Adaptive Control of Traffic Signalling System (ACTSS)

Abhishek Y S S[1], Amar Bahadur Thapa[2], Dinesh Singh[3]

[1, 2, 3] *Student, Dept. of Computer Science and Engineering, New Horizon College of Engineering, Karnataka, India*

---***---

Abstract - *Traffic congestion is a serious problem in developing and developed cities equally alike. While the vehicle count continues to grow, the infrastructure has not improved. Therefore this problem persisted and has not been an easy challenge to tackle. These traffic problems lead loss of precious time, excessive pollution, faster depletion of natural resources such as fuel, etc. Modifying the existing infrastructure is a herculean task with lot of constraints and complications.*

An adaptive traffic signalling system is necessary that minimizes the need to upgrade the existing infrastructure, yet provides an efficient way to handle the traffic on the road. This is achieved by making the ordinary traffic system dynamic by considering real time data. Traffic density on the roads are considered and service time for respective roads are determined from it. The scenarios that are handled by the system are, avoiding time wastage by providing service to an empty road, and avoiding the provision of excess time to a single road.

The system is also aimed at detecting the presence of emergency vehicles, and if present, the regular service cycle is interrupted and the road with the emergency vehicle is serviced. This adds a life-saving feature to the roads, which has not been implemented in most of the countries.

Key Words: Real Time Data, Image Capturing, Image Processing, Edge Detection, Traffic Density, Service Time, Emergency Vehicle Detection

1. INTRODUCTION

Traffic is the movement of various vehicles on the road. Traffic management is the process of planning, coordinating, controlling and organizing traffic to achieve efficiency and effectiveness of the existing road capacity.

Traffic flow on the road is regulated by the use of traffic signals, which coordinate the flow of traffic from various roads, to avoid incidents and also clogging up of the vehicles. A traditional traffic signalling system uses three colored signal, with the colors Red, Orange and Green. Red signifies stop, orange signifies wait and green signifies go.

The amount of time during which the traffic of a road is allowed to go is called as the service time of that road. In India we use a traffic system that is static in nature and hence uses a predefined service time allotted to each road at the time of installation of the signal. The service time never changes until it is changed by the technician by modifying the code.

1.1 Traffic related problems and statistics

Traffic lights provide service time to the roads that do not have any traffic on them because of their fixed time cycle of working. Also the traffic system allots a fixed duration of service to each road irrespective of the count of vehicles on the road. This could mean that a road with more traffic in comparison to another road will get the same service time as the latter. This leads to the problems mentioned below:

1. **Time wastage**: Due to the fixed time cycle, we will be wasting precious time on a road that may not have traffic or give equal time to roads with varying traffic density. According to the article "*Indians spend more time behind the wheel than Chinese and Aussies : Survey*" [1], in India, 49% of respondents in a survey spend more than 100 minutes everyday for commuting and 14% spend up to three hours a day.

2. **Noise Pollution**: Due to stagnation of cars at one location, with the engines running, and also due to the unnecessary honking we generate noise pollution. According to the article "*Noise levels high on MG Road*" [2], noise levels at Bangalore a metropolitan city in India, noise are at alarmingly high levels with the major contribution being from traffic signals.

3. **Air Pollution due to excessive fuel consumption**: Burning fuel standing at a point with no work being done, we contribute to the air pollution and also spend a lot on fuel. According to the article *"India's rising obsession with diesel worries govt."* [3], in India, 175 million litres of diesel and 57 million litres of diesel was consumed on a daily basis in the year 2012.

Brinda R. B., et. al. [4] stated that these issues in fact originate at the traffic signals. So it is lucid that to solve these issues it is necessary to solve them at the traffic signal itself.

2. REQUIRED SYSTEM

We need a traffic signalling system that addresses all the issues of the traditional traffic signalling system. This is achieved by developing a system that takes in real time data like the count of vehicles on the road and then modifies its behavior so that efficiency is achieved in the way traffic on the roads are serviced. The system should identify the traffic density on each road and depending on that, allot service time for the respective roads.

3. RELATED WORK

Vismay Pandit, et. al. [5], proposed a system of capturing an image of the empty road and then capture an image of the road, compare the two images to identify the traffic density on the road using which service time can be calculated for each road. But this system would be limited as slight change in the position of camera after capturing the reference image, the empty road would lead to inaccurate results.

