Proceedings of the Technical Sessions, 24 (2008) 6-12
Institute of Physics – Sri Lanka


# Estimation of the Speeds of Moving Vehicles from Video Sequences

L.G.C Wimalaratna[1] and D.U.J. Sonnadara[2]
[1]*Virtusa (Pvt.) Ltd., Trans Asia Commercial Complex, Colombo 2*
[2]*Department of Physics, University of Colombo, Colombo 3*

**ABSTRACT**

In this paper, the results of a pilot project carried out to estimate the speeds of moving vehicles from video sequences are presented. The software pre-processes video images using gray scale and median filters. To extract moving vehicles from a traffic scene, the difference between three consecutive video frames were taken and filtered through an edge detection filter. Blob counting technique was applied to identify the position and size of the moving vehicles in the images. To estimate the trajectories of moving vehicles, the predicted area technique was applied to the blobs. Preliminary tests carried out on local roads show that the individual vehicular speeds can be estimated to an accuracy of better than ±3 km/h.

## 1. INTRODUCTION

Application of computer vision paves the way for automatic identification of vehicles from a traffic scene since it allows the identification of individual vehicles and extract the parameters related to each vehicle such as location, registration number, color, type etc. simultaneously. The additional parameters that can be extracted from an automatic monitoring system such as traffic flow rate, queue length, vehicle count and speed of vehicles are very useful for traffic control systems. In general, the information that can be extracted through an automated system can be used effectively to control the flow of traffic and provide security.

Although a number of research studies have been conducted in many countries in the area of automatic vehicle identification [1-4], to our knowledge there are no research studies conducted in Sri Lanka in this area. However, the use of vehicles, especially in the metropolitan areas has increased dramatically in recent years. Road traffic congestion has become a major issue in populated cites such as Colombo. Security issues has created further burden on the commuters.

The main objective of this pilot project was to detect moving vehicles in a traffic scene, track the detected vehicles and estimate their speeds. The traffic analyzer software developed under this project uses video frames of a traffic scene captured in a local road as the input and estimate the speed of individual vehicles and display the path taken by the vehicles in a predefined region.

## 2. METHODOLOGY

This section presents the technology inputs and processing steps needed in the transition from raw video frames to moving vehicle identification, tracking and speed extraction. Figure 1 shows the key stages in the vehicle identification and tracking process. Each of these steps is described in detail below.

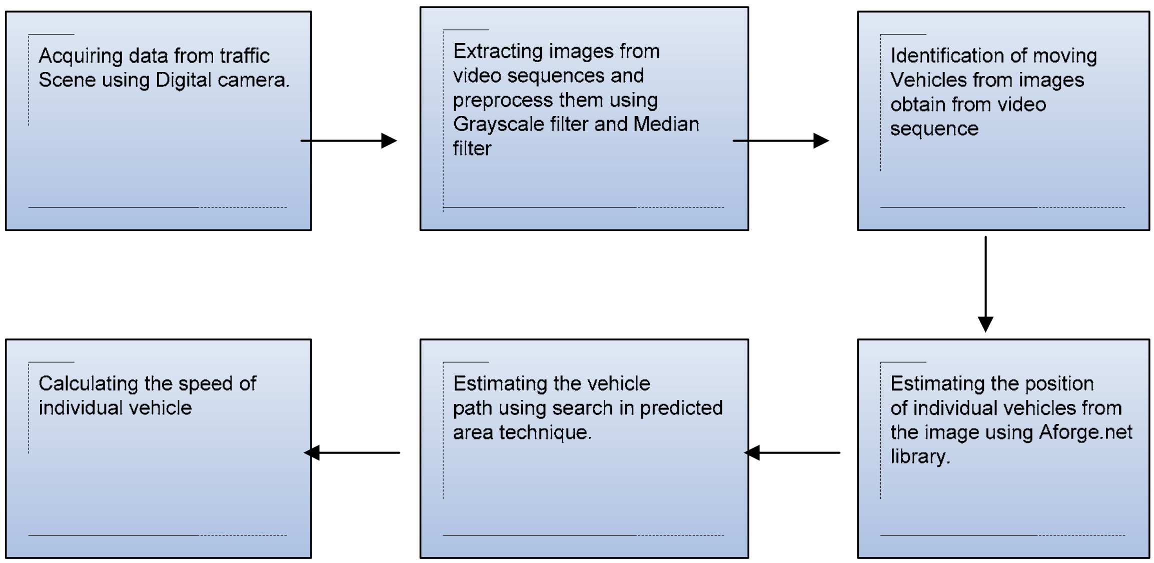


**Figure 1:** Key stages of vehicle identification and tracking process

### 2.1 Acquiring data from traffic scenes

Fuji FinePix 5500 camera mounted on a tripod was used to collect the data. The camera was placed and trial video images were taken at different distances from traffic flow line in several locations with different viewing angles having varying traffic flow conditions. Many of the locations had the common problem of overlapping vehicles in the camera view. Especially when the camera was placed on the side of the road perpendicular to the traffic flow line, vehicles tend to overlap in the video image. Solution to this problem is to place the camera down tilted on the traffic route. Overhead bridges were found to be the ideal places for the camera placement. It was also decided to select a location with less traffic density for the preliminary work.

