

SIMULINK based Moving Object Detection and Blob Counting Algorithm for Traffic Surveillance

Mayur Salve
B.E(Electronics)
RAIT

Dinesh Repale
B.E(Electronics)
RAIT

Sanket Shingate
B.E(Electronics)
RAIT

Divya Shah
Asst. Professor
RAIT

ABSTRACT

The main objective of this paper is to propose a SIMULINK MODEL to detect moving vehicles. Background subtraction is the technique used in this algorithm. Based on the retrieved information, automatic traffic surveillance can be done. Initially, a recorded video is given directly to the blocks. The main logic is then implemented using various Embedded MATLAB blocks using individual algorithms. The algorithm takes into consideration three main techniques namely Background Subtraction, Edge Detection and Shadow Detection. Background Subtraction block is sub-divided into Selective and Non-selective parts to improve the sensitivity and give accurate background. Edge detection helps to detect the exact boundaries of moving vehicles. This is followed by the shadow detection block that removes the falsely detected pixels that are generated due to shadow of the vehicle. By analyzing the output of the blocks discussed above, the final mask is generated. The mask along with the input frame processed to give the final video output with the detected object. Furthermore, using a Blob analysis block, parameters such as number of blobs per frame (vehicles) and the area of blobs can be used directly for traffic surveillance. Finally a Blob counting block is used to count and display the total number of cars.

Keywords

SIMULINK, MATLAB, Moving Object Detection, Blob Counting Algorithm, Traffic Surveillance.

1. INTRODUCTION

Complex traffic surveillance systems are often used for controlling traffic and detecting unusual situations, such as traffic congestion or accidents. This paper describes an approach which detects moving and recently stopped vehicles using the novel technique of background subtraction [1] [2]. Only the SIMULINK Utility of MATLAB Software is used to realize image processing operations. The result of the algorithm is a binary mask image of blobs representing the detected objects. The background is updated slowly with Selective and Non-selective algorithm. The use of two updating blocks improves the sensitivity of the algorithm.

1.1. About MATLAB

MATLAB is software used to compute numerical functions. All the work related to image processing is first done using

MATLAB. Image processing is any form of signal processing for which the input is a digital image, such as a photograph or video frame. The output of image processing may be either an image or parameters related to the image. Unlike other software MATLAB does pixel by pixel calculation of the input image as per the algorithm. As image sizes grow larger, software becomes less useful in the video processing realm.

1.2. About SIMULINK

SIMULINK [8] is a utility of MATLAB [8] used to represent a process in the form of blocks. A block diagram is a representation of the process consisting of inputs, the system and its outputs. Rather than writing codes as done for MATLAB, blocks are simply placed and connected to make the complete system. In short it provides a graphical user interface (GUI) for building blocks, performing simulations and finally analyzing results. In this paper, SIMULINK is used for performing Image Processing operations and to obtain appropriate results. Also, a VHDL code of SIMULINK model can be generated using the HDL Encoder tool. This code can be used to program FPGA, ASIC or other equivalent ICs.

1.3. Algorithm

Each pixel is modified independently using the statistical procedure of Gaussian distribution [3] and the pixels of the moving object is detected using the inequality mentioned below:

$$|I_t - \mu_t| < k \cdot \sigma_t \quad (1)$$

Where μ_t and σ_t are mean and standard deviation matrices of Gaussian distribution for image pixel intensities and constant k typically has a value between 2 and 3. The updating background image is calculated as shown by the following equations:

$$\mu_t(x, y) = \alpha \cdot I_{t-1}(x, y) + (1 - \alpha) \cdot \mu_{t-1}(x, y) \quad (2)$$

$$\sigma_t^2(x, y) = \alpha [I_{t-1}(x, y) - \mu_{t-1}(x, y)]^2 + (1 - \alpha) \cdot \sigma_{t-1}^2(x, y) \quad (3)$$

Where I_{t-1} and μ_{t-1} is the intensity of previous image frame and previous frame and α is learning ratio.

