Penerbit Akademia Baru

Journal of Advanced Research in Applied Mechanics

Journal homepage: www.akademiabaru.com/aram.html
ISSN: 2289-7895

Vehicle Speed Estimation using Image Processing

Open Access

Fatima Afifah[1,*], Sharmin Nasrin[1], Abdul Mukit[1]

[1] University of Asia Pacific, 74/A, Green Rd, Dhaka 1215, Bangladesh

ARTICLE INFO

Article history:
Received 5 October 2018
Received in revised form 4 December 2018
Accepted 8 December 2018
Available online 5 January 2019

ABSTRACT

This paper aims at determining vehicle speed which is necessary for traffic surveillance systems. These systems are very much useful to monitor and manage various traffic conditions such as traffic management, prevention of accident, also secure transportation. This paper describes a comprehensive approach to localizing target vehicles in video under various environmental conditions. The extracted geometry features from the video are continuously projected onto a profile and are constantly tracked. We rely on temporal information of features and their motion behaviors for vehicle identification, which compensates for the complexity in recognizing vehicle shapes, colors, and types. In speed determination vehicle detection and vehicle tracking are the key steps. To overcome the disadvantages of traditional methods, here, vehicle speed determination using image processing is done with Python language and OpenCV library. In this paper, the proposed method consists of few basic steps-— background subtractions, feature extraction, vehicle tracking etc. The speed is determined using distance travelled by vehicle over number of frames and frame rate.

Keywords:
Speed, background subtraction, openCV, morphological transformation, python

1. Introduction

Nowadays, with the continuous increase of vehicle in the road, traffic management authority requires better traffic surveillance system. With more vehicles, the number of accidents on the road rises up each year. Speed is now the single biggest cause of road accidents. Enforcing speed limit is one of the ways to eliminate speed related accidents. Traffic surveillance systems for vehicle detection and speed measurement play an important role in enforcing speed limits. They also provide relevant data for traffic control such as vehicle speed, traffic count etc. Those systems are divided in intrusive and non-intrusive sensors [1]. Intrusive sensors are usually based on inductive loop detectors. Although these sensors are used widely, they have complex installation and high maintenance, promotes asphalt deterioration and also can be damaged by wear and tear. It also requires line sight connection between vehicle and the equipment. Apart from this, due to high cost of equipment and less accuracy, it is losing its popularity. Non-intrusive sensors, which include laser

* *Corresponding author.*
E-mail address: afifah.fatima@uap-bd.edu (Fatima Afifah)

meters and doppler radars, avoid these problems, but are usually more expensive and require frequent maintenance. The work presented in this paper aims to overcome this challenging task of speed determination by providing an economic solution.

Due to the availability of cheaper cameras which are able to produce images with higher quality, video-based systems are by far the most inexpensive alternate for non-intrusive speed measurement. In fact, existing systems are often connected to video cameras [2] that record the license plates of vehicles that exceed the speed limit — thus; the infrastructure for such systems is already available in most cases. Image processing is a convenient technique to analyze videos and extract information. The next section has covered the work done previously by many researchers on use of image processing for speed determination.

2. Related Work and Contributions

Due to the hard work of many researchers and scientists, we now have various methods of determining moving vehicle speed. They are Inductive loops, RADAR gun, LASER Gun, Manual count [3]. But these intrusive and non-intrusive sensors are either too expensive and requires high maintenance or damages the already existing road. So to overcome the limitations in existing methods, several video-based approaches were proposed for estimating or measuring the speed of vehicles in roadways. Almost all the methods include a background/foreground segmentation step to detect image regions containing motion. The methods of background extraction from color image [4] based on average value, median filter [9-11] and common region. Few other common approaches for this task include simple frame differences [5-8], as well as statistic models based on Gaussian distributions [12,13] or other measures [14]. Image tracking is one of the steps to estimate vehicle speed. Some of the common tracking method involves image patches [12], edges [5], [9], blobs [10], corners [6,7]. Few researchers have successfully tracked vehicles based on the license plate region [15-17], or a combination of such features [11]. Some methods are sensitive to conditions such as shadows, perspective and illumination variation. Blob analysis [5,6,8,10,12,14] is one such method which can have high tracking error. Moreover, these methods produce satisfactory results only when the camera is positioned high above the roadway, with the blobs being tracked for many frames.