Before taking video images from the selected location, the entrance zone and the exit zone were marked. Several identification marks were also placed at 10 m distance from each other parallel to the entrance and exit zones. The frame speed of the camera was limited to 30 images per second. Image resolution was 640x480 pixels.

### 2.2 Extracting images from video source

Initially, SC Video Decompiler [5] was used to extract images from the video files. Since automation of extraction method was needed, the JockerSoft.Media.dll [6] was used. This library allows extracting frames from a video source simply by giving the position of the image to be extracted.

After extracting images from the video source, they were converted to grayscale images. In gray scale images each pixel consists of an 8 bit number that represents a value ranging from 0 to 255; 0 values representing the Black and 255 representing the White. Aforge.net imaging library [7] was used to convert RGB images to grayscale images.

In image processing, it is recommended to perform high degree of noise reduction at the initial stages. Thus, a median filter was applied at the image preprocessing stage. The median filter is a non-linear digital filtering technique, often used to remove noise from images and other signals.

### 2.3 Identification of moving vehicles

To identify moving vehicles from a traffic scene, there are two popular techniques. One is the background subtraction technique. However, this technique requires background image to be updated frequently. Since this is not a desirable feature, the consecutive image subtraction technique was used in this work.

In the consecutive image subtraction method, initially three consecutive images are input to the system. Difference images are taken by processing these consecutive image frames. Figure 2 shows three consecutive images extracted from the video source separated by 1/30 sec from each other. These three images are already preprocessed by the grayscale and median filters.

(a) (b) (c)

**Figure 2:** Three consecutive images of a traffic scene

The difference between two images can be calculated by finding the difference between pixels in each image and generating a new image based on the result. For this technique to work, the two images must be aligned so that corresponding points coincide, and their photometric values must be made compatible, either by careful calibration, or by post-processing.

To extract the moving vehicles from the difference images, edge detection was carried out. In this work, Sobel edge detection operator was chosen since it gave better results than other edge detection operators such as Canny and Roberts during the preliminary tests were carried out using Matlab.

In the edge detected images, in addition to the moving vehicles, noise was present. This noise was created due to moving trees and other changes in the environment. By intersecting the two edged images, the extra noise can be removed.

Output images were then binarized and subjected to dilation and erosion [8]. Binary images are images whose pixels have only two possible intensity values. They are normally displayed as black and white images. Numerically, the two values are 0 for black and 1 for white. The basic effect of the dilation operator on a binary image is to gradually enlarge the boundaries of regions of foreground pixels (i.e. white pixels). Thus, areas of foreground pixels grow in size while holes within those regions become smaller. The dilation operator takes two pieces of data as inputs. The first is the image which is to be dilated. The second is a set of coordinate points known as a structuring element (also known as a kernel). The structuring element determines the precise effect of the dilation on the input image. The effect of the erosion operator on a binary image is to erode away the boundaries of regions of foreground pixels (i.e. white pixels). Thus areas of foreground pixels shrink in size, and holes within those areas become larger.

### 2.4 Identifying vehicle positions from images

First, a simple technique based on projections was used to estimate the vehicle positions (blob positions). In this technique image projections were taken in $X$ direction and $Y$ direction. Peaks in the histograms indicate the positions of the blobs and the corresponding pixel position can be used as the $X$ or $Y$ coordinate of the blobs. However, this is not a good technique to identify the vehicle positions from images due to overlapping when several blobs are present in the image.

Thus, Aforge.net library was used to estimate the position of the vehicles. Aforge.net library use an algorithm called connected component labeling algorithm [7] to estimate the position of the blobs (see Figure 3).

**Figure 3:** Identified vehicles using connected component algorithm

Connected components labeling scans an image and groups its pixel into components based on pixel connectivity, i.e. all pixels in a connected component share similar pixel intensity values and are in some way connected with each other. Once all groups are determined, each component was labeled with a color code.

