Vehicle Speed Detection in Video frames using Corner detection

Kiran Kumar KV[#1], Pallavi Chandrakant[*2], Santosh Kumar[#3], Kushal KJ[#4]

[#]*Electronics & Communication Dept.,*

Peoples Education Society Institute of Technology (BSC)
Hosur Road, Bangalore-560100, INDIA

kvkiran@pes.edu

***Abstract*— vehicle speed detection is used to estimate the velocity of the moving vehicle using image and video processing techniques. Without any camera calibrations video is captured and analyzed for speed in real time. By employing frame subtraction and masking techniques, moving vehicles are segmented out. Speed is calculated using the time taken between frames and segmented object traversed in that frames. Finally frame masking is used to differentiate between one or more vehicles. With an average error of +/- 2 km/h speed detection was achieved for different video sequences.**

***Keywords*— Speed detection, Video processing, Computer vision, Frames subtraction, Edge detection, Image segmentation, Corner Detection, Frame masking.**

I. INTRODUCTION

With the increase in urban population in many cities, amounts of vehicles have also been drastically increased. In a recent study over-speeding caused most of the accidents, followed by drunken driving. Over-speeding of two-wheelers and three-wheelers is one of the major reasons of accidents. In order to support traffic management system in our country we need to build economical traffic monitoring systems. In recent times image and video processing has been applied to the field of traffic management system. This paper explicitly concentrates on the speed of the vehicles, which is one of the important parameters to make roads safe.

Relatively few efforts have been attempted to measure speed by using video images from uncalibrated cameras. Similarly, several other papers suggest estimating speed by first placing two detection lines (separated by a known distance) and then measuring travel times between the lines. This paper provides a low cost and versatile vehicle speed detection using a computer vision based approach. In this setting, the speed is detected using video cameras commonly available.

II. LITERATURE SURVEY

One of the technologies our law enforcement department uses to measure the speed of a moving vehicle is Doppler radar. It beams a radio wave at a vehicle, and then estimate the vehicles speed by measuring change in reflected wave frequency. It is a fixed or hand-held device and is reliable when a moving object is in the field of view and no other moving objects are nearby. Cosine error has to be taken care if the gun is not in the line of sight. Also Radio interference which causes errors in speed detection has to be taken care.

Some of the previous works using image and video processing applied for vehicle detection and speed measurements are vehicle detection based on frame difference, calibrated camera, motion trajectories, Optics and digital aerial images.

Also, blurred images were used to find out the vehicle speed along with high-end camera motion detection for automated speed measurements and feature point tracking for vehicle speed measurements were used.

Currently highly reliable GPS systems are used to track vehicle speeds in US. Cost-effectiveness is a concern in such a case.

In our method moving vehicle video from any video camera or mobile source is utilized. The algorithms are implemented in 'C' language using OpenCV and Visual Studio. Later this code can be ported to a simple processor where vehicle speed can be measured. Example: a simple smart phone with average processing capacity. Our aim was to implement real- time vehicle speed detector.

III. VIDEO SEGMENTATION

A video signal is the term used to describe any sequence of time varying images. A still image is a spatial distribution of intensities that remain constant with time while a time varying image has a spatial intensity distribution that varies with time. Videos can be in various formats based on the different cameras or mobile phones used. The video format used is an AVI file with the extension .avi.

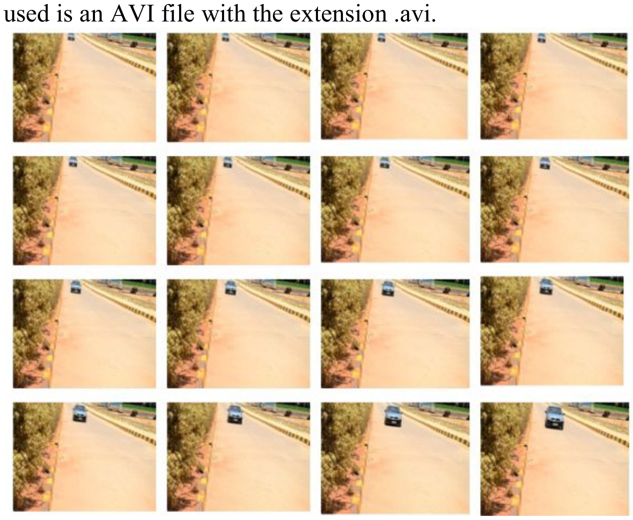

Figure1. Frames extracted from the video.

