Real Time Traffic Estimation Using Live Video and Image Processing

Date: 29/06/2014

Saisubramaniam G
Computer Science and Engineering
Sri Venkateswara College of Engineering, Department of Computer Science and Engineering
Sriperumbudur, India

Yeshwanth Raja
Computer Science and Engineering
Sri Venkateswara College of Engineering, Department of Computer Science and Engineering
Sriperumbudur, India

Shyam Ravichandran
Computer Science and Engineering
Sri Venkateswara College of Engineering, Department of Computer Science and Engineering
Sriperumbudur, India.

A Vignesh
Computer Science and Engineering
Sri Venkateswara College of Engineering, Department of Computer Science and Engineering
Sriperumbudur, India.

Abstract:

Computer vision based approach to monitor roads and analyze traffic is gaining popularity in recent times due to the advancement in the field of image processing. In this paper we propose the various techniques of image processing to determine the traffic rate in a scale of low, medium and high. We also determine the area covered by the vehicles in the road under observation and present the same in a customized window.

The proposed framework has been implemented using the OpenCV library for C++ and it aims to classify the traffic on the road. Moreover, for implementing this project following steps must be considered: 1) image acquisition 2) RGB to grayscale transformation 3) image enhancement and 4) morphological operations. The framework implemented reports results from real traffic videos obtained from the Indian Institute of Technology, Madras (IITM).

A comparison is made with Google Maps, and WAZE, both built on crowdsourcing concepts. The results are based on the estimation of traffic with reliable data vs crowd-source data.

Keywords: Image processing, Traffic analysis, Real time data streaming, decision making.

Background:

It is important to know the road traffic density real time especially in mega cities for signal control and effective traffic management. In recent years, video monitoring and surveillance systems have been widely used in traffic management. Hence, traffic estimation and area coverage can be achieved using video monitoring systems.

Input is in the form of live stream taken from the cameras placed in different areas around Chennai city. Each camera is given a separate IP address and the input is cropped in order to focus only on the lane of interest. The techniques employed include Motion Detection which is used to detect the traffic at the instance by computing the absolute difference between the consecutive frames. Another technique, namely, Edge Detection is used to find the area covered by the vehicles. By using these two methods, we can determine the traffic effectively.

Comparison is made with Google Maps, which uses crowd sourcing. Google Maps offers traffic data in real-time, using a colored map overlay to display the speed of vehicles on particular roads. Crowdsourcing is used to obtain the GPS-determined locations of a large number of cellphone users, from which live traffic maps are produced.

Our input is based on reliable sources (live footage obtained from the camera fixed at the area). Since it is meant specifically for Traffic Estimation, the data obtained is much more reliable than the data used by Google Maps.

Motion Detection:

Motion Detection is the process of detecting a change in position of an object relative to its surroundings or the change in the surroundings relative to an object. We detect motion by optics (video and camera systems). We use a simple algorithm for motion based on the method of Collins et al. as this method erases the phenomenon "ghosting".

Let's consider 3 images - called previous, current and next. First subtract the images previous and next and then the images current and next. After this a logical AND operation is executed on the results of both and a threshold is chosen to make it accurate for large changes only. Pre-processing is useful to reduce the number of false positive alarms that arises from factors like varying lighting, camera flicker, and CCD dark currents. If no motion is detected, it is assumed that the space is empty.

Due to the simplicity of this method, it has a high performance. However it can generate lots of false positives, if the wrong threshold has been chosen. Picking the right threshold is crucial, and some motion detection algorithms suggest to pick a dynamic/adaptive threshold. However a dynamic threshold can still fail and give false positives either.

Robust motion algorithms tackle these problems with ease, but need a lot of extra CPU power to succeed. That's why methods like Differential Subtracting come in handy. A simple trick to erase false positives is assuming motion only occur in a sequence of images larger than one.

Another parameter used to neglect false positives is the standard deviation**.** It tells us something about the distribution of motion. When motion is specific at a single point let's say a human which is relative small against the size of the viewport (far away from the camera) then the motion will be concentrated in a single point/pixel, in this case the standard deviation will be near to zero. When a lot of motion is detected and is distributed over the entire image, than the standard deviation will be very high.

