

# Vehicle Speed Measurement and Number Plate Detection using Real Time Embedded System

N.Arun Prasath<sup>1</sup> G.Sivakumar<sup>2</sup> Mr.N.Kumaresan<sup>3</sup>

1.M.E. (Embedded Systems and Technologies) Student, Department of Electronics and Communication Engineering, Anna University of technology, Coimbatore  
arunprasathest@gmail.com

2.M.E. (Embedded Systems and Technologies) Student, Department of Electronics and Communication Engineering, Anna University of technology, Coimbatore

3.Assistant Professor, Department of Electronics and Communication Engineering, Anna University of technology, Coimbatore

## ABSTRACT

A real time system is proposed to detect moving vehicles that violate the speed limit. A dedicated digital signal processing chip is used to exploit computationally inexpensive image-processing techniques over the video sequence captured from the fixed position video camera for estimating the speed of the moving vehicles. The moving vehicles are detected by analysing the binary image sequences that are constructed from the captured frames by employing the inter-frame difference or the background subtraction techniques. The detected moving vehicles are tracked to estimate their speeds. This project deals with the tracking and following of single object in a sequence of frames and the velocity of the object is determined. The proposed method varies from previous existing methods in tracking moving objects, velocity determination and number plate detection. From the binary image generated, the moving vehicle is tracked using image segmentation of the video frames. The segmentation process is done by using the thresholding and morphological operations on the video frames. The object is visualized and its centroid is calculated. The distance it moved between frame to frame is stored and using this velocity is calculated with the frame rate of video. The images of the speeding vehicles are further analysed to detect license plate image regions. The entire simulation is done in matlab and simulink simulation software.

**Keywords:** morphological; thresholding; segmentation; centroid

## Introduction

This project deals with the single vehicle object tracking and velocity of that object and also detecting license plate regions. The evaluation is performed on matlab /Simulink software. The vehicle speed detectors could be categorized into two types: 1) hardware-based detectors and 2) software-based detectors. The former is based on the electromagnetic principles and requires a dedicated hardware. The latter, which is adopted in this paper, performs vehicle detection using complex image- processing techniques over images or videos captured from camera and it requires high power computing processors. Software-based vehicle detector is nonintrusive and does not require roadway disruption for installation or maintenance, but the performance of image analysis algorithms can suffer from poor weather, darkness, glare, or shadows. Furthermore, high power computing processors are needed to perform computationally complex image processing algorithms in real time.

Images of speeding vehicles are further analyzed to detect the license plate region according to the color of license plates. We avoid employing computationally complex license plate recognition techniques. In general, license plate recognition techniques consist of two main parts: 1) detecting license plate regions and 2) recognizing license numbers. For license plate region detection, several methods have been proposed, such as template matching, Hough transform, histogram analysis, morphological operations and neural network. They have been developed to detect license plates based on the features of license plate characteristics or statistical representations, including edge detection, and color. License plate number recognition methods generally employ approaches that are similar to those used in license plate detection by only focusing on the character characteristics of license plates. All of the above techniques are computationally expensive and require high power computing processor whose energy consumption is very high. To save energy and keep the system functional without any interruption caused from power break, only license plate regions are detected by using computationally inexpensive image processing techniques

## OBJECTIVE

The objective of this project is to develop an algorithm for tracking of an object and determining the velocity of moving object in sequence of frames. All the evaluation has to be performed on MATLAB/SIMULINK software, which is capable of doing onboard calculations. It has special functions for handling graphics up to ten-times faster than others. Algorithms can be extended for real time applications.

### III. EXISTING METHOD

In this section, image processing techniques, which are used for binary image extraction for moving vehicle object, is introduced. To keep the energy consumption as low as possible, image processing techniques with high computational costs are avoided. Practical solutions that result in low energy consumption are developed and employed.

#### PARTITIONING THE ROAD INTO DISJOINT REGIONS

In order to reduce the computational cost and alleviate the measurement problems caused by multiple vehicles moving side by side or one after each other, the road area is partitioned into disjoint regions. The parts of the image that do not belong to any region are not considered in the calculations to reduce the computational cost, and each region is processed separately. The road partitioning layout used in this paper is shown in Figure. 3.1. The road is partitioned into two regions to represent each lane independent from the other lane, i.e.,  $R_o$  (region painted with light-blue color) and  $R_i$  (region painted with dark-blue color).

