COYOTE - Computer Vision Based Speed Camera

A.D.M.P Rathnayaka, B.A.S Induwara, M.P.S Udayangana, T.S Samarakoon and C.R. De Silva
Department of Computer Science and Engineering, University of Moratuwa, Sri Lanka.

Abstract - **This paper describes about a new way of vehicle speed detection which uses computer vision and image processing techniques. The paper proposes how we could use a real time video stream and calculate speeds of vehicles that are present in the video, by overcoming issues like partial and full occlusions. Also this paper contains descriptions of each module of the proposed system and their functionalities.**

I. INTRODUCTION

Traffic monitoring systems using computer vision techniques have taken a tremendous part in the area of researching. When it comes to speed detection, available mechanisms have significant issues, which are stated under problem description. This causes those systems to be unreliable, hence to be unusable due to lack of transparency.

Computer vision based techniques can be used to ensure required trustworthiness, accuracy and transparency of the process. This paper describes a computer vision based speed detection system which overcomes above mentioned drawbacks.

A. *Problem Description*

The problem that has been addressed in this paper is specifically focused on Sri Lanka. Present, vehicle speed detection mechanism in Sri Lanka is the use of "Radar Gun", and this mechanism has several significant drawbacks that cause this mechanism to be suspended.

Even though calibrated radar gun gives accurate speed readings, that can only detect one vehicle's speed at a time. This leads to a loose end where several vehicles can pass through without being detected when speed detection processes in operation. Another issue with the Radar Gun readings is that we cannot guarantee which vehicle's speed is detected by the radar gun, if there was two or more vehicles on the road at the time of speed detection.

This mechanism is lacking the transparency since there is no evidence that can be used to prove once a vehicle is detected as violating the speed limit. This has led the vehicle speed detection process to a questionable situation and hence, the high speed detection in Sri Lanka has been suspended and relevant low has been seized.

This paper proposes a new vehicle speed detection mechanism that uses computer vision and image processing techniques which is addressing all the above mentioned issues.

II. METHODOLOGY

This section presents the technology inputs and processing steps needed in the moving vehicle identification, tracking and speed calculation.

Figure 1 shows the key stages in the vehicle identification and tracking process. Each of these steps is described in detail below.

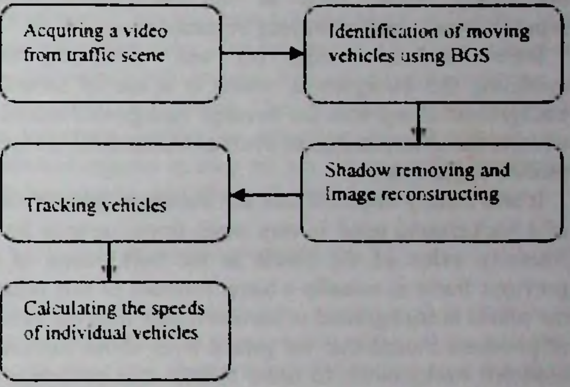

Fig. 1. Key stages of vehicle Identification and tracking process

A. *Acquiring a video of traffic scene*

Camera with a high frame rate and a high picture quality should be used to acquire the video of the traffic to ensure the required accuracy level of results is met. Placement of the camera should be done in a way the following requirements are met. Visible range of the camera must include at least 75 meters of a flat roadside where the vehicles that are being monitored travel on. Height of the camera from the ground level should be 3-4 meters.

B. *Identification of moving vehicles*

Moving object identification is the first stage of this system. Generally, there are three types of methods [1] optical flow method, temporal difference method and background subtraction method. Optical flow [2-3] methods aim at computing an approximation of the 2D motion filed by means of spatio-temporal changes of image intensity. They can be used in the presence of camera motion, but most optical flow methods are computationally complex, and it cannot be applied to full-frame video streams in real-time without specialized hardware. In contrast, temporal difference [4-5] method takes into account differences between two consecutive frames. This method is strictly dependent on the velocity of moving object and it is subject to the foreground aperture problem. Comparatively to above two methods background subtraction method is more effective and practical. Modeling background is crucial in this method. Generating and maintaining of the reference background is done using image sequence itself. Straightforward background models include average background model and median background model. This paper proposes a model that uses mean background and also a secondary background model where the immediate previous image frame with respect to the current frame being analyzed.

What is promising of this method is it can effectively detect objects in the scene background which are not completely static, such as the background include repetitive motion like swaying vegetation.

Here an improvement [6] was introduced when modeling the background, which is a use of secondary background along with the average background model, to address the shortcomings of average background model for motion segmentation.

