Contemporary Engineering Sciences, Vol. 8, 2015, no. 6, 247 - 255
HIKARI Ltd, www.m-hikari.com
http://dx.doi.org/10.12988/ces.2015.5123

Observation of Vehicle-Related Events on Highways Using OpenCV

Nigel Field David Tariang

Computer Science and Engg., Sathyabama University, Chennai, India

P. Asha

Computer Science and Engg., Sathyabama University, Chennai, India

Abstract

Monitoring and observation of moving vehicles is beneficial for the analysis of traffic flow and cognition of other events. For identifying motion from the video feed, the system applies adaptive background subtraction using the OpenCV library for obtaining the threshold images containing blobs which represent moving vehicles. Applying certain operations on the blobs, the speed, class and number of vehicles can be procured.

Keywords: Vehicle monitoring, blob tracking, video analysis, adaptive background subtraction

1. Introduction

Modern highways are developing into areas of a high toll of vehicles with class as a variant. Intelligent Transportation System (ITS) are conducting operations to manage knowledge based on the events like the count of the vehicles, speed and type. The acquired data can be utilized for better traffic monitoring, prediction and prevention. These video-based solutions have received compelling attention from researchers [2] [3] for the past decade.

The detection of vehicles can be obtained by applying video-based operations whose feed is captured by a mounted camera over the highway. The concerned operation can be effected by various environmental conditions but other advantages like unruffled traffic flow, ease of installation and level of customization can be considered.

A Computer vision technique for identifying moving entities in a video feed is 'background subtraction', which computes the absolute difference between the background and the following frames. An effective technique is Otsu's thresholding mechanism [8]. The resulting frame is prompted for required operations. An adaptive background [4] subtraction mechanism is adopted for dealing with real-time conditions.

For real-time detection of vehicles, the foreground blobs [6] [7] obtained from the absolute background subtraction are monitored over the Region-of-interest (ROI). The ROIs can be initialized as the camera position is fixed. Problems arise when the shadows of the cars overlap with other cars which can be considered by the system as a single vehicle. The HSI shadow removal mechanism [10] [11] can be used to tackle the problem to produce a more robust system.

This paper is organized as the following. Section 2 describes in detail about the adaptive background subtraction and foreground extraction algorithm. Section 3 describes the tracking of the vehicles and Section 4 present the methods of observation of the speed, vehicle count and class categorization of vehicles. Section 5 gives the conclusion.

2. Adaptive Background Subtraction and Foreground Extraction

The adaptive background and foreground extraction operations are required for generating threshold images which contain the blobs of moving vehicles that are essential for conducting further procedures. Fig.1 describes the algorithmic procedure for obtaining the segmented vehicles.