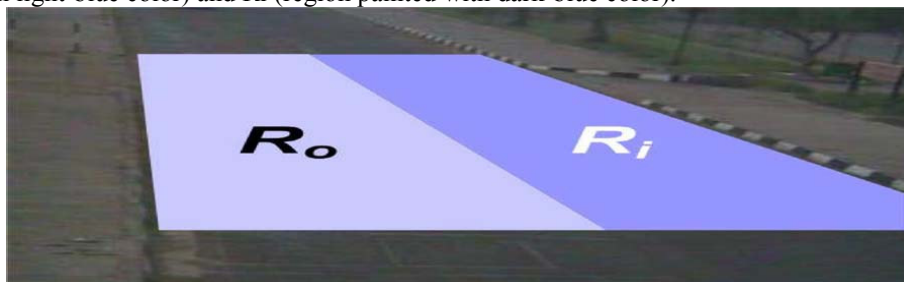


Figure 3.1 Partitioning the road area into two regions, i.e., the light blue region labeled with  $R_o$  and the dark blue region with  $R_i$

The setup shown in Figure.3.1 has other advantages besides less computational costs of mathematical calculations. Such a partitioning makes the parameter estimation to be performed on a local region, which has almost constant illumination on all over the region. This makes the parameter estimation robust. Furthermore, the setup confines only one car in each region for the cases of vehicles was passing nearby or vehicles following each other in the same lane. The distances between start and end of the regions are set as short as possible to not let any vehicle change its region before the speed measurement is performed. Such a method also benefits from automatically handling multi-vehicles on the road.

#### BINARY IMAGE GENERATION

The speed measurement is performed in binary image domain, i.e., each pixel is transformed into either “1” or “0” according to its motion information. To binarize the incoming input image and only detect the moving pixels, two different techniques are used: 1) inter-frame difference and 2) background subtraction.

##### Inter-frame Difference Technique

Let us assume that the input RGB image from the camera through a video card is represented as  $X_t(\text{RGB})$  of size  $X_H \times X_W$  at time  $t$ . Furthermore, considering the region of interest, as shown in Figure. 3.1 (e.g.,  $R_o$ ), let  $I_t^{(\text{RGB})}$  be the sub-image of size  $H \times W$  referring to the pixel intensity values of the spatial region labeled with  $R_o$ . The difference image between consecutive frames at times  $t$  and  $t - 1$  is computed by taking the absolute valued difference of gray scale representations of frames. The RGB image  $I$  (RGB) is converted to gray scale image  $I_t$  by using simple averaging of color channels,

$$(I_t^{(R)} + I_t^{(G)} + I_t^{(B)}) / 3 \quad (3.1)$$

Where  $I_t^{(R)}$ ,  $I_t^{(G)}$  and  $I_t^{(B)}$  are the intensity values for R,G and B channels respectively.

After the conversion to gray scale, the previous frame  $I_{t-1}$  is subtracted from the current frame  $I_t$  to create an absolute- valued difference  $DI_{t,t-1}$ , i.e.,

$$DI_{t,t-1} = |I_t - I_{t-1}| \quad (3.2)$$

Once the difference image is obtained, thresholding is applied to differentiate the moving pixels from the non-moving pixels. This process generates a binary difference image  $BDI_{t,t-1}$  using a threshold value

$\tau(DI_{t,t-1})$ . The threshold value  $\tau(DI_{t,t-1})$  is statistically obtained as

$$T(DI_{t,t-1}) = \mu(DI_{t,t-1}) + \sigma(DI_{t,t-1})$$

$$\mu(DI_{t,t-1}) = \frac{\sum_{y=1}^H \sum_{x=1}^W DI_{t,t-1}(y,x)}{HW} \quad (3.3)$$

$$\sigma(DI_{t,t-1}) = \frac{\sqrt{(\sum_{y=1}^H \sum_{x=1}^W (DI_{t,t-1}(y,x) - \mu(DI_{t,t-1}))^2)}{HW-1}$$

Where  $\mu(DI_{t,t-1})$  and  $\sigma(DI_{t,t-1})$  are the mean and the standard deviation of the difference pixels, respectively, and  $(y, x)$  is the spatial coordinate. The definition of standard deviation given in (3.3) requires mathematical operations of square root and square. These operations are expensive in computation and require more energy consumption. Thus we modify it as

$$\sigma(DI_{t,t-1}) = \frac{\sum_{y=1}^H \sum_{x=1}^W |(DI_{t,t-1}(y,x) - \mu(DI_{t,t-1}))|}{HW-1} \quad (3.4)$$

