Al Al-Bayt University

Prince Hussein bin Abdullah Information Technology College

Computer Science Department

Thesis Title

Traffic Light Control Utilizing Queue Length and Object Segmentation

By

Obadah "Mohammad Adnan" Al ayesh

2010

"Traffic Light Control Utilizing Queue Length and Object Segmentation "

"التحكم بالإشارة الضوئية من خلال استعمال طول الطابور واكتشاف الأجسام"

By

Obadah " Mohammad Adnan " Ayesh

"0620901001"

Supervisor : Dr. Venus W. Samawi

A Thesis Submitted to the

Scientific Research and Graduate Faculty in partial fulfillment of the

Requirements for the Degree of Master of Science

in Computer Science

<u>**Members of the Committee**</u>	<u>**Approved**</u>
Prof. Adnan Al-Smadi	
Dr. Venus W. Samawi	
Dr. Azzam Sleit	
Dr . Akram hamarshi	

Al al-Bayt University

Mafraq, Jordan

2009

Dedication

To

My parents who gifted me their lives
And to
My wife "WALA"

I dedicate this work.

Acknowledgment

I would like to present my thanks and appraisal to Dr. Venus Samawi , who gave me her supports, advices and for her feeling of the responsibility to guide me to the correct approach for completing this work within the required time.

Also, I would like to thank Professors and Head of the Computer Science department for their support to postgraduate students, especially Dr. Jehad Alkhaldi.

Finally, great thanks and appreciation to my friends for their assistance.

Obadah Ayesh

Table of Contents

Subject	Page
Front page	**I**
Dedication	**II**
Acknowledgment	**III**
List of Contents	**IV**
List of Tables	**VI**
List of Figures	**VII**
List of Chart	**IX**
Abstract	**X**
Chapter One : Introduction	
1.1 Overview	**2**
1.2 Traffic Light Control Methodologies	**3**
1.2.1. Fixed Time Control	**3**
1.2.2 Dynamic Traffic Control	**4**
1.3 Problem Description	**5**
1.4 Literature Review	**6**
1.5 Research Objectives and Statement of The Problem	**9**
1.6 Thesis Layout	**10**
Chapter Two : Main Principles of Image Processing	
2.1 Introduction	**12**
2.2 Main Principles of Image Processing	**12**
2.3 Image Subtraction	**14**
2.4 Noise Removal and Filtering	**15**
2.5 Edge Detection	**15**
2.6 Morphological Processing	**19**
2.7 Image Segmentation	**21**

Subject	Page
Chapter Three : System Development	
3.1 Introduction	**24**
3.2 Camera View	**25**
3.3 Image Description	**26**
3.4 System Model	**26**
3.4.1 Preprocessing Phases	**27**
3.4.2 Module Counting Vehicle	**34**
3.4.3 Module Queue Length	**38**
3.4.4 Timing	**42**
Chapter Four : Experimental Results	
4.1 Introduction	**45**
4.2 Image Data	**46**
4.3 Experimental Results : Analyses & Comparison	**49**
Chapter Five : Conclusions And Future Works	
5.1 Conclusions	**58**
5.2 Future Works	**58**
References	**59**
Appendix (A)	**64**
الخلاصة	**67**

List of Tables

Table Number	Description	Page Number
Table 3.1	Photo Specifications	26
Table 4.1	Low Density	50
Table 4.2	High Density	52
Table 4.3	All pictures with Mean Square Error MSE	54

List of Figures

Figure Number	**Description**	**Page Number**
Figure 1.1	Traffic Jam	2
Figure 1.2	Fixed Traffic	4
Figure 1.3	Dynamic Traffic Control	5
Figure 2.1	Boolean Logical Rules	14
Figure 2.2	Canny Edge Detection	16
Figure 2.3	Prewitt Edge Detection	17
Figure 2.4	Sobel Edge Detection	18
Figure 2.5	Morphological Operations	20
Figure 3.1	Camera View	25
Figure 3.2	System Model	26
Figure 3.3	Sample of Input Image	27
Figure 3.4	RGB Image Converted to Grayscale	28
Figure 3.5	Image Masking	29
Figure 3.6	Image Subtraction	29
Figure 3.7	Convert Image to Binary	30
Figure 3.8	Image after Filter Images and Close	31
Figure 3.9	Image After Sobel Edge Detection	31
Figure 3.10	Image After Dilate	32
Figure 3.11	After Fill Image Regions and Holes	32
Figure 3.12	Image After Remove Small Object	32

Figure Number	Description	Page Number
Figure 3.13	Preprocessing Phase	33
Figure 3.14	Model Counting Vehicle	35
Figure 3.15	Implementation Outcome	38
Figure 3.16	Road Map	39
Figure 3.17	Low Density Road	41
Figure 3.18	Queue Module	42
Figure 4.1	Image Low Density	47
Figure 4.2	Image High Density	48
Figure 4.3	Image is Divided into Levels	49

List of Chart

Chart Number	Description	Page Number
Chart 4.1	Number of Vehicle in Low Density	51
Chart 4.2	Estimated Time in Low Density Pictures	52
Chart 4.3	Number of Vehicle in High Density	53
Chart 4.4	Estimated Time for High Density	54
Chart 4.5	Number of Vehicles for All Pictures	56
Chart 4.6	Estimated Time for All Pictures	56

Abstract

As the social economic activities grow variously in recent days, an advanced traffic control system corresponding to the changes of road traffic circumstances becomes an urgent matter. This research suggests two techniques based on picture taken by digital camera placed at fixed position in the traffic light with specific resolution and distance. The resulted image will be preprocessed (subtract image from its background, remove noisy small objects) then use two techniques ***queue length*** (find out the length of car queue on road depending on the distance between the two end points on road line), and ***vehicle counting*** (by performing image segmentation and count out segments that satisfy the proper size of vehicle).To measure the queue length or to count the number of the vehicles, edge detection (in this research Sobol edge detection method is used) and segmentation (based on region growing) are needed. Finally, an equation to find out the estimated time and actual time is suggested to determine the estimated time reference for optic Green.

The two suggested techniques compared from two points of views: time needed to make decision of green-light period, and the suggested estimated time by the technique with respect to the actual time needed.

As a result, it was found the vehicle counting technique is recommended to be used since it shows better performance from decision making time and estimated time with respect to actual. Also the Impact of image quality (low and high density of vehicles on road) is studied, where it was found that the queue suit high density road vehicle counting suit low density road, but generally segmentation suitable for both types.

Chapter One

Introduction

1.1 Overview

As driving around town on daily travel, one may find himself stuck in traffic and receiving poor gas mileage. One of the main reasons could be the poor design of the traffic light system. A report card released in 2001 shows traffic signals across the world are largely inefficient and could increase road rage and pollution. The report says Out-of-whack signals cause too much stop-and-go traffic, which bother drivers, wastes gas and increases air fumes. [GRE 05].

Most areas don't have: ***a feasible plan for traffic signal operation***, ***staff to monitor traffic before and after normal work hours***, and ***don't conduct*** (see Figure 1.1). Traffic signals must be instructed when to change phase. They can also be coordinated so that the phase changes called for occur in some relationship with traffic monitoring, and nearby signals. Mainly there are two types of traffic control: fixed time control and dynamic time control.

Figure 1.1 Traffic Jam.[INT 09]**.**

The traffic congestion occurs when too many vehicles attempt to use a common city road with limited capacity. The efficient, safe, and less polluting transportation of persons and goods calls for an optimal utilization of the available infrastructure via suitable application of a variety of traffic control measures. Traffic control directly depends on the efficiency and relevance of the employed control methodologies, [MAR 03]. Due to the importance of real time (dynamic) traffic control, more and more researcher investigated the real time vision based transportation surveillance system .They deliberate to analysis and detect the objects first then counting the number of cars after that, extrapolate the transportation information of the main urban road. [HSU 03] [HAR 00] [GIU 02].

1.2 Traffic Light Control Methodologies

Generally, there are two types of controllers: fixed time controller at which the interval of each phase is fixed and determined manually by traffic center, and dynamic traffic controller at which duration of each phase depends on transportation surveillance system.

