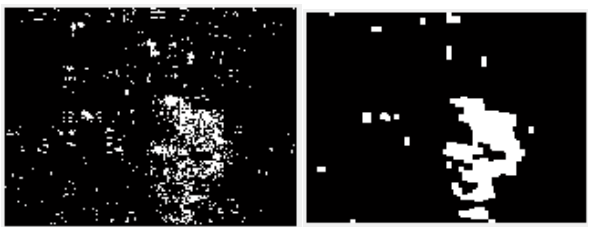
- Postprocessing= morphological closing and opening to remove salt pepper noise.

```
videoFReader = vision.VideoFileReader('TDVideo.avi');

H = vision.ColorSpaceConverter('Conversion', 'RGB to intensity');

medianFilter1 = vision.MedianFilter([3 3]);

hclosing = vision.MorphologicalClose;
hclosing.Neighborhood = strel('square',3);

hclosing = vision.MorphologicalClose;
hclosing.Neighborhood = strel('square',3);

hfdet = vision.ForegroundDetector('NumTrainingFrames',200,'NumGaussians',3,'LearningRate',0.05,'MinimumBackgroundRatio',0.7);

while ~isDone(videoFReader)
frame = step(videoFReader);
frame1=step(H,frame);
frame2=step(medianFilter1,frame1);
fgMask = step(hfdet,frame2);
Det1=step(hclosing,fgMask);
Det2=step(hclosing,Det1);
end

release(H);
release(hfdet);
release(medianFilter1);
release(hclosing);
release(videoFReader);
```

In this sample code, 'frame' is input frame. 'frame1' grayscale conversion of input frame. ' frame2' is median filtered output. 'fgMask' is foreground mask generated by GMM based object detection method. 'Det2' is post processed output. Sample code output is shown in Fig. 2

a. **b**

c. **d**



e. **f**

Fig.2 Output at various level (a) input sequence (b) Ground truth (c) Gray-scale conversion (d) Median filtering (e)Foreground mask after Object Detection (f) Post-processed output

5. DISCUSSION AND CONCLUSION

This paper presents the basic object detection system. MATLAB platform(MATLAB 2012) is used to carry implementation of the system. Different Toolboxes has been explored and useful MATLAB functions and objects are collected which can be useful at various stages. Toolboxes mainly includes image acquisition, image processing and computer vision. Sample MATLAB coding is presented for object detection. Each stage in the system has been implemented by available functions/objects in toolbox. It shows that implementation is easy and code is being short due to use of predefined objects/functions in MATLAB. This study may help new student and research in this field to study, implement and experiment established research.

REFERENCES

[1] Video Analytics: http://www.dspdesignline.com/videoanalytics.html

[2] Jun-Wei Hsieh, Shih-Hao Yu, Yung-Sheng Chen, An Automatic Traffic Surveillance System for Vehicle Tracking and Classification, IEEE Transactions on Intelligent Transportation Systems, Vol. 7

[3] Shireen Y. Elhabian, Khaled M. El-Sayed, Moving Object Detection in Spatial Domain using Background Removal Techniques - State-of-Art , *Recent Patents on Computer Science* 2008, *1*, 32-54

[4] K. Toyama, J. Krumm, B. Brumiit, and B. Meyers. Wallflower: Principles and practice of background maintenance. International Conference on Computer Vision, pages 255–261, September 1999.

[5] N. Goyette, P. Jodoin, F. Porikli, J. Konrad, and P. Ishwar. changedetection.net: A new change detection benchmark dataset. IEEE Workshop on Change Detection, CDW 2012 at CVPR 2012, June 2012.

[6] A. Vacavant, T. Chateau, A. Wilhelm, and L. Lequievre. A benchmark dataset for foreground/background extraction. International Workshop on Background Models Challenge, ACCV 2012, November 2012.

[7] I. Kavasidis, S. Palazzo, and C. Spampinato. An innovative web-based collaborative platform for video annotation. Multimedia Tools and Applications, pages 1–20, 2013.

[8] Y. Sheikh and M. Shah. Bayesian modeling of dynamic scenes for object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27:1778–1792, 2005.

[9] Ruolin Zhang, Jian Ding, "Object Tracking and Detecting Based on Adaptive Background Subtraction", International Workshop on Information and Electronics Engineering, 2012, 1351-1355.

[10] J.JoshanAthanesious, P.Suresh, "Systematic Survey on Object Tracking Methods in Video",International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) October 2012, 242-247

BIOGRAPHIES

Lajari S. Alandkar –Ph.D. student (Electronics) at Walchand Institute of Technology, Solapur, MH, India. Her areas of interest include Image Processing, and Computer Vision.

Sachin R. Gengaje – He is presently working as a Head and Professor, Department of Electronics Engineering at Walchand Institute of Technology, Solapur, MH, India. His areas of interest include Soft Computing, Image & Speech Processing, and Biomedical Expert Systems.