978-0-7695-5100-5/13 $31.00 © 2013 IEEE
DOI 10.1109/ICSIP.2014.46

This video is converted into frames. Since the video had 30 frames per second, extracting all the frames would lead to unwanted redundancy and these increases the delay time to execute the program. Hence we sampled the frames so that we get 3 frames per second, which serves the need.
Further above reference frames which are consecutive in nature are selected and converted into grayscale. Conversion to grayscale further reduces the amount of computation.

Next steps include preprocessing, moving edge detection, morphological operations, edge detection, vehicle segmentation and corner detection.

A. Pre-processing

The traffic images have a noise component from several interference sources. The Types of noise include the following:

1) Salt-and-pepper noise, which occurs when an image is coded and transmitted over a noisy channel or degraded by electrical sensor noise, as in video cameras.

2) Convolution noise (blurring), which produces images that are degraded by lens misfocus, motion, or atmospheric turbulence, such as adverse weather conditions.

Both noise sources contribute to high-frequency noise components. In our process, median filtering is used to reduce this high-frequency noise. It preserves the edge information required by our algorithm. Edges are a key image feature, as they remain prominent despite the variations in the traffic scenes and ambient lighting.

B. Background Subtraction

A reference frame is required for subtraction is in some ways similar to calibration. Here calibration is with the input image and not with the camera. Reference frame is used to remove the background of the image which is out of our interest.

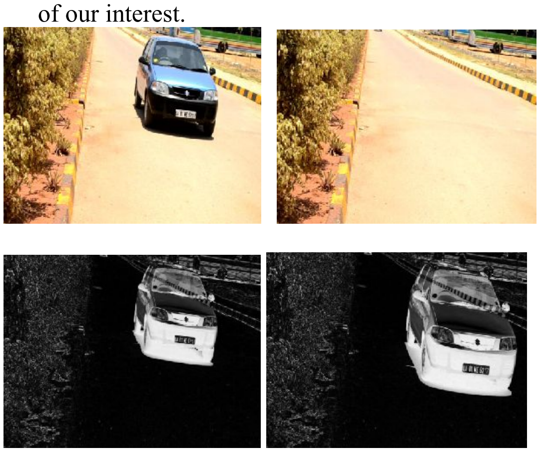

Figure 2. Left to Right and Top to Bottom: a. Frame 600, B. Frame 00, C & D. Frames Numbering 600 and 675 are subtracted from Frame 000 to subtract background.

As shown in figure2. a frame is subtracted from a reference frame which has only the background. Finally we get only the part which has moved. Since the leaves are also moving further processing of the image is done to get region of interest using masks.

C. Edge Detection

Since there is a background which contains the moving leaves. Further pre-processing and post-processing steps are involved in edge detection. Those steps are
1. *Smoothing*: Blurring of the image to remove noise.
2. *Gradients:* The edges should be marked where the gradients of the image has large magnitudes.
3. *Non-Maximum Suppression:* Only local maxima should be marked as edges.
4. *Double Theresholding*: Potential edges are determined by thresholding.
5. *Edge tracking by Hysteresis:* Final edges are determined by suppressing all edges that are not connected to a very certain (strong) edge.

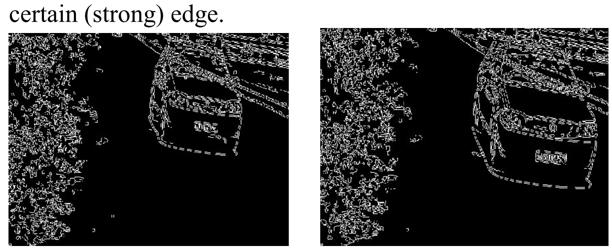

Figure 4. Canny Edge Detected Images.

D. Morphological Operation

In order to remove gaps obtained along the edges, we need to enhance the moving edges. This enhancement uses the morphological operator's dilation and erosion with an appropriate structural element. The result of sequentially applying dilation (first) and erosion is to remove specific image features smaller than the structural element without affecting the large features of interest. The structural element used is a line of size 8*1.