In order to provide accurate traffic, we employ smoothing techniques so that the traffic is gradually estimated, as how a human being will deduct traffic in the area.

Exponential smoothing is a technique that can be applied to time series data, either to produce smoothed data for presentation, or to make forecasts. Exponential smoothing also gives an understanding of the current and ongoing trend. The time series data themselves are a sequence of observations. Each timestamp (key frame in our case) provides information about the environment which depends on its past activities. Whereas in the simple moving average the past observations are weighted equally, exponential smoothing assigns exponentially decreasing weights over time. Recent observations are given more weightage when compared to the least-recent observations over time.

The raw data sequence is often represented by $\{x_t\}$ beginning at time t=0, and the output of the exponential smoothing algorithm is commonly written as $\{s_t\}$, which may be regarded as a best estimate of what the next value of *x* will be. When the sequence of observations begins at time *t* = 0, the simplest form of exponential smoothing is given by the equation:

$$s_0 = x_0$$
$$s_t = \alpha x_{t-1} + (1-\alpha)s_{t-1},\ t > 0$$

Where α is the *smoothing factor*, and 0 < α < 1.

Using this method, we are able to detect the traffic on the road and classify them into the following three categories-Low, Medium and High.

Edge Detection:

Edge detection is the name for a set of mathematical methods which aim at identifying points in a digital image at which the image brightness changes sharply or, more formally, has discontinuities. The points at which image brightness changes sharply are typically organized into a set of curved line segments termed edges. Edge detection is a fundamental tool in image processing, machine vision and computer vision, particularly in the areas of feature detection and feature extraction.

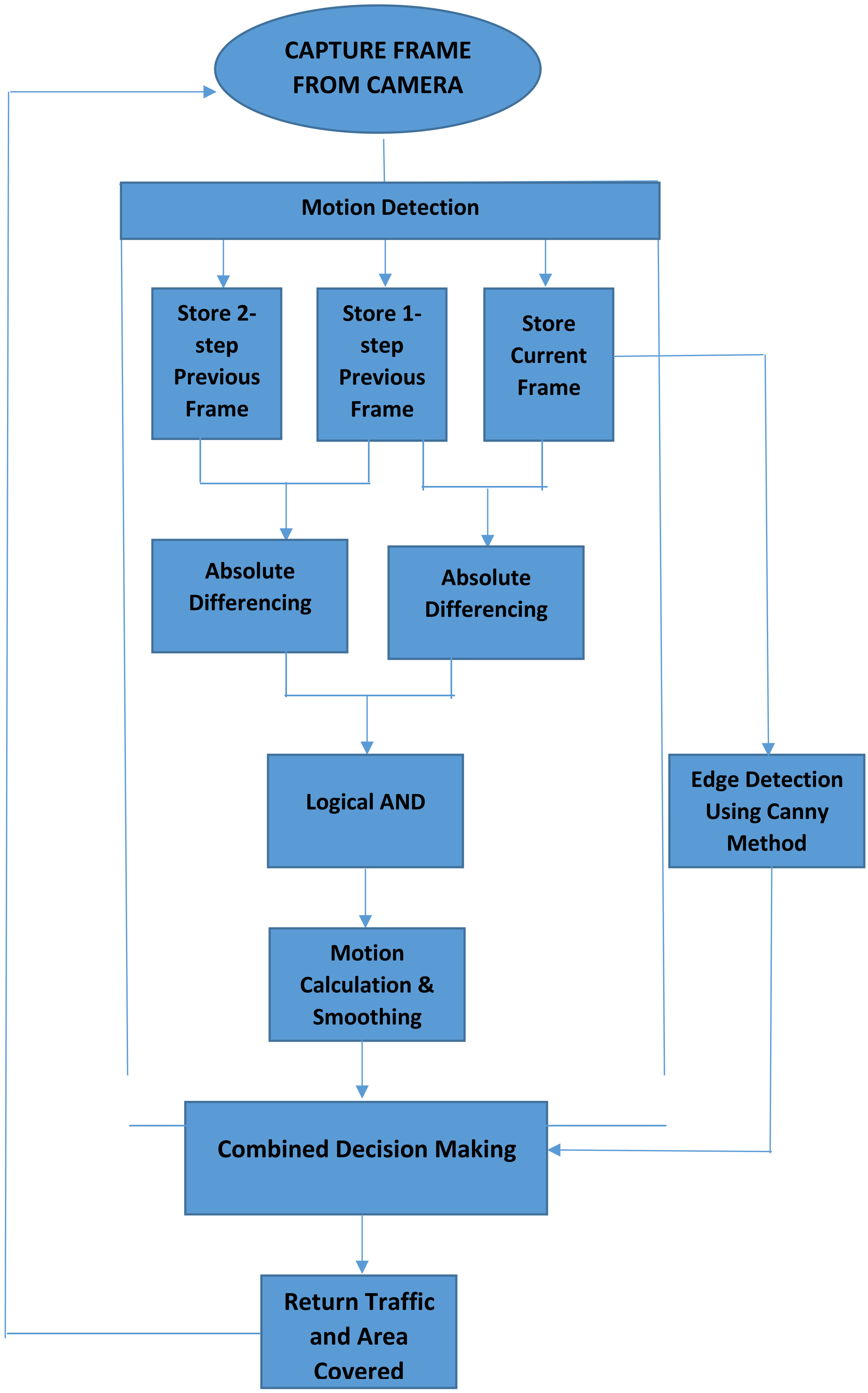
The Canny Edge detector was developed by John F. Canny in 1986. Also known to many as the optimal detector, canny algorithm aims to satisfy three main criteria:

