INTELLIGENT TRAFFIC CONTROL USING IMAGE PROCESSING

DATTA SAINATH DWARAMPUDI

MANISH H. SIRWANI

SHASHANK PANJALA

Department of Electronics and Communication Engineering

MAHATMA GANDHI INSTITUTE OF TECHNOLOGY
(Affiliated to Jawaharlal Nehru Technological University, Hyderabad, T.S.)

Chaitanya Bharathi P.O., Gandipet, Hyderabad – 500 075

2016

INTELLIGENT TRAFFIC CONTROL USING IMAGE PROCESSING

MINI PROJECT REPORT

SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

BY

DATA SAINATH DWARAMPUDI (12261A0409)
MANISH H. SIRWANI (12261A0430)
SHASHANK PANJALA (12261A0441)

Department of Electronics and Communication Engineering

MAHATMA GANDHI INSTITUTE OF TECHNOLOGY
(Affiliated to Jawaharlal Nehru Technological University, Hyderabad, T.S.)

Chaitanya Bharathi P.O., Gandipet, Hyderabad – 500 075

2016

MAHATMA GANDHI INSTITUTE OF TECHNOLOGY

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad, T.S.)

Chaitanya Bharathi P.O., Gandipet, Hyderabad-500 075

Department of Electronics and Communication Engineering

CERTIFICATE

Date: April 22, 2016.

This is to certify that the Major Project Work entitled "**Intelligent Traffic Control Using Image Processing**" is a bonafide work carried out by

Datta Sainath D.	**(12261A0409)**
Manish H. S.	**(12261A0430)**
Shashank P.	**(12261A0441)**

in partial fulfillment of the requirements for the degree of **BACHELOR OF TECHNOLOGY** in **ELECTRONICS & COMMUNICATION ENGINEERING** by the Jawaharlal Nehru Technological University, Hyderabad during the academic year 2015-16.

The results embodied in this report have not been submitted to any other University or Institution for the award of any degree or diploma.

Dr. T. D. Bhatt, Assc. Professor
Faculty Advisor/Liaison

Dr. S P Singh
Professor & Head

ACKNOWLEDGEMENT

We express our deep sense of gratitude to our Guide Dr. T. D. Bhatt for his invaluable guidance and encouragement in carrying out our Project.

We wish to express our sincere thanks to Dr. S.P.Singh, Head of the Department of Electronics and Communication Engineering, M.G.I.T., for giving us all the necessary technical guidance and encouraging us throughout the Project.

Finally, we thank all the people who have directly or indirectly helped us through the course of our Project.

Datta Sainath D.
Manish H. S.
Shashank P.

Abstract

In modern life we have to face with many problems one of which is traffic congestion becoming more serious day after day. It is said that the high tome of vehicles, the scanty infrastructure and the irrational distribution of the development are main reasons for augmented traffic jam. The major cause leading to traffic jam is the high number of vehicle which was caused by the population and the development of economy. To unravel this problem, the government should encourage people to use public transport or vehicles with small size such as bicycles or make tax on personal vehicles. Particularly, in some Asian countries such as Viet Nam, the local authorities passed law limiting to the number of vehicles for each family. The methods mentioned above are really efficient in fact. That the inadequate infrastructure cannot handle the issue of traffic is also a decisive reason. The public conveyance is available and its quality is very bad, mostly in the establishing countries. Besides, the highway and roads are incapable of meeting the requirement of increasing number of vehicle. Instead of working on roads to accommodate the growing traffic various techniques have been devised to control the traffic on roads like embedded controllers that are installed at the junction.

"Intelligent traffic control using image processing" technique that we propose overcomes all the limitations of the earlier (in use) techniques used for controlling the traffic. Earlier in automatic traffic control use of timer had a drawback that the time is being wasted by green light on the empty. This technique avoids this problem. Upon comparison of various edge detection algorithms, it was inferred that Canny Edge Detector technique is the most efficient one. The project demonstrates that image processing is a far more efficient method of traffic control as compared to traditional techniques. The use of our technique removes the need for extra hardware such as sound sensors. The increased response time for these vehicles is crucial for the prevention of loss of life. Major advantage is the variation in signal time which control appropriate traffic density using Image matching. The accuracy in calculation of time due to single moving camera depends on the registration position while facing road every time. Output of our code clearly indicated some expected results. It showed matching in almost every interval that were decided as lots of traffic, more traffic and less traffic.

Table of contents

LIST OF FIGURES

CHAPTER 1

CHAPTER 1
INTRODUCTION

In modern life we have to face with many problems one of which is traffic congestion becoming more serious day after day. It is said that the high tome of vehicles, the scanty infrastructure and the irrational distribution of the development are main reasons for augmented traffic jam. The major cause leading to traffic jam is the high number of vehicle which was caused by the population and the development of economy. To unravel this problem, the government should encourage people to use public transport or vehicles with small size such as bicycles or make tax on personal vehicles. Particularly, in some Asian countries such as Viet Nam, the local authorities passed law limiting to the number of vehicles for each family. The methods mentioned above are really efficient in fact. That the inadequate infrastructure cannot handle the issue of traffic is also a decisive reason. The public conveyance is available and its quality is very bad, mostly in the establishing countries. Besides, the highway and roads are incapable of meeting the requirement of increasing number of vehicle. Instead of working on roads to accommodate the growing traffic various techniques have been devised to control the traffic on roads like embedded controllers that are installed at the junction. These techniques are briefly described in next section.

1.1 STANDARD TRAFFIC CONTROL SYSTEM

1.1.1 MANUAL CONTROLLING

Manual controlling the name instance it require man power to control the traffic. Depending on the countries and states the traffic polices are allotted for a required area or city to control traffic. The traffic polices will carry sign board, sign light and whistle to control the traffic. They will be instructed to wear specific uniforms in order to control the traffic.

1.1.2 AUTOMATIC CONTROLLING

Automatic traffic light is controlled by timers and electrical sensors. In traffic light each phase a constant numerical value loaded in the timer. The lights are automatically getting ON and OFF depending on the timer value changes. While using electrical sensors it will capture the availability of the vehicle and signals on each phase, depending on the signal the lights automatically switch ON and OFF.