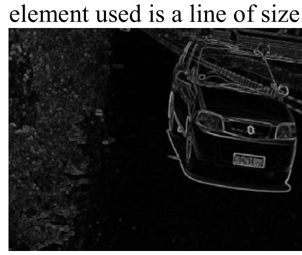

Figure 3: Morphed Image

IV. VEHICLE SEGMENTATION

A. *Auto Masking*

Due to heavy background and static interference, vehicle segmentation is implemented using simple masking techniques. The masks vary with respect to the background, Based on the correlation between pixels in the neighbourhood and by thresholding, three different masks are implemented has shown in the figure 5. Masks can be obtained after the edge detected image. Finally logical ANDing operation with image is performed to obtain the segmented image of the vehicle has shown in the figure 6.Similarly masking can also be used to differentiate between multiple vehicles in different lanes.

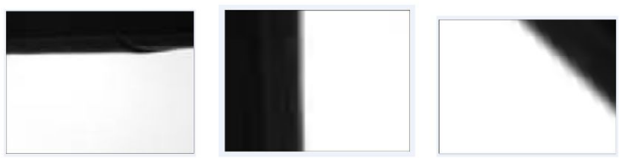

Figure 5. Masks obtained from Edge Images.

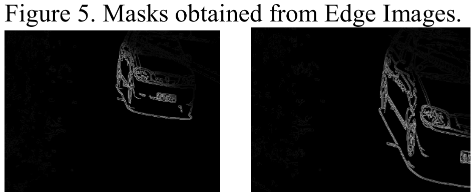

Figure 6. Masked frames.

V. SPEED DETECTION

After ANDing, the background is not completely removed. But some or the other background noise will be involved and edge detection leads to wrong answers. Hence we apply corner detection in order to find the corners from the vehicle. Coners are detected as shown in the figure 7.

Corner Detection

There are a significant number of different approaches for detecting corners; however we use Moravec operator for better speed of operation (mainly), detection rate, localization and robustness to noise. The operator considers a local window in the image and determines the average change of intensity resulting from shifting the window by a small amount in various directions. This operation is repeated for each pixel position which is assigned an *interest value* equal to the *minimum* change produced by these shifts. Points of interest are the local maximum of the interest values. Since corners exhibit a large intensity variation in every direction, this operator is a corner detector - although, with a more relaxed definition for corner. Moravec implemented this approach by computing the unnormalized local **autocorrelation** in the 4 principle directions, which results in an anisotropic response.

Harris improved upon Moravec's corner detector by considering the differential of the corner score with respect to direction directly, instead of using shifted patches.
Let this image be given by **I(u,v),** consider taking an image patch over the area and shifting it by **(x,y).** The weighted *sum of squared differences* (SSD) between these two patches, denoted **S**, is given by:

$$S(x,y) = \sum_u \sum_v w(u,v)\,(I(u+x,v+y) - I(u,v))^2$$

Outputs of the corner detection are as shown in the figure 6.

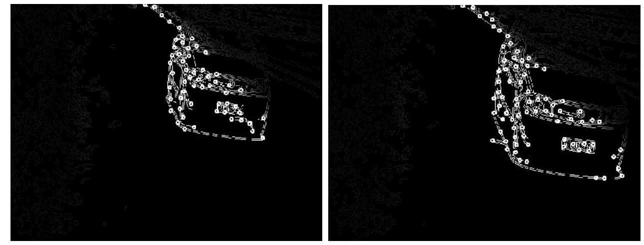

Figure 7: Corner Detected masked output Images.

A. *Maximum Corner.*

Once we get all the corners in the frame as shown before we must use a technique to find the distance moved by the vehicle. This distance measured here is, the pixel distance. To find the pixel distance we should find a unique way of getting the same corner coordinates of the vehicle in both the frames. There are many ways to recognize the same corner in different frames irrespective of their positions. So we found the y-coordinate of the maximum corner present in the frame. Now the maximum corners in both the frames are the same point or corner of the vehicle and we know the x and y coordinates. Some of the other methods are finding the corner with maximum/minimum x or y coordinates, finding the centroid of the vehicle which is obtained by drawing a rectangular Bounding Box. The next step is to find the distance (dp) and then speed.