1.2.1. Fixed Time Control

With this type of control the duration and the order of all green phases is fixed. Traffic signal phase changes are based on one of three systems: Pre-timed, semi-actuated, and fully-actuated. The simplest control system uses a timer (fixed-time). Each phase of the signal lasts for a specific duration before the next phase occurs; this pattern repeats itself regardless of traffic. Many older traffic light installations still use this type. The main advantage is that the simplicity of the control enables the use of simple and inexpensive equipment.

Today, most major cities use fixed-time traffic control systems, which divide a typical day into several sessions according to prevailing traffic conditions. In each session, the timing of the traffic signals is fixed and unresponsive to real-time traffic conditions. Some modern systems work using detectors installed beneath the road surface. When the system detects that the traffic pattern has changed substantially, it selects another pre-calculated timing program. Nevertheless, the timing of such programs is still fixed.

The weakness in these static control systems and big disadvantage is that they cannot adapt itself to the dynamic nature of changing traffic conditions. They only provide reasonable performance when actual traffic conditions match the predefined conditions which were used to calculate the original fixed-time programs.

In the real world, it is not uncommon for these timing programs to under-utilize the green time, allowing situations where there is no vehicle crossing the road junction when the light is green (see Figure 1.2), [CHA 96] [GRE 05] .

Figure 1.2 Fixed Traffic Control.[INT 09]**.**

For example, the system may set the green time for 25 vehicles to pass through a junction by assuming that all 25 vehicles have already built up behind the traffic light. However, in reality, the road section may only accommodate 8 vehicles. The remaining 17 vehicles may still not have arrived at the junction, and thus the green time reserved for these 17 vehicles will be wasted. In addition, oncoming vehicles may gradually accumulate behind the junction and create further delays.

Subsequently, in rush hours, if traffic flow is effectively coordinated and green time utilized to allow just a few more vehicles to cross the junction, much of the traffic congestion can be significantly relieved.

1.2.2 Dynamic Traffic Control

More sophisticated control systems use electronic detector loops, which are sensors buried in the pavement to detect the presence of traffic waiting at the light, and thus can avoid giving the green light to an empty road while motorists on a different route are stopped. A timer is frequently used as a backup in case the sensors fail; an additional problem with sensor-based systems is that they may fail to detect vehicles such as motorcycles or bicycles and cause them to wait forever (or at least until a detectable vehicle also comes to wait for the light). The sensor loops typically work in the same fashion as metal detectors; small vehicles or those with low metal content may fail to be detected. It is also commonplace to alter the control strategy of a traffic light based on the time of day and day of the week, or for other special circumstances (such as a major event causing unusual demand at an intersection), (See Figure 1.3). [CHA 96], [GRE 05].

Image processing Vehicle detector. Terminal

Travel time Measuring

Figure 1.3 Dynamic Traffic Controls.[INT 09]**.**

In 1993, the synthetic and epoch-making traffic management system, Universal Traffic Management Society of Japan (UTMS), was established. Among the purposes of UTMS, especially Integrated Traffic Management System (ITCS) aims at the development of a more optimum signal control algorithm which can corresponds to the every moment changing traffic situation.

1.3 Problem Description

One of the major problems concerning traffic control is to try to provide a dynamic system that makes decision when to change the traffic signal phase through specifying the jam points in the road. The main advantages of such systems are to:

1. Reduces traffic congestion: such system substantially reduces traffic congestion by accurate signal controller operation to suit prevailing traffic conditions ensure traffic safety and smooth traffic flow.

2. Reduces traffic accidents: The system reduces traffic accidents by smoothing traffic flow.

3. Reduces traffic pollution: reduces the number of vehicle's stopping due to traffic congestion to cut exhaust gas and noise emitted in stopping and starting vehicles, thus reduces traffic pollution.

4. Saves energy and preserves the environment: makes traffic flow smoother and reduces travel time to destinations to save energies such as gasoline and light oil.

5. Reduce travel time.

Such system can be implemented through identifying the traffic jam on road. One of methods used to do so is based on image processing at which number of vehicles in the road should be determined.

1.4 Literature Review

With respect to the research on traffic control based on image processing, attracted a growing number of computer scientists and they have proposed several different intelligent systems. This section is devoted to survey and to present some of the related research works.

1. M. Fathy and M.Y. Siyal, [FAT 95], the real-time measurement of various traffic parameters including queue parameters is required in many traffic situations such as accident and congestion monitoring and adjusting the timings of the traffic lights. In case of the queue detection, at least two algorithms have been proposed by previous researchers. Those algorithms are used for queue detection and are unable to measure queue parameters. The authors propose a method based on applying the combination of noise insensitive and simple algorithms on a number of sub-profiles (a one-pixel-wide key-region) along the road. The proposed queue detection algorithm consists of motion detection and vehicle detection operations, both based on extracting edges of the scene, to reduce the effects of variation of lighting conditions. To reduce the computation time, the motion detection operation continuously operates on all the sub-profiles, but the vehicle detection is only applied to the tail of the queue. The proposed algorithms have been implemented on an 80386-based microcomputer system and the whole system works in real-time.

2. Z. Zhu, B. Yang, G. Xu, and D. Shi, [ZHU 96], the authors present a novel approach using 2D spatio-temporal images for automatic traffic monitoring. A TV camera is mounted above the highway to monitor the traffic through two slice windows for each traffic lane. One slice window is along the lane and the other perpendicular to the lane axis. Two types of 2D spatio-temporal (ST) images are used in the system: the panoramic view image (PVI) and the epipolar plane image (EPI). The real-time vision system for automatic traffic monitoring, VISATRAM, an inexpensive system with a PC 486 and an image

frame grabber has been tested with real road images. Not only can the system count the vehicles and estimate their speeds, but it can also classify the passing vehicles using 3D measurements (length, width and height). The VISATRAM works robustly under various light conditions including shadows in the day and vehicle lights at night, and automatically copes with the gradual and abrupt changes of the environment.

3. Markus Ebbecke, Majdi Ben Hadj Ali, Andreas Dengel [MAR 97], in cooperation with the traffic and transport research group at the University of Kaiserslautern the researchers conceived and implemented an automatic real time object detection, tracking and classification system working on color image sequences, taken by a static camera. The system is employed by researcher to detect conflict situations in traffic scenes. The main requirements for the system were real time ability and implementation on cheap standard hardware. In this paper, the writer proves that it is possible to develop a system that runs in real time on standard hardware. For traffic researchers, such an automatic system is the basis for the examination of traffic situations over a long period of time.

4. Rita Cucchiara, Member, IEEE, Massimo Piccardi, Member, IEEE, and Paola Mello [RIT 00], this research presents an approach for detecting vehicles in urban traffic scenes by means of rule-based reasoning on visual data. The strength of the approach is its formal separation between the low-level image processing modules and the high-level module, which provides a general-purpose knowledge-based framework for tracking vehicles in the scene. The image-processing modules extract visual data from the scene by spatio-temporal analysis during daytime, and by morphological analysis of headlights at night. The high-level module is designed as a forward chaining production rule system, working on symbolic data, i.e., vehicles and their attributes (area, pattern, direction, and others) and exploiting a set of heuristic rules tuned to urban traffic conditions. The synergy between the artificial intelligence techniques of the high-level and the low-level image analysis techniques provides the system with flexibility and robustness.

5. Lipton, H. Fujiyoshi, and R. Patil, [LIP 98] this paper describes an end-to-end method for extracting moving targets from a real-time video stream,

classifying them into predefined categories according to image-based properties, and then robustly tracking them. Moving targets are detected using the pixelwise difference between consecutive image frames. A classification metric is applied these targets with a temporal consistency constraint to classify them into three categories: human, vehicle or background clutter. Once classified, targets are tracked by a combination of temporal differencing and template matching. The resulting system robustly identifies targets of interest, rejects background clutter, and continually tracks over large distances and periods of time despite occlusions, appearance changes and cessation of target motion.

6. Y. L. Murphey, H. Lu, S. Lakshmanan, R. Karlsen, G. Gerhart, and T. Meitzler, [MUR 99] , the researchers present an intelligent system, Dyta (dynamic target analysis), for moving target detection. Dyta consists of two levels of processes. At the first level it attempts to identify possible moving objects and compute the texture features of the moving objects. At the second level, Dyta inputs the texture features of each moving object to a fuzzy intelligent system to produce the probability of moving targets. The three major algorithms of Dyta, the moving target tracking algorithm, the Gabor multi-channel filtering, and fuzzy learning and inference, are presented in the paper. They conducted extensive experiments on the Dyta system using images captured in outdoor environments.