### 2.5 Tracking and estimating the vehicle path

After identifying moving vehicles from the traffic scene, positions of the individual blobs were identified and a recording table was created. The number of rows in the recording table represents the number of blobs found in a given image. Columns represent parameters pertaining to each of the identified blobs. When the vehicular identification and tracking is in progress, the blobs are processed regularly and match with the parameters in the table. Thus the path of individual vehicles can be tracked separately. By recording the $X$ and $Y$ positions, trajectory of the individual vehicles can be traced.

It should be noted that the vehicles are tracked only if they are in a predefined region of the image. Due to the depth, the predefined region is a trapezoid along the road. As shown in the figure 4, contours of individual vehicle are enclosed in boxes and path of vehicles are drawn along the trapezoid. The tracking area is also shown in the processed image.

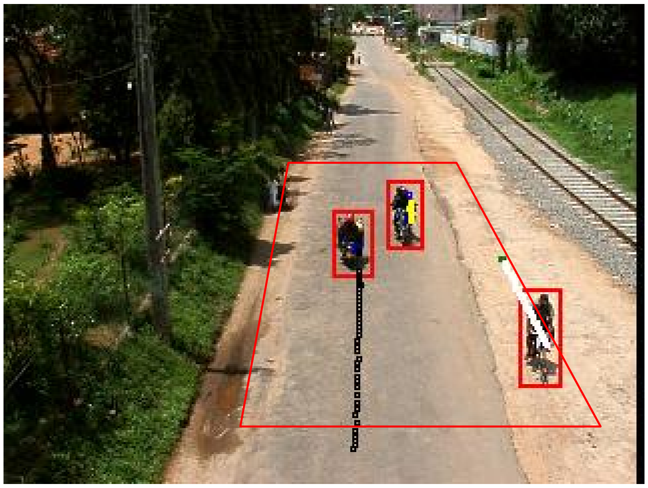

**Figure 4:** Tracking of individual vehicles

## 3. PERFORMANCE

To study the performance of the tracking algorithm and to test the developed software, a separate experimental data set was collected. The data was collected in daytime using the same camera setup that was described earlier. A vehicle with known speeds from 10 km/h to 50 km/h was used and video frames were recorded together with the speed settings. The software estimated the speeds to within ±3 km/h of actual speeds.

The developed pilot system had its own drawbacks too. When big vehicles such as buses and lorries were in the traffic scene, often they were not identified clearly as expected. There is a tendency to identify long vehicles as several vehicles.

It was also noted that, noise generated by moving trees, shadows of trees and moving pedestrian's effect the accuracy of the system. Shadows of trees and motion of trees may be misidentified as vehicles. Pedestrians are also identified by the system as moving objects and tend to track them.

When two vehicles travel in close proximity to each other (partially hidden in the camera view), software tends to identify these as a single vehicle. It was also noted that when the vehicles are at the far-end of the tracking region (other end) they are not identified and tracked accurately.

## 4. CONCLUSIONS

Results of a pilot study carried out to identify, track and estimate speed of individual vehicles from video source was presented. The developed Traffic Analyzer software package identified and tracked individual vehicles with a reasonable performance (within ±3 km/h) for most of the cases encountered in this work. The performance of the system can be further improved to develop a robust system that can be used as an aid to manage traffic related problems in busy roadways.

**Acknowledgements:** Financial assistance given by the National Science Foundation, Sri Lanka for the research grant number RG/07/E/03 is greatly acknowledged.

## REFERENCES

1. D.J. Dailey, F.W. Cathey and S. Pumrin, An Algorithm to Estimate Mean Traffic Speed Using Uncalibrated Cameras, IEEE Transactions on Intelligent Transportation Systems, 1-2 (2000) 98-107
2. E. Atkociunas, R. Blake, A. Juozapavicius, M. Kazimianec, Image Processing in Road Traffic Analysis, Nonlinear Analysis: Modelling and Control, 10-4 (2005) 315–332
3. B. Coifman, D. Beymer, P. McLauchlan and J. Malik, A Real-Time Computer Vision System for Vehicle Tracking and Traffic Surveillance, Transportation Research Part C: Emerging Technologies, 6-4 (1998) 271-288
4. D.M. Ha, J.M. Lee and Y.D. Kim, Neural-edge-based vehicle detection and traffic parameter extraction, Image and Vision Computing, 22-11 (2004) 899-907
5. Softpedia, http://www.softpedia.com/get/Multimedia/Video/Other-VIDEO-Tools/
6. Image Processing Toolbox User's Guide, http://www.mathworks.com/access/
7. AForge.NET: Image processing library, http://code.google.com/p/aforge/
8. Hypermedia Image Processing Reference, http://www.cee.hw.ac.uk/hipr/html/