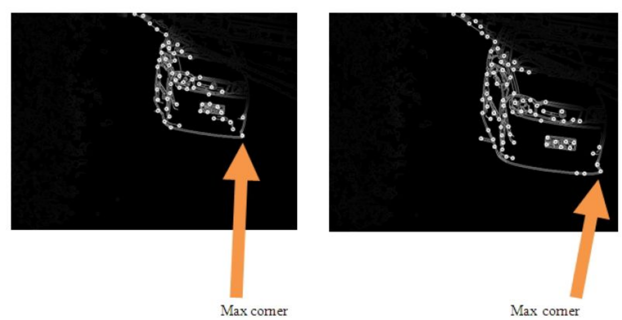

Figure7. Max Corner detected image.

The Distance formula is used to calculate distance between the two corners. Distance is equal to

$$\mathbf{dp} = \sqrt{(X1 - X2)^2 + (y1 - y2)^2}$$

Where
X1- x co-ordinates of corner in frame 1
X2- x co-ordinates of corner in frame 2
y1- y co-ordinates of corner in frame 1
y2- y co-ordinates of corner in frame 2

The speed of the vehicle can be calculated using the value of distance calculated above

Speed = (dp * c * 5) / (18 *0.344)
Where **dp**- distance calculated, **c**- sampling constant
5/18- conversion factor to represent speed in kmph from mps and **0.344**- sampling rate,. The values of sampling constant (c) for the varying ranges of pixels are for
Y coordinate < 135 then c=1.46
135 < Y coordinate < 205 then c=0.65
205 < Y coordinate < 260 then c=0.47
260 < Y coordinate < 390 then c=0.26
Else c=0.2

VI. ADVANTAGES AND DISADVANTAGES

Advantages

1. Since we are using C-language and less computational algorithms, speed detection can be implemented on high-end smart phone.
2. The cost of vehicle speed detection is very less when compared to the present day technology of radar guns and speed cameras used by the police department which costs a lot.
3. Distance and angle covered by this paper is more when compared to radar gun.

Disadvantages

1. Cloudy days results in wrong answers.
2. High end processors like 1.2 GHz core2duo are required for real time analysis.
3. Efficiency goes down to multiple vehicles. Focus should be on a single vehicle.

VII. FLOW CHART

The flow chart is has shown in figure 8B does selected frame extraction. The video given at the start contains 30 frames /sec. Out of which every 10 frame is selected for speed measurement. In addition to it reference frame is selected for initial algorithm calibration. Since it is faster to work on gray scale images rather than color images, color frames are converted into gray scale images. Initial image wont contain the vehicle as shown in the figure 8A. This can be presumed to be initial calibration for the given Image.

Figure 8A. Input frame from video.