Omkar Ramdas Gaikwad, et. al. [6] put forth a system that involves painting stripes on the road at different distances from the traffic signal. The number and the percentage of the stripes covered on each road would be helpful in estimating the traffic density on that road. But this technique would lead to excess costs in terms of modifying the infrastructure of each road by putting stripes on every road, and also there is need for the people to be educated about the purpose of these stripes to avoid confusion.

Rashishet, et. al. [7], proposed a system that captures images of the traffic using a camera mounted on a motor, so that a single camera can take the image of the roads and a series of image processing steps are performed on the image of the traffic to calculate the density of traffic on the road. This method would free us from the restriction of having an absolutely stationary camera, which was essential for the solution proposed by Vismay Pandit, et. al. [5]. But this solution has the camera rotating on a motor, which makes the system complex to implement and maintain compared to a system with a single camera taking a picture of all roads without moving.

This paper also provides a way of detecting emergency vehicles on the roads using the red beacon as an indicator, by processing the image for identifying the presence of red dot. But this method is limited if a large vehicle is obstructing the emergency vehicle from the view of the camera. We propose a system which identifies the presence of an emergency vehicle by listening for the siren of the same.

4. ADAPTIVE CONTROL OF TRAFFIC SIGNALLING SYSTEM

Keeping in mind the issues being faced and the necessary steps to be taken in order to tackle these issues a system is proposed. Adaptive Control of Traffic Signalling System (hereafter addressed as ACTSS) is designed to take real time information of the traffic conditions of the road through an image. A camera is installed at the junction that takes a picture of all the roads at the junction and the density of traffic on each road is determined from this image. If a road is empty, it will not be given any service time. This allows us to save time. If a road has more cars than the other roads, then that road is given more service time, as long as the calculated service time is within the threshold value. If the calculated service time is greater than the threshold value, then the threshold value is given as the service time to avoid starvation of service.

At the same time, the system constantly monitors the surroundings, listening for emergency vehicles, so that the service can be given to them immediately. This would add a life-saving feature to ACTSS. The sound that falls in a particular range of frequency indicates the presence of emergency vehicle. If an emergency vehicle is detected, the road containing the emergency vehicle is prioritized and is serviced immediately. The sound is detected by a microphone installed on each road.

4.1 Modules of the System

A monolithic system is restricted in nature and cannot be designed easily to be adaptive. ACTSS is designed with multiple modules with each module being specifically designed to perform a task. The components in ACTSS are:

a. **Image processing using MATLAB®**

An image of the traffic is captured using MATLAB® and then the image is processed on MATLAB®. The count of vehicles is extracted from this image by performing various image processing functions available from the Image Processing Toolkit. This information is then used for allotting the required service time for each road.

b. **Signal and timer control using Arduino™ Mega**

Once the count of vehicles is determined by MATLAB® on each road, that information is passed onto the microcontroller, Arduino™ Mega. Arduino™ Mega is responsible with displaying the appropriate signals at the appropriate roads and also displaying the service time of a road being serviced using 7-Segment Displays. We write a program for Arduino™ Mega on Arduino™ IDE using C language.

c. **Emergency vehicle detection by Arduino™ Uno**

To implement a lifesaving feature in this system, where we allow the road with emergency vehicles like ambulances, fire-trucks, etc., we use detection of frequency of various sounds in the environment. An Arduino™ Uno has been programmed to detect the various frequencies, and if the frequency of a particular sound falls within the range of that of an emergency vehicle, an interrupt is generated and sent to Arduino™ Mega, that will halt its regular operation and service the road with the emergency vehicle first. Once the emergency vehicle is out of the place, routine operation will resume.