Many researchers have found ways to avoid the problem associated with blob analysis. Dogan *et al.*, [7], in his work directly tracked distinctive features using the Lucas-Kanade optical flow algorithm [18]. The disadvantage of using this algorithm is that it can handle only a single vehicle at a time. Moreover, they require a side view of the vehicles and do not take perspective into account. Equation of spherical projection is used in [19] to estimate the vehicle speed. Lucas-Kanade-Tomasi algorithm is used for motion tracking. Vehicle detection and tracking is one of the most important steps to estimate vehicle speed. The various algorithms which are developed for vehicle detection and tracking has reviewed in [20].

While using different methods and algorithms, vehicles might go undetected or wrong-fully detected. To remove misdetection of vehicle [21] due to vehicle travelling from other lane, or other small movements such as tree waving can be avoided using ROI extraction. The background is multiplied with ROI mask. So that vehicles are detected accurately. Furthermore thresholding and morphological operations are used to reduce noise. In thresholding selection of threshold value is based on various methods. The threshold value can be selected manually or automatically by using thresholding. The histogram thresholding is explained in [22]. The next section has covered the proposed method for determining vehicle speed.

3.Proposed Technique

This section details the main steps of the proposed technique. We use the basics such as background subtraction, thresholding and other morphological transformation to detect the vehicles. After that detecting contour and the center of it, we find the speed of each vehicle.

3.1 Pre-Processing

A camera has been used for recording vehicle movement. Before even beginning the image processing, it is vital to convert the video into the frames. In pre-processing the video has converted into the frames. At first, *cv2.VideoCapture* function is used where the video is passed as an argument. This helps load up the video for image processing. In OpenCV, the read function is used to get each frame. The various parameters such as number of frames, frame rate, frame size are extracted in pre-processing. There are a total of 1244 frames in this video. It has a frame rate 24 frames per second. The frame size is of 1920x1080 pixels, which has been retrieved using the shape function. The *get* function has been used to obtain these information but with different flags. The flags *cv2.CAP_PROP_FPS* and *cv2.CAP_PROP_FRAME_COUNT* are for frame rate and frame count respectively.

3.2 Perspective Transformation

Perspective transformation helps to reduce error in speed estimation. A video can be taken from any angle but it is essential to align it with the global coordinate. Since we can find the pixel covering the length and width of an image and pixel value for any angular dimension cannot be found, it is necessary to perform perspective transformation. The purpose of this is to transform each frame into a plain sheet so that position of any vehicle can be found from the pixel values. In OpenCV it is done with *cv2.warpPerspective* function. The result of perspective transformation is shown in Figure 1.

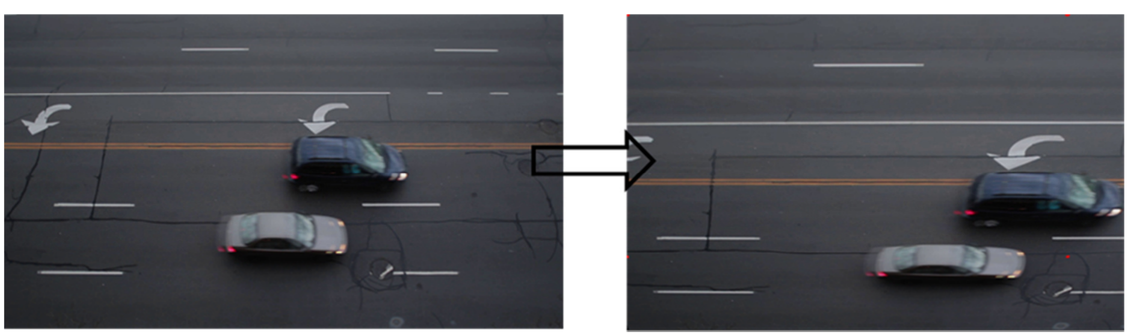

Fig. 1. Example of perspective transformation

3.3 Grayscale Image Generation

Generating grayscale image is a vital step towards detecting single or multiple vehicles. With grayscale image, not only speed of image processing can be improved but also unnecessary noise contributed by colored images can be avoided. Therefore, an RGB to grayscale conversion is performed on each video frame. Among several grayscale transformations, the simplest form of grayscale transformation has been used in this work. For color conversion we use the function *cv2.cvtColor (input_image, flag)* where flag determines the type of conversion. To convert to grayscale we use flag *cv2.COLOR_BGR2GRAY*. The result of grayscale image is shown in Figure 2.

