Dynamic Time Allocation for Traffic Control System using Image Processing

[1]V.Viswanadha, [2]S. Chiranjeevi Reddy, [3]B.Venkatesu
[1]Associate Professor, [2]Assistant Professor, [3]Assistant Professor,
[1]Electronics & Communication Engineering
[1]Siddharth Institute of Engineering & Technology, Puttur, Chittoor Dt, AP, 517583

Abstract – *In today's life we have to face many problems one of which is traffic congestion becoming more serious every day. It is said that the high tome of vehicles, the flimsy infrastructure and the irrational distribution of the resources are main reasons for increased 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.*

In present traffic control, fixed time is allotted for each road irrespective of its traffic density which wastes the time on low or empty road. We proposed a system where time is allotted for each road based on its traffic density. We implemented image processing algorithm for real time traffic light control which will control the traffic efficiently. A web camera is placed in each stage of traffic light that will capture the still images of the road where we want to control the traffic. Then those captured images are successively matched using image matching with a reference image which is an empty road image. The traffic is governed according to percentage of matching.

Keywords – Traffic light control, Cameras, Image Processing, Edge detection and Image matching.

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.

1.1. Standard Traffic Control Systems:

1.1.1 Manual Controlling

Manual controlling the name instance it requires 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.

1.3 Need for 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.

2. 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. 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, space crafts and military reconnaissance flights. Image Processing systems are becoming widely popular due to easy availability of powerful personnel computers, large memory devices, graphics software and many more.

2.1 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 element and each finite element is called a pixel.

Fig: Input image

The figure is the captured by a sensor. Here photo diode sensor is used. The sensor is constructed with silicon material. The output voltage waveform of sensor is proportional to light. We can also use filter to improve selectivity. We can also make a output which has one strong color than remaining visible colors using filter. We can generate a 2-D image using single sensor with a displacement in both directions of plane. The arrangement used here is for high precision scanning where film negative is mounted on to a drum which produce mechanical rotation. This mechanical rotation provides displacement in one direction. A sensor mounted on a lead screw is used as it provides motion in perpendicular direction. Using this we can control mechanical motion effectively and images are obtained with high resolution. The sensors a arranged as strips to provide imaging in both direction. The strip provides image in one direction while motion takes care about perpendicular direction. This method is effectively used in airborne imaging. The arrangement is attached to aircrafts during their flights. One dimensional imaging sensor strips that respond to various bands of electromagnetic spectrum are mounted perpendicular to provide perpendicular image so as form a 2-D image.

2.2. 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. $0< f(x, y) < \infty$.

2.3. Image Formed Due to Reflection:

The function f(x, y) is characterized by two components.

1. The amount of source illumination incident on the scene being viewed, which is called illumination components denoted by i(x, y) and
2. The amount of illumination reflected by the objects in the scene which is called as reflectance components denoted by r(x, y).

We can get the image as product of intensity

$$f(x, y)=i(x, y)r(x, y) \quad (1)$$

$$0< i(x, y) < \infty \quad (2)$$

$$0< r(x, y) <1 \quad (3)$$

(reflection bounded by total absorption and reflection)

The nature of i(x, y) is determined by illumination and r(x, y) nature is determined by characteristics of imaged object.

2.4 RGB To Gray Conversion:

Humans perceive color 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.

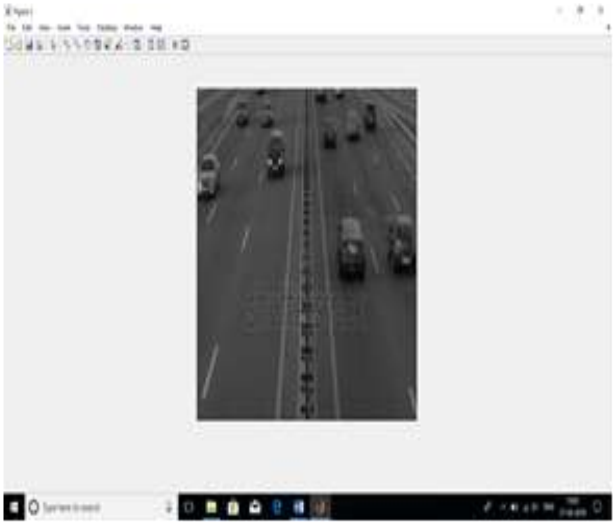


Fig: RGB to gray conversion.