1.2 DRAWBACKS:

In the manual controlling system we need more man power. As we have poor strength of traffic police we cannot control traffic manually in all area of a city or town. So we need a better solution to control the traffic. On the other side, automatic traffic controlling a traffic light uses timer for every phase. Using electronic sensors is another way in order to detect vehicles, and produce signal that to this method the time is being wasted by a green light on an empty road. Traffic congestion also occurred while using the electronic sensors for controlling the traffic. All these drawbacks are supposed to be eliminated by using image processing.

1.3 IMAGE PROCESSING IN TRAFFIC LIGHT CONTROL

We propose a system for controlling the traffic light by image processing. The vehicles are detected by the system through images instead of using electronic sensors embedded in the pavement. A camera will be placed alongside the traffic light. It will capture image sequences. Image processing is a better technique to control the state change of the traffic light. It shows that it can decrease the traffic congestion and avoids the time being wasted by a green light on an empty road. It is also more reliable in estimating vehicle presence because it uses actual traffic images. It visualizes the practicality, so it functions much better than those systems that rely on the detection of the vehicles' metal content.

1.4 INTRODUCTION TO IMAGE PROCESSING

Image Processing is a technique to enhance raw images received from cameras/sensors placed on space probes, aircrafts and satellites or pictures taken in normal day-today life for various applications. An Image is rectangular graphical object. Image processing involves issues related to image representation, compression techniques and various complex operations, which can be carried out on the image data. The operations that come under image processing are image enhancement operations such as sharpening, blurring, brightening, edge enhancement etc. Image processing is any form of signal processing for which the input is an image, such as photographs or frames of video; the output of image processing can be either an image or a set of characteristics or parameters related to the image. Most image-processing techniques involve treating the image as a two-dimensional signal and applying standard signal-processing techniques to it. Image processing usually refers to digital image processing, but optical and analog image processing are also possible.

CHAPTER 2

CHAPTER 2

DETAILED DESCRIPTION OF OUR PROJECT

Many techniques have been developed in Image Processing during the last four to five decades. Most of the methods are developed for enhancing images obtained from unmanned space probes, spacecrafts and military reconnaissance flights. Image Processing systems are becoming widely popular due to easy availability of powerful personnel computers, large memory devices, graphics softwares and many more.

Image processing involves issues related to image representation, compression techniques and various complex operations, which can be carried out on the image data. The operations that come under image processing are image enhancement operations such as sharpening, blurring, brightening, edge enhancement .Traffic density of lanes is calculated using image processing which is done of images of lanes that are captured using digital camera. We have chosen image processing for calculation of traffic density as cameras are very much cheaper than other devises such as sensors.

Making use of the above mentioned virtues of image processing we propose a technique that can be used for traffic control. The block diagram of the proposed algorithm is given on next page.

2.1 BLOCK DIAGRAM

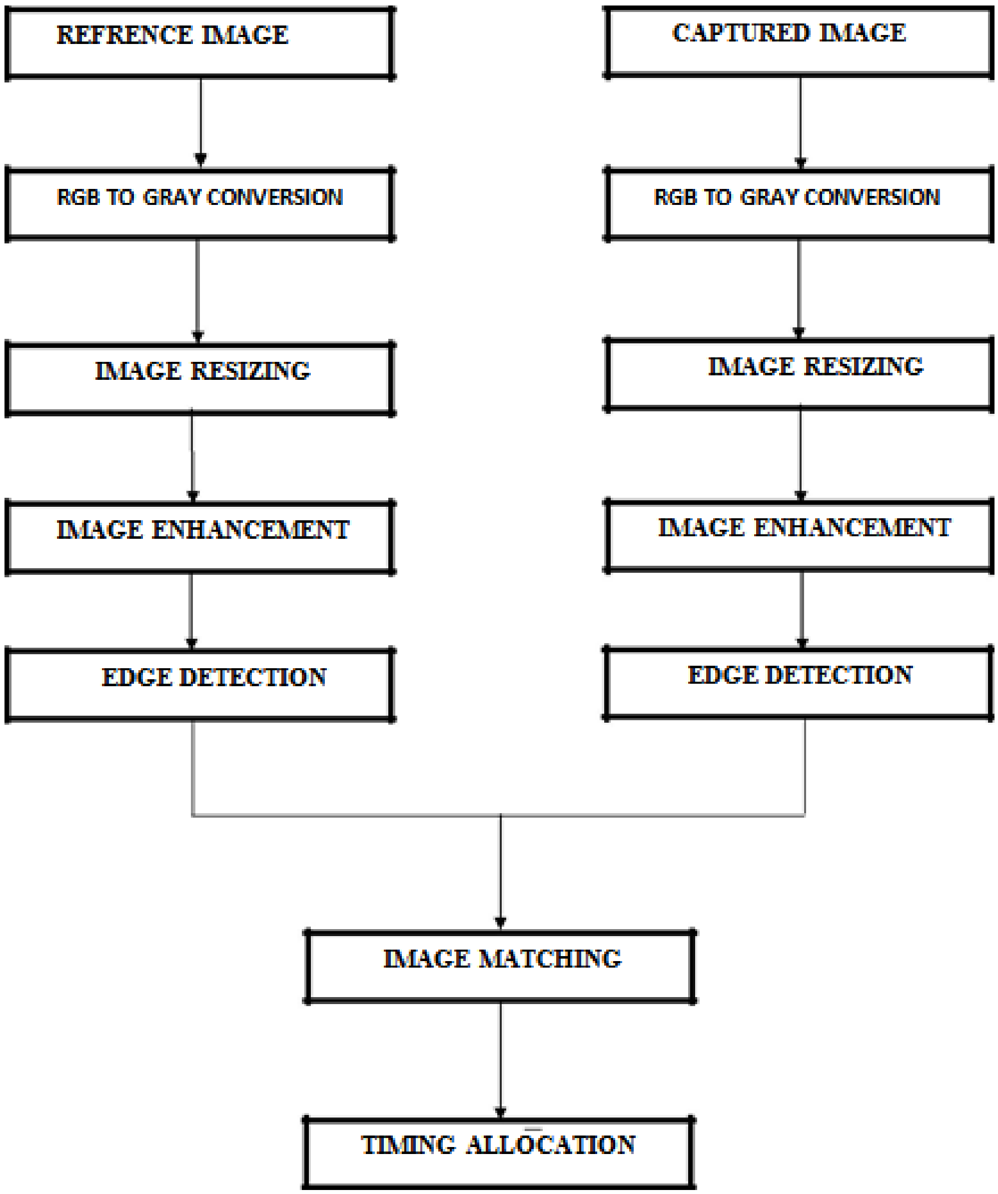

Figure 2.1 shows Block Diagram of "Traffic Control Using Image Processing" (proposed algorithm)