The system architecture of the ACTSS can be seen in figure 'Fig. - 1'

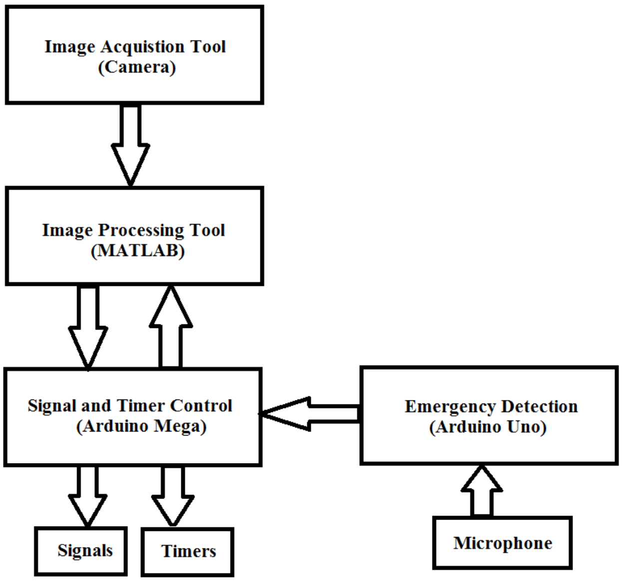

Fig. – 1: System Architecture of ACTSS

5. WORKING OF ACTSS

There a series of steps that will be repeated in an infinite cycle. The steps are as follows.

- Image acquisition
- Image processing
- Displaying the output
- Emergency vehicle detection

In the following sections the activities of each of these steps are elaborated.

5.1 Image acquisition

Image acquisition is the concept of capturing an image using a hardware source such as a camera and storing it in memory. In this step we are capturing the images of road with the varying traffic using a USB webcam and MATLAB® inbuilt tools and storing in a hard disk, so that it can later be used to extract the required data i.e. vehicle count.

'Fig.2' shows the captured image of a model platform with varying number of cars on each road.

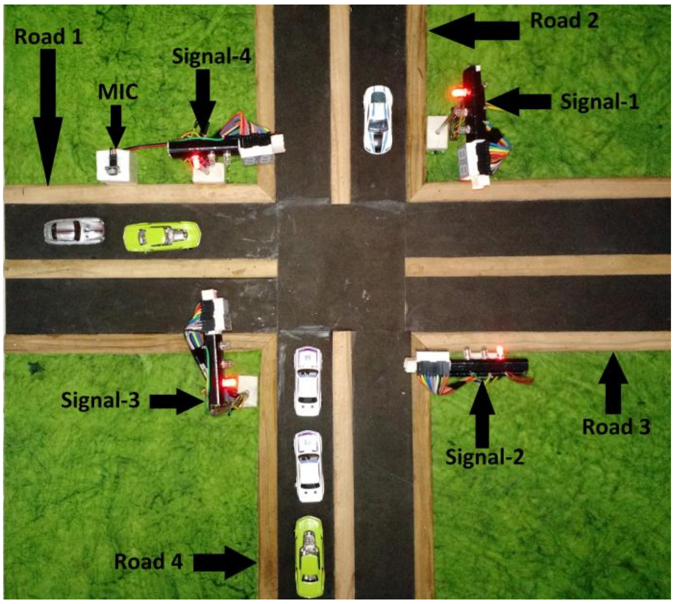

Fig. – 2: Model platform with cars used for simulation

5.2 Image processing

Once the image has been captured and stored in memory the core part of the system comes into action. A series of techniques are implemented in order to detect traffic count on road.

It includes:-

1. **Cropping the image.**

   The intention of this step is to be selective with the area of interest and only consider that part of the image that is needed. So the area on which we will be further working on is selected and cropped in order to increase the efficiency of the system.

   'Fig.3' shows the cropped image of road 1 that will be considered for further processing.

Fig. – 3: Cropped image of road 1

2. **Conversion of the RGB to Gray-scale**

   A 24-bit RGB image has 16,777,216 possible values and for every pixel we have 3 values to work upon. Whereas an 8-bit Grayscale image has only 256 possible values and only one value per pixel on which we need to work. Since the colors in the image serve no specific purpose in our system, we convert the RGB image to a Grayscale image in order to increase the efficiency of the system.

'Fig. - 4' shows the Grayscale image obtained from the cropped RGB image seen in 'Fig. - 3'

Fig. – 4: Grayscale image of road 1

3. **Edge Detection**

To identify the vehicles on the road, we perform the edge detection on the cropped image. This would return an image with only the edges in it. We can see the edges of the vehicles which we use later to identify the vehicle count. We use the inbuilt function of edge detection of MATLAB® using the Sobel algorithm.