By emitting a restricted combination of these three colors (red, green and blue), and hence stimulate the three types of cones at will, we are able to generate almost any detectable color. This is the reason behind why color 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 color images as stored in an RGB format. In grayscale images, however, we do not differentiate how much we emit of different colors, 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, you can eliminate noise, which will make it easier to identify basic characteristics.

- Decorrelation stretches.
- Histogram equalization
- Linear contrast adjustment
- Median filtering
- Unsharp mask filtering
- Noise-removal Wiener filtering

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.

2.6.1 Edge Detection Techniques:

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

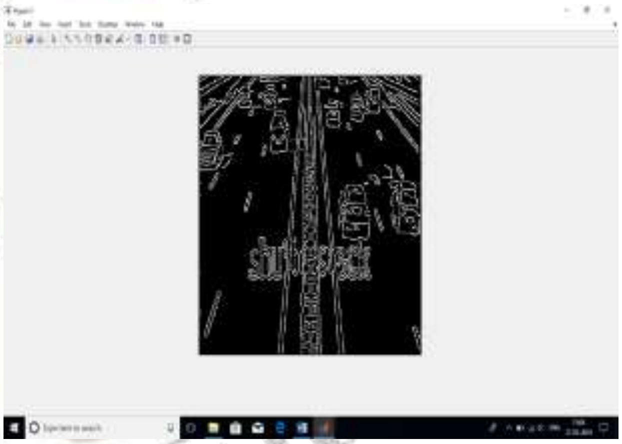


Fig: Edge detected Picture from MATLAB

In MATLAB by default, edge uses the Sobel method to detect edges but the following provides a complete list of all the edge-finding methods supported by this function:

Matlab Syntaxes: BW = edge(I, 'sobel').

BW = edge(I,'prewitt').

BW = edge(I,'roberts').

BW = edge(I,'log').

BW = edge(I,'zerocross',thresh,h).

BW = edge(I,'canny').

In all the above syntaxes I is the Input Image.

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 center of the true edge should be minimum.

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. Summarizing, the Canny edge detection algorithm consist of the following basic steps;
Smooth the input image with Gaussian filter.

i. Compute the gradient magnitude and angle images.
ii. Apply non maxima suppression to the gradient magnitude image.
iii. Use double thresholding and connectivity analysis to detect and link edges.

2.7 Image Matching:

Edge based matching is the process in which two representatives of the same objects are paired together. Any edge or its representation on one image is compared and evaluated against all the edges on the image. Edge detection of reference and the real time images has been done using Sobel operator. Then these edges detected images are matched and accordingly the traffic light durations can be set.

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

%match=No. of pixels matched successfully total/
total no. of pixels

3. IMPLEMENTATION

As mentioned earlier the components required for this system are classified by two different modules and mentioned. They are explained below.

- Hardware and interfacing
- Software module

The software module has been finished with the reference and captured images. Remaining the hardware module and interfacing the software module with hardware module has to be done in future.

MATLAB version 7.8 as image processing software comprising of specialized modules that perform specific tasks has been used.

Calculation of Matching and Timing Allocation:

After edge detection procedure both allusion and real time images are matched and traffic lights can be controlled based on percentage of matching.

Block Diagram:

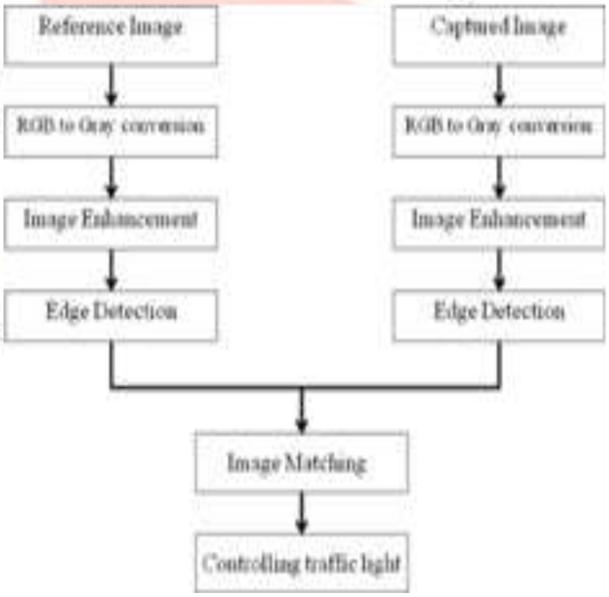


Fig: Block diagram

- If the matching is between 0 to 10% - green light is on for 90 seconds.
- If the matching is between 10 to 50% green light is on for 60 seconds.
- If the matching is between 50 to 70% green light is on for 30 seconds.
- If the matching is between 70 to 90% green light is on for 20 seconds.
- If the matching is between 90 to 100% - red light is on for 90 seconds.