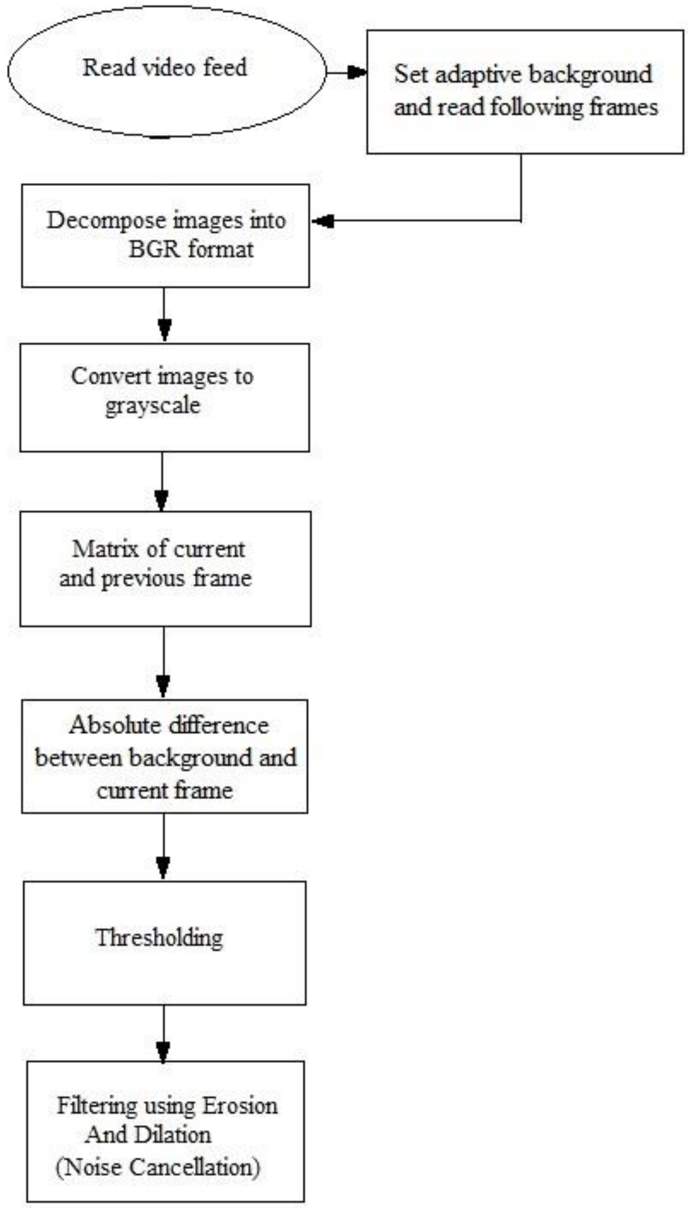


Fig. 1- Flowchart of the algorithm for obtaining blobs.

A. Adaptive background subtraction

Firstly, the algorithm reads the video feed from a mounted camera. Each frame of the video is decomposed into a BGR format which needs further processing. For the purpose of obtaining threshold images, frames in grayscale provide better results. The BGR format images are converted to grayscale for reaping robust 'difference' images. Converting the images to grayscale eliminates the possibility of obtaining unnecessary noise which could be attained if the images were to be in the BGR format. Following the conversion from BGR to grayscale is the assignment of the images' pixels to a matrix of pixels. The matrices for all images contain pixel values from the corresponding images. These values are the ones we use for the actual 'frame differencing' operation. Considering two matrices, one for the background and the other for the following frames, the 'absolute difference' function is called to attain the difference between the two frames, which is the resulting 'difference image'.