The Block diagram above gives an overview of how traffic will be controlled using image processing. Various boxes in Block diagram are explained below:

2.2 IMAGE ACQUISITION

Generally an image is a two-dimensional function f(x,y)(here x and y are plane coordinates).The amplitude of image at any point say f is called intensity of the image. It is also called the gray level of image at that point. We need to convert these x and y values to finite discrete values to form a digital image. The input image is a fundus taken from stare data base and drive data base. The image of the retina is taken for processing and to check the condition of the person. We need to convert the analog image to digital image to process it through digital computer. Each digital image composed of a finite elements and each finite element is called a pixel.

2.3 FORMATION OF IMAGE

We have some conditions for forming an image f(x,y) as values of image are proportional to energy radiated by a physical source. So f(x,y) must be nonzero and finite.

i.e. $\mathbf{0 < f(x,y) < \infty}$.

2.4 IMAGE PRE-PROCESSING

2.4.1 IMAGE RESIZING/SCALING

Image scaling occurs in all digital photos at some stage whether this be in Bayer demosaicing or in photo enlargement. It happens anytime you resize your image from one pixel grid to another. Image resizing is necessary when you need to increase or decrease the total number of pixels. Even if the same image resize is performed, the result can vary significantly depending on the algorithm.

Images are resized because of number of reasons but one of them is very important in our project. Every camera has its resolution, so when a system is designed for some camera specifications it will not run correctly for any other camera depending on specification similarities. so it is necessary to make the resolution constant for the application and hence perform image resizing.

2.4.2 RGB TO GRAY CONVERSION:

Humans perceive colour through wavelength-sensitive sensory cells called cones. There are three different varieties of cones, each has a different sensitivity to electromagnetic radiation (light) of different wavelength. One cone is mainly sensitive to green light, one to red light, and one to blue light. By emitting a restricted combination of these three colours (red, green and blue), and hence stimulate the three types of cones at will, we are able to generate almost any detectable colour. This is the reason behind why colour images are often stored as three separate image matrices; one storing the amount of red (R) in each pixel, one the amount of green (G) and one the amount of blue (B). We call such colour images as stored in an RGB format. In grayscale images, however, we do not differentiate how much we emit of different colours, we emit the same amount in every channel. We will be able to differentiate the total amount of emitted light for each pixel; little light gives dark pixels and much light is perceived as bright pixels. When converting an RGB image to grayscale, we have to consider the RGB values for each pixel and make as output a single value reflecting the brightness of that pixel. One of the approaches is to take the average of the contribution from each channel: (R+B+C)/3. However, since the perceived brightness is often dominated by the green component, a different, more "human-oriented", method is to consider a weighted average, e.g.: 0.3R + 0.59G + 0.11B.

2.5 IMAGE ENHANCEMENT

Image enhancement is the process of adjusting digital images so that the results are more suitable for display or further analysis. For example, we can eliminate noise, which will make it more easier to identify the key characteristics.

In poor contrast images, the adjacent characters merge during binarization. We have to reduce the spread of the characters before applying a threshold to the word image. Hence, we introduce “**POWER- LAW TRANSFORMATION**” which increases the contrast of the characters and helps in better segmentation. The basic form of power-law transformation is

$$s = cr^{\gamma},$$

where r and s are the input and output intensities, respectively; c and γ are positive constants. A variety of devices used for image capture, printing, and display respond according to a power-law. By convention, the exponent in the power-law equation is referred to as gamma. Hence, the process used to correct these power-law response phenomena is called gamma correction. Gamma correction is important, if displaying an image accurately on a computer screen is of concern. In our experimentation, γ is varied in the range of 1 to 5. If c is not equal to '1', then the dynamic range of the pixel values will be significantly affected by scaling. Thus, to avoid another stage of rescaling after power-law transformation, we fix the value of c = 1.With $\gamma = 1$, if the power-law transformed image is passed through binarization, there will be no change in the result compared to simple binarization. When $\gamma > 1$, there will be a change in the histogram plot, since there is an increase of samples in the bins towards the gray value of zero. Gamma correction is important if displaying an image accurately on computer screen is of concern.

2.6 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 technically, has discontinuities or noise. The points at which image brightness alters sharply are typically organized into a set of curved line segments termed edges.

The same problem of detecting discontinuities in 1D signal is known as step detection and the problem of finding signal discontinuities over time is known as change detection. Edge detection is a basic tool in image processing, machine vision and computer envisage, particularly in the areas of feature reveal and feature extraction.

a. EDGE DETECTION TECHNIQUES

Different colours have different brightness values of particular colour. Green image has more bright than red and blue image or blue image is blurred image and red image is the high noise image.

Following are list of various edge-detection methods:-

- Sobel Edge Detection Technique
- Perwitt Edge Detection
- Roberts Edge Detection Technique
- Zero cross Threshold Edge Detection Technique
- Canny Edge Detection Technique

In our project we use “CANNY EDGE DETECTION TECHNIQUE” because of its various advantages over other edge detection techniques.

b. CANNY EDGE DETECTION

The Canny Edge Detector is one of the most commonly used image processing tools detecting edges in a very robust manner. It is a multi-step process, which can be implemented on the GPU as a sequence of filters. Canny edge detection technique is based on three basic objectives.

I. Low error rate:-

All edges should be found, and there should be no spurious responses. That is, the edges must be as close as possible to the true edges.

II. Edge point should be well localized:-

The edges located must be as close as possible to the true edges. That is , the distance between a point marked as an edge by the detector and the centre of the true edge should be minimum.

III. Single edge point response:-

The detector should return only one point for each true edge point. That is, the number of local maxima around the true edge should be minimum. This means that the detector should not identify multiple edge pixels where only a single edge point exists.