Fig. 2. Example of grayscale image generation

3.4 Image Blurring

Image blurring is used for removing noise and edges, which is achieved by convolving the image with a low-pass filter kernel. Among all the filtering methods offered in OpenCV, Gaussian Filter has been used here. It uses a Gaussian function for calculating the transformation to apply to each pixel in the image.

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{1}$$

It is done with the function, *cv2.GaussianBlur (image, kernel size, sigmaX,sigmaY).* The width and height of the kernel should be positive and odd. The standard deviation in the X and Y directions are sigmaX and sigmaY respectively. For our proposal, *cv2.GaussianBlur (image, (5, 5), 0)* has been used.

3.5 Image Segmentation

Image segmentation is a major step in image processing to detect vehicles. It helps to extract the moving foreground from static background. For proposed method, a widely used technique for detecting moving objects from a video, called background subtraction has been used. . It provides good results in segmentation, and allows automating the process when foreground color of images is not constant, as well as speeds it up significantly. A Gaussian Mixture-based Background/Foreground Segmentation Algorithm, called *cv2.createBackgroundSubtractorMOG2* has been used in this work. As a part of background subtraction, average image scene is created by accumulating all images. The difference between the Current Frame and the Existing Average has been kept to further assist in vehicle detection.

3.6 Thresholding

Thresholding is one of the ways for image segmentation. It converts grey scale image to binary image. A binary image can be converted to any format. The function used is *cv2.threshold*. First argument is the source image, which should be a grayscale image. Second argument is the threshold value which is used to classify the pixel values. Third argument is the maxVal which represents the value to be given if pixel value is more than (sometimes less than) the threshold value. The result of image thresholding is shown in Figure 3.

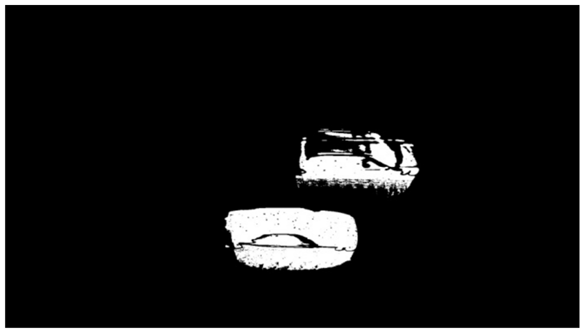

Fig. 3. Example of thesholding

3.7 Morphological Operations

They are generally used to remove noise from imperfect segmentation. Morphological operations are especially suited for binary images. So they are performed on output image of thresholding. Here opening, closing and dilation are performed. Opening and closing is used to remove holes in the detected foreground. Dilation is interaction of structuring element and foreground pixels. The structuring element is nothing but a small binary image. In the process of dilation the size and shape determination of structuring element is very important. Dilation is performed before thresholding. Opening and closing are done after thresholding. The results of these morphological operations are shown in Figure 4. After this, the selected object pixels are applied for connected component analysis.

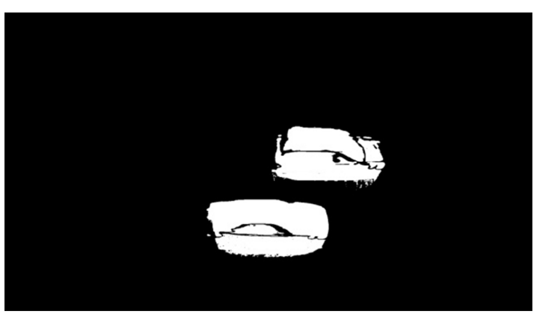

Fig. 4. Example of morphological operations

The vehicle detection using the technique can be shown by the following pipeline in Figure 5.