7. Lawrence Y. Deng, Nick C. Tang, Dong-liang Lee, Chin Thin Wang and Ming Chih Lu [LAW 05], The adaptive urban traffic signal control (TSC) system became a development trend of intelligent transportation system (ITS). they investigated the vision based surveillance and to keep sight of the unpredictable and hardly measurable disturbances may perturb the traffic flow. they integrated and performed the vision based methodologies that include the object segmentation, classify and tracking methodologies to know well the real time measurements in urban road. According to the real time traffic measurement, the adaptive traffic signal control algorithm to settle the red–green switching of traffic lights both in “go straight or turn right” and “turn left” situations is derived. By comparing the experimental result obtained by original traffic signal control system which improves the traffic

queuing situation, they confirm the efficiency of vision based adaptive TSC approach. In the experiment results, they diminished approximately 20% the degradation of infrastructure capacities.

8. Richard Lipka, Pavel Herout, [RIC 08] , This paper describes behavior and implementation of light signalization in urban traffic simulator JUTS. Traffic lights are necessary part of urban traffic network and they allow using JUTS in experiments dealing with impact of time plans to traffic situation.

9. Alvaro Soto and Aldo Cipriano, [ALV 96], This paper describes a computer vision system for the real time measurement of traffic flow. The traffic images are captured by a video camera and digitized into a computer. The measuring algorithms are based on edges detection and comparison between a calculated reference without vehicles and the current image of traffic lanes. Tests under real traffic conditions were satisfactory, with over 90% of accuracy and error below 5%.

10. Michele Zanin, Stefano Messelodi , Carla Maria Modena[MIC 03], This paper describes a method for the real-time measurement of vehicle queue parameters in a video-based traffic monitoring experimental system. The method proposed here is based on vehicle presence detection and movement analysis in video sequences acquired by a stationary camera. Queues are detected and characterized by a severity index. Intensive experiments show the robustness of the method under varying illumination and weather conditions. The system is presently undergoing an on-field testing phase in a double ways road near Trento, Italy, where queues frequently occur.

1.5 Research Objectives and Statement of The Problem.

Automatic traffic scene analysis is very interesting in the context of traffic planning and monitoring. The main objective of the proposed research is to construct a system that makes dynamic decision on traffic control (when to change phase of traffic signal) through analyzing the road image and identify the traffic load on a road. The main points that will be covered are:

Perform some preprocessing operations on road image (image resizing, edge detection and segmentation).

Propose two algorithms to specify traffic load through:

A. Measure the length of the queues on road using edge detection of road ends (front and rear), then specify number of vehicles.

B. Count number of vehicles after performing image segmentation.

Find out the estimated time for both algorithms

Analyze and test the proposed algorithms by comparing the results (number of vehicles and estimated time) of each method with the actual number of vehicles and actual time. Finally, suggest the best algorithm based on the comparison results.

1.6 Thesis Layout

Chapter two describes the Main principles of image processing used in the proposed technique. Chapter three describes the technique developed in this thesis. Chapter four presents the experimental results. The last chapter summarizes the conclusion and direction for future work.

Chapter Two

Main Principles of Image Processing

2.1 Introduction

The traffic control consideration is an old problem at which many methods are used to solve it. Counting the number of vehicles in the road has become one of the most important technologies during the last decades.

Over the last few decades, road traffic analysis has attracted several researchers, who conducted relevant studies and contributed their experiences in applying image processing and computer vision techniques to analyze the road traffic and address several associated problems.

The main issue of this thesis is to use a camera to count the vehicles within a road during a certain period of time based on image processing at which certain image morphology, edge detection and segmentation are need to develop traffic control system.

This Chapter describes the main principles of image processing used in the proposed technique to detect vehicle at the image and count them. Some of the Morphology of image processing used to analysis the images will be illustrated. Next section covers some of image enhancement operations used in the proposed technique like ***image subtraction***, ***image negatives***, ***intensity thresholding***, in addition to ***edge detection***, and ***image segmentation*** with segment labeling.

2.2 Main Principles of Image Processing

The field of image processing has grown considerably during the past decade with the increased utilization of imagery in myriad applications coupled with improvements in the size, speed, and cost effectiveness of digital computers and related signal processing technologies. Image processing has found a significant role in scientific, industrial, space, and government applications.

An image may be defined as a two-dimensional function, $f(x, y)$, where x and y are spatial (plane) coordinates, and the amplitude of f at any pair of coordinates (x, y) is called the intensity or gray level of the image at that point. When x, y, and the amplitude values of f are all finite, discrete quantities, we call the image a digital image [GON 08]. Image could be classified as *palette images* at which images contains palette which provide certain number of colors (binary, grayscale, colored images) and the pixel represents an index to the palette, or *true color image* each pixel consists of 3 bytes (Red, Green, Blue). In this research, true color images need to be transformed to and grayscale images and binary image (black and white).

- **Image Grayscale:** In photography and computing, a grayscale or grayscale digital image is an image in which the value of each pixel is a single sample, that is, it carries only intensity information. Images of this sort, also known as black-and-white, are composed exclusively of shades of gray, varying from black at the weakest intensity to white at the strongest.

  Grayscale images are distinct from one-bit black-and-white images, which in the context of computer imaging are images with only the two colors, black, and white (also called bilevel or binary images). Grayscale images have many shades of gray in between. Grayscale images are also called monochromatic, denoting the absence of any chromatic variation.

- **Binary Image:** The identification of objects within an image can be a very difficult task. One way to simplify the problem is to change the grayscale image into a binary image, in which each pixel is restricted to a value of either 0 or 1. Numerically, the two values are often 0 for black, and either 1 or 255 for white.

  Binary images are often produced by thresholding a grayscale or color image, in order to separate an object in the image from its background. The color of the object (usually white) is referred to as the foreground color. The rest (usually black) is referred to as the background color. However, depending on the image which is to be thresholded, this polarity might be inverted, in which case the object is displayed with 0 and the background is with a non-zero value.

Some morphological operators assume a certain polarity of the binary input image so that if we process an image with inverse polarity the operator will have the opposite effect. For example, if we apply a closing operator to a black text on white background, the text will be opened.

2.3 Image Subtraction

An image subtraction algorithm is used to separate an object in the image from its background (in this project to recognize a moving object). The major reason for using this algorithm is that it is simple and can be implemented by the limited real-time processing capabilities of the image processing board. It turned out to work acceptably well. It simply compares the previous frame image with the current one. The result of the subtraction is visible from the monitor, which shows the same data as is ``seen'' by the computer, i.e. if something is moving, then a white pixel shows up on the screen in that position. It could be obtained by computing the difference between all pairs of corresponding pixels from *f* and *h* (were f and h are the two image that are to be subtracted). The key usefulness of subtraction is the enhancement of differences between images. Subtraction of two images is often used to detect motion. Consider the case where nothing has changed in a scene; the image resulting from subtraction of two sequential images is filled with zeros-a black image. If something has moved in the scene, subtraction produces a nonzero result at the location of the movement [GON 08].