Using the threshold  $\tau(DI_{t,t-1})$ , the binary difference image  $BDI_{t,t-1}$  is created as

$$BDI_{t,t-1}(y,x) = \begin{cases} 1 & \text{if } DI_{t,t-1}(y,x) \geq \tau(DI_{t,t-1}) \\ 0 & \text{otherwise} \end{cases} \quad (3.5)$$

Figure 3.2 shows the binarization process of the frame difference using two consecutive RGB frames. Figure 3.2(a) and (b) shows the current and previous frames, respectively. The binary difference image according to (3.2) is shown in Figure 3.2(c). The thresholding process according to (3.5) generates a binary image as shown in Figure 3.2(d) for further processing. Generating a binary difference image using the inter-frame difference technique only requires a single frame memory to store the previous frame.

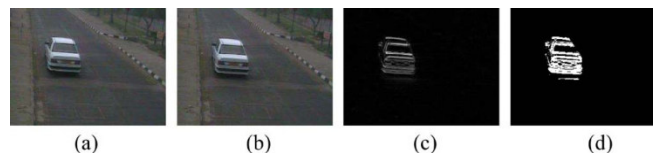


Figure.3.2 Illustration of generating difference and binary difference images using inter-frame difference technique, (a) Current RGB image. (b) Previous RGB image (c) Difference image using (3.2). (d) Binary difference image using (3.5)

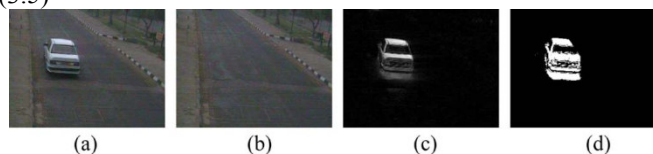


Figure.3.3 Illustration of generating difference and binary difference images using background subtraction technique, (a) Current RGB image. (b) Background RGB image (c) Difference from the background image using (6). (d) Binary difference image using (3.7)

#### Background Subtraction Technique

Inter-frame difference-based binary image generation produces an abstract representation of moving objects. A better approach is to perform background subtraction, which identifies moving objects from the portion of a video frame that significantly differs from a background model. It involves comparing an observed image with an estimate of the background image to decide if it contains any moving objects. The pixels of the image plane where there are significant differences between the observed and estimated images indicate the location of the objects of interest.

Let us assume that the background image  $B_t$  of a specific scene is constructed. The current gray scale image  $I_t$  at time  $t$  is subtracted from the background image  $B_t$  as

$$BS_t = |I_t - B_t| \quad (3.6)$$

To find the absolute-valued difference image between  $I_t$  and  $B_t$ . The difference image is binarized using the same approach in (5), i.e.

$$BDI_{t,t-1}(y,x) = \begin{cases} 1 & \text{if } BS_t(y,x) \geq \tau(BS_t) \\ 0 & \text{otherwise} \end{cases} \quad (3.7)$$

Where  $\tau(BS_t)$  is a statistical threshold that is calculated according to (3.3), and  $BDI_{t,t-1}$  is a binary difference image.

Figure.3.3 shows the binarization process of the background subtraction technique. Figure 3.3(a) and 3.3(b) shows the current and background images, respectively. The difference image according to (3.6) is shown in Figure 3.3(c). The thresholding process according to (3.7) generates a binary image that is shown in Figure 3.3(d) for further processing.

The background image is updated according to BBSt using BSt and It as

$$B_{t+1}(y,x) = \begin{cases} \alpha B_t(y,x) + (1-\alpha)I_t(y,x), & \text{if } BBS_t(y,x) = 0 \\ B_t(y,x), & \text{otherwise} \end{cases} \quad (3.8)$$

Where  $B_{t+1}$  is the updated background image, and  $\alpha \in [0, 1]$  is an update factor that is settled as  $\alpha = 0.95$  in this method.

#### IV.ALGORITHMS FOR TRACKING MOVING OBJECT TRACKING THE MOVING OBJECT

This section explains the algorithms to track the single object and to estimate the velocity of moving object. The sequential approach to track the moving objects and the velocity of objects are as follow.