It was clearly observed that the change of intensity level of a background pixel is very small from frame to frame. Intensity value of the pixels in the background of the previous frame is actually a better estimate of that value of the pixels in background of current frame, than the average of previous frames that we gained from above mentioned modeled background. In order to take this advantage we use another model of the background called Secondary background where the actual intensity level of pixels that are classified as background in the previous frame. By using a method that uses combination of these two models, a great segmentation of the moving objects is obtained. The average model is used to identify moving regions in a frame and the Secondary background model is used to identify the valid moving object within these regions. By this way, major drawback in the average model based method; the "Tailing Effect" of identified moving regions can be avoided.

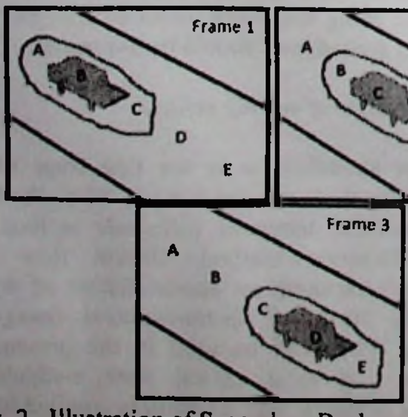

Fig. 2. Illustration of Secondary Background Concept

This secondary background concept is illustrated in the Fig. 2. The oval shaped region around the car is the moving region that is identified using the average background model. In the third frame we can differentiate between the car region (D) and the background regions (C and E) that are all identified as moving regions. This is done by using secondary background for region E from frame 2, region C from frame 1 and region D from frame 2.

After the segmentation, the background model is updated by calculating the average over N frames. The Secondary Background is updated by storing the intensity levels of pixels that were classified as background as the new Secondary Background. For pixels which were classified as foreground, the Secondary Background is left unaltered so that it holds the value of intensity from the frame where the pixel was last classified as background.

Secondary Background for frame k is denoted by sbg^k. For initialization, sbg is simply set to the background calculated for the First frame:

$$sbg^1 = bg^1(y, x) \qquad (1)$$

sbg^1 is the initial value for Secondary Background.

The segmented output based on the Secondary Background is denoted by:

$$seg2^k(y,x) = \begin{cases} 1, & if\ (|sbg^k(y,x) - I^k(y,x)| > T2\ and\ seg1^k(y,x) = 1) \\ 0, & Otherwise \end{cases} \qquad (2)$$

T2 is a threshold. The Secondary Background is updated for the next frame:

$$sbg^{k+1}(y,x) = \begin{cases} I^k, & if\ seg2^k(y,x) = 0 \\ sbg^k, & Otherwise \end{cases} \qquad (3)$$

As explained in above equations, the segmentation and background modeling processes are interdependent. Secondary background model is fed back with the segmentation results.

The improved background that uses the Secondary Background model does not exhibit the error merging of vehicles together and forming a continuous stream in segmented foreground.

C. *Shadow removing and Image reconstruction*

Shadow removing and Image reconstruction modules are introduced to enhance the quality of identified moving objects. Image Dilate and Erode [7] methods are used in Image reconstruction.

1) *Color and texture-based shadow detection*

A shadow is normally an area that is partially irradiated or illuminated because of the interception of radiation by an opaque object between the area and the source of radiation [8]. Assuming that the irradiation consists only of white light, the chromaticity in a shadowed region should be the same as when it is directly illuminated. This applies same to lightened areas in the image. Based on this assumption, a normalized chromatic color space,

$$r = R/(R + G + B),\ g = G/(R + G + B) \qquad (4)$$

Keeping lightness information is important in order to avoid errors such as ambiguous situations with a white car and the gray road. Another important issue is that we are only interested in detecting shadows that form part of the foreground objects. Shadows that form part of the background are not a problem as they don.t have to be tracked. Specifically, a shadow removal algorithm needs to analyze foreground pixels and detect those that have similar chromaticity but lower brightness to the corresponding region when it is directly illuminated. The adaptive background reference image provides the needed information.

i. Color-based detection

Based on the fact that both brightness and chromaticity are very important, a good distortion measure between foreground and background pixels has to be decomposed into its brightness and chromaticity components. Brightness distortion (BD) can be defined as a scalar value that brings expected background close to the observed chromaticity line. Similarly, color distortion (CD) can be defined as the orthogonal distance between the expected color and the observed chromaticity line. Both measures are shown in Fig. 3 and formulated bellow.