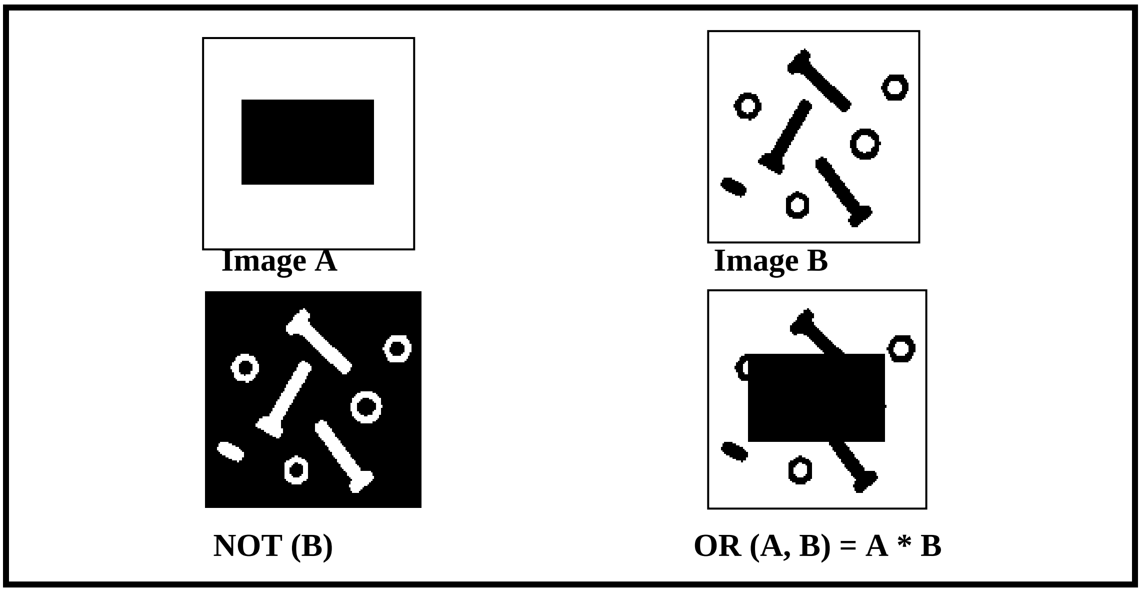


Figure 2.1 Boolean Logical Rules. [CON08].

As result of image subtraction, noise objects could appear, for that, it is usually necessary to perform a high degree of noise reduction on the image before performing higher-level processing steps, such as edge detection, segmentation, etc.

2.4 Noise Removal and Filtering

Digital images are prone to various noise types. Noise is the result of errors in the image acquisition process that result in pixel values that do not reflect the true intensities of the real scene. Linear filtering could be used to remove certain types of noise. Certain filters, such as averaging or Gaussian filters, are appropriate for this purpose. For example, an averaging filter is useful for removing grain noise from a photograph. Because each pixel gets set to the average of the pixels in its neighborhood, local variations caused by grain are reduced.

Median filtering is another type of filtering that acts almost like average filter. In average filtering, each output pixel is set to an average of the pixel values in the neighborhood of the corresponding input pixel. However, with median filtering, the value of an output pixel is determined by the *median* of the neighborhood pixels, rather than the mean. The median is much less sensitive than the mean to extreme values (called *outliers*). Median filtering is therefore better able to remove these outliers without reducing the sharpness of the image, [GON 08] [MAT 09].

The median filter is a non-linear digital filtering technique, often used to remove noise from images or other signals. It is particularly useful to reduce speckle noise and salt and pepper noise. Its edge-preserving nature makes it useful in cases where edge blurring is undesirable. ***The idea is to pick a pixel, calculate the median of neighboring pixels' values by sort the neighboring pixels in numerical order, pick the median value and make it the pixels value***. Repeat this process for each pixel in the image.

2.5 Edge Detection

Edges are significant local changes of intensity in an image. Typically occur on the boundary between two different regions in an image [MIK 96]. Edge is the boundary between two regions with relatively distinct gray-level properties. The goal of edge detection is to mark the points in a digital image at which the luminous intensity changes sharply**,** [TIN 05].

Goal of edge detection is to produce a line drawing of a scene from an image of that scene and important features that can be extracted from the edges of an image (e.g., corners, lines, curves). These features are used by higher-level computer vision algorithms (e.g., recognition).

There are many ways to perform edge detection. However, most of them may be grouped into two categories, gradient and Laplacian. The gradient method detects the edges by looking for the maximum and minimum in the first derivative of the image. The Laplacian method searches for zero-crossings in the second derivative of the image to find edges**,** [MIK 96].

Edge detection of an image reduces significantly the amount of data and filters out information that may be regarded as less relevant, preserving the important structural properties of an image. There are different algorithm that could be used to perform edge detection, such as:

1-Canny: The Canny edge detection operator was developed by John F. Canny in 1986 and uses a multi-stage algorithm to detect a wide range of edges in images. Most importantly, Canny also produced a computational theory of edge detection explaining why the technique works. The Canny Edge Detection Algorithm is illustrated in **Appendix (A),** and The behavior of canny algorithm is shown in figure 2.2**.**

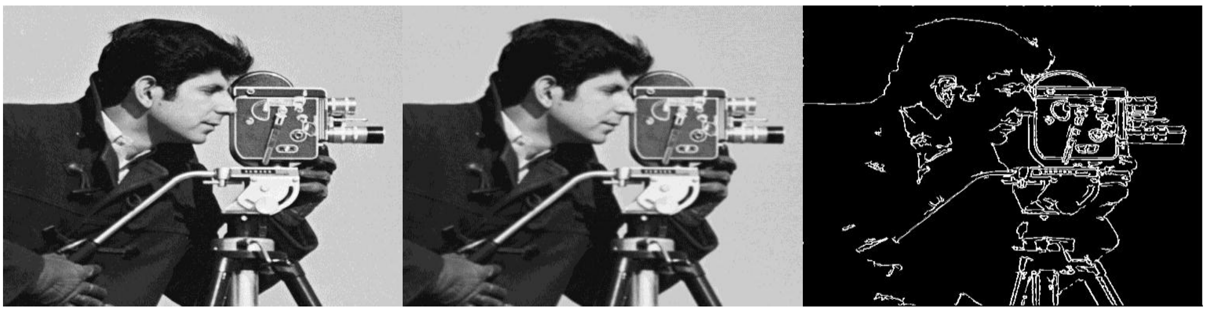


Figure 2.2 Canny Edge Detection. [KOB 07].

2-Prwitte: Prewitt is a method of edge detection in computer graphics which calculates the maximum response of a set of convolution kernels to find the local edge orientation for each pixel. Various kernels can be used for this operation. The Prewitt operator measures two components. The vertical edge component is calculated with kernel Kx and the horizontal edge component is calculated with kernel Ky. |Kx| + |Ky| gives an indication of the intensity of the gradient in the current pixel.

Kx=

-1	0	+1
-1	0	+1
-1	0	+1

Ky=

+1	+1	+1
0	0	0
-1	-1	-1

The behavior of Prewitt algorithm is shown in Figure 2.3.

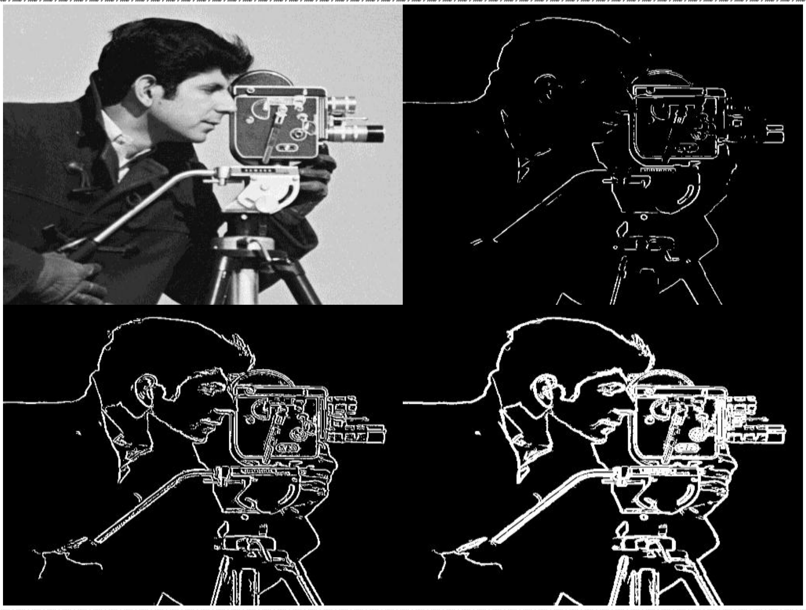


Figure 2.3 Prewitt Edge Detection.[KOB 07].

3-Sobel operator is an operator used in image processing, particularly within edge detection algorithms. Technically, it 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 behavior of Sobel algorithm is shown in figure 2.4 .