4. 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. GUI is explained in next section.

4.1. GUI:

A graphical user interface (GUI) is a graphical display in one or more windows containing controls, called components, which enable a user to perform interactive tasks. The user of the GUI does not have to create a script or type commands at the command line to accomplish the tasks. Unlike coding programs to accomplish tasks, the user of a GUI need not understand the details of how the tasks are performed. GUI components can include menus, toolbars, push buttons, radio buttons, list boxes, and sliders—just to name a few. GUIs created using MATLAB tools can also perform any type of computation, read and write data files, communicate with other GUIs, and display data as tables or as plots. The following figure illustrates a simple GUI that you can easily build

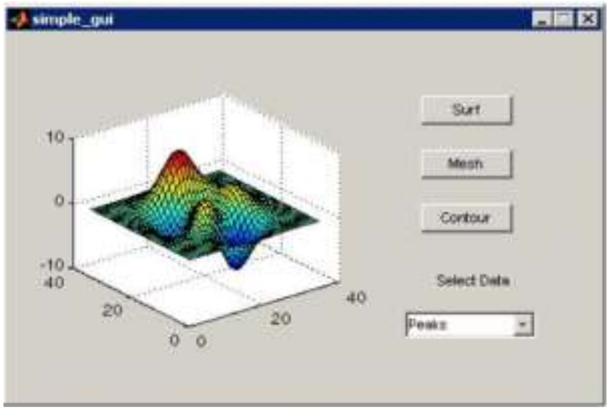


Fig: ***GUI***

The GUI contains

•An axes component

•A pop-up menu listing three data sets that correspond to MATLAB

Functions: peaks, membrane, and sinc.
•Astatic text component to label the pop-up menu

•Three buttons that provide different kinds of plots: surface, mesh, and contour

When you click a push button, the axes component displays the selected data set using the specified type of 3-D plot.
Typically, GUIs wait for an end user to manipulate a control, and then respond to each user action in turn. Each control, and the GUI itself, has one or more call-backs, named for the fact that they "call back" to MATLAB to ask it to do things. A particular user action, such as pressing a screen button, or passing the cursor over a component, triggers the execution of each call back. The GUI then responds to these events. You, as the GUI creator, write call-backs that define what the components do to handle events. This kind of programming is often referred to as event-driven programming. In event-driven programming, call back execution is asynchronous, that is, events external to the software trigger call back execution. In the case of MATLAB GUIs, most events are user interactions with the GUI, but the GUI can respond to other kinds of events as well, for example, the creation of a file or connecting a device to the computer.

You can code call-backs in two distinct ways:

•As MATLAB language functions stored in files

•As strings containing MATLAB expressions or commands (such as 'c = sqrt (a*a + b*b); 'or' print') Using functions stored in code files as call-backs is preferable to using strings, because functions have access to arguments and are more powerful and flexible. You cannot use MATLAB scripts (sequences of statements stored in code files that do not define functions) as call-backs. Although you can provide a call back with certain data and make it do anything you want, you cannot control when call-backs execute. That is, when your GUI is being used, you have no control over the sequence of events that trigger particular call-backs or what other call-backs might still be running at those times. This distinguishes event-driven programming from other types of control flow, for example, processing sequential data files.

A MATLAB GUI is a figure window to which you add user-operated components. You can select, size, and position these components as you like using call-backs you can make the components do what you want when the user clicks or manipulates the components with keystrokes.

You can build MATLAB GUIs in two ways:

•Use GUIDE (GUI Development Environment), an interactive GUI construction kit.

This approach starts with a figure that you populate with components from within a graphic layout editor. GUIDE creates an associated code file containing call-backs for the GUI and its components. GUIDE saves both the figure (as a FIG-file) and the code file. Opening either one also opens the other to run the GUI.

•Create code files that generate GUIs as functions or scripts (programmatic GUI construction).

Using this approach, you create a code file that defines all component properties and behaviours. When a user executes the file, it creates a figure, populates it with components, and handles user interactions. Typically, the figure is not saved between sessions because the code in the file creates a new one each time it runs.

The code files of the two approaches look different. Programmatic GUI files are generally longer, because they explicitly define every property of the figure and its controls, as well as the call-backs. GUIDE GUIs define most of the properties within the figure itself. They store the definitions in its FIG-file rather than in its code file. The code file contains call-backs and other functions that initialize the GUI when it opens.