The essence of Canny's work was in expressing the preceding three criteria mathematically and then attempting to find optimal solution to these formulations, in general, it is difficult to find a close form solution that satisfies all the preceding objectives. However, using numerical optimization with 1-D step edges corrupted by additive while Gaussian noise led to the conclusion that a good approximation to the optimal step edge detector is the first derivative of Gaussian:

$$\frac{d}{dx} e^{\frac{-x^2}{2\sigma^2}} = \frac{-x}{\sigma^2} e^{\frac{-x^2}{2\sigma^2}} \tag{1}$$

Generalizing this result to 2-D involves recognizing that the 1-D approach still applies in the direction of the edge normal. Because the direction of the normal is unknown beforehand, this would require applying the 1-D edge detector in all possible directions. This task can be approximated by first smoothing the image with circular 2-D Gaussian function, computing the gradient of the result, and then using the gradient magnitude and direction to estimate edge strength and direction at every point.

Let f(x,y) denote the input image and G(x,y) denote the Gaussian function:

$$G(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{2}$$

We form a smoothed image, $f_s(x, y)$, by convolving G and f:

$$\mathbf{f_s(x,y)} \quad = \quad \mathbf{G(x,y) * f(x,y)} \tag{3}$$

This operation is followed by computing the gradient and direction (angle)

$$\mathbf{M(x,y)} \quad = \quad \sqrt{\mathbf{g_x}^2 + \mathbf{g_y}^2} \tag{4}$$

And

$$\alpha(x,y) \quad = \quad \tan^{-1}\frac{g_y}{g_x} \tag{5}$$

With

$$\mathbf{g_x} = \frac{\partial \mathbf{f_s}}{\partial \mathbf{x}} \ \text{ and } \mathbf{g_y} = \frac{\partial f_s}{\partial y}$$

Equation (2) is implemented using an n x n Gaussian mask. Keep in mind that M(x, y) and (x, y) are arrays of the same size as the image from which they are computed. Because it is generated using the gradient M(x, y) typically contains wide ridges around local maxima.

The next step is to thin those ridges. One approach is to use non maxima suppression. This can be done in several ways, but the essence of the approach is to specify a no. of discrete orientations of edge normal (gradient vector). For example, in 3x3 region we can define four orientation for an edge passing through the centre point of the region: horizontal, vertical, +45° and -45°.

The final operation is to threshold $g_N(x, y)$ to reduce false edge point. We do it by using a single threshold, in which all value below the threshold were set to 0. If we set the threshold too low, there will still be some false edge (called false positives). If the threshold id set too high, then actual valid edge points will be eliminated (false negatives). Canny's algorithm attempts to improve on this situation by using hysteresis threshold. We use two thresholds, A low threshold, TL, and a high threshold, TH. Canny suggested that the ratio of high to low threshold should be two or three to one.

We visualize the thresholding operation as creating two additional images

$$\mathbf{g_{NH}(x, y) \quad = \quad g_N(x, y) \geq T_H} \tag{6}$$

$$\mathbf{g_{NL}(x, y) \quad = \quad g_N(x, y) \geq T_{\mathit{L}}} \tag{7}$$

The nonzero pixels in $g_{NL}(x, y)$ and $g_{NH}(x, y)$ may be viewed as being “strong”. And “weak” edge pixels, respectively.

Where, initially, both $g_{NH}(x, y)$ $g_{NL}(x, y)$ are set to 0. After thresholding, $g_{NH}(x, y)$ will have fewer nonzero pixels than $g_{NL}(x, y)$ in general, but all the nonzero pixels in $g_{NH}(x, y)$ will be contained in $g_{NL}(x, y)$ because the latter image is formed with lower threshold. We eliminate from $g_{NL}(x, y)$ all the nonzero from $g_{NH}(x, y)$ by letting

After the thresholding operations, all strong pixels in $g_{NH}(x, y)$ are assumed to be valid edge pixels and are so marked immediately. Depending on the value of T_H, the edges in $g_{NH}(x, y)$ typically have gaps.

Longer edges are formed using the following procedure

a. Locate the next unvisited pixel p in $g_{NH}(x, y)$.

b. Mark as valid edge pixels all the weak pixels in $g_{NL}(x, y)$ that are connected to p using say 8 connectivity.

c. If all nonzero pixels in $g_{NH}(x, y)$ have been visited go to step d. else return to step a.

d. Set to zero all pixels in $g_{NL}(x, y)$ those were not marked as valid edge pixels.

At the end of this procedure, the final image output by the Canny is formed by appending to $g_{NH}(x, y)$ all the nonzero pixels from $g_{NI}(x, y)$.

We use two additional images, $g_{NH}(x, y)$ and $g_{NI}(x, y)$, to simplify the discussion. In practice, hysteresis threshold can be implemented directly during non-maxima suppression, and thresholding can be implemented directly on $g_{NL}(x, y)$ by forming a list of strong pixels and the weak pixels connected to them.

Summarizing, the Canny edge detection algorithm consist of the following basic steps;

i. Smooth the input image with Gaussian filter.

ii. Compute the gradient magnitude and angle images.

iii. Apply non-maxima suppression to the gradient magnitude image.

iv. Use double thresholding and connectivity analysis to detect and link edges.

2.7 IMAGE MATCHING

Recognition techniques based on matching represent each class by a prototype pattern vector. An unknown pattern is assigned to the class to which is closest in terms of predefined metric. The simplest approach is the minimum distance classifier, which, as its name implies, computes the (Euclidean) distance between the unknown and each of the prototype vectors. It chooses the smallest distance to make decision. There is another approach based on correlation, which can be formulated directly in terms of images and is quite intuitive.

We have used a totally different approach for image matching. Comparing a reference image with the real time image pixel by pixel. Though there are some disadvantages related to pixel based matching but it is one of the best techniques for the algorithm which is used in the project for decision making. Real image is stored in matric in memory and the real time image is also converted in the desired matric. For images to be same their pixel values in matrix must be same. This is the simplest fact used in pixel matching. If there is any mismatch in pixel value it adds on to the counter used to calculate number of pixel mismatches. Finally percentage of matching is expressed as

$$\%\text{match} = \frac{\text{No.of pixels matched sucessfully}}{\text{total no.of pixels}}$$

The block diagram of the proposed algorithm discussed above is implemented in MATLAB R2013a. So it is necessary to gain an insight of MATLAB.

2.8 INTRODUCTION TO MATLAB

The name MATLAB stands for MATrix LABoratory. MATLAB was written originally to provide easy access to matrix software developed by the LINPACK (linear system package) and EISPACK (Eigen system package) projects.