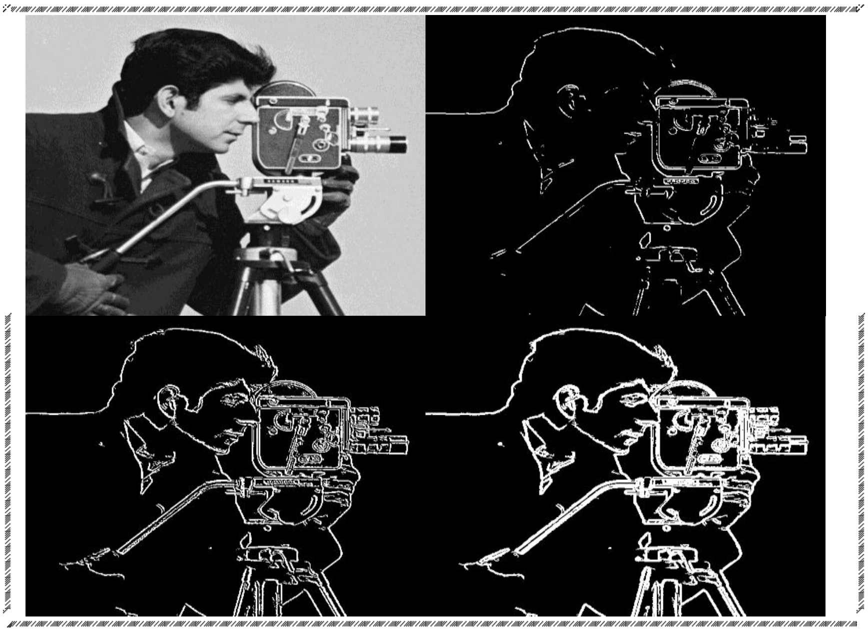


Figure 2.4 Sobel Edge Detection. [KOB 07].

The Sobel operator is the magnitude of the gradient computed by:

$$M\sqrt{s_x^{\,2} + s_y^{\,2}}$$ ………………….(٢.1), [CON 08].

where the partial derivatives are computed by:

$$s_x = (a_2 + ca_3 + a_4) - (a_0 + ca_7 + a_6)$$
$$s_y = (a_0 + ca_1 + a_2) - (a_6 + ca_5 + a_4)$$ ……………….(٢.2), [CON 08].

with the constant c = 2.

Sobel edge detection operation extracts all of edge in an image, regardless of direction. The Sobel operation has the advantage of providing both a differencing and smoothing effect. Sx and Sy can be implemented using convolution masks is defined as follows.

$s_x =$

-1	0	1
-2	0	2
-1	0	1

$s_y =$

1	2	1
0	0	0
-1	-2	-1

It 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 Sobel operator is based on convolving the image with a small, separable, and integer valued filter in horizontal and vertical direction and is therefore relatively inexpensive in terms of computations.

On the other hand, the gradient approximation which it produces is relatively crude, in particular for high frequency variations in the image, [GON 08].

Morphological operations may be needed after edge detection to fill holes and closing short gaps in strokes.

2.6 Morphological Processing

Morphological image processing is a type of processing in which the spatial form or structure of objects within an image is modified. Morphological image processing techniques are useful for extracting image components that are helpful in representing and describing region shapes. The filters can be described using set of theoretic notation and implemented using simple computer algorithms, [GON 08].

Morphology is a broad set of image processing operations that process images based on shapes. These operations apply a structuring element to an input image, creating an output image of the same size. In a morphological operation, the value of each pixel in the output image is based on a comparison of the corresponding pixel in the input image with its neighbors. By choosing the size and shape of the neighborhood, you can construct a morphological operation that is sensitive to specific shapes in the input image.

The most basic morphological operations are *dilation, erosion, open and close* (see figure 2.5).

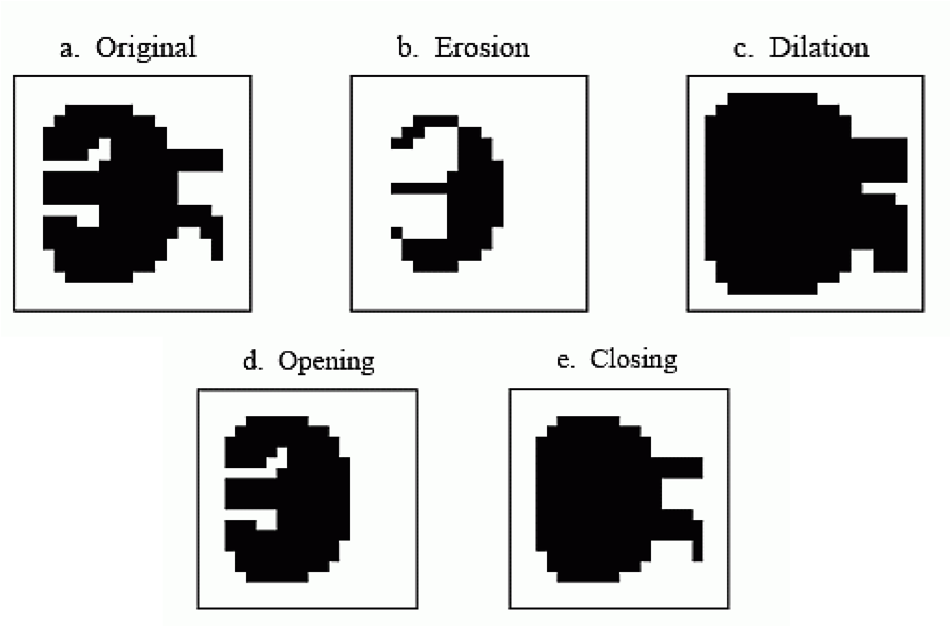


Figure 2.5 Morphological Operations. [CON 08].

Dilation adds pixels to the boundaries of objects in an image, while ***erosion*** removes pixels on object boundaries. The number of pixels added or removed from the objects in an image depends on the size and shape of the structuring element used to process the image. In the morphological dilation and erosion operations, the state of any given pixel in the output image is determined by applying a rule to the corresponding pixel and its neighbors in the input image. The rule used to process the pixels defines the operation as dilation or erosion, [GON 08][MAT 09]

Dilation and erosion are often applied to an image in concatenation. Dilation followed by erosion is called a ***close operation***, where it should be noted that erosion is performed with the reflection of the structuring element. Closing of an image with a compact structuring element without holes (zeros), such as a square or circle, smooth contours of objects, eliminates small holes in objects, and fuses short gaps between objects.

An ***open operation*** consists of erosion followed by dilation. Where again, the erosion is with the reflection of the structuring element. Opening of an image smoothes contour of objects, eliminates small objects, and breaks narrow strokes, [GON 08] [TIN 05].

2.7 Image Segmentation

The goal of image segmentation is to cluster pixels into salient image regions, i.e., regions corresponding to individual surfaces, objects, or natural parts of objects. Image segmentation is one of the primary steps in image analysis for object identification. The main aim is to recognize homogeneous regions within an image as distinct and belonging to different objects.

Partitioning of an image into several constituent components is called segmentation. Segmentation is an important part of practically any automated image recognition system, because it is at this moment that one extracts the interesting objects, for further processing such as description or recognition. Segmentation of an image is in practice the classification of each image pixel to one of the image parts ,[GON 08].

A central problem, called segmentation, is to distinguish objects from background. For intensity images four popular approaches are: threshold techniques, edge-based methods, region-based techniques, and connectivity-preserving relaxation methods.

- Threshold technique: Threshold techniques, which make decisions based on local pixel information, are effective when the intensity levels of the objects fall squarely outside the range of levels in the background. Because spatial information is ignored, however, blurred region boundaries can create havoc.
- Edge-based methods: center around contour detection: their weakness in connecting together broken contour lines make them, too, prone to failure in the presence of blurring.
- Region-based method: the image is partitioned into connected regions by grouping neighboring pixels of similar intensity levels. Adjacent regions are then merged under some criterion involving perhaps homogeneity or sharpness of region boundaries. Over stringent criteria create fragmentation; lenient ones overlook blurred boundaries and over merge. Hybrid techniques using a mix of the methods above are also popular.
- Connectivity-preserving relaxation-based segmentation method: usually referred to as the active contour model, was proposed recently. The main idea

is to start with some initial boundary shape represented in the form of spline curves, and iteratively modifies it by applying various shrink/expansion operations according to some energy function. Although the energy-minimizing model is not new, coupling it with the maintenance of an ``elastic" contour model gives it an interesting new twist. As usual with such methods, getting trapped into a local minimum is a risk against which one must guard; this is no easy task, [GON 08].