- **Low error rate:** Meaning a good detection of only existent edges.
- **Good localization:** The distance between edge pixels detected and real edge pixels have to be minimized.
- **Minimal response:** Only one detector response per edge.

It is an edge detection method that uses a multi-stage algorithm to detect a wide range of edges in images. Large intensity gradients are more likely to correspond to edges than small intensity gradients. It is in most cases impossible to specify a threshold at which a given intensity switches from corresponding to an edge into no doing so. Therefore canny uses thresholding with hysteresis. Thresholding with hysteresis requires two thresholds- high and low. Making the assumption that important edges should be along continuous curves in the image allows us to follow a faint section of a given line and to discard a few noisy pixels that do not constitute a line but have produced large gradients.

The threshold values tend to change with illumination, hence we have used the system clock as an indicator to update the values from time to time.

Algorithm:


CAPTURE FRAME FROM CAMERA
Motion Detection
Store 2-step Previous Frame
Store 1-step Previous Frame
Store Current Frame
Absolute Differencing
Absolute Differencing
Logical AND
Edge Detection Using Canny Method
Motion Calculation & Smoothing
Combined Decision Making
Return Traffic and Area Covered

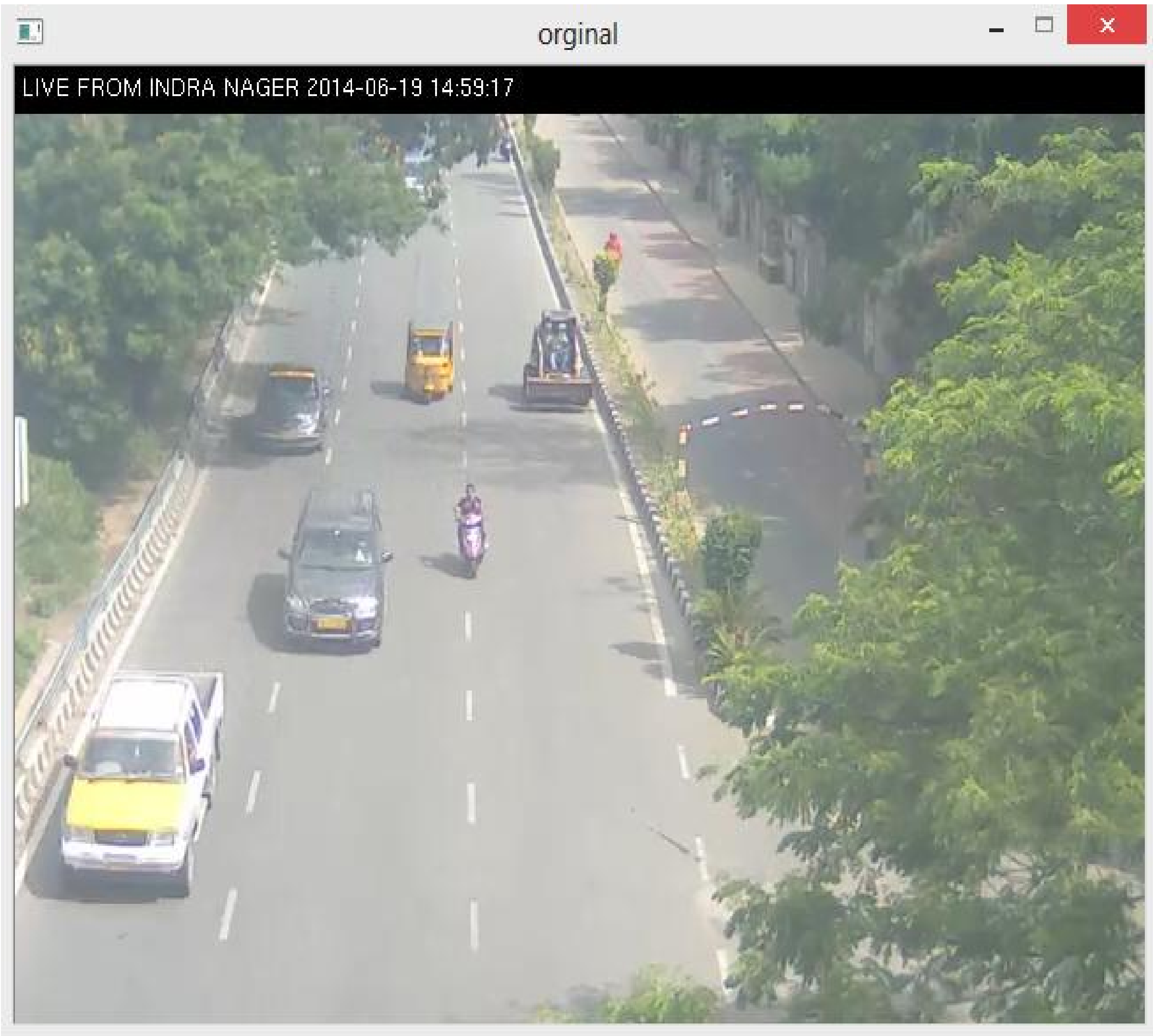
Experimental Results:

We implemented our proposed framework on live video streams received from the cameras fitted at certain areas in Chennai city. Samples of the work are presented in this section.

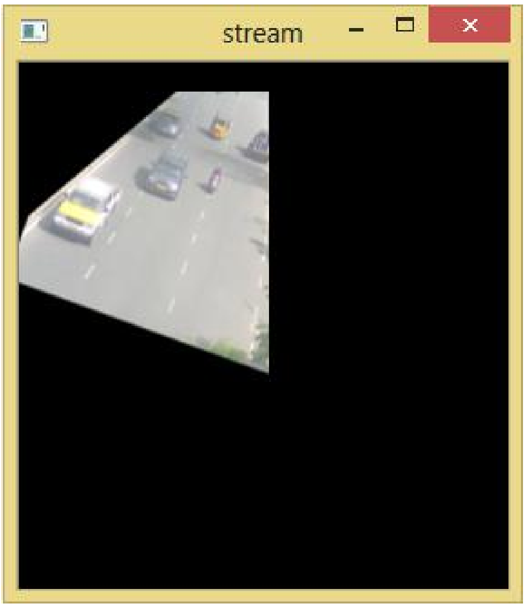
Results from the Motion Detection module

A sample footage of Indira Nagar, Chennai dated 19th June, 2014

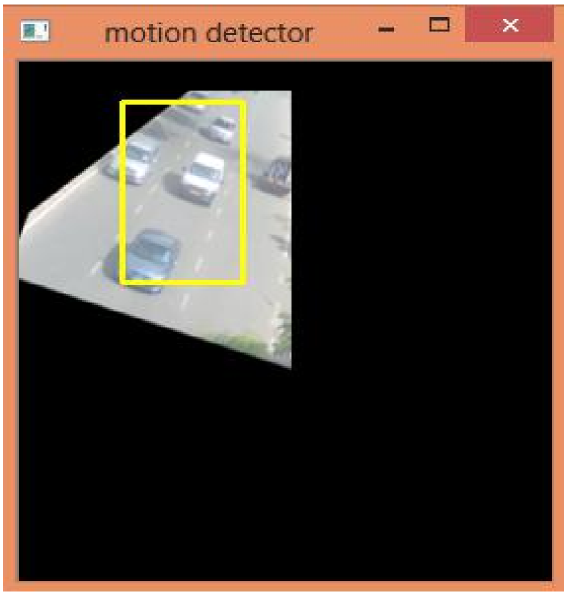
A sample – the captured camera footage (Indira Nagar)