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming environment. Furthermore, MATLAB is a modern programming language environment: it has sophisticated data structures, contains built-in editing and debugging tools, and supports object-oriented programming. These factors make MATLAB an excellent tool for teaching and research.

MATLAB has many advantages compared to conventional computer languages (e.g., C, FORTRAN) for solving technical problems. MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. The software package has been commercially available since 1984 and is now considered as a standard tool at most universities and industries worldwide.

It has powerful built-in routines that enable a very wide variety of computations. It also has easy to use graphics commands that make the visualization of results immediately available. Specific applications are collected in packages referred to as toolbox. There are toolboxes for signal processing, symbolic computation, control theory, simulation, optimization, and several other fields of applied science and engineering.

There are various tools in Matlab that can be utilized for image processing, such as Simulink, GUI etc. Simulink contains various toolboxes and image processing toolbox is one such example. Simulink is used for simulation of various projects. GUI is another important tool in Matlab. It can be designed either by manual programming which is tedious task or by using guide

2.9 MATLAB CODE

2.9.1 MAIN PROGRAM

```
%%%%%%%%%%Code preapared for project work%%%%%%%%%%
%%%%%%%%%%Author: Datta Sainath D%%%%%%%%%%
clc;
close all;
clear all;

%%%%%%%%%%Reference Image%%%%%%%%%%
RGB1 = imread('No Traffic.jpg');
%figure; imshow(RGB1); title('Reference Image');

%RGB to Gray Conversion
I1 = rgb2gray(RGB1);
ID1=im2double(I1);
%figure; imshow(ID1); title('Reference Image: Gray Image');

%Image Resizing
IR1 = imresize(ID1, [512 512]);
%figure; imshow(IR1); title('Resized Reference Image');

%Image Enhancement Power Law Transformation
c = 2;
g =0.9;
for p = 1 : 512
   for q = 1 : 512
      IG1(p, q) = c * IR1(p, q).^ 0.9;
   end
end
```

```
%figure; imshow(IG1); title('Enhanced Reference Image');

%Edge Detection
% The algorithm parameters:
% 1. Parameters of edge detecting filters:
%    X-axis direction filter:
    Nx1=10;Sigmax1=1;Nx2=10;Sigmax2=1;Theta1=pi/2;
%    Y-axis direction filter:
    Ny1=10;Sigmay1=1;Ny2=10;Sigmay2=1;Theta2=0;
% 2. The thresholding parameter alfa:
    alfa=0.1;
% Get the initial Reference Image
figure;
subplot(3,2,1);
imagesc(IG1);
title('Image: Reference Image');
% X-axis direction edge detection
subplot(3,2,2);
filterx=d2dgauss(Nx1,Sigmax1,Nx2,Sigmax2,Theta1);
Ix= conv2(IG1,filterx,'same');
imagesc(Ix);
title('Ix');
% Y-axis direction edge detection
subplot(3,2,3)
filtery=d2dgauss(Ny1,Sigmay1,Ny2,Sigmay2,Theta2);
Iy=conv2(IG1,filtery,'same');
imagesc(Iy);
title('Iy');
% Norm of the gradient (Combining the X and Y directional derivatives)
subplot(3,2,4);
NVI1=sqrt(Ix.*Ix+Iy.*Iy);
```

```
imagesc(NVI1);
title('Norm of Gradient');
% Thresholding
I_max=max(max(NVI1));
I_min=min(min(NVI1));
level=alfa*(I_max-I_min)+I_min;
subplot(3,2,5);
Ibw=max(NVI1,level.*ones(size(NVI1)));
imagesc(Ibw);
title('After Thresholding');
% Thinning (Using interpolation to find the pixels where the norms of
% gradient are local maximum.)
subplot(3,2,6);
[n,m]=size(Ibw);
for i=2:n-1,
for j=2:m-1,
   if Ibw(i,j) > level,
   X=[-1,0,+1;-1,0,+1;-1,0,+1];
   Y=[-1,-1,-1;0,0,0;+1,+1,+1];
   Z=[Ibw(i-1,j-1),Ibw(i-1,j),Ibw(i-1,j+1);
     Ibw(i,j-1),Ibw(i,j),Ibw(i,j+1);
     Ibw(i+1,j-1),Ibw(i+1,j),Ibw(i+1,j+1)];
   XI=[Ix(i,j)/NVI1(i,j), -Ix(i,j)/NVI1(i,j)];
   YI=[Iy(i,j)/NVI1(i,j), -Iy(i,j)/NVI1(i,j)];
   ZI=interp2(X,Y,Z,XI,YI);
      if Ibw(i,j) >= ZI(1) & Ibw(i,j) >= ZI(2)
      I_temp(i,j)=I_max;
      else
      I_temp(i,j)=I_min;
      end
   else
```

```
  I_temp(i,j)=I_min;
  end
end
end
imagesc(I_temp);
title('After Thinning');
colormap(gray);
I_temp1 = I_temp;

%%%%%%%%%%Captured Image%%%%%%%%%%
RGB2 = imread('Lots of Traffic.jpg');
%figure; imshow(RGB2); title('Captured Image');

%RGB to Gray Conversion: Captured Image
I2 = rgb2gray(RGB2);
ID2=im2double(I2);
%figure; imshow(ID2); title('Captured Image:Gray Image');

%Image Resizing: Captured Image
IR2 = imresize(ID2, [512 512]);
%figure; imshow(IR2); title('Resized Captured Image');

%Image Enhancement Power Law Transformation: Captured Image
for p = 1 : 512
  for q = 1 : 512
    IG2(p, q) = abs(c * IR2(p, q).^ 0.9);
  end
end
%figure; imshow(IG2); title('Enhanced Captured Image');

%Edge Detection
```

```
% Get the initial Captured Image
figure;
subplot(3,2,1);
imagesc(IG2);
title('Image: Captured Image');
% X-axis direction edge detection
subplot(3,2,2);
filterx=d2dgauss(Nx1,Sigmax1,Nx2,Sigmax2,Theta1);
Ix= conv2(IG2,filterx,'same');
imagesc(Ix);
title('Ix');
% Y-axis direction edge detection
subplot(3,2,3)
filtery=d2dgauss(Ny1,Sigmay1,Ny2,Sigmay2,Theta2);
Iy=conv2(IG2,filtery,'same');
imagesc(Iy);
title('Iy');
% Norm of the gradient (Combining the X and Y directional derivatives)
subplot(3,2,4);
NVI2=sqrt(Ix.*Ix+Iy.*Iy);
imagesc(NVI2);
title('Norm of Gradient');
% Thresholding
I_max=max(max(NVI2));
I_min=min(min(NVI2));
level=alfa*(I_max-I_min)+I_min;
subplot(3,2,5);
Ibw=max(NVI2,level.*ones(size(NVI2)));
imagesc(Ibw);
title('After Thresholding');
% Thinning (Using interpolation to find the pixels where the norms of
```

```
% gradient are local maximum.)
subplot(3,2,6);
[n,m]=size(Ibw);
for i=2:n-1,
for j=2:m-1,
    if Ibw(i,j) > level,
    X=[-1,0,+1;-1,0,+1;-1,0,+1];
    Y=[-1,-1,-1;0,0,0;+1,+1,+1];
    Z=[Ibw(i-1,j-1),Ibw(i-1,j),Ibw(i-1,j+1);
      Ibw(i,j-1),Ibw(i,j),Ibw(i,j+1);
      Ibw(i+1,j-1),Ibw(i+1,j),Ibw(i+1,j+1)];
    XI=[Ix(i,j)/NVI2(i,j), -Ix(i,j)/NVI2(i,j)];
    YI=[Iy(i,j)/NVI2(i,j), -Iy(i,j)/NVI2(i,j)];
    ZI=interp2(X,Y,Z,XI,YI);
        if Ibw(i,j) >= ZI(1) & Ibw(i,j) >= ZI(2)
        I_temp(i,j)=I_max;
        else
        I_temp(i,j)=I_min;
        end
    else
    I_temp(i,j)=I_min;
    end
end
end
imagesc(I_temp);
title('After Thinning');
colormap(gray);
I_temp2 = I_temp;


%%%%%%%%%%Image Matching%%%%%%%%%%
```

```
match = 0;
BW1 = im2bw(NVI1);
BW2 = im2bw(NVI2);
for p = 1 : 511
    for q = 1 : 511
        if (BW1(p, q) == BW2(p,q))
            match = match +1;
        end
    end
end

match;


%%%%%%%%%%Output Display%%%%%%%%%%

if(match>233000)
    disp('Green signal will be displayed for 10 second');
    disp('Red signal will be displayed for 50 seconds');
elseif(match>232000 && match <233000)
    disp('Green signal will be displayed for 20 second');
    disp('Red signal will be displayed for 40 seconds');
else
    disp('Green signal will be displayed for 30 second');
    disp('Red signal will be displayed for 30 seconds');
end
```