Segmentation stage does not worry about the identity of the objects. They can be labeled later. The first region growing method was the seeded region growing method. This method takes a set of seeds as input along with the image. The seeds mark each of the objects to be segmented. The regions are iteratively grown by comparing all unallocated neighboring pixels to the regions. The difference between a pixel's intensity value and the region's mean, δ, is used as a measure of similarity. The pixel with the smallest difference measured this way is allocated to the respective region. This process continues until all pixels are allocated to a region.

Chapter Three

System Development

3.1 Introduction

Pattern recognition and computer vision receive as an input image and produces descriptive information concerning the received image. Pattern recognition and computer vision are important activities in many domains such as: medicine (for example it can be used in cancer-cells detection), space, handwriting recognition, biometric identification, document classification, industry, optical character recognition, internet search engines. Pattern recognition could be used to discriminate objects in an image through image segmentation, edge detection and classification. One of the applications that make use of object discrimination is in traffic control.

The traffic control problem considered as an old problem at which there are a variety of monitoring systems used for counting vehicles, such as infrared detectors and cable to counting wheels, or through road image analysis which is the core this research.

In this research, traffic jam could be identified through analyzing the image shut taken by digital camera, with specific resolution and distance. Two approaches are suggested to find out the traffic jam and make decision about the time needed for green light that will reduce traffic jam, the first one through detecting the queue-length of vehicles on a track, while the second one by counting number of the vehicles on a track based on object discrimination. Image processing techniques are used to analysis current traffic situation form received image, then a signal is send to control traffic light's scope to advance to the higher class the traffic condition.

Counting vehicles or knowing cars-queue-length from an image in noisy environment is a challenging task. This research proposes a technique that uses one camera to detect the vehicle position and to count number of cars. The proposed technique is mainly based on image segmentation.

The proposed technique to vehicle counting is to detect vehicles to find the number of vehicles at every image. At first, compare current image with background to calculate the final number of entering vehicles in area of traffic.

This chapter discusses the camera view of the proposed technique, and then describes how to detect vehicles including queuing and segmentation model. The next section describes the vehicles counting algorithm associated with the flow Chart of the proposed techniques. The chapter also points out the main steps used in the suggested system. Finally, the two suggested approaches will be explained and there behavior will be discussed.

3.2 Camera View

As well known that roads could be of different forms and different number of tracks, also the traffic light could be used to control competing flows of traffic in the road. In this research, the suggested system is used to control traffic flow on one way of road, (One camera is needed) .so to achieve full road control, four cameras are needed. This technique depends mainly upon the vehicles and the road tracks, so that it is necessary to pay attention to the location of the camera. Camera should be located in a central road that allows it to take images clearly as shown in (Figure 3.1).

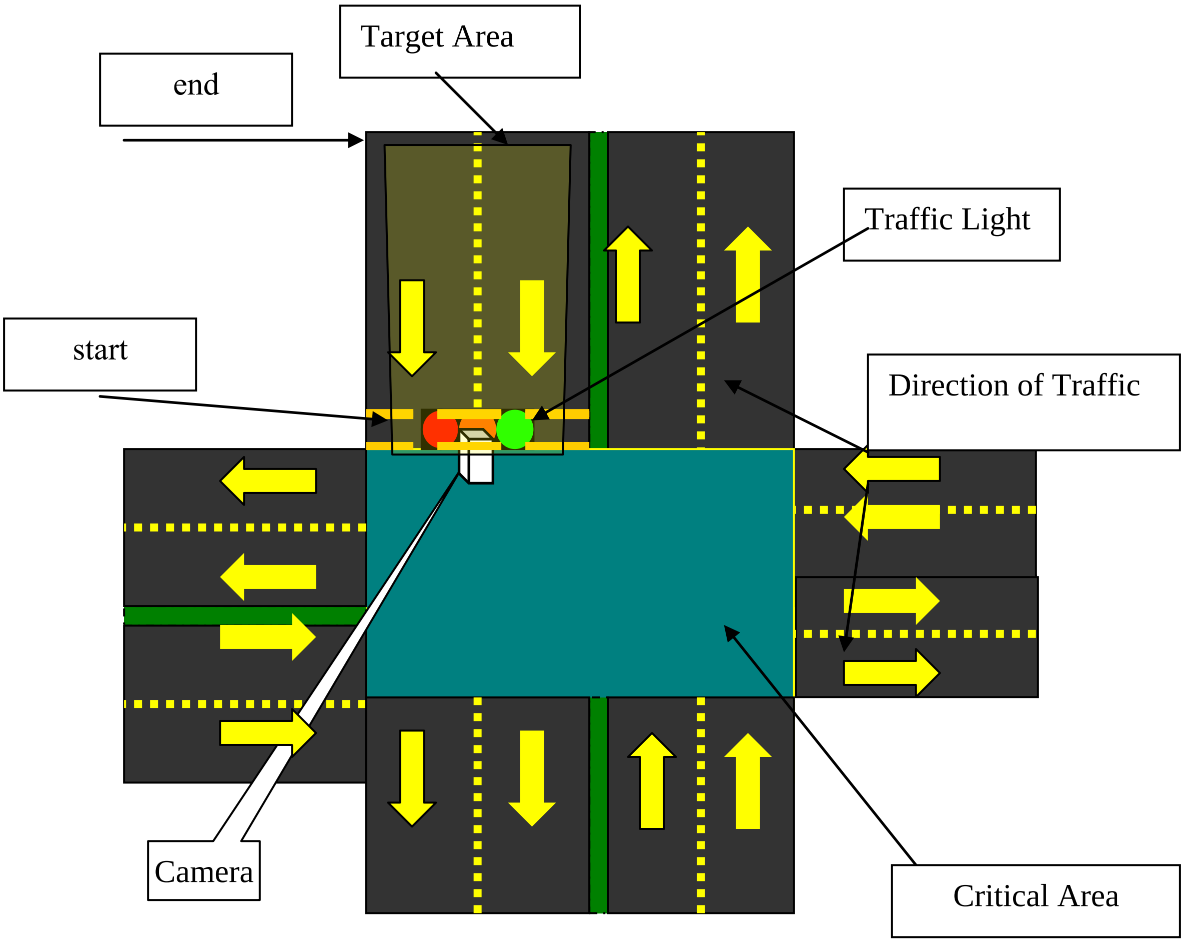


Figure 3.1 Camera View

The proposed technique doesn't need any other sensors except the fixed camera to capture images and also there is no need to any equipment other than a PC to analyze data.

3.3 Image Description

To evaluate the performance of the suggested approaches, about 50 image is taken. Photo specifications are as shown in Table 3.1 .

Table 3.1 Photo Specifications

File format	Width	Height	Bit-Depth	Color Type	Number of Samples	Coding Method	Coding Process
Jpg'	830	897	24	'truecolor'	3	'Huffman'	Sequential

3.4 System Model

Detecting Vehicles in images is a fundamental task for realizing surveillance systems or intelligent vision based human computer interaction. The proposed system is divided in to two techniques: the first depends on Vehicle detection to count the number of cars, and second depends on length of Vehicle-queue to compute number of cars on road track. The suggested system mainly consists of three phases: preprocessing phase, image analysis phase (consists of two modules queue length and Counting Vehicles), and the timing decision phase (as illustrated in Figure 3.2).