The transformation of the road – Cropping the lane of interest:



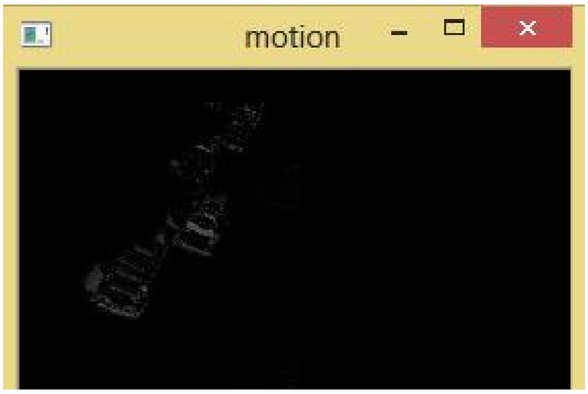
Detecting the motion



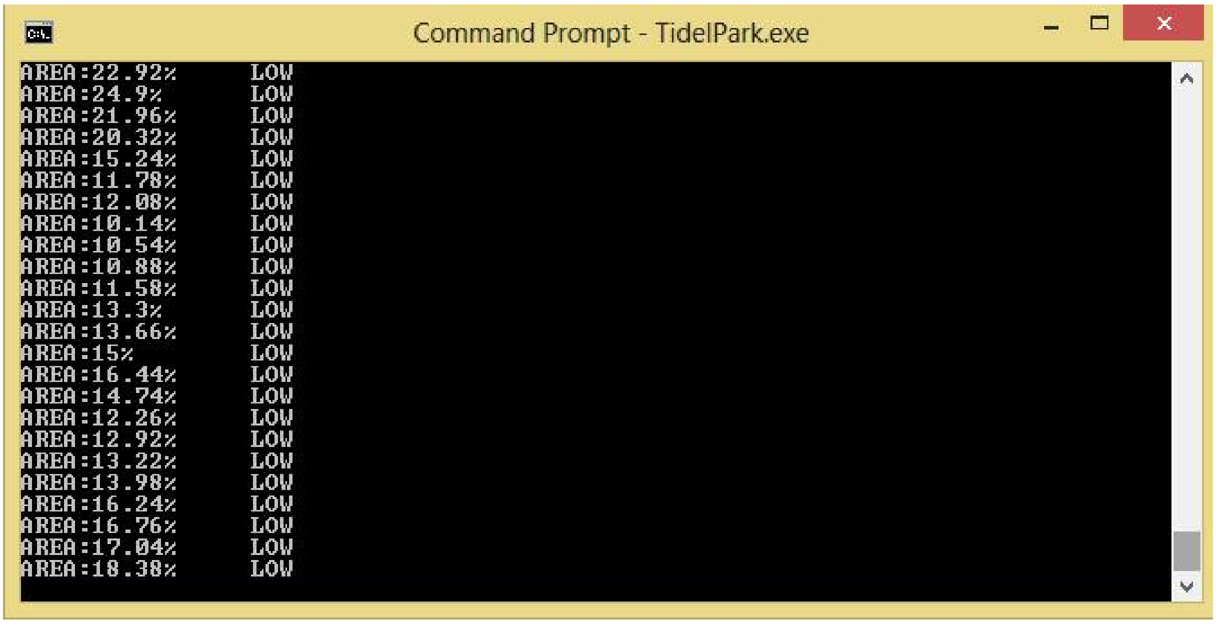
Absolute differencing between the previous and current frame

Absolute differencing between the previous and current frame

Motion Detection by using Logical AND on both the frames



The traffic estimation:

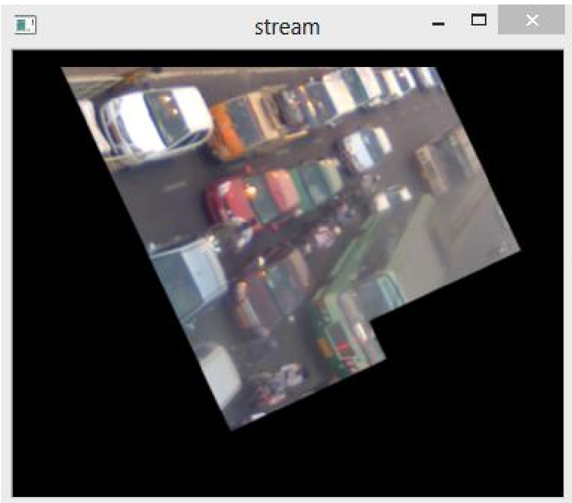


The motion detection technique classifies the traffic rate into one of the three categories – low, medium or high.

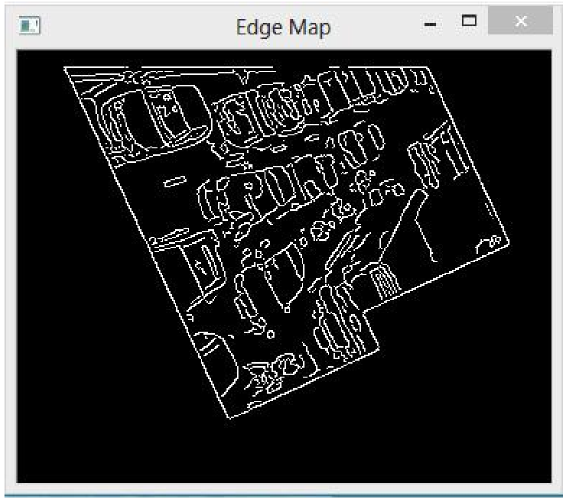
Results of the Edge Detection module

Tidel Park road:

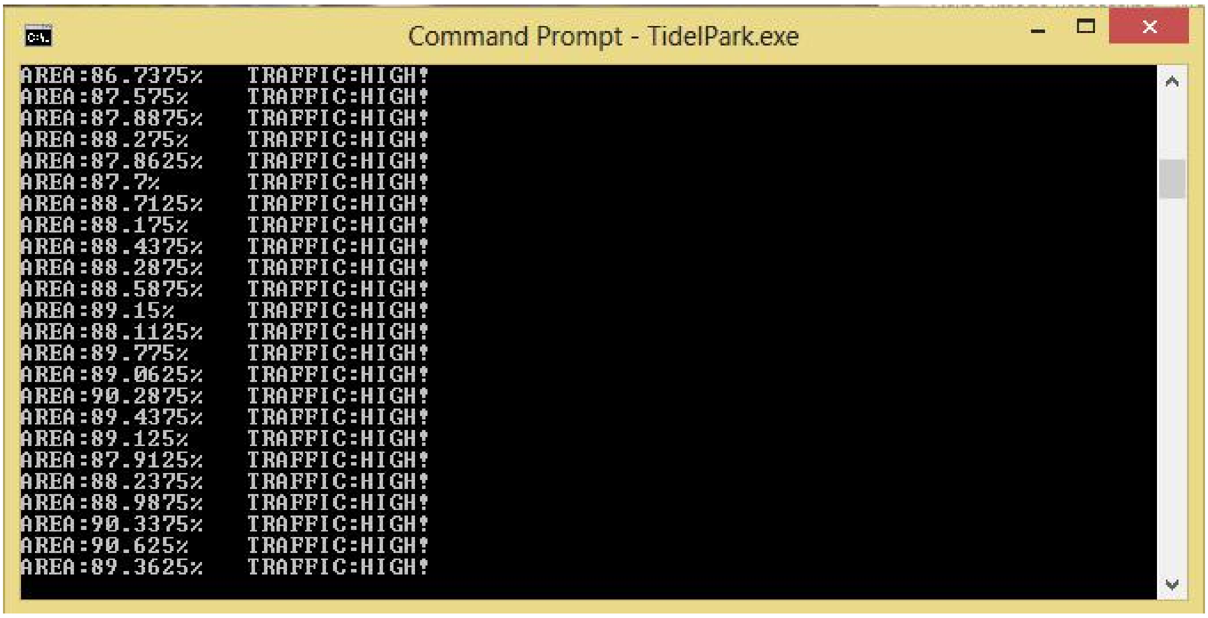
The transformation of the road



The Area covered-using Edge Detection



Area coverage:

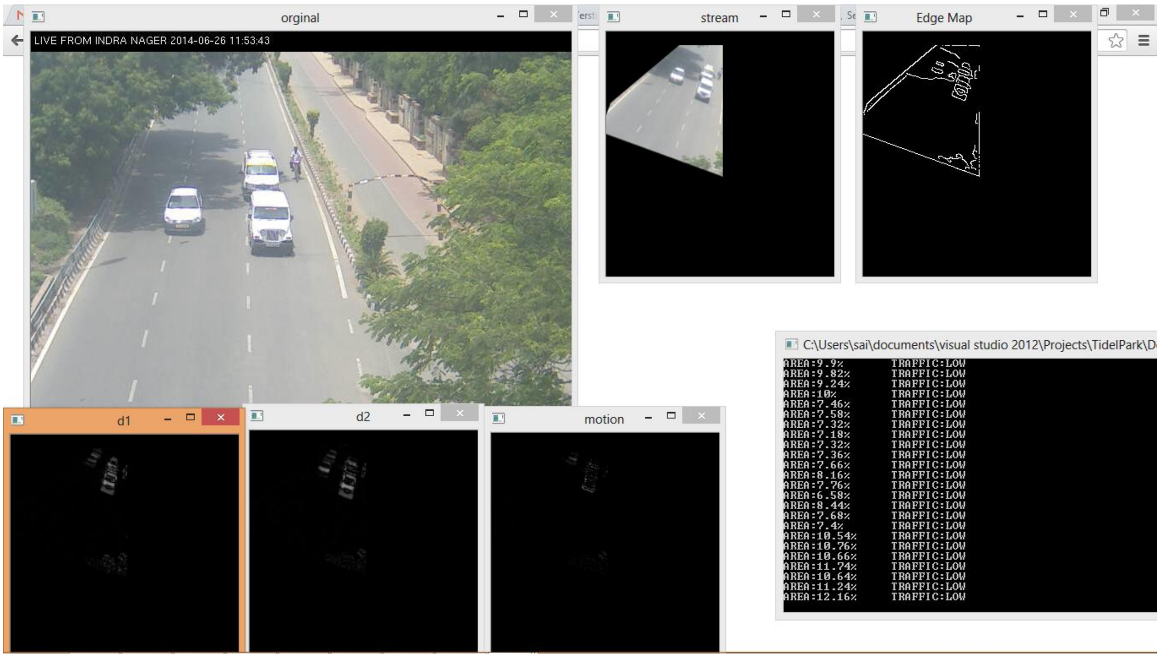


The results of Edge Detection give us the area covered on the road under observation