Segmentation: To separate multiple regions in image

Feature Extraction: To analyse the regions in image

Tracking: Analysing the position, velocity and moving direction of object

##### Segmentation

To perform the segmentation operation, frame difference algorithm is implemented as it takes less processing time. Frame difference algorithm performs separation of two sequential frames and the frame difference algorithm for the segmentation is explained as follow

1. Read the input images
2. For (present position=initial position: final position)
  - (a) Difference between the pixels values at present position of two images is calculated
  - (b) Calculate the absolute value.
  - (c) Store the difference in new image at same pixel position that is at present position

##### Feature Extraction

Every object has a specific feature which is used to visualize the object and used for tracking. After performing the segmentation, a rectangular bounding box is plotted with the dimensions of the object produced in the residual image. Algorithm for the bounding box is as followed

1. Read the image difference
2. For (present position=initial value: final value) of Y resolution
3. For (present position=initial value: final value) of X resolution
  - (a) Calculate the sharp change in intensity of image from top and bottom
  - (b) Store the values in an array
4. Height of the bounding box is = bottom value- top value
5. For (present position=initial value: final value) of X resolution
6. For (present position=initial value: final value) of Y resolution
  - (a) Calculate the sharp change in intensity of image from left and right
  - (b) Store the values in an array
7. Width of the bound box = right value - left value
8. Using the dimensions, draw boundary to the image  
Initial value: the starting position of the pixel in an image.  
Final value: the ending position of the pixel in an image.  
Height = top value – bottom value  
Width = right value – left value
9. Add the height value with the top value and store it in a variable like mid top
10. Add the width value to the left value and store it in a variable like mid left
11. Assign the max intensity to the pixel at pixel value at (mid top, mid left)

##### Distance Calculation

The distance travelled by the object is determined by using the centroid. It is calculated by using the Euclidean distance formula. The variables for this are the pixel positions of the moving object at initial stage to the final stage. Algorithm for calculating distance is explained as follow

1. Read the centroid position of each image.
2. Calculate the distance between two centroid images
3. For (present position = initial value - final value) of X resolution
4. For (present position = initial value - final value) of Y resolution
5. Calculate change in distance by  
 $Distance = \sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2}$

Where  $X_1$ =previous pixel position and  $X_2$ =present pixel position in width

$Y_1$ =previous pixel position and  $Y_2$ =present pixel position in height

6. Store all the distance values in an Array.

### Determination of Velocity

Velocity of moving object is determined using the distance travelled by the centroid to the frame rate of the video. Algorithm for calculating velocity is explained as follow

1. Read the distance travelled by the object

Velocity = distance travelled/frame rate

2. Save the value in an array

3. The velocity of moving object in the sequence frames is defined in pixels / second

The velocity of the moving vehicle objects can also be calculated as shown in the equations below.

Distance between vehicle and starting point measured in kilometer

$$Distance = Df * (D / D_x) * (P_n - P_0)$$

Time that vehicle spent in order to move to  $P_n$  in unit of hour

$$Time = Tf * (t_n - t_0)$$

Vehicle speed measured in format of kilometer per hour

$$Speed = Distance / Time \text{ (Kilometer per Hour)}$$

Where  $D$  is the real distance between two marking points (start point and end point) measured in meter

$D_x$  is the distance between two marking points measured in pixels

$X$  is the width of the video scene measured in pixels

$Y$  is the height of the video scene measured in pixels

$P_0$  is the right most of the vehicle position at time  $t=0$  measured in unit of pixels

$P_n$  is the right most of the vehicle position at time  $t=n$  measured in unit of pixels.

$t_n$  is the tickler (timestamp) saved at time  $t = n$  measured in unit of milliseconds

$t_0$  is the tickler (timestamp) saved at time  $t = 0$  measured in unit of milliseconds

$Df$  is the distance conversion factor from meter to kilometer, which is  $(1.00/(1000.00*60.00*60.00))$

$Tf$  is the time conversion factor. In this case, the conversion is from millisecond to hour, which is  $(1.00/1000.00)$

### V. NUMBER PLATE REGION EXTRACTION

The image of a vehicle with number plate is taken. Then the part of the image containing the number plate is extracted by using the image edges detection, image masking, dilation process and the image of the number plate is plotted. From the plotted image, the image of the number plate is extracted. Then the binary image of the number plate is extracted from the image of the number plate

