

July 2012

## Image Processing Based Intelligent Traffic Controller

Vikramaditya Dangi

*Electronics and Telecommunication Dept., Sardar Patel Institute of Technology, Mumbai, India,*  
vikdangi@hotmail.com

Amol Parab

*Electronics and Telecommunication Dept., Sardar Patel Institute of Technology, Mumbai, India,*  
amol\_aldiablo@yahoo.co.in

Kshitij Pawar

*Electronics and Telecommunication Dept., Sardar Patel Institute of Technology, Mumbai, India,*  
kpawar90@gmail.com

S.S Rathod

*Electronics and Telecommunication Dept., Sardar Patel Institute of Technology, Mumbai, India,*  
rathod\_spce@yahoo.com

Follow this and additional works at: <https://www.interscience.in/uarj>



Part of the [Business Commons](#), [Education Commons](#), [Engineering Commons](#), [Law Commons](#), [Life Sciences Commons](#), and the [Physical Sciences and Mathematics Commons](#)

---

### Recommended Citation

Dangi, Vikramaditya; Parab, Amol; Pawar, Kshitij; and Rathod, S.S (2012) "Image Processing Based Intelligent Traffic Controller," *Undergraduate Academic Research Journal*: Vol. 1 : Iss. 1 , Article 5.  
DOI: 10.47893/UARJ.2012.1004

Available at: <https://www.interscience.in/uarj/vol1/iss1/5>

This Article is brought to you for free and open access by the Interscience Journals at Interscience Research Network. It has been accepted for inclusion in Undergraduate Academic Research Journal by an authorized editor of Interscience Research Network. For more information, please contact [sritampatnaik@gmail.com](mailto:sritampatnaik@gmail.com).

# Image Processing Based Intelligent Traffic Controller

Vikramaditya Dangi, Amol Parab, Kshitij Pawar & S.S Rathod

Electronics and Telecommunication Dept., Sardar Patel Institute of Technology, Mumbai, India  
E-mail : vikdangi@hotmail.com, amol\_aldialblo@yahoo.co.in, kpawar90@gmail.com, rathod\_spce@yahoo.com

**Abstract** - The frequent traffic jams at major junctions call for an efficient traffic management system in place. The resulting wastage of time and increase in pollution levels can be eliminated on a city-wide scale by these systems. The paper proposes to implement an intelligent traffic controller using real time image processing. The image sequences from a camera are analyzed using various edge detection and object counting methods to obtain the most efficient technique. Subsequently, the number of vehicles at the intersection is evaluated and traffic is efficiently managed. The paper also proposes to implement a real-time emergency vehicle detection system. In case an emergency vehicle is detected, the lane is given priority over all the others.

**Keywords** - Adaptive Background Subtraction, Edge Detection, Emergency vehicles, Image Processing, Traffic Management.

## I. INTRODUCTION

Current traffic control techniques involving magnetic loop detectors buried in the road, infra-red and radar sensors on the side provide limited traffic information and require separate systems for traffic counting and for traffic surveillance.

Inductive loop detectors do provide a cost-effective solution, however they are subject to a high failure rate when installed in poor road surfaces, decrease pavement life and obstruct traffic during maintenance and repair. Infrared sensors are affected to a greater degree by fog than video cameras and cannot be used for effective surveillance.

In contrast, video-based systems offer many advantages compared to traditional techniques. They provide more traffic information, combine both surveillance and traffic control technologies, are easily installed, and are scalable with progress in image processing techniques. This paper tries to evaluate the process and advantages of the use of image processing for traffic control. Implementation of our project will eliminate the need of traffic personnel at various junctions for regulating traffic. Thus the use of this technology is valuable for the analysis and performance improvement of road traffic.

Also priority to emergency vehicles has been the topic of some research in the past. A proposed system for detection of these vehicles as in [1] is based on Radio-Frequency Identification (RFID). However, the use of this technology necessitates unnecessary extra hardware to be installed both at every junction and in every vehicle. There have also been studies to recognize

these vehicles by analysis of the sound of their siren as shown in [2]. However, this technology is also easily influenced by noise and requires additional hardware at every traffic signal.

## II. PROPOSED SYSTEM

### A. System Overview

The various steps of our proposed system are described in Fig 1. A camera is fixed on polls or other tall structures to overlook the traffic scene as seen in [3]. Images extracted from the video are then analysed to detect and count vehicles. Then depending on the signal-cycle (we have taken it to be 3 minutes), time is allotted to each lane. For example, if the number of vehicles in a four-lane intersection is found to be 10, 30, 20 and 20, then time allotted to each lane is in the ratio 1:3:2:2. The system also takes into account the emergency vehicles at the intersection. If such a vehicle is detected, the lane is given priority over the others.