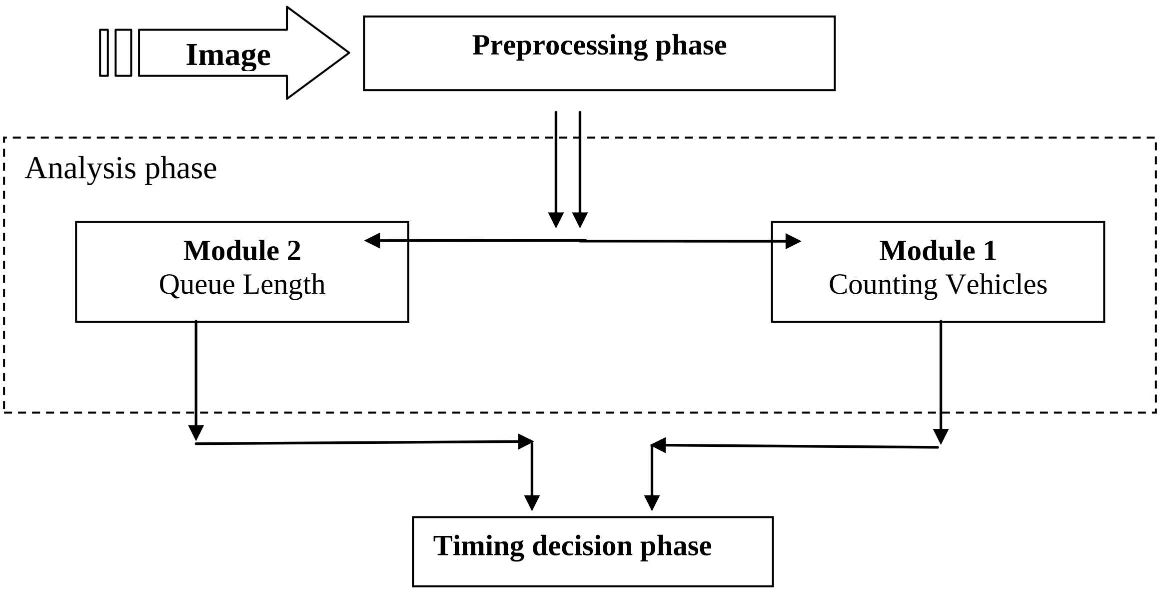


Figure 3.2 System Model

3.4.1 Preprocessing Phase

In this research, the image of road when it is empty is needed (call it Background image) in addition to the image taken every one day and if any image don’t have any vehicles or any thing its background and take it in memory (update background). For the road (call it traffic image) to perform background subtraction which is a simple and effective technique to extract the foreground objects from the scene. Background subtraction is considered one of the major steps to perform image analysis to control traffic. Before starting with image analysis, the following steps are needed:

1. Read image data for base image and traffic image. (See Figure 3.3).

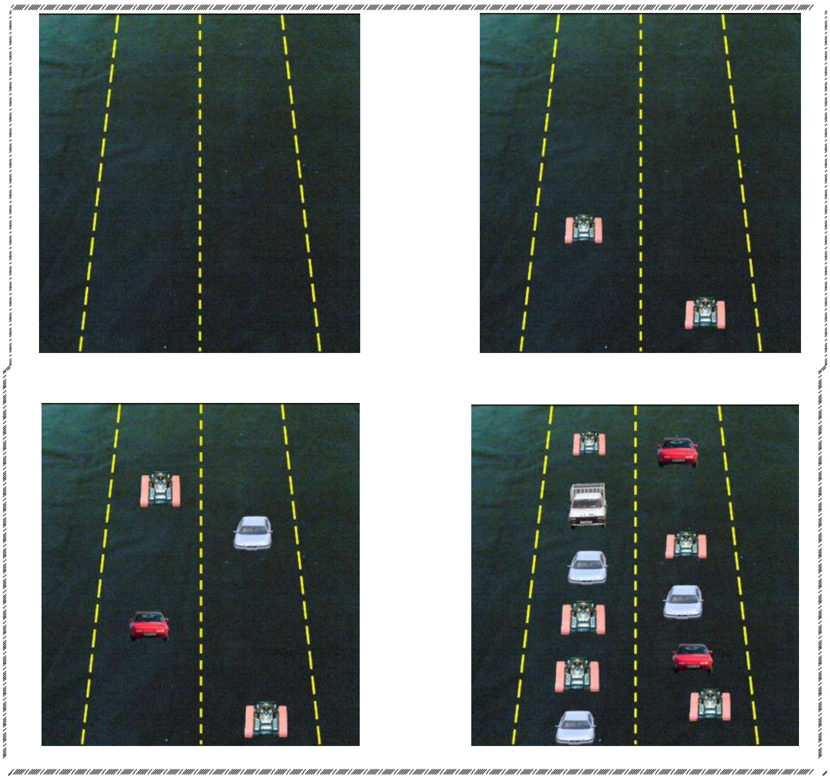


Figure 3.3 Sample Of Input Image

The system will accept as input an image of any size, the image will be resized to 897×830 to have standard image size.

2. Convert both image background and traffic-image to grayscale(as shown in Figure 3.4) RGB colors are specified in terms of the three primary colors: red (R), green (G),

and blue (B). RGB values could be converted to grayscale values by forming a weighted sum of the R, G, and B components:

Grayscale = 0.2989 * R + 0.5870 * G + 0.1140 * B ---- (3.1). [CON 08],[MAT 09].

Grayscale is a data matrix whose values represent intensities within some range.

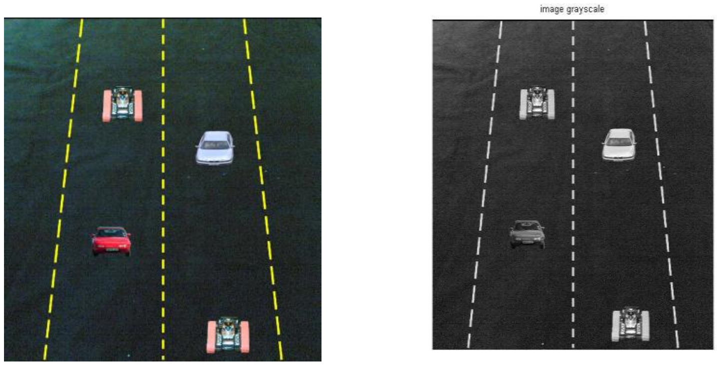


Image before convert image after convert

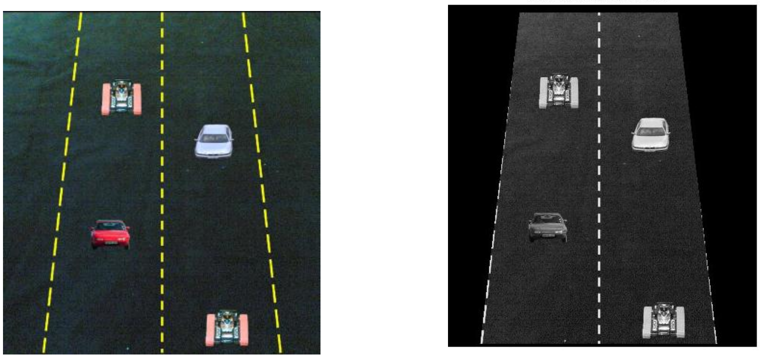
Figure 3.4 RGB Image Converted To Grayscale

3. Create Mask: A Region Of Interest (ROI) is a portion of an image that you want to filter or perform some other operation on. The resulting image contains the entire region of interest, where region of interest signifies the region in the image where the queues of vehicles are expected.

This method is intended to separate the part of the road where vehicles are moving in one direction Figure 3.5 (a). This action is essential because it simplify an information processing extracted from more than one image. Masking algorithm is given by formula (3.2):

$$N(p) = M(p) \times V(p) \qquad \text{--------------------(3.2). [MAT 09].}$$

where M(p) is an image point value in primary frame, N(p) is a new image point in the output image Figure 3.5 (b), V (p) is mask value for point p: V (p) = 0 if corresponding pixel is eliminated, otherwise V(p)=1. Masking is applied to each RGB color separately.



(a) Before Masking. (b) After Masking.

Figure 3.5 Image Masking

It is convenient to constrain lane mask by using graphical editor especially created for system configuration. This feature connects sharp points given by user into persistent curve. As a first step, one should select a portion of the image (either by machine or manually) such that it contains entire region of interest. [ATK 05].

4. Image Subtraction: It is the method to calculate a difference between the input image and the background image (Subtract Background image from traffic image) which is a road surface image including no vehicles. By processing this method, a background difference image which is only of vehicles is made(see Figure 3.6). The subtraction of two images is performed straightforwardly in a single pass. The output pixel values are given by Equation (3.3):

$$D(X,Y) = C(X,Y) - B(X,Y) \qquad \text{-----------(3.3). [MAT 09]}$$

Where D(X,Y) is the difference image, C(X,Y) is the traffic image, B(X,Y) is the background image.