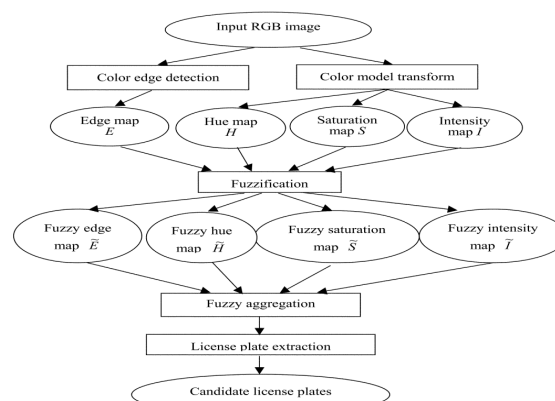


Figure.5.1 The process of number plate region extraction

## VI.RESULTS AND DICUSSION

Object tracking, the main application for security, surveillance and vision analysis, is evaluated on MATLAB/SIMULATION. In this, a video is recorded using digital camera. The recorded video frames are converted into individual frame and then converted into Gray Scale (Intensity) format images. The images are used as input for the frame difference for the separation of foreground and background objects. A rectangular bounding box is plotted around the foreground objects produced from frame difference. By using the dimensions of rectangular bounding box, a centroid is plotted. The position of the centroid is stored in the array and the distance is calculated using Euclidean distance formula. The velocity of the object movement from frame to frame is calculated by using the distance and frame rate of the recorded video. The velocity is stored in the array. Velocity of moving object is in 2-dimension (since camera is static). Figure 7.1 shows the step by step process of tracking a moving object. The simulated output acquired from matlab and Simulink for velocity determination is shown in Figure 7.2