'Fig. – 5' shows the image of the edges of the vehicles obtained from 'Fig. – 4'

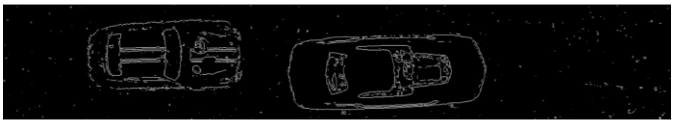

Fig. – 5: Edges of the vehicles on road 1

4. **Removing stray lines**

As any picture captured of the road can have discrepancies like shadows, etc. we intend to remove these stray lines before further processing. So we remove any line whose length is less than or equal to 5 pixels. The pixel count is determined by the size of the model.

'Fig. – 6' shows the image of road 1 with stray lines removed.

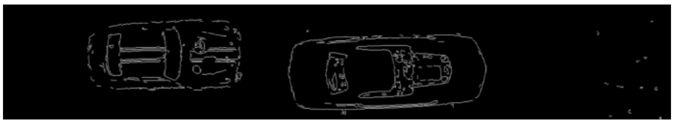

Fig. – 6: Road 1 with no stray lines

5.3 Signal and timer control

5. **Closing the disconnected edges**

Due to certain unfavorable lighting conditions, it might so happen that even when the complete vehicle is present on the road, its edges may not be completely closed. But for counting the number of vehicles, we need completely bounded objects. To achieve this, we need to connect the disconnected edges. This is achieved by connecting disconnected edges. To identify which two edges can be connected, at every pixel we create a disk of radius 7 pixels, and if two edges exist in this disk that are not connected, then those two edges would be connected. This happens for every pixel in the image. The radius of the disk is determined considering the size of the model.
This also leads to filling up certain portions of the object, which will not influence the results in any manner.

'Fig. – 7' shows the image with connected edges

Fig. – 7: Image of road 1 with connected edges

6. **Filling the holes**

After connecting disconnected edges, certain holes will always be left out in the objects. This may lead to erroneous results and therefore need to be taken care of. We use the above built in function to fill all the holes in the image within a bounded object.

'Fig. – 8' is the processed version of the image in 'Fig. – 7' where the holes have been filled.

Fig. – 8: Image of road 1 with the holes filled

7. **Smoothening the edges**

The objects that have been filled might have jagged edges unlike the actual object. To smoothen the edges out, we perform this step that creates a disk of radius 1 pixel and sees if within that radius, can the object be smoothen by filing up the space. The disk radius is dependent on the size of the model.
'Fig. – 9' shows the image of road 1 with smoothened edges of the objects within it.

Fig. – 9: Image of road 1 with smoothened edges

8. Remove the noise

There are always stray objects on the roads like animals, litter, etc., which can be considered as an object. The universal fact about these objects is that they are small in size when compared to any vehicles. So we can remove any objects below a certain pixel count as they obviously cannot be a vehicle, therefore are a stray, unwanted object. In this case, we remove any objects below the size of 400 pixels, a measure determined from the average size of the vehicles on the model. Now this is the final image that we use to find the count of vehicles on each road.

'Fig. – 10' shows the final image with no noise.

Fig. – 10: Image of road 1 with no noise

9. Counting the vehicles on each road

Once all these steps are performed, we have an image that has only vehicles as the objects. Now using the first statement, we return each object as an individual element to an array. Since each element in the array is an object, length of array is the number of vehicles on that road. We use the length function to get the appropriate traffic density on each road.

5.3 Displaying the output

Once the number of vehicles, i.e. the traffic density has been identified, data has to be sent from MATLAB® to Arduino™ Mega, the microcontroller that controls the signals and the timers, through which we display the output.

We establish a connection with Baud rate 115200 bits per second over a COM port and use a variable to identify this connection. We have to open s1 to open the pipe file in which we write the data.

Later after identifying the density of traffic on an individual road, we immediately write it into a 4 separate strings each for a road.

The densities stored in the individual strings is sent to Arduino™ Mega along with the road number by concatenating the strings in an order.

Once the Data is received by Arduino™ Mega, the signals and timers start working accordingly. The Arduino™ Mega has the connections already made for each red, yellow and green signals, along with the timers.