(a) Input frame

(b) Grayed background

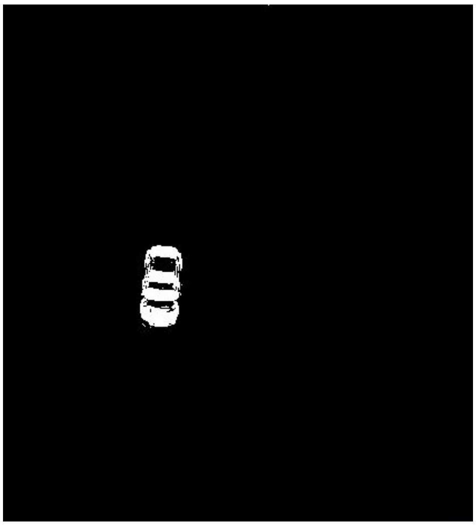
(c) Grayed input frame

(d) Difference image

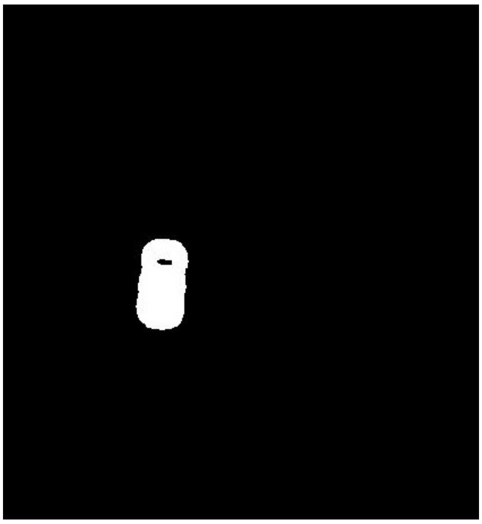
Fig.2 Adaptive background subtraction

B. Foreground extraction

For extraction of the foreground features, the difference image obtained from the background subtraction is utilized. The concerned image is usually not rich on the basis of blob filling and consists a substantial amount of holes. Firstly, the image is converted into a threshold or binary image whose pixel values are either '1' or '0' i.e., 'white' or 'black'. All the pixel values from the difference image which aren't registered as black are converted to white. This gives rise to a bi-level image which is easier to work with as the complexity of dealing with other color codes is diminished. Otsu's thresholding technique gives the best results for obtaining binary images as it is not affected by brightness and contrast. The noise level in the threshold images can be eliminated and the blob areas can be made richer by further processing. Erosion and Dilation are the Morphological operations to be used. To eliminate noise and achieve clearer boundaries for blobs, erosion is done on the threshold image to remove smaller blobs which are not desired and enrich the region of foreground pixels which are of interest. Dilation on the other hand, increases the size of the white pixels to fill up the blobs and reduce the size of holes that are present with the white blob. The resulting threshold image is one which has been enriched with white pixels over the regions of interest.



(e) Initial threshold image



(f) Final threshold image

Fig.3 Filtering foreground features

III. Vehicle Tracking

For the actual tracking of the vehicles, the frame which the algorithm works with is the final threshold image. Hence it is critical to obtain well defined binary images which incorporate the blobs in them without the presence of noise. The system proceeds to identify the blobs and store them in a vector of blobs which simulates an array of blobs. For each blob, the largest possible rectangle is drawn surrounding it's area which gives way to finding the centroid of the blob. The vector of blobs stores the area and the centroid of the blobs which also represent their position on the pixel-plane. Every vehicle that enters the video frames are considered as blobs and their area, along with their centroid (position) are updated constantly after every frame that has been processed. Every vehicle that leaves the pixel-plane is discarded from the vector.

Fig.4 Tracked vehicle

IV. Tracking the Speed, Count and Class of Vehicles

A. Speed tracking

Speed tracking and trapping can be implemented for security issues in cases of road accidents and for preventing them. The system does not require complex hardware for tracking the speed of a vehicle which is convenient in terms of cost. For performing speed tracking, the system has to initialize a certain amount of distance on the pixel plane which corresponds to the distance of the real-world coordinates. Two trigger points can be setup, one for initiating the timer when the centroid of the vehicle crosses it and one for stopping the timer when the centroid crosses it. The timer keeps track of the time that the vehicle takes to cross the two pre-determined points. This recorded time, along with the assumed distance is utilized to attain the speed of the vehicle by using the traditional 'distance/time' formula.

B. Vehicle counter

Keeping track of the vehicle count can be beneficial in the field of traffic prediction and controlling. Implementing the vehicle counter requires the centroid of the vehicle to cross a predetermined point on the pixel-plane. For each vehicle that crosses it, the vehicle count is incremented by one. With the increase in the number of vehicles each year, roads can tend to get crowded which can lead to traffic. Predicting the traffic conditions on a street can provide a better initiative towards monitoring the flow of vehicles. Table-1 shows the experimental results for the vehicle counter.

C. Categorization of vehicle class

With toll gates being setup on various highways, information on the type of vehicles crossing a highway can help in the research of profit prediction for upcoming toll gates as the prices for different classes varies. For classifying the type of vehicles, the system can obtain the areas of the vehicle-blobs and segregate them accordingly. The system keeps track of the count for each vehicle, whether they are two-wheelers, passenger vehicles, trucks or buses.

Table 1- Vehicle Counter

No. of input frames	**Actual No. of Vehicles**	**Vehicles Tracked**	**Accuracy Rate (%)**
70	2	2	100
320	7	7	100
470	13	13	100
850	27	28	96.42
1440	39	40	97.50
2050	46	48	95.83

V. Conclusion

With increasing demands in the ITS sector, a huge amount of potential and feasible applications can be designed for the monitoring of traffic flow, vehicle counting and classification.