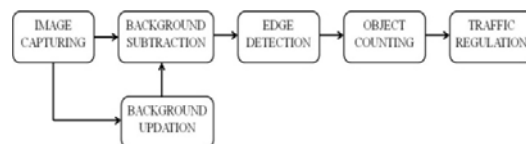


Fig. 1 : System Overview.

### B. Background Subtracion

Static background subtraction' has been the traditional method for real-time segmentation of an object in video based system. The technique is based on computing the error between a constant background frame and the current one. Video-based techniques for

outdoor environments are easily influenced by factors such as weather, change in illumination and motion. Hence, a static background proves insufficient and a robust background model is necessary to deal with change of luminance.

We propose the use of the adaptive background technique as described in [4]. Generating the current background image based on segmentation results extracted from differencing the image with the previous extracted background is the basic idea of our method. The updated background ( $B_{new}$ ) is computed as a function of current background ( $B_o$ ) and current frame  $I$  through the equation:

$$B_{new} = \begin{cases} (\alpha * B_o) + (1 - \alpha) * I & \text{if } RES = 0 \\ B_o, & \text{otherwise} \end{cases} \quad (1)$$

Where RES is the result of subtraction of consecutive frames, and the value of  $\alpha$  is 0.5.

The model hence accounts for the changes in background and reduces the error caused by them as shown in (1).

Once we have developed an adaptively changing background model, our next step is to separate the foreground from the background of the image. This is done by a pixel-by-pixel comparison of the current frame with the background at that instant. A pixel would be part of the foreground, when its value is different enough from its corresponding value in the background model. The edges and objects are then recognized on the basis of a predefined threshold.

### C. Edge Detection

After separating the foreground objects, we need to define their edges in the subtracted image. This is done by using an edge detection algorithm. There are a variety of edge detection techniques that have been used in the past [5]. Simple techniques such as the Boolean edge detector converts a window of pixels into a binary pattern based on a local threshold, and then applies masks to determine if an edge exists at a certain point or not. In the Marr-Hildreth Edge Detector, we smoothen the image using a Gaussian and Laplacian function. This takes the second derivative of an image. If there is a step difference in the intensity of the image, it will be represented in the second derivative by a zero crossing.

The Sobel operator is a discrete differentiation operator, computing an approximation of the gradient of the image intensity function. At each point in the image, the result of the Sobel operator is either the corresponding gradient vector or the norm of this vector. The operator consists of a pair of  $3 \times 3$  convolution kernels designed to respond maximally to edges running

vertically and horizontally relative to the pixel grid, one kernel for each of the two perpendicular orientations. The method finds edges using the Sobel approximation to the derivative. It then returns edges at those points where the gradient of the image is the maximum. Prewitt operator is similar to the Sobel operator used for detecting vertical and horizontal edges in images. It is a fast method only suitable for well-contrasted noiseless images.

The Canny edge detector is considered to be one of the most widely used edge detection algorithms in the industry. It works by first smoothing the image and finds the image gradient to highlight regions with high spatial derivatives. It then tracks along these regions to suppress any pixel that is not at the maximum. Finally, through hysteresis, it uses two thresholds and if the magnitude is below the first threshold, it is set to zero. If the magnitude is above the high threshold, it is made an edge and if the magnitude is between the two thresholds, it is set to zero unless there is a path from this pixel to a pixel with a gradient above the second threshold. That is to say that the two thresholds are used to detect strong and weak edges, and include the weak edges in the output only if they are connected to strong edges.

### D. Background Subtracion

After finding the edges the next stage is to count the number of objects as defined by the edges. There have been many algorithms suggested for object detection and contour tracing. These include the commonly used Radial Sweep method, Theo Pavlidis' Algorithm and Square Tracing Algorithm. However, in this paper we have implemented the Moore-neighbourhood algorithm based on a similar method as in [6]. The algorithm starts by choosing a random start point. When the current pixel 'p' is black, the Moore neighbourhood of 'p' is examined in clockwise direction starting with the pixel from which 'p' was entered and advancing pixel-by-pixel until a new black pixel in 'p' is encountered. The algorithm terminates when the start pixel is visited for the second time. The black pixel walked over will be the contour of the pattern. The efficiency of the algorithm improves greatly when we stop only after entering the start pixel in the same manner as entered initially. This is known as Jacob's stopping criteria. We have implemented this algorithm which does a decent job of identifying the number of cars in a given picture.