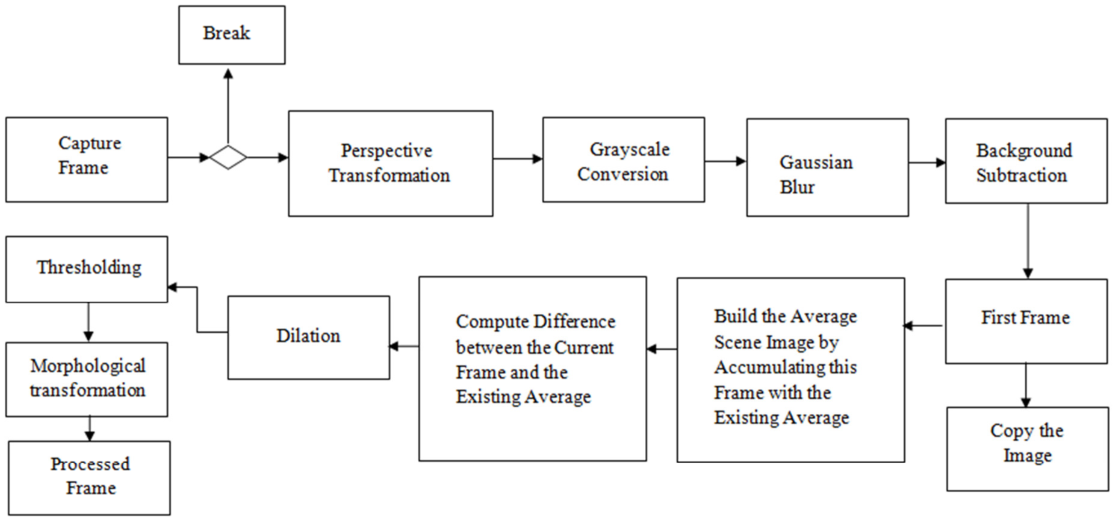

Fig. 5. Pipeline showing the steps towards detecting vehicles

3.8 Feature Extraction & Vehicle Tracking

Feature extraction is the key aspect in moving vehicle tracking. The more literature is available on various methods of feature extraction. Features are nothing but some of the characteristics of detected vehicle such as position, speed, color, shape, centroid, edges etc. the steps to feature extraction and vehicle tracking are given below.

(i) Firstly, contour is defined using the function *cv2.findcontour*. This function also helps to provide center for the contour area.

(ii) As the primary purpose of this work is to determine the vehicle speed, other objects like trees, pedestrians etc. need to be avoided while processing. This can be possible by defining a minimum area to be detected and contoured.

(iii) Vehicles are tracked from the distance of centers from one frame to another frame.

The results of vehicle tracking are shown in Figure 6.

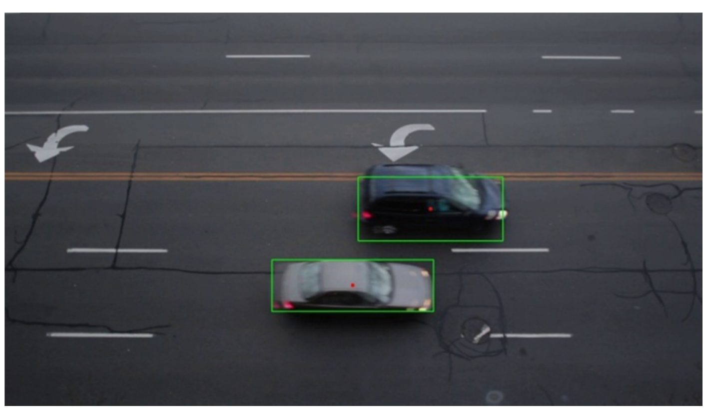

Fig. 6. Example of vehicle tracking

3.9 Speed Determination

Finally, once the vehicles are tracked, we can determine the speed of vehicles on the road. Vehicles are tracked from the distance of centers from one frame to another frame. The real world distance is needed to be mapped on the image. It is convenient to take the real world distance as the same length as the image width since image width can be found using OpenCV library. Euclidean distance of two consecutive center of a blob is taken as the pixel distance. Frame rate of the video can be found from OpenCV library. Total time taken for a vehicle to pass the screen can be found from dividing image width by frame rate.