Incorporation of different techniques together can provide a robust and working implementation of the paper. Using Otsu's thresholding technique, rich binary images are attainable. Blobs from the threshold image are processed to monitor the speed, count and class of vehicles. The quality of the results depend on the implementation of the system.

Acknowledgements. I would like to thank Mrs. P. Asha, Asst. Prof. of the Computer Science Department of Sathyabama University, Chennai for her guidance throughout this paper and constant support.

References

[1] Da Li; Bodong Liang; Weigang Zhang, "Real-time moving vehicle detection, tracking, and counting system implemented with OpenCV",4th IEEE International Conference on Information Science and Technology (ICIST), pp. 631 – 634, 2014. http://dx.doi.org/10.1109/icist.2014.6920557

[2] S. Sivaraman and M. Trivedi, "Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis", IEEE Transactions on Intelligent Transportation Systems, vol. 14, no. 4, pp. 1773-1795, 2013. http://dx.doi.org/10.1109/tits.2013.2266661

[3] Y. Liu, B. Tian, S. Chen, F. Zhu, and K. Wang, "A survey of vision-based vehicle detection and tracking techniques in ITS", 2013 IEEE International Conference on Vehicular Electronics and Safety (ICVES), pp. 72-77. http://dx.doi.org/10.1109/icves.2013.6619606

[4] M. Lei, D. Lefloch, P. Gouton, and K. Madani, "A Video-Based Real-Time Vehicle Counting System Using Adaptive Background Method", 2008 IEEE International Conference on Signal Image Technology and Internet Based Systems, pp. 523-528. http://dx.doi.org/10.1109/sitis.2008.91

[5] G. D. Sullivan, K. D. Baker, A. D. Worrall, C. I. Attwood, and P. M. Remagnino, "Model-based vehicle detection and classification using orthographic approximations", Image and Vision Computing, vol. 15, no. 8, pp. 649-654, 1997. http://dx.doi.org/10.1016/s0262-8856(97)00009-7

[6] S. Gupte, O. Masoud, R.F.K. Martin, and N.P. Papanikolopoulos, "Detection and classification of vehicles", IEEE Transactions on Intelligent Transportation Systems, vol. 3, no. 1, pp. 37-47, 2002. http://dx.doi.org/10.1109/6979.994794

[7] S. C. Lee and R. Nevatia, "Speed Performance Improvement of Vehicle Blob Tracking System", in Multimodal Technologies for Perception of Humans, Lecture Notes in Computer Science, vol. 4625, pp: 197-202, 2008. http://dx.doi.org/10.1007/978-3-540-68585-2_17

[8] N. Otsu, "A threshold selection method from gray-level histograms", IEEE Transactions on Systems, Man, and Cybernetics, vol. 9, no. 1, pp. 62-66, 1979. http://dx.doi.org/10.1109/tsmc.1979.4310076

[9] P. S. Liao, T. S. Chen, and P. C. Chung, "A Fast Algorithm for Multilevel Thresholding", Journal of Information Science and Engineering, vol. 17, no. 5, pp. 713-727, 2001.

[10] B. Sun, S. Li, "Moving Cast Shadow Detection of Vehicle Using Combined Color Models", 2010 Chinese Conference on Pattern Recognition (CCPR), pp. 1-5. http://dx.doi.org/10.1109/ccpr.2010.5659321

[11] A. Sanin, C. Sanderson, and B.C. Lovell, "Shadow detection: A survey and comparative evaluation of recent methods", Pattern Recognition, vol. 45, no. 4, pp. 1684-1695, April 2012. http://dx.doi.org/10.1016/j.patcog.2011.10.001

[12] OpenCV User Site: http://opencv.org/

[13] OpenCV on SourceForge: http://sourceforge.net/projects/opencvlibrary

Received: February 11, 2015; Published: March 5, 2015