Each cycle begins by servicing road 1 first. First all red signals will be glowing, then after a delay of 2 seconds, red signal for road 1 changes to yellow, indicating to look. After another two minutes, the yellow signal turns off and the green signal will turn on along with the timer. After the calculated amount of service time is given by calling the timer functions, the green signal turns to yellow. After another 2 seconds, the yellow signal turns to red. Then road 2, followed by road 3, and then finally road 4 is serviced in a cycle.

5.4 Emergency vehicle detection

The system detects emergency vehicles by measuring frequencies of the sounds that enter the mic, and by comparing it to the siren frequencies. If the frequencies are in the range of the siren sound, the system will be interrupted, then all the traffic lights on the intersection will become red, and the traffic light on the side of the emergency vehicle will be made green.

The system needs an Arduino™ Uno microcontroller connected to a microphone for each road to listen to the sounds and identify the presence of an emergency vehicle.

Every country has a predefined range of frequencies for all the emergency vehicles. In India, the frequency range for ambulances is between 1100 to 1300Hz or between 1600 to 1900Hz. We record the frequencies of all the sounds and see if the frequency falls within this predefined range. If it does fall within the range, an interrupt is raised at Arduino™ Mega. The regular function of the Arduino™ Mega is then paused and an Interrupt Service Routine is executed. Once the ISR is executed, the normal function will be resumed at the point where it was left.

A sample scenario has been shown in Table – 1, with different vehicle count on each road. Road 1 and 2 show roads with vehicle count well within the threshold. Road 3 represents a road with no vehicle, thereby does not get any service time allotted to it. Road 4 has vehicle count above the threshold value so threshold service time of 18 seconds is only allotted.

Table -1: Table for sample traffic scenario

Vehicle count and service time allotment		
Road No.	**Count**	**Service Time (sec)**
1	2	12
2	1	6
3	0	0
4	4	18

6. CONCLUSIONS

We intended to develop a system that would tackle the issues of time loss, noise pollution, and air pollution along with fuel wastage caused due to the traditional traffic signalling system. The system works on real-time information obtained from the immediate surroundings and then changes its behavior according to it. With the ability to detect the presence of emergency vehicles on the roads, we can also make the system supportive in the mission to save lives by prioritizing the emergency services.

All this is done with the least modifications to the existing infrastructure and at very less procurement of equipment.

REFERENCES

[1] Dipak K Dashi (2015, February 23) Indians spend more time behind the wheel than Chinese, Aussies: Survey. Retrieved from http://timesofindia.indiatimes.com/

[2] Soumita Majumdar (2011, May 21) Noise levels high on MG road. Retrieved from http://www.dnaindia.com/

[3] Anupama Airy (2012, April 17) India's rising obsession with diesel worries govt. Retrieved from http://www.hindustantimes.com/

[4] Brinda R.B., Namratha Venkatesh Murthy, B.M. Ramya, Dr. Vijaya Prakash A M, "Edge Detection (Image Processing) Smart Traffic Control", International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering, Vol.3 Issue 11, Novemeber 2015, Pg. No. 128

[5] Vismay Pandit, Jinesh Doshi, Dhruv Mehta, Ashay Mhatre, Abhilash Janardhan, "Smart Traffic Control System Using Image Processing", International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), Jan-Feb 2014, Pg. No. 280

[6] Omkar Ramdas Gaikwad , Anil Vishwasrao , Prof Kanchan Pujari, Tejas Talathi, "Image Processing Based Traffic Light Control", International Journal of Science, Engineering and Technology Research (IJSETR), Vol.3 Issue 4, April 2014, Pg. No. 1010

[7] Rahishet, Aparajita Saho, Aparna Indore, Vaibhav Deshmukh, Pushpa US, "Intelligent Traffic Light Control Using Image Processing", 21st IRF International Conference, Mar 2015, Pg. No. 1

BIOGRAPHIES

Abhishek Y S S is a final year B.E. student of Computer Science and Engineering at New Horizon college of Engineering, Bangalore.

Amar Bahadur Thapa is a final year B.E. student of Computer Science and Engineering at New Horizon college of Engineering, Bangalore.

Dinesh Singh is a final year B.E. student of Computer Science and Engineering at New Horizon College of Engineering, Bangalore