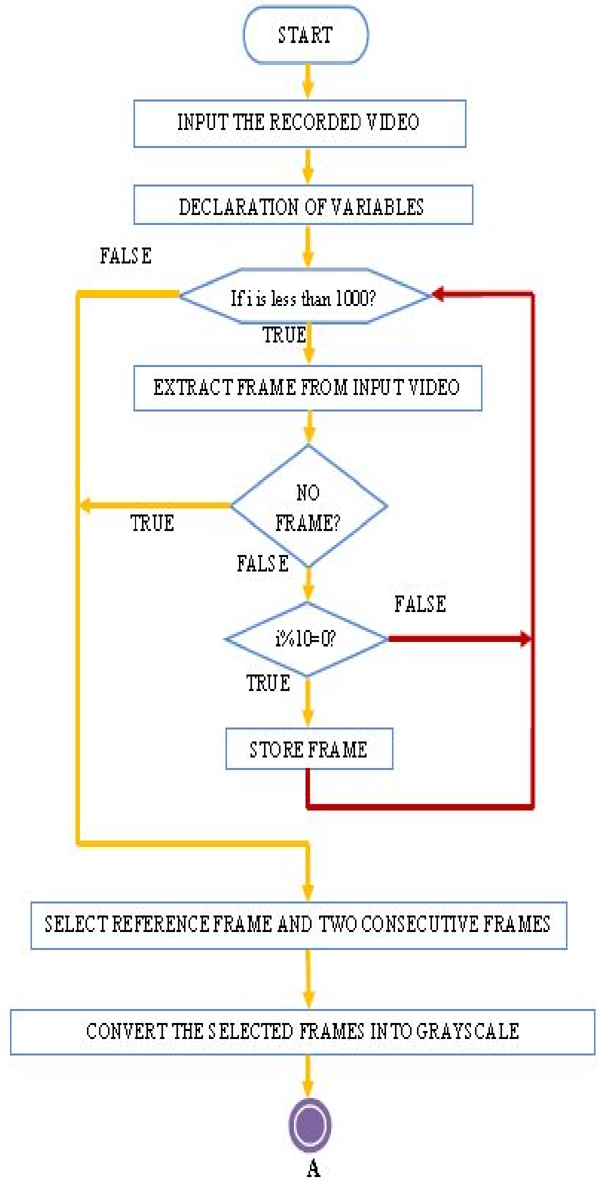

Figure 8B: Flow chart for Frame Extraction

The flow chart in figure 9A gives the flow about the algorithms discussed, to give the speed of the vehicle. Corner detection helps

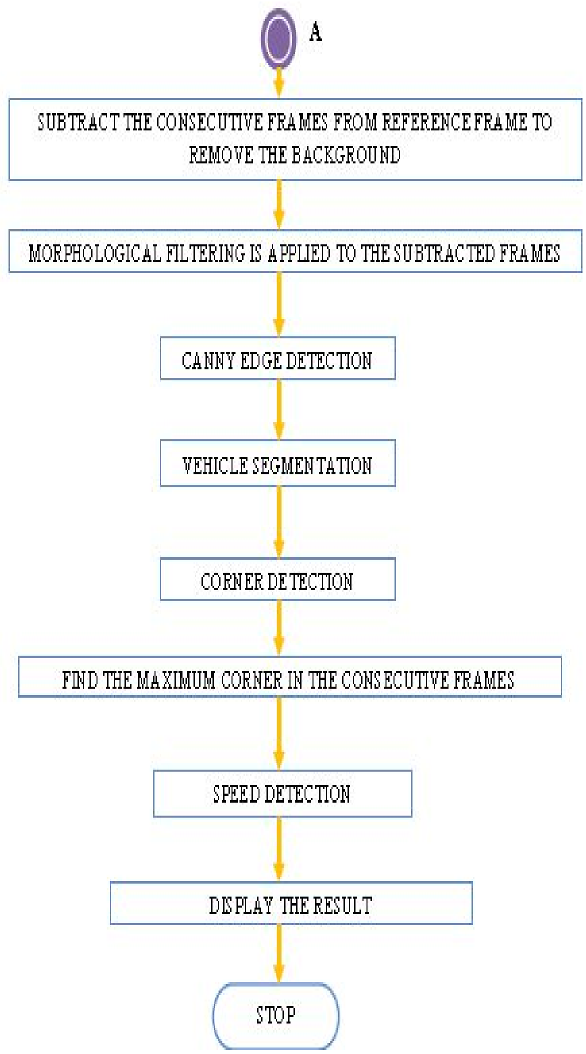

Figure 9A: Vehicle segmentation and Vehicle Speed Detection

VIII. RESULTS

Results displayed below are for a single vehicle travelling in finite speed. Figure 9A includes a frame from video. Figure 9B displays the visual studio with output dos screen. Figure 9C and figure 9D displays the output DOS screen, with speed of 46 kmph and 19 kmph.

Code is ran on a dual core processor of 1.2 GHz, 2GB RAM. OpenCV library is used Visual Studio compiler. OpenCV is a open source software.

Project gives the correct result under normal conditions and cloudy conditions. But fails if shadows creep up during sunset and sunrise.

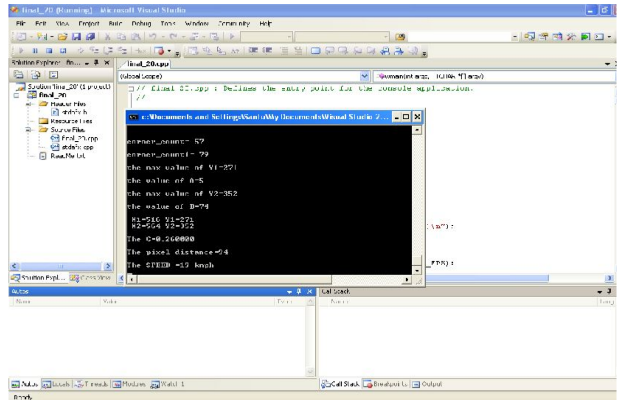

Figure 9B: Visual Studio Window.