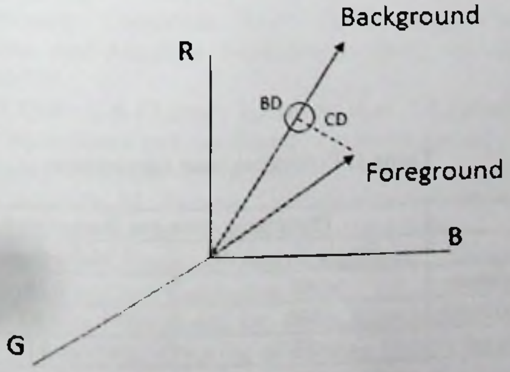

Fig. 3. Brightness and Color distortions

\

$$BD = \arg_a \min (Foreground - a \cdot Background)^2$$

(5)

$$CD = \| Foreground - a \cdot Background \|$$

Brightness distortion values over 1.0 correspond to lighter foreground. On the other hand, the foreground is darker when *BD* is below 1.0. The brightness distortion can be easily obtained by computing the derivative of the first expression, *i.e.*

$$BD = \frac{Foreground \cdot Background}{Background^2} \qquad (6)$$

Finally, a set of thresholds can be defined to assist the classification into foreground, highlighted or shadowed pixel.

It is still possible to achieve more precise results by normalizing variations in color bands increasing computational cost. But since our system has many other parts with high computational costs, we don't use the normalized color space for shadow removing. Many other approaches [8] are also based on the same underlying idea of decomposing color and brightness. Our reconstruction process doesn't rely on any particular implementation so any approach can be used. The last thing to mention is that the technique fulfills its objective not to remove self shadowed regions as they do not share similar brightness and chromaticity with the background reference image.

2) Foreground reconstruction

The cast shadow/highlights removal algorithm is a destructive process in a sense that, despite the assertion process described above, original object shapes are more likely to be distorted and some pixels will remain misclassified.

Mathematical morphology theory can be employed in order to reconstruct the original image without cast shadow or highlights. Mathematical morphology reconstruction filter uses an image called "marker" image as a mark to rebuild an object inside in an original image called "mask" image. In our case the "marker" image is a binary image where a pixel is set at "1" when it corresponds to a foreground, not cast shadow/highlight pixel. On the other hand, the "mask" is also a binary image where a "1" pixel can correspond to a foreground pixel, or cast shadow/highlight pixel, or speckle noise.

It is highly desirable that the "marker" image, ***K***, contains only real foreground object pixels, *i.e.*, not any shadow/highlight pixels so that those regions will not be reconstructed. Therefore, the use of very aggressive thresholds is necessary in the foregoing color-based removal process to assure that all the shadow/highlight pixels are removed.

A noise removal filter is also applied to suppress isolated noisy foreground pixels that remain and obtain a good quality "marker" image, ***K***. We used Gaussian smooth algorithm as our noise removal.

$$K = M \cap (Gaussian(M, r)) \qquad (7)$$

M is the binary image generated after shadow removal and assertion process; r denotes the smoothing radius which is using in Gaussian method. As a result of this, only the regions not affected by noise which are clearly free of shadows/highlights are subject to the shape reconstruction process shown as following expression.

$$R = L \cap (K \oplus SE) \qquad (8)$$

L is the mask, I the marker, $\oplus$ defined the dilation operation and SE the structuring element whose size usually depends on the size of the objects of interest, although a 5 x 5 square element proved to work in all our tests. Basically this process consists of a dilation of the "marker" image, followed by the intersection with the "mask" image. The underlying idea is that the shadow removed blobs keep at least a number of points that have been robust to erroneous shadow removal. These robust points are appropriate for leading the reconstruction of neighboring points as long as they form part of the silhouette in the original blob.

D. *Tracking vehicles*

There are four major aspects of object tracking: 3D model based tracking, Region based tracking, Active contour based tracking, and feature based tracking [9].

3D model based tracking requires predefined models of vehicles which makes that unpractical and active contour based has advantage over region based tracking in reduced complexity. But it is unable to segment vehicles that are partially occluded.

We implemented Feature based tracking method that goes along with the background subtraction. This method first extracts best feature points of each blob- identified foreground moving objects, in a given image frame and then find and match extracted features in successive image frame in the video. Then those features are grouped and compared with the blobs in the successive image frame. Most satisfactory blob to a given feature group is mapped to the previous blob which was mapped to the same group of features. This reduces region of the frame that must be analyzed and hence performance is improved which is crucial in real time processing.

E. *Speed calculation*

Speed calculation from a video requires mapping of real world coordination to 2D image plane and calibrating of length information. Calibration of the system require user to enter the region of interest from the visible scenery of the camera and specific length information with respect to pixels. Garibotte et al. [10] estimated individual vehicle speeds and read license plates. They placed the camera at a specific position and orientation to track and read the license plates as they passed by.

Another approach et al. [11] uses trained data to calibrate the system for speed detection. Here system is fed by several filmed motion at the same road length for some car drives with known speeds. Then form the estimation curve that allows estimating a speed of any vehicle that moves within calibrated road segment. But the constraints and assumptions used by our algorithms are vastly different from both of above approaches, because of the positioning of our cameras.