2. Main Block

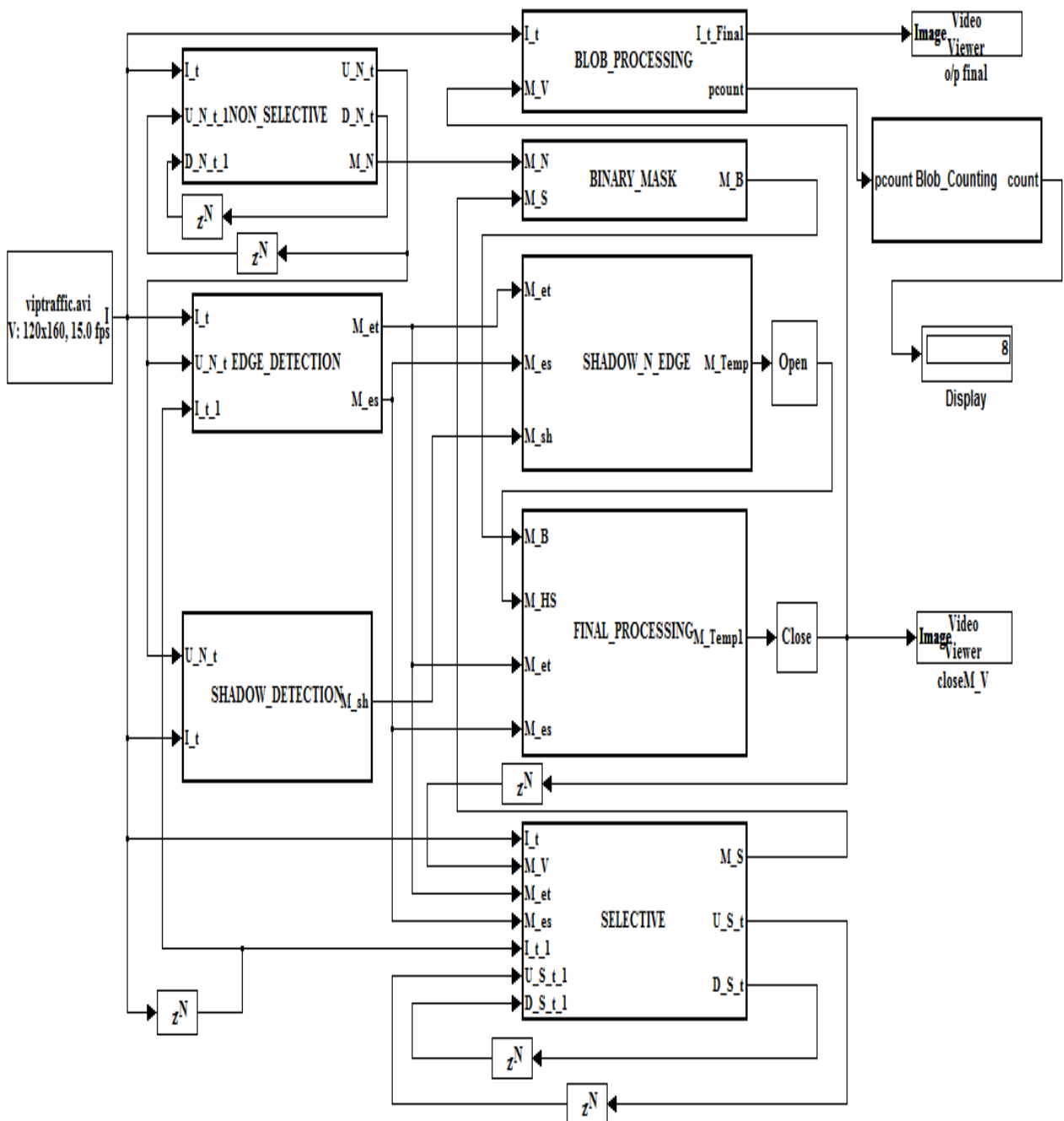


Figure 1: Main Block Diagram

3. NON-SELECTIVE BLOCK

The task of detecting the moving and recently stopped vehicles is done by the non-selective block. The equation for non-selective background updating [2] [4] is given by

$$\mu_{N,t}(x,y) = \begin{cases} \mu_{N,t-1}(x,y) + \delta_{N1} & \text{if } I_t(x,y) > \mu_{N,t-1}(x,y) \\ \mu_{N,t-1}(x,y) - \delta_{N1} & \text{if } I_t(x,y) < \mu_{N,t-1}(x,y) \\ \mu_{N,t-1}(x,y) & \text{otherwise} \end{cases} \quad (4)$$