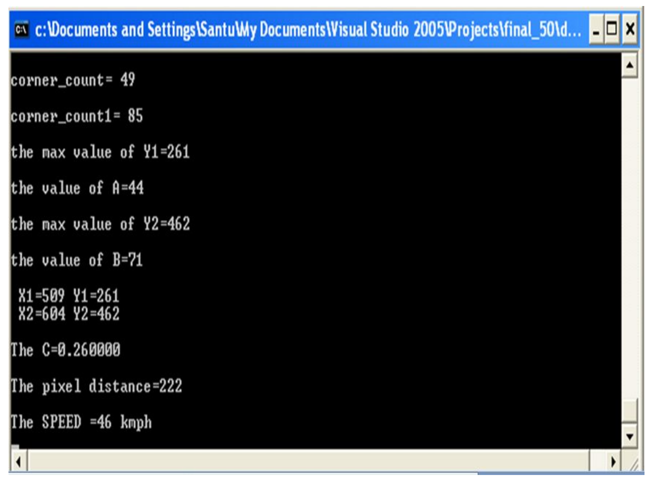

Figure 9C: Result displayed for a pre-recorded video of speed 46 kmph.

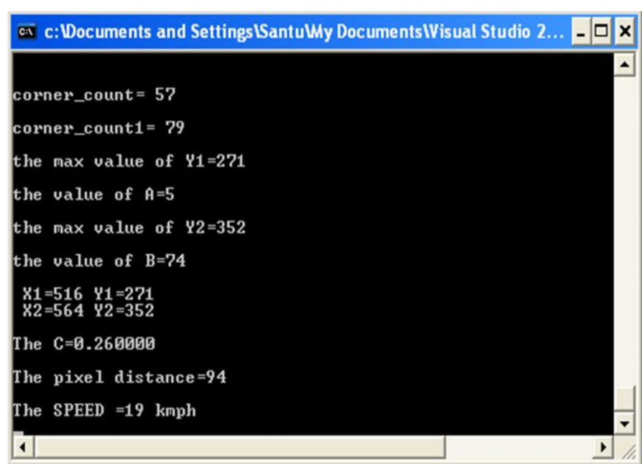

Figure 9C: Result displayed for a pre-recorded video of speed 20kmph

IX. CONCLUSION

The speed detection system is able detect vehicles speed even with shadows also. With fast processors or high-end smart phones, it can be seen as a future vehicles speed detector. Since the cost of this system is many times less. It can used manage traffic and avoid accidents at a cheaper price.

ACKNOWLEDGMENT

Thanks to our HOD for motivating me to publish this project as a paper.

REFERENCES

[1] E. Atko ciˉunas, R. Blake, A. Juozapavi cius, M. Kazimianec, Image Processing in Road Traffic Analysis, Nonlinear Analysis: Modelling and Control, 2005, Vol. 10, No. 4, 315–33

[2] Arash Gholami Rad, Abbas Dehghani and Mohamed Rehan Karim, "Vehicle speed detection in video image sequences using CVS method", at International Journal of the Physical Sciences Vol. 5(17), pp. 2555-2563, 18 December, 2010 , ISSN 1992 - 1950 ©2010 Academic Journals

[3] H. P. Moravec. Towards Automatic Visual Obstacle Avoidance. Proc. 5th International Joint Conference on Artificial Intelligence, pp. 584, 1977

[4] Z.Zheng, H.Wang and EKTeoh. Analysis of Gray Level Corner Detection. Pattern Recognition Letters , Vol. 20, pp. 149-162, 1999

[5] Rafael C.Gonzalez and Richard E.Woods "Digital Image Processing" Prentice Hall, 2001

[6] D.A. Forsyth, J. Ponce. "Computer Vision. A Modern Approach", Prentice Hall, 2003

[7] Open source computer vision library - reference manual, INTEL Corporation, 1999-2001.

[8] News Article: http://articles.timesofindia.indiatimes.com/2012-06-27/chandigarh/32439930_1_road-accidents-nakas-internal-roads