The contour tracing algorithm enables us to define the boundary of the object as well as their size. We specify different size ranges to classify the various types of vehicles. This gives us a measure of the traffic density on each road at the intersection (refer Fig. 3(d)). The traffic light is then regulated by allotting variable time according to the measured density and size of the vehicles.

### E. Emergency Vehicle Condition

In case a red beacon is detected, the next step is to identify whether it is from an emergency vehicle or not. This is done by identifying the blinking frequency of red light detected in the image sequence and comparing it to the standard used by the emergency vehicles.

The conditions for detection of red light beacon during various periods of the day are shown below. Once they are satisfied, we scan the intermediate frames for the absence of the beacon by the condition as shown below.

Night time conditions:

For red light:  $R > 230$ ,  $G < 250$ ,  $B < 250$

In the intermediate frames:  $R < 230$ ,  $G > 230$ ,  $B > 230$

Day time conditions:

For red light beacon:  $R > 230$ ,  $G < 250$ ,  $B < 250$

In the intermediate frames:  $R < 230$ ,  $G < 230$ ,  $B < 230$

If matched, the normal system is overridden and the lane is given priority over all the others. The lane is turned green until the vehicle has passed the intersection.

### III. RESULTS AND DISCUSSIONS

To compare between various types of edge detection algorithms we tested their performance for ten images taken from real traffic intersections. After finding the edges, the picture was subjected to an object counting algorithm. The performance of the edge detector algorithms was defined by the number of vehicles accurately detected. The results are shown in Table I. Canny Edge detector was found to be the best among those compared (93.47%).

The Boolean edge detector performs a decent job of marking the locations of edges, however it failed to complete the edges making object detection difficult. The Sobel and Prewitt operators are more adept at recognizing edges that are horizontal or vertical and are susceptible to noise (refer Fig 2), as also found in [7]. The Marr-Hildreth was found to be the most susceptible to noise and gave a lot of false results. The use of two thresholds by Canny edge detector makes it less likely to be fooled by noise, and more likely to detect true weak edges, providing a better and fairly noise resistant method for the detection of edges. Hence we have used this method of detection in the paper, along with Moore neighbourhood method to count the vehicles marking the final step of our system.

TABLE I : COMPARISON OF EDGE DETECTION TECHNIQUES

Image no	Actual no. of objects	Boolean	Marr Hildreth	Sobel	Prewitt	Canny
1	4	2	6	2	2	4
2	3	0	4	1	1	2
3	4	2	3	2	3	4
4	5	2	3	2	3	6
5	5	2	3	3	3	5
6	7	3	5	3	2	6
7	4	1	5	1	1	4
8	5	2	5	3	2	5
9	3	0	3	0	1	2
10	6	4	3	2	3	6
<b>Accuracy %</b>		39.13	84.78	41.30	45.65	93.47

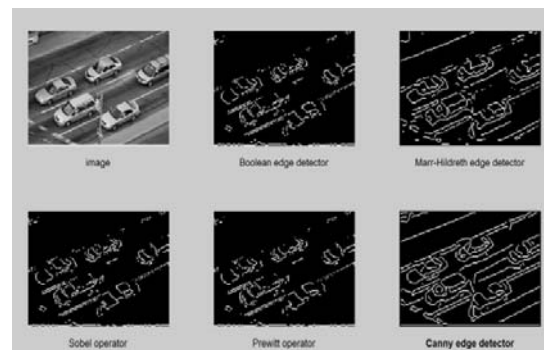


Fig. 2 : Output of various edge detection techniques.

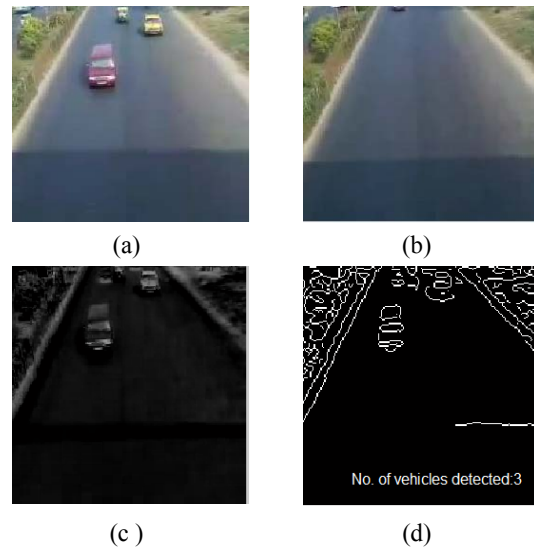


Fig. 3 : (a)Real-time image (b)Background image (c)Subtracted image (d)No. of vehicles = 3

The proposed system is used to analyze a real time traffic scene for a road (Fig 3(a)). The adaptive background, updated from the scenes is shown in Fig.