Where $I_t(x,y)$ is the brightness of a pixel situated at coordinates (x,y) of input monochrome image at the time t ; $\mu_{N,t}(x,y)$ the brightness of a pixel situated at coordinates (x,y) of background image, updated non-selectively;

$\delta_{N1} = 2-5 = 0.03125$ is a small constant evaluated experimentally. It is assumed that the brightness of the input image is in the range: $I_t(x,y) \in \langle 0,255 \rangle$.

The updating of $\sigma_{N,t}$ which is σ_{τ} from (1) for non-selective model [2] is given by (5):

$$\sigma_{N,t}(x,y) = \begin{cases} \sigma_{N,t-1}(x,y) + \delta_{N1} & \text{if } |I_t - \mu_{N,t-1}| > \sigma_{N,t-1} \\ \sigma_{N,t-1}(x,y) - \delta_{N1} & \text{if } |I_t - \mu_{N,t-1}| < \sigma_{N,t-1} \\ \sigma_{N,t-1}(x,y) & \text{otherwise} \end{cases} \quad (5)$$

where δ_{N2} is experimentally evaluated to be 0.00390625 (i.e. 2^{-8}). The values of $\mu_{N,t}$ and $\sigma_{N,t}$ are used to calculate the pixel values of m_N from equation (1).



Figure 2: Showing $I_t(x,y)$ and $m_N(x,y)$ respectively.

4. SELECTIVE BLOCK

This block is similar to the non-selective block, but it depends on the final mask output [4] as shown by (6):

$$\mu_{S,t}(x,y) = \begin{cases} \mu_{S,t-1}(x,y) + \delta_{S1} & \text{if } I_{t-1}(x,y) > \mu_{S,t-1}(x,y) \\ & \text{and } m_{VTS,t-1}(x,y) = 0 \\ \mu_{S,t-1}(x,y) - \delta_{S1} & \text{if } I_{t-1}(x,y) < \mu_{S,t-1}(x,y) \\ & \text{and } m_{VTS,t-1}(x,y) = 0 \\ \mu_{S,t-1}(x,y) & \text{otherwise} \end{cases} \quad (6)$$

$$\sigma_{S,t}(x,y) = \begin{cases} \sigma_{S,t-1}(x,y) + \delta_{S2} & \text{if } |I_{t-1}(x,y) - \mu_{S,t-1}(x,y)| \\ & > \sigma_{S,t-1}(x,y) \text{ and } m_{VTS,t-1}(x,y) = 0 \\ \sigma_{S,t-1}(x,y) - \delta_{S2} & \text{if } |I_{t-1}(x,y) - \mu_{S,t-1}(x,y)| \\ & < \sigma_{S,t-1}(x,y) \text{ and } m_{VTS,t-1}(x,y) = 0 \\ \sigma_{S,t-1}(x,y) & \text{otherwise} \end{cases} \quad (7)$$

where $m_{VTS,t}(x,y) = m_{V,t}(x,y) \vee m_{ET,t}(x,y) \vee m_{ES,t}(x,y)$; $\mu_{S,t}(x,y)$ – the brightness of a pixel at coordinates (x,y) of background image updated using selectivity; $m_V(x,y)$ – the element of the detected vehicle mask image of value equal to 0 or 1, where 1 denotes the detected moving objects. The values of constants δ_{S1} and δ_{S2} were established experimentally: $\delta_{S1} = 0.25$, $\delta_{S2} = 0.03125$ for $I_{t-1}(x,y) \in \langle 0,255 \rangle$. From (1), the image frame m_s comes out as shown below:



Figure 3: Showing $m_s(x,y)$

5. BINARY MASK COMBINATION BLOCK

The output from both selective and non-selective block is combined into a single binary mask. A simple AND operation is not enough to detect all the pixels. Hence a special combination of AND and OR operations [5] are used to improve the detection as shown (8):

$$m_B(x,y) = \begin{cases} m_S(x,y) \vee m_N(x,y) & \text{if } ((m_S(x-1,y) \wedge m_N(x-1,y)) \\ & \vee (m_S(x-1,y-1) \wedge m_N(x-1,y-1)) \\ & \vee (m_S(x,y-1) \wedge m_N(x,y-1)) \\ & \vee (m_S(x+1,y-1) \wedge m_N(x+1,y-1))) \\ m_S(x,y) \wedge m_N(x,y) & \text{otherwise} \end{cases} \quad (8)$$



Figure 4: Showing $m_B(x,y)$

6. TEMPORAL AND SPATIAL EDGE DETECTION

During a dark scene, True Positive pixels [2] may not be detected and a major portion of the car may be neglected. So to improve the segmentation quality, the TP pixels need to be increased. This is done by using two type of edge detection: Temporal and Spatial Edge Detection blocks.

In temporal edge detection block, the edges are detected by taking the difference of the current and previous frame:

$$\Delta I_T = |I_t - I_{t-1}| \quad (9)$$

In spatial edge detection, the difference of current image and background is taken as below:

$$\Delta I_S = |I_t - \mu_{N,t}| \quad (10)$$

Temporal and spatial edge image mask m_{ET} and m_{ES} respectively is given by the following equation:

$$m_{ET}(x, y) = \begin{cases} 1 & \text{if } |\Delta I_T(x, y) - \Delta I_T(x - 1, y)| > \theta_{ET} \vee \\ & |\Delta I_T(x, y) - \Delta I_T(x, y - 1)| > \theta_{ET} \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

$$m_{ES}(x, y) = \begin{cases} 1 & \text{if } |\Delta I_S(x, y) - \Delta I_S(x - 1, y)| > \theta_{ES} \vee \\ & |\Delta I_S(x, y) - \Delta I_S(x, y - 1)| > \theta_{ES} \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

where θ_{ET} and θ_{ES} are constant threshold values.

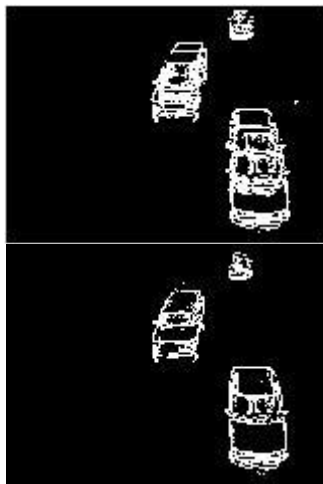


Figure 5: Showing m_{ET} and m_{ES} respectively.

7. SHADOW DETECTION

Shadows get detected and may be misunderstood by the system to be an object itself. Hence to remove any such discrepancies, the Shadow detection technique is used. By comparing the decrease in brightness [5] [6] [7], shadows are detected by the following equation:

$$m_{SH}(x, y) = \begin{cases} 1 & \text{if } \alpha \leq \frac{I_t(x, y)}{\mu_{N,t}(x, y)} \leq \beta \\ 0 & \text{otherwise} \end{cases} \quad (13)$$

where α and β are constant coefficients: $\alpha=0.55$, $\beta=0.95$, both evaluated experimentally.



Figure 6: Showing m_{SH} .

8. FINAL PROCESSING

The masks obtained by implementing the previous algorithms are combined into a single mask M_{Temp1} :

$$m_{HS} = dil(ero(\sim(m_{ET} \wedge m_{ES}) \wedge m_{SH})) \quad (14)$$

$$M_{Temp1} = ero(dil((m_B \wedge \sim m_{HS}) \vee (m_{ET} \wedge m_{ES}))) \quad (15)$$

where $dil()$ and $ero()$ denote 2×2 morphological dilation and erosion operation [2] respectively.

The mask thus obtained usually contains False negative and False positive pixels. Hence to improve the shape of the blobs, the inbuilt opening block is added to obtain the final frame of mask m_v .



Figure 7: Showing m_v .