Total time = Image width / Frame rate

Speed of the vehicle is then found from dividing the multiplication of real world distance and pixel distance by total time.

Speed = (Pixel distance * real world distance)/ Total time.

Finally, the speed detection using the technique can be shown using by the pipeline in Figure 7.

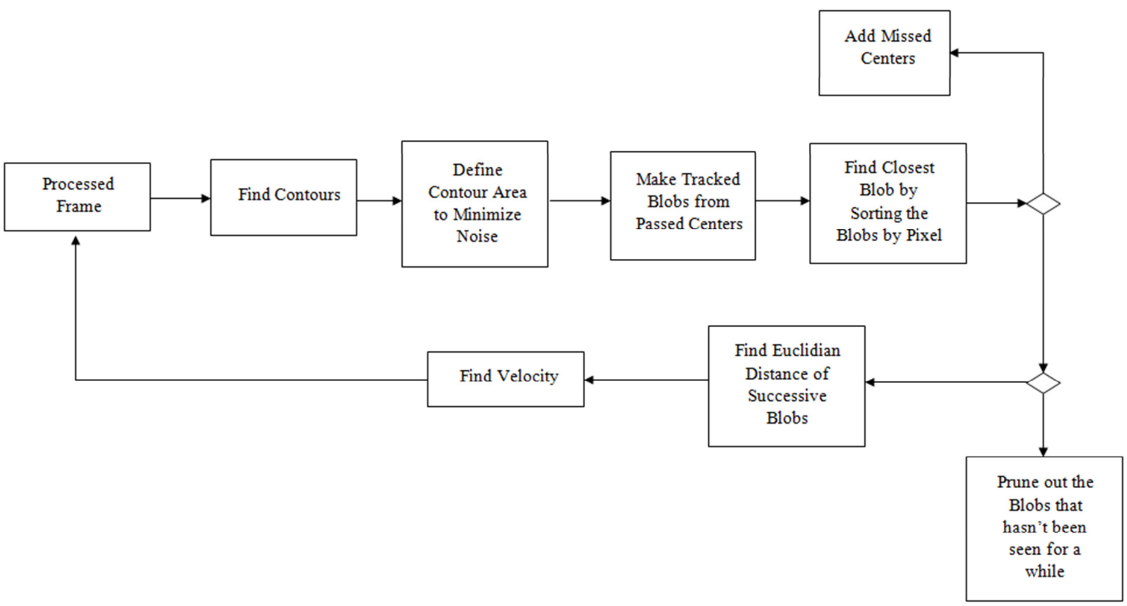

Fig. 7. Pipeline showing the steps towards determining speed of vehicles

References

[1] Mathew ,T., V.(2014). Intrusive and non-intrusive technologies, Indian Institute of Technology Bombay
[2] Buch, Norbert, Sergio A. Velastin, and James Orwell. "A review of computer vision techniques for the analysis of urban traffic." *IEEE Transactions on Intelligent Transportation Systems* 12, no. 3 (2011): 920-939.

[3] Adnan, Muhammad Akram, Norliana Sulaiman, Nor Izzah Zainuddin, and Tuan Badrul Hisyam Tuan Besar. "Vehicle speed measurement technique using various speed detection instrumentation." In *Business Engineering and Industrial Applications Colloquium (BEIAC), 2013 IEEE*, pp. 668-672. IEEE, 2013.

[4] Chen, Peijiang. "Moving object detection based on background extraction." In *Computer Network and Multimedia Technology, 2009. CNMT 2009. International Symposium on*, pp. 1-4. IEEE, 2009.

[5] Dailey, Daniel J., Fritz W. Cathey, and Suree Pumrin. "An algorithm to estimate mean traffic speed using uncalibrated cameras." *IEEE Transactions on Intelligent Transportation Systems* 1, no. 2 (2000): 98-107.