### SPEED CALCULATION

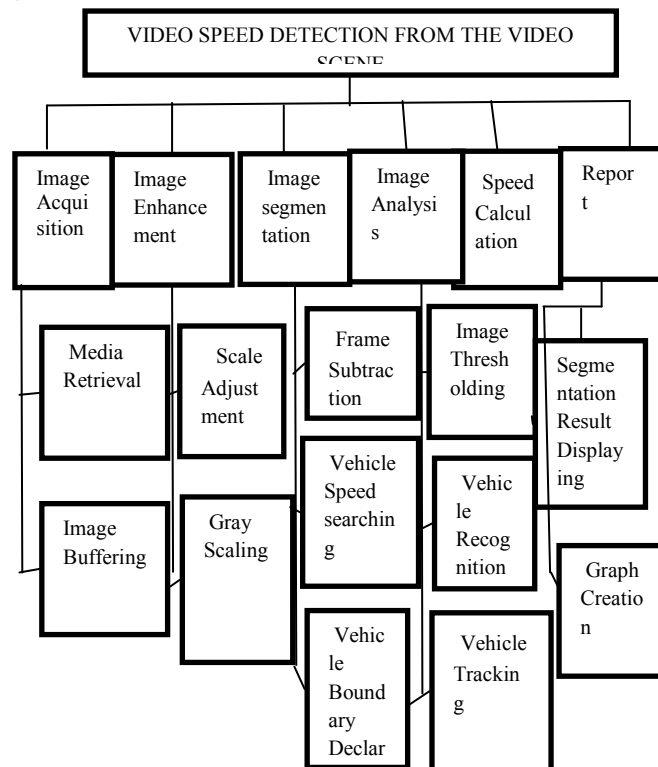


Figure 6.1. Step by step diagrams during evaluation and implementation

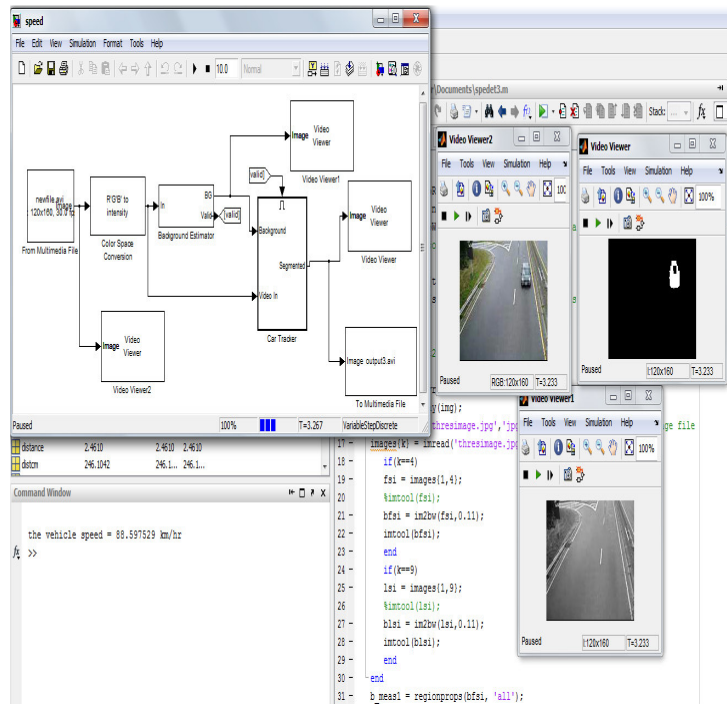


Figure 6.2 The simulated output for vehicle speed estimation using the matlab and simulink software.

The vehicle speed estimation using the previous existing method and the proposed method using both background subtraction and inter-frame difference technique for binary image generation is shown in the table. The proposed method has less error rate and more accuracy in detecting vehicle speed.

Table 6.1 comparison of average error for estimating vehicle speed using existing and proposed method

Vehicle speed estimation technique	Average Error	
	Background Subtraction	Inter-frame difference
Existing method	1.43	1.35
Proposed method	0.45	1.26

### NUMBER PLATE REGION EXTRACTION

In general, license plate recognition techniques consist of two main parts: Detecting license plate regions and Recognizing license numbers. A mathematical model is proposed for license plate detection in the proposed method. Number plate region is extracted by using the image edges detection, image masking, dilation process and the image of the number plate is plotted. The binary image of the number plate region is extracted from the image of the plotted number plate. The simulated output from matlab is shown in Figure. In the Figure, the image of a vehicle with number plate is taken and the part of the image containing the number plate is extracted by using the image edges detection, image masking, dilation process and the image of the number plate is plotted as shown in Figure. From the plotted image, the image of the number plate is extracted and the binary image of the number plate is extracted from the image of the number plate as shown in Figure 7.3.

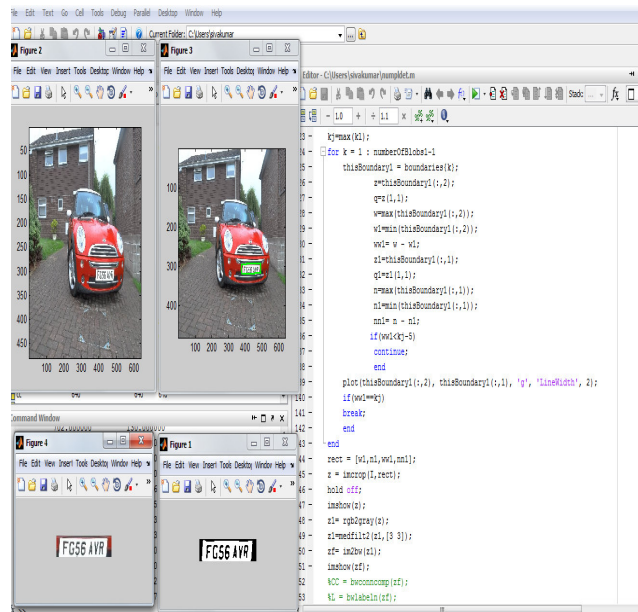


Figure 6.3 The extracted number plate binary image using the matlab software

## VII.CONCLUSION

Tracking of moving object is evaluated on MATLAB/SIMULATION which is a major application in security, surveillance and vision analysis. Object tracking, the main application for security, surveillance and vision analysis, is evaluated on MATLAB/SIMULINK and by using the code composer studio, the interface we can deploy the algorithms from the workstation to TMS320C6416T processor. Algorithms are developed for improving the image quality, segmentation, feature extraction and for determining the velocity. The implementation is performed by removing noise in image and separating foreground and background objects. The object is visualized and its centroid is calculated. The distance it moved between frame to frame is stored and using this velocity is calculated with the frame rate of video. The velocity of moving object in 2-dimension (since the camera is static) is determined.