9. BLOB PROCESSING

This block is used to process the information that is retrieved from afore proposed algorithm which can be used to calculate various elements such as area, eccentricity, centroid, orientation and the number of blobs detected. It is also used to form rectangle around the detected blobs which can be shown in the final image. These elements are of prime purpose to carry out the process of traffic surveillance.

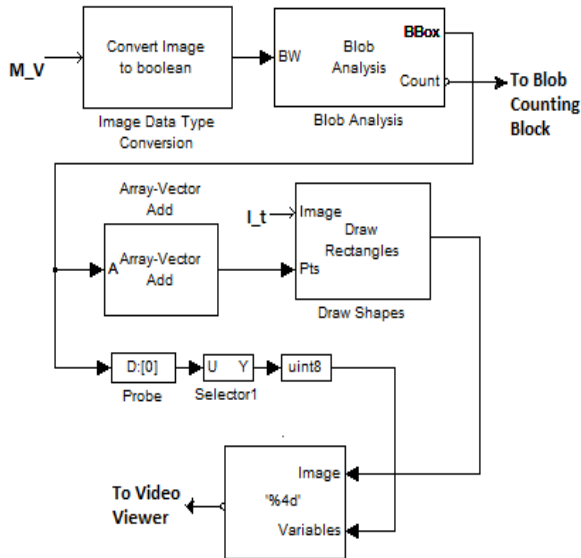


Figure 8: Inside the Blob Processing Block.



Figure 9: Image showing the detected objects.

10. BLOB COUNTING BLOCK

The Blob Processing block helps to find the number of vehicles visible per frame. But this number might not be particularly helpful for traffic surveillance. So an additional Embedded MATLAB function block is added to count the total number of cars that are passing. The result is displayed using the inbuilt display block as shown in figure 1.

11. CONCLUSION

The system realizes real-time motion segmentation and morphological operations. The important advantage of using SIMULINK is that a complete design of the system can be generated, visualized and analyzed as required. Also, a VHDL code of SIMULINK model can be generated using the HDL Encoder tool provided in SIMULINK. This code can be used to program FPGA, ASIC or other equivalent ICs. As a future scope, the output of the proposed algorithm can be used for automatic speed detection of vehicles and estimate direction of

motion. Number plate detection algorithm can be used along with the discussed algorithm to generate a log of the necessary information such as number plate, time of passing, speed of the vehicle, etc.

12. ACKNOWLEDGEMENTS

We are thankful to Dr. M. D. Patil, HOD(electronics) for their support and encouragement. We also want to thank college staff and family members for their constant support.

13. REFERENCES

- [1]. Sen-Ching S. Cheung and Chandrika Kamath, "Robust Background Subtraction With Foreground Validation For Urban Traffic Video" in Proc. SPIE Video Communication and Image Processing, 2004, pp.881-892.
- [2]. Marek Wójcikowski, Robert Zaglewski, "FPGA-Based Real-Time Implementation of Detection Algorithm for Automatic Traffic Surveillance Sensor Network".
- [3]. A. Franois, G. Medioni, "Adaptive Color Background Modeling for Real-Time Segmentation of Video Streams", in Proc. of Int. Conf. Imaging Science, Systems and Technology, Las Vegas, NV, 1999, pp. 227-232.
- [4]. F. Porikli, Y. Ivanov, T. Haga, "Robust abandoned object detection using dual foregrounds", EURASIP J. Adv. Signal Process, Jan.2008.
- [5]. D. Duque, H. Santos, P. Cortez, "Moving Object Detection Unaffected by Cast Shadows, Highlights and Ghosts", in Proc. IEEE Int. Conf. Image Processing, 2005, pp. 413-416.
- [6]. R. Cucchiara, C. Granna. M. Piccardi, A. Prati, S. Sirotti, "Improving Shadow Suppression in Moving Object Detection with HSV Color Information", in Proc. IEEE Intell. Transp. Syst. Conf., Oakland, CA, 2001, pp. 334-339.
- [7]. R. Cucchiara, C. Granna. M. Piccardi, A. Prati, "Detecting Moving Objects, Ghosts and Shadows in Video Streams", IEEE Trans. Pattern Anal. Machine Intell., vol. 25, no. 10, pp. 1337-1342, Oct. 2003.
- [8]. <http://www.mathworks.in>