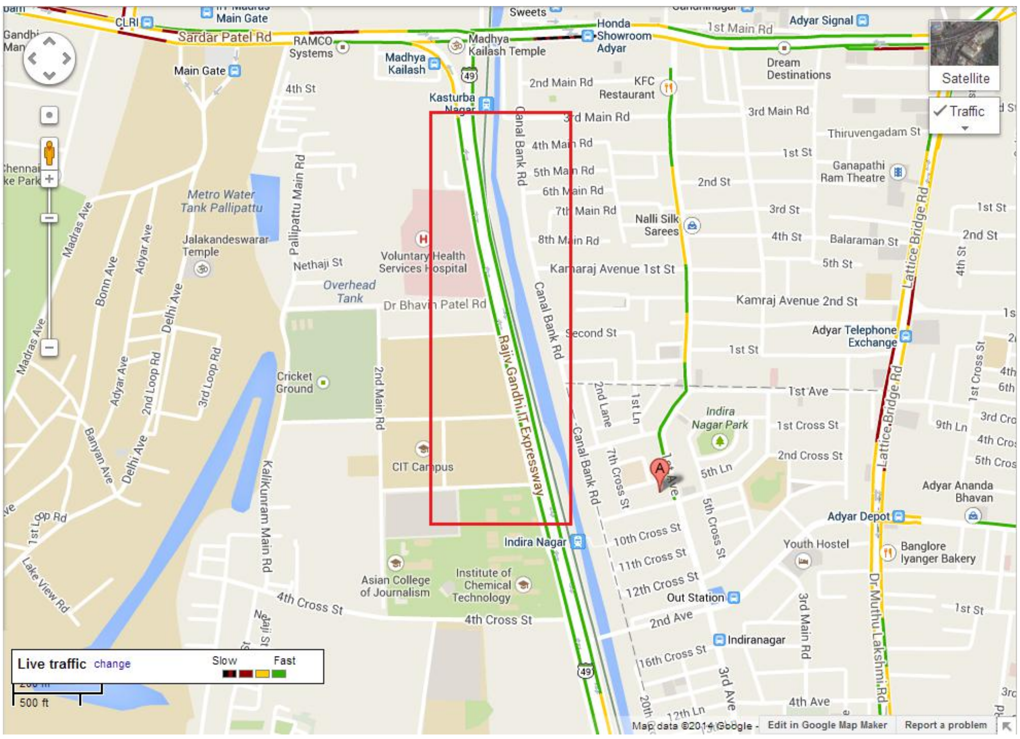
Comparison with Google Maps and WAZE:

We performed a test to check the accuracy of our project against Google Maps and WAZE. The test was performed on the Indira Nagar Road (from Tidel Park to Madhya Kailash, Chennai). The results are shown to be more or less similar, our work had the added advantage of estimating the area covered in addition to estimating the correct traffic.

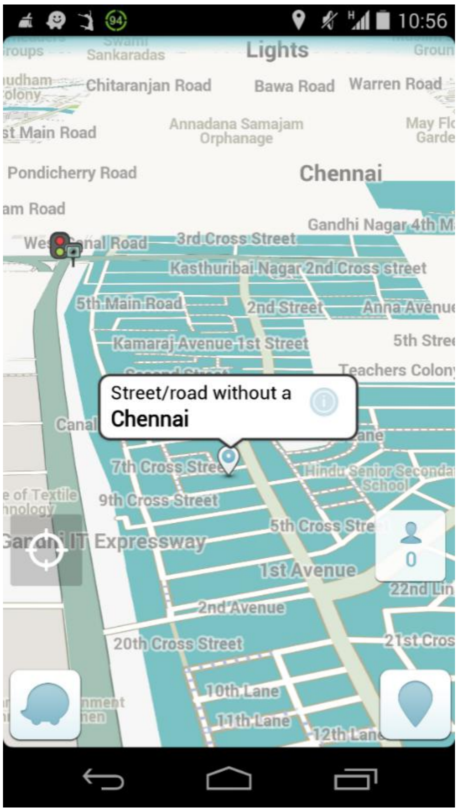
Traffic Estimation using the proposed framework



Google Maps



WAZE



All the three applications show the same traffic, which proves that the module works properly. Thus our project shows the correct traffic on the road, using reliable information rather than crowdsourcing.

Future Enhancements

The following enhancements are proposed for future work to make the system even better:

- Achieving better accuracy rates when there is heavy rainfall.
- Automatic lane detection.
- Using a sensor to record brightness levels and update threshold value, independent of the system clock.

Conclusion

The study presented explains computer vision based approach to road monitoring and traffic analysis problem. Tasks such as vehicle motion detection and area coverage calculation are considered. It is also more consistent in detecting vehicle presence because it uses actual traffic images captured from live video stream. It visualizes the reality so it functions much better than those systems that rely only on crowdsourcing. System implementation confirmed theoretical and design findings, and performed well in comparison with Google Maps and WAZE. Overall, a system that can be relied for traffic estimation purposes.

Acknowledgment

We would like to thank the Transportation Engineering Division of the Department of Civil Engineering at the Indian Institute of Technology, Madras for their support and guidance through the tenure of the project. We would also like to thank our department - Computer Science and Engineering, Sri Venkateswara College of Engineering and our guide and head, Dr. Susan Elias for the exposure and opportunity provided.