[6] Madasu, Vamsi Krishna, and Madasu Hanmandlu. "Estimation of vehicle speed by motion tracking on image sequences." In *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pp. 185-190. IEEE, 2010.

[7] Doğan, Sedat, Mahir Serhan Temiz, and Sıtkı Külür. "Real time speed estimation of moving vehicles from side view images from an uncalibrated video camera." *Sensors* 10, no. 5 (2010): 4805-4824.

[8] He, Xiao Chen, and Nelson HC Yung. "A novel algorithm for estimating vehicle speed from two consecutive images." In *Applications of Computer Vision, 2007. WACV'07. IEEE Workshop on*, pp. 12-12. IEEE, 2007.

[9] Zhiwei, He, Liu Yuanyuan, and Ye Xueyi. "Models of vehicle speeds measurement with a single camera." In *Computational Intelligence and Security Workshops, 2007. CISW 2007. International Conference on*, pp. 283-286. IEEE, 2007.

[10] Maduro, Cristina, Katherine Batista, Paulo Peixoto, and Jorge Batista. "Estimation of vehicle velocity and traffic intensity using rectified images." In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pp. 777-780. IEEE, 2008.

[11] Palaio, Hélio, Cristina Maduro, Katherine Batista, and Jorge Batista. "Ground plane velocity estimation embedding rectification on a particle filter multi-target tracking." In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pp. 825-830. IEEE, 2009.

[12] Grammatikopoulos, Lazaros, George Karras, and Elli Petsa. "Automatic estimation of vehicle speed from uncalibrated video sequences." In *Proceedings of International Symposium on Modern Technologies, Education and Profeesional Practice in Geodesy and Related Fields*, pp. 332-338. 2005.

[13] Wang, Weiqiang, Jie Yang, and Wen Gao. "Modeling background and segmenting moving objects from compressed video." *IEEE transactions on circuits and systems for video technology* 18, no. 5 (2008): 670-681.

[14] Schoepflin, Todd N., and Daniel J. Dailey. "Dynamic camera calibration of roadside traffic management cameras for vehicle speed estimation." *IEEE Transactions on Intelligent Transportation Systems* 4, no. 2 (2003): 90-98.

[15] Garibotto, Giovanni, Paolo Castello, E. D. Ninno, Pietro Pedrazzi, and Gianfranco Zan. "Speed-vision: Speed measurement by license plate reading and tracking." In *Proc. IEEE Int. Conf. Intell. Transp. Syst*, pp. 585-590. 2001.

[16] Czajewski, Witold, and Marcin Iwanowski. "Vision-based vehicle speed measurement method." In *International Conference on Computer Vision and Graphics*, pp. 308-315. Springer, Berlin, Heidelberg, 2010.

[17] Garg, Mayank, and Sahil Goel. "Real-time license plate recognition and speed estimation from video sequences." *ITSI Transactions on Electrical and Electronics Engineering* 1, no. 5 (2013): 1-4.

[18] Lucas, Bruce D., and Takeo Kanade. "An iterative image registration technique with an application to stereo vision." (1981): 674-679.

[19] Madasu, Vamsi Krishna, and Madasu Hanmandlu. "Estimation of vehicle speed by motion tracking on image sequences." In *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pp. 185-190. IEEE, 2010.

[20] Wang, Guolin, Deyun Xiao, and Jason Gu. "Review on vehicle detection based on video for traffic surveillance." In *Automation and Logistics, 2008. ICAL 2008. IEEE International Conference on*, pp. 2961-2966. IEEE, 2008.

[21] Huang, Chieh-Ling, and Heng-Ning Ma. "A Moving Object Detection Algorithm for Vehicle Localization." In *Genetic and Evolutionary Computing (ICGEC), 2012 Sixth International Conference on*, pp. 376-379. IEEE, 2012.

[22] Raju, P. Daniel Ratna, and G. Neelima. "Image segmentation by using histogram thresholding." *International Journal of Computer Science Engineering and Technology* 2, no. 1 (2012): 776-779.