You can create a GUI with GUIDE and then modify it programmatically. However, you cannot create a GUI programmatically and then modify it with GUIDE. The GUI-building technique you choose depends on your experience, your preferences, and the kind of application you need the GUI to operate. This table outlines some possibilities.

5. FINAL RESULT

If the matching is 90 to 100%, red light is on for 90 seconds

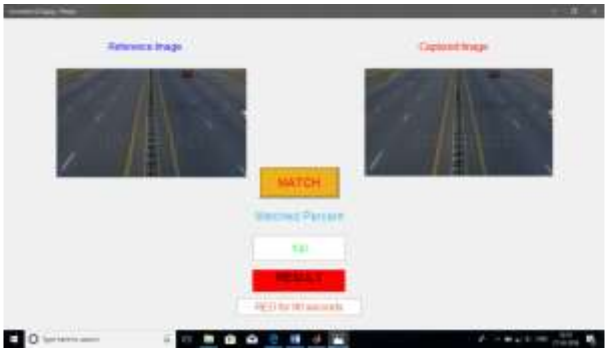


Fig: Results when matching is 90 to 100%

If the matching is 70 to 90%, green light is on for 20 seconds.

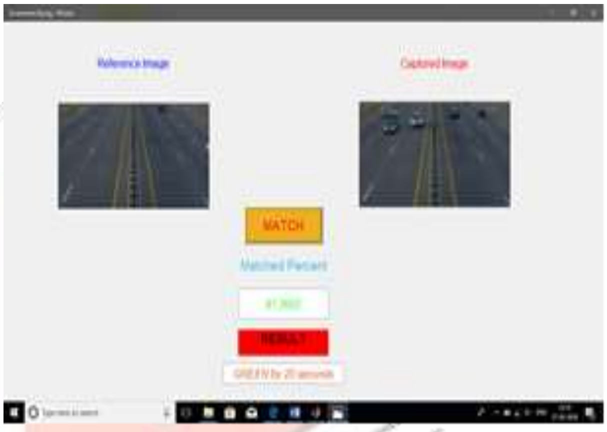


Fig: Results when matching is 70 to 90%

If the matching is between 50 to 70%, green light is on for 30 seconds.

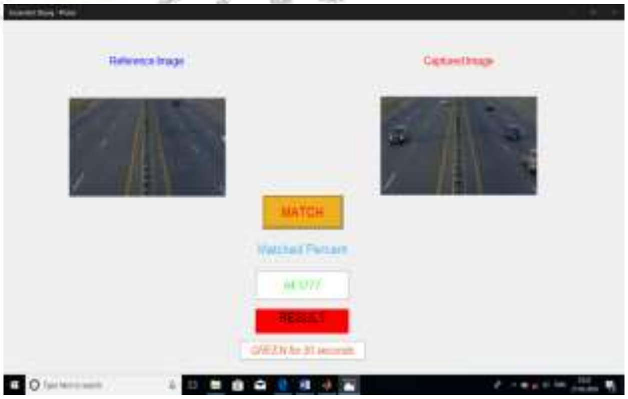
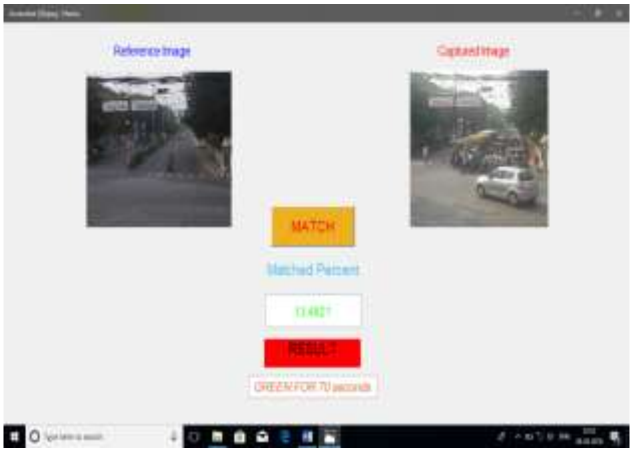


Fig: Results when matching is 50 to 70%

If the matching is between 30 to 10%, green light is on for 30 seconds.



13.4821

GREEN FOR 70 seconds

Fig: Results when matching is 30 to 10%.

6. CONCLUSION

"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 GUI clearly indicated some expected results. It showed matching in almost every interval that were decided as boundaries like 10%, 35%, 68% etc.

7. 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.