3(b). The subtracted image then contains only the foreground objects (vehicles) as seen in Fig 3(c). Using image processing algorithms (Fig 3(d)), the number of vehicles in the lane can be found out. In this case, the number of vehicles is 3.

The video is also analysed for the detection of emergency vehicles through their flashing red lights. By specifying a threshold, we have isolated the areas with high intensity of red light and comparatively lesser intensity of blue and green colour. The resultant image is shown in Fig. 4(b). As we can see, the headlights of the vehicle were also detected, which led to an erroneous output. Hence the red light must satisfy the additional condition of blinking. This is achieved by taking account for the fact that the red light shall appear in every third frame only. The other lights do not appear in the image sequence with this frequency and hence are eliminated. This leads to the conclusion of the presence of an emergency vehicle as shown in Fig. 4(c).

Our model was tested for ambulance during various times of the day and was found to be successful. In addition, the beacon can be identified even if the emergency vehicle is in an inclined position with respect to the camera as seen in Fig. 4(c).



Fig. 4 : a) Image of a vehicle during daytime, (b) Detection of all lights, (c) Emergency vehicle detected

#### IV. CONCLUSION

In this project, we have successfully implemented an algorithm for a real-time image processing based traffic controller. Upon comparison of various edge detection algorithms, it was inferred that Canny Edge Detector technique is the most efficient one. Analysis of various contour tracing and object counting methods revealed the Moore neighbourhood technique to be more robust when compared to the others. The paper demonstrates that image processing is a far more

efficient method of traffic control as compared to traditional techniques. We have also implemented a system for emergency vehicle detection based on image processing techniques. The use of our algorithm removes the need for extra hardware such as sound sensors or RFID tags. The increased response time for these vehicles is crucial for the prevention of loss of life.

#### V. FUTURE WORK

The focus shall be to implement the controller using DSP as it can avoid heavy investment in industrial control computer while obtaining improved computational power and optimized system structure. The hardware implementation would enable the project to be used in real-time practical conditions. More information about this method can be found in [8]. In addition, we propose a system to identify the vehicles as they pass by, giving preference to emergency vehicles and assisting in surveillance on a large scale.

#### REFERENCES

- [1] Ahmed S. Salama, Bahaa K. Saleh, Mohamad M. Eassa, "Intelligent Cross Road Traffic Management System (ICRTMS)," 2nd Int. Conf. on Computer Technology and Development, Cairo, Nov 2010, pp. 27-31.
- [2] B. Fazenda, H. Atmoko, F. Gu, L. Guan1 and A. Ball, "Acoustic Based Safety Emergency Vehicle Detection for Intelligent Transport Systems," ICCAS-SICE, Fukuoka, Aug 2009, pp.4250-4255.
- [3] Y. Wu, F. Lian, and T. Chang, "Traffic monitoring and vehicle tracking using roadside camera," IEEE Int. Conf. on Robotics and Automation, Taipei, Oct 2006, pp. 4631– 4636.
- [4] Z. Jinglei, L. Zhengguang, and T. Univ, "A vision-based road surveillance system using improved background subtraction and region growing approach," Eighth ACIS Int. Conf. on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, Qingdao, August 2007, pp. 819-822.
- [5] M. Siyal, and J. Ahmed, "A novel morphological edge detection and window based approach for real-time road data control and management," Fifth IEEE Int. Conf. on Information, Communications and Signal Processing, Bangkok, July 2005, pp. 324-328.
- [6] K. Wang, Z. Li, Q. Yao, W. Huang, and F. Wang, "An automated vehicle counting system for traffic surveillance," IEEE Int. Conf. on Vehicular Electronics and Safety, Japan, Dec 2007, pp. 1-6.

- [7] M. Juneja, P. Sandhu, "Performance evaluation of edge detection techniques for images in spatial domain," *Int. Journal of Computer Theory and Engineering*, Singapore, vol. 1, no. 5, Dec 2009, pp. 1793-8201.
- [8] K. Hang, "Real-time image acquisition and processing system design based on DSP," *The 2nd IEEE Int. Conf. on Computer and Automation Engineering (ICCAE)*, Singapore, Feb 2010, pp. 492-496.