## References

- [1] Turgay Celik and Huseyin Kusetogullari , (2010) "Solar-Powered Automated Road Surveillance System for Speed Violation Detection", Ieee Transactions on Industrial Electronics, pp. 3216-3226.
- [2] Ahmad I. R. B, Rahimt H. A, Sheikh2 U. U., and Zaint A. S. M(2010), "Vehicle speed detection using frame differencing for a surveillance system", 10th International Conference on Information Science, Signal Processing and their Applications, pp. 630-632.
- [3] Alper Yilmaz, Mubarak Shah, and Omar Javed.,(2006) "Object tracking: A survey". ACM Computer. Survey, pp. 564-567
- [4] Baik D.-K and Ki Y.-K., (2006), "Model for accurate speed measurement using double-loop detectors," IEEE Trans on Veh Technol", pp. 1094-1101.
- [5] Chellappa R, Venkateswar V, and Zhou Y. T.,(1989) "Edge detection and linear feature extraction using a 2-d random field model", IEEE Trans. Pattern Anal. Mach. Intell, pp. 243-246.
- [6] Cheng H, Chen B, Palen J, Lin B, Shaw B, and Wang Z., (2005), "Development and field test of a laser-based nonintrusive detection system for identification of vehicles on the highway," IEEE Trans. Intell. Transp. Syst., pp. 147-155.
- [7] Chomtip Pornpanomchai S and Kaweepap Kongkittisan T,(2009) "Vehicle speed detection system", IEEE International Conference on Signal and Image Processing Applications. pp. 194-201.
- [8] Crawford C and Doran V.,(2004) "Trial and evaluation of intelligent road studs for hazard warning," in Proc. IEE Int. Conf. Road Trans. Inf. Control, pp. 132-136.
- [9] Cucchiara R, Mello P, and Piccardi M.,(2000) "Image analysis and rule-based reasoning for a traffic monitoring system," IEEE Trans. Intell. Transp. Syst., pp. 119-130.
- [10] Davis L, Haritaoglu I, and Harwood D.,(2000), "Real-time surveillance of people and their activities," IEEE Trans. Pattern Anal. Mach. Intell., pp. 809-830.
- [11] Gupte S, Martin R, Masoud O, and Papanikolopoulos N.,(2002) "Detection and classification of vehicles," IEEE Trans. Intell. Transp. Syst., pp. 37-47.
- [12] Lee H.J and Wangand S.Z.,(2003), "Detection and recognition of license plate characters with different



- appearances ,”in Proc. IEEE Intell. Transp. Syst, pp.979-984.
- [13] Li Z and Yang Z.M.,(2006), “Application of cement-based piezoelec- tric sensors for monitoring traffic flows,” J. Trans. Eng., pp. 565–573.

#### **AUTHORS PROFILE**

**ARUN PRASATH N** received B.E(E.C.E) degree from Krishansamy College of Engineering and Technology.He is currently doing M.E(Embedded Systems Technologies) in Anna University of Technology ,Coimbatore.He worked has a maintainance engineer in Electronics Controls Power Systems Pvt Limited,bangalore.Now his interest researches in the field of embedded networks,real time operating systems and image processing.

**SIVAKUMAR G** received B.E(E.C.E) degree from Amirtha University. He is currently doing M.E(Embedded Systems Technologies) in Anna University of Technology ,Coimbatore. Now his interest researches in the field of linux,real time operating systems and image processing.

The IISTE is a pioneer in the Open-Access hosting service and academic event management. The aim of the firm is Accelerating Global Knowledge Sharing.

More information about the firm can be found on the homepage:

<http://www.iiste.org>

### CALL FOR JOURNAL PAPERS

There are more than 30 peer-reviewed academic journals hosted under the hosting platform.

**Prospective authors of journals can find the submission instruction on the following page:** <http://www.iiste.org/journals/> All the journals articles are available online to the readers all over the world without financial, legal, or technical barriers other than those inseparable from gaining access to the internet itself. Paper version of the journals is also available upon request of readers and authors.

### MORE RESOURCES

Book publication information: <http://www.iiste.org/book/>

Academic conference: <http://www.iiste.org/conference/upcoming-conferences-call-for-paper/>

### IISTE Knowledge Sharing Partners

EBSCO, Index Copernicus, Ulrich's Periodicals Directory, JournalTOCS, PKP Open Archives Harvester, Bielefeld Academic Search Engine, Elektronische Zeitschriftenbibliothek EZB, Open J-Gate, OCLC WorldCat, Universe Digital Library, NewJour, Google Scholar