2.9.2 EDGE DETECTIN FUNCTION PROGRAM

```
%%%%%%% The functions used in the main.m file %%%%%%%
% Function "d2dgauss.m":
% This function returns a 2D edge detector (first order derivative
% of 2D Gaussian function) with size n1*n2; theta is the angle that
% the detector rotated counter clockwise; and sigma1 and sigma2 are the
% standard deviation of the gaussian functions.
function h = d2dgauss(n1,sigma1,n2,sigma2,theta)
r=[cos(theta) -sin(theta);
   sin(theta)  cos(theta)];
for i = 1 : n2
    for j = 1 : n1
        u = r * [j-(n1+1)/2 i-(n2+1)/2]';
        h(i,j) = gauss(u(1),sigma1)*dgauss(u(2),sigma2);
    end
end
h = h / sqrt(sum(sum(abs(h).*abs(h))));

% Function "gauss.m":
function y = gauss(x,std)
y = exp(-x^2/(2*std^2)) / (std*sqrt(2*pi));

% Function "dgauss.m"(first order derivative of gauss function):
function y = dgauss(x,std)
y = -x * gauss(x,std) / std^2;
%%%%%%%%%%%%%% end of the functions %%%%%%%%%%%%%
```

CHAPTER 3

CHAPTER 3

RESULTS AND CONCLUSION

3.1 RESULTS

3.1.1Original Images taken for Image Processing:

NO TRAFFIC SITUATION

Figure 3.1 shows Original Image taken for No Traffic Situation

VERY LESS TRAFFIC SITUATION

Figure 3.2 shows Original Image taken for Very Less Traffic Situation

LESS TRAFFIC SITUATUION

Figure 3.3 shows Original Image taken for Less Traffic Situation

MORE TRAFFIC SITUATION

Figure 3.4 shows Original Image taken for More Traffic Situation

LOTS OF TRAFFIC SITUATION

Figure 3.5 shows Original Image taken for Lots of Traffic Situation

3.1.2 Processed Images

Image Processing techniques were used on the input images and the following results were obtained:

NO TRAFFIC PROCESSED IMAGES

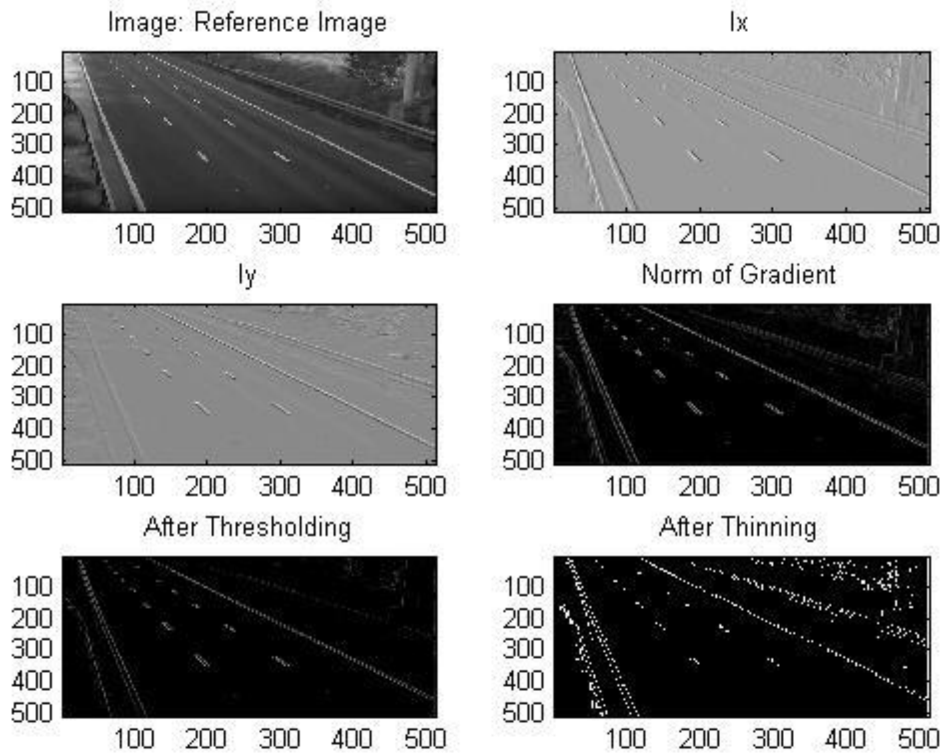

Figure 3.6 shows Processed Images for No Traffic Image

VERY LESS TRAFFIC PROCESSED IMAGES

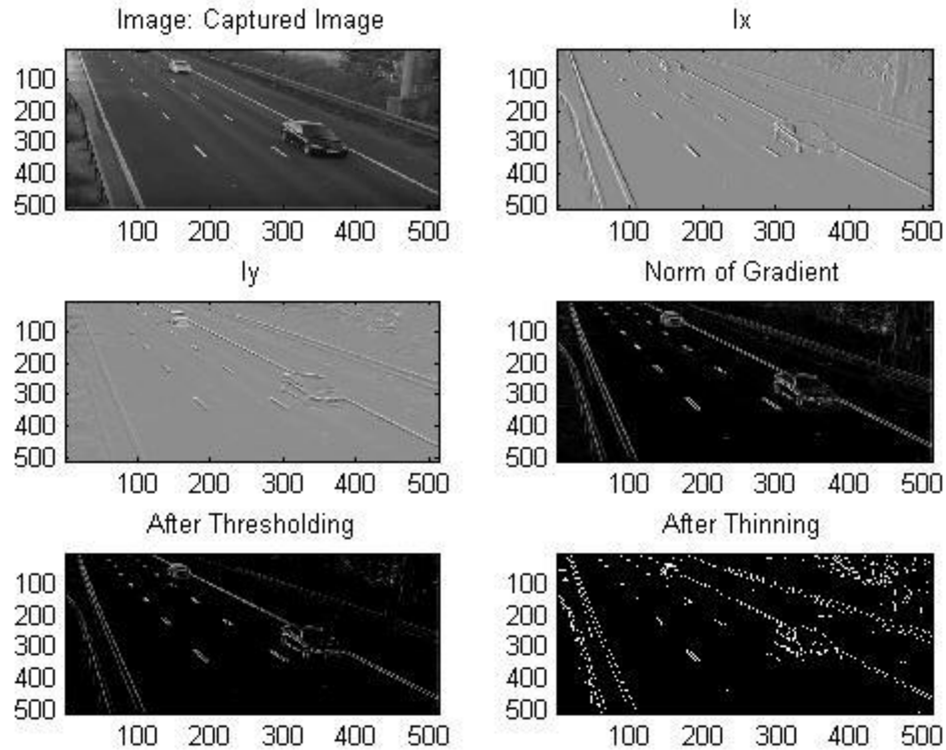

Figure 3.7 shows Processed Images for Very Less Traffic Image

LESS TRAFFIC PROCESSED IMAGES

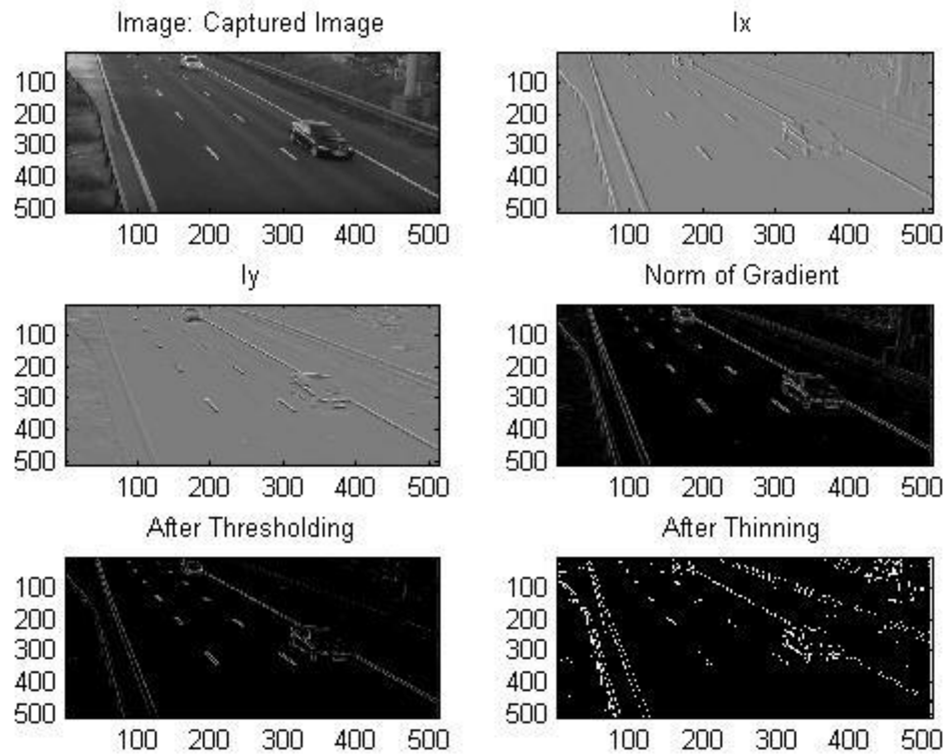

Figure 3.8 shows Processed Images for Less Traffic Image

MORE TRAFFIC PROCESSED IMAGES

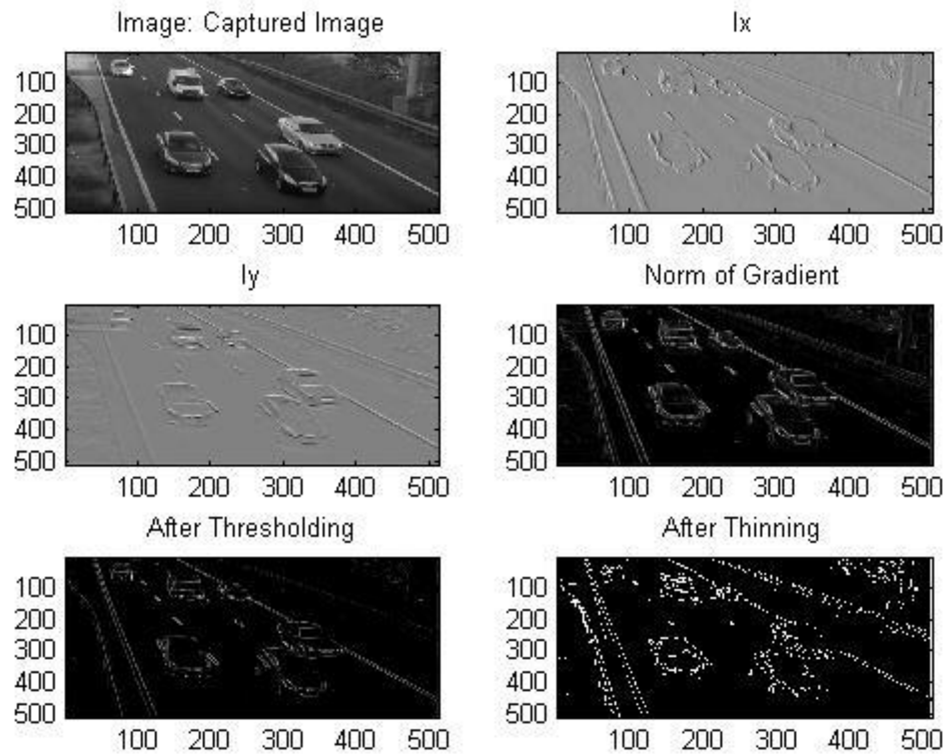

Figure 3.9 shows Processed Images for More Traffic Image

LOTS OF TRAFFIC PROCESSED IMAGES

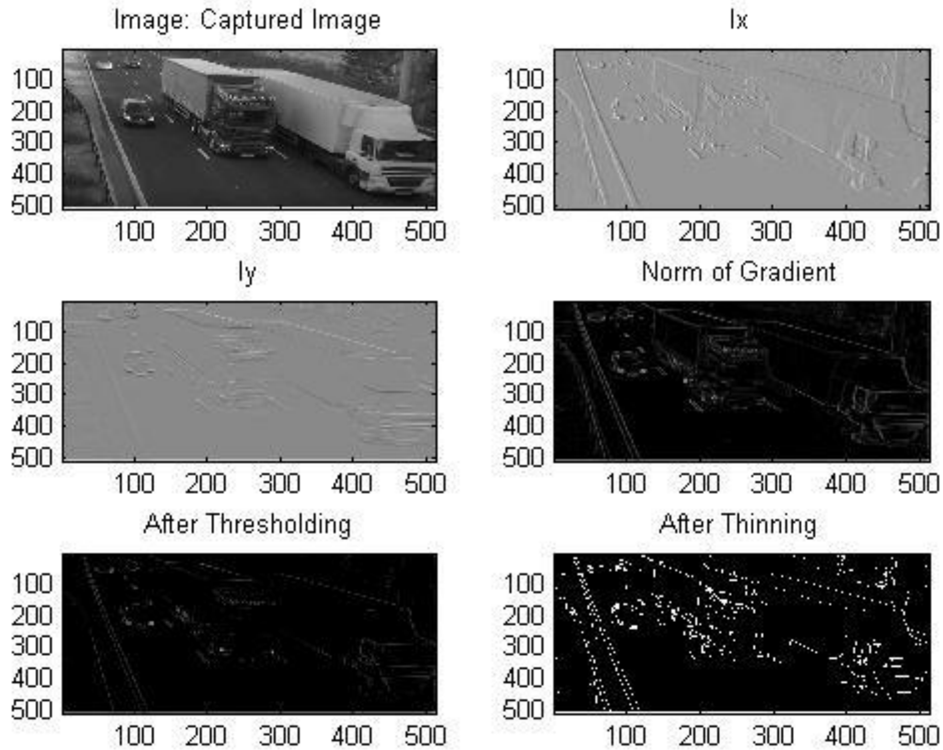

Figure 3.10 shows Processed Images for Lots of Traffic Image

3.1.3 Outputs

Following results were displayed from the processed images:

Very Less Traffic Situation Output:

Green signal will be displayed for 10 second

Red signal will be displayed for 50 seconds

Less Traffic Situation Output:

Green signal will be displayed for 10 second

Red signal will be displayed for 50 seconds

More Traffic Situation Output:

Green signal will be displayed for 20 second

Red signal will be displayed for 40 seconds

Lots of Traffic Situation Output:

Green signal will be displayed for 30 second

Red signal will be displayed for 30 seconds

3.2 CONCLUSION

“Intelligent traffic control using image processing” technique that we propose overcomes all the limitations of the earlier (in use) techniques used for controlling the traffic. Earlier in automatic traffic control use of timer had a drawback that the time is being wasted by green light on the empty. This technique avoids this problem. Upon comparison of various edge detection algorithms, it was inferred that Canny Edge Detector technique is the most efficient one. The project demonstrates that image processing is a far more efficient method of traffic control as compared to traditional techniques. The use of our technique removes the need for extra hardware such as sound sensors. The increased response time for these vehicles is crucial for the prevention of loss of life. Major advantage is the variation in signal time which control appropriate traffic density using Image matching. The accuracy in calculation of time due to single moving camera depends on the registration position while facing road every time. Output of our code clearly indicated some expected results. It showed matching in almost every interval that were decided as boundaries like lots of traffic, more traffic and less traffic.

3.3 FUTURE SCOPE

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. 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] Digital image processing by Rafael C. Gonzalez and Richard E. Woods.

[2] 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.

[3] 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.

[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] 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.

[7] Reulke, S. Bauer, T. D'oring, F. Meysel, "Traffic surveillance using multi-camera detection and multi-target tracking", Proceedings of Image and Vision Computing New Zealand 2007, pp. 175–180, Hamilton, New Zealand, December 2007. www.nzta.govt.nz live traffic webcams / NZ Transport Agency

[8] http://www.mathworks.com: Web

[9]Traffic light control system simulation through vehicle detection using image processing By Mac Michael B. Reyes and Dr Eliezer A. Albaccea