We proposed the system with a specific calibration system that meets the requirements of our constraints and assumptions. Here we used 2D Homography [12] to transform image coordinates to the real world coordination system. Actually here we create matrix to transform one coordination system to the other. So we can easily transform image coordinates to the real world coordination system. Using the transformation matrix we can get the real distance between two pixel points which we have been tracked on several frames. Then we can obtain the speed simply by dividing the real distance by the time difference between those frames.

III. RESULTS AND DISCUSSIONS

Vehicle speed detection for the purpose of traffic speed law enforcement is currently achieved using RADAR based methods. But that method has been questioned for its transparency and accuracy. In this work, we have proposed a novel approach for vehicle speed detection and identification using computer vision techniques. This ensures reliability, accuracy and most importantly the transparency of the speed detection process.

Performance of the system is a major concern since this is a real time application. Hence we introduced pipelining and multi threading to the system. Following table compares the performance by means of execution time per frame.

Table 1: Processing time comparison

	Processing time per frame (ms)	
	Single threaded	Multi threaded
Minimum	44	28
Maximum	108	102
Average	73	31

According to the observation that we have made, overall performance of the system has increased nearly two folds.

Following graph shows the summarized information of results of field experiment tests that are carried out to evaluate the system and its accuracy level. The graph shows detected speeds of vehicles using this system and compares those with actual speed values.

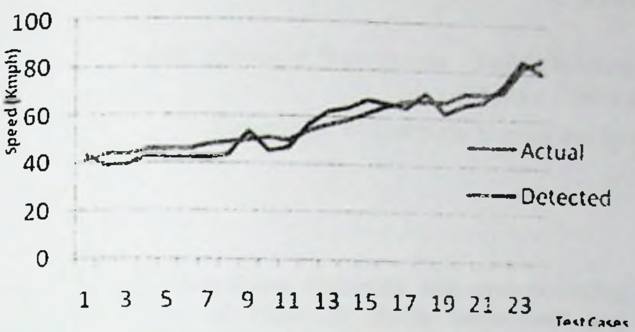

Fig. 4. Calculated speed using Coyote system and actual speed of the vehicle for several test cases

From the statistics of test results, we can conclude that the system detect the speeds with a 4 kmph error range which is an acceptable range in detecting high speeding vehicles.

References

[1] H. Wang, M. Ren, J. Yang , "Object tracking based on Genetic algorithm and Kalman filter" in *International Conference on Computational Intelligence and Security,*2008

[2] A.Talukder, L.Matthies. Real-time detection of moving object from moving vehicles using dense stereo and optical flow. *IEEE Conference on Intelligent Robots and Systems*, 2004.

[3] L.Wixson. Detecting Sailient" Motion by Accumulating Directionally Consistent flow". *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2000, vol. 22(8), pp.774-780.

[4]R.T.Collins, A.J.Lipton, T.Kanade, et al.," A system for video surveillance and monitoring. Technical Report", Carnegie Mellon University, 2000

[5]R.Cucchiara, M. Piccardi, "Vehicle detection under day and night illumination". *Proceedings of the Third International Symposia on intelligent industrial Automation and Soft Computing*, 1999.

[6] Dr. Donna Haverkamp, Dr. Arvin Agah and Dr. James Miller, "Automatic Tracking of Moving Objects in Video for Surveillance Applications", Department of Electrical Engineering & Computer Science and the Faculty of the Graduate School of the University of Kansas, July 2007

[7] Erode and Dilate, Available: http://www.sidefx.com/docs/houdini9.1/nodes/cop2/dilateerode, [Accessed: Dec, 2009]

[8] A Elgamal, R Duraiswami, D. Harwood and L. Davis, "Background and Foreground Modeling Using Nonparametric Kernel Density Estimation for Visual Surveillance". Proceedings of the IEEE, Vol. 90, No. 7, 2002

[9] B Coifman, D Beymer, P McLauchlan, J Malik, "Real Time Computer vision system for Vehicle tracking and Traffic Surveillance", 1998

[10] G. Garibotto, P. Castello, E. Del Ninno, P. Pedrazzi, G. Zan, "Speed-vision: speed measurement by license plate reading and tracking," in *IEEE 5th Int. Conf. Intelligent Transportation Sys.* pp. 585-590, 2002.

[11] E. Atkoci, R. Blake, A. Juozapavi, M. Kazimianec, "Image Processing in Road Traffic Analysis"

[12] Gary Bradski and Adrian Kaehler, *Learning OpenCV*,Sebastopol: O'Reilly Media,2008