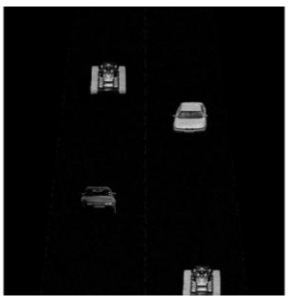
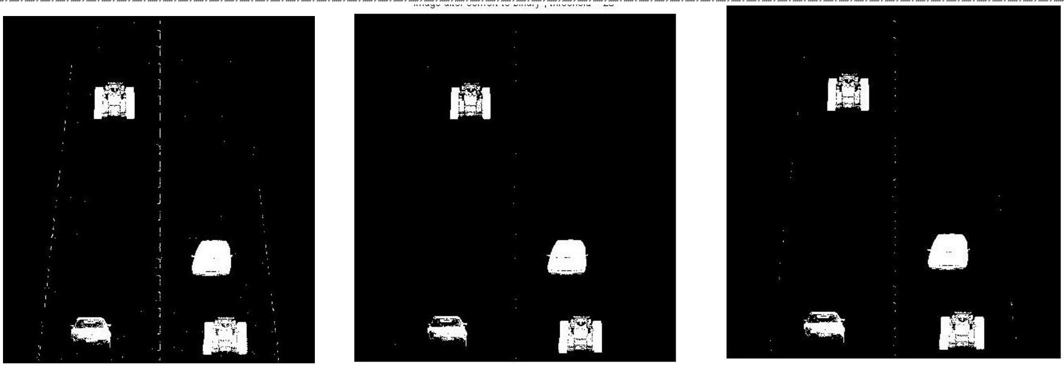


Figure 3.6 Image Subtraction

5. Convert result image to binary image, based on threshold, applying binary image thickening to prevent total erasure and to ensure connectivity of edges(see Figure 3.7) The difference image needs to be transformed into binary image by:

$$R_i(x,y)= \begin{cases} 0 & , \textit{if } |D_i(x,y)| < T \\ 255 , \textit{ otherwise} & \end{cases}$$ ----------(3.4). [CON 08],[MAT 09].

where $\mathbf{D_i(X,Y)}$ is the difference image, $\mathbf{R_i(x,y)}$ is a binary image and T is a threshold. In this research, by experiments, it was found that the best T value is 20.



(a) T=15 (b) T=20 (c) T=25

Figure 3.7 Convert Image To Binary

6.Noise Filtration: Image, after subtraction has a lot of speckles caused by noise, which could be removed only by means of filtration. An effective method is threshold filtration just after background elimination. Noise removal using to process the binary image depending on binary image Close and Open operations. To perform noise filtration, median filter is used Figure 3.8 illustrate the resulted image after filtering and close operation .

The ***median filter*** replaces the center value in the window with the median of all the pixel values in the window. The kernel is usually square but could be any shape. An example of median filtering of a single 3×3 window of values is shown below.

$Y(n) = med (X_{n-k} , X_{n-k} ,\dots, X_n ,\dots,X_{n+k})$ ------(٣.5). [MAT 09][CON 08].

1. order X_j .
2. choose middle element.

The idea is to calculate the median of neighboring pixels' values. This can be done by repeating these steps for each pixel in the image. Store the neighboring pixels in an array. The neighboring pixels can be chosen by any kind of shape, for example a box or a cross. The array is called the window, and it should be odd sized. Sort the window in numerical order; pick the median from the window as the pixels value.

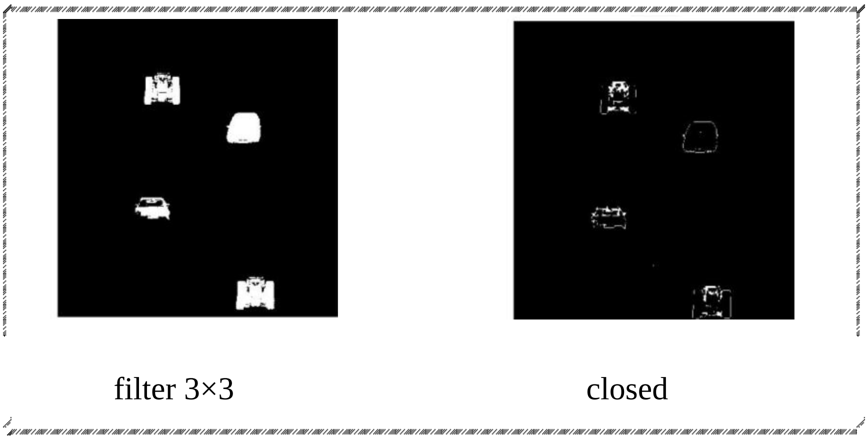


Figure 3.8 Image After Filter Images And Close

7. Using Sobel edge detection filter's to determine the edges in the image as shown in Figure 3.9 .

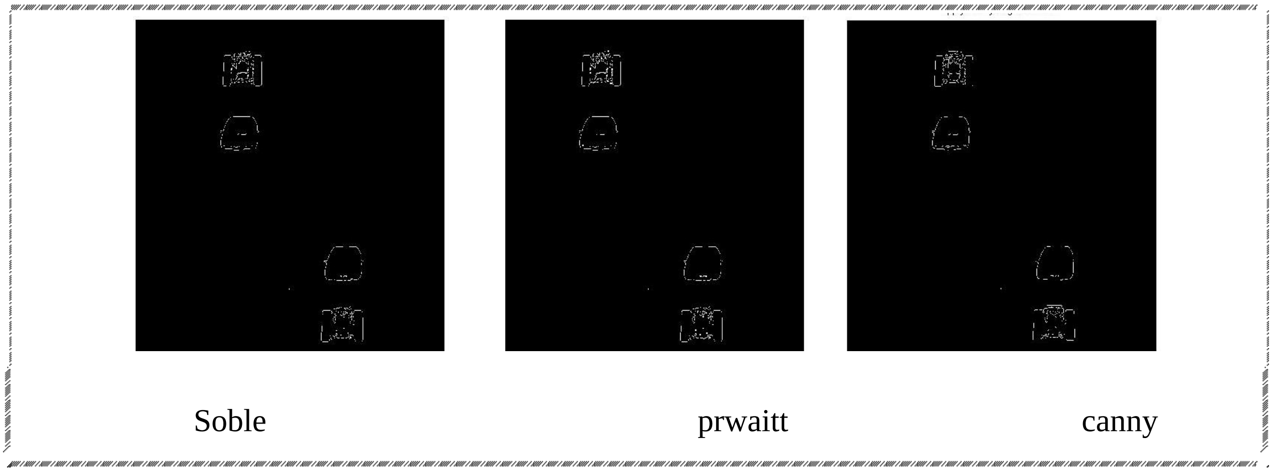


Figure 3.9 Image After Sobel Edge Detection

8. The main morphological operations are dilation and erosion. Dilation and erosion are related operations, although they produce very different results. Dilation adds pixels to the boundaries of objects (i.e., changes them from off to on), while erosion removes pixels on object boundaries (changes them from on to off). Figure 3.10 illustrates the resulted image after dilate operation.

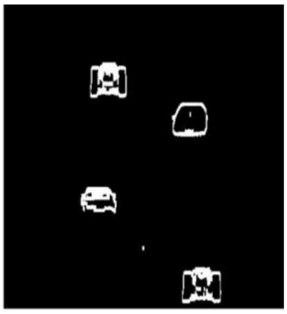


Figure 3.10 Image After Dilate

9. Filling small holes in objects and closing short gaps in strokes using the majority black operator. Figure 3.11 illustrates the Image After Fill Image Regions And Holes.

Figure 3.11 Image After Fill Image Regions And Holes

10. Remove small objects less than 3000 pixels, assuming that smaller objects could be human, animals, rock or any other thing but vehicle. Removes from a binary image all connected components (objects) that have fewer than 3000 pixels. Figure 3.12 illustrates the resulted image after Image After Remove Small Object.

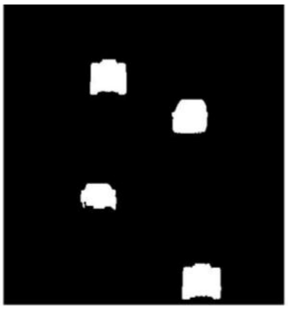


Figure 3.12 Image After Remove Small Object

The flow-graph of preprocessing phase is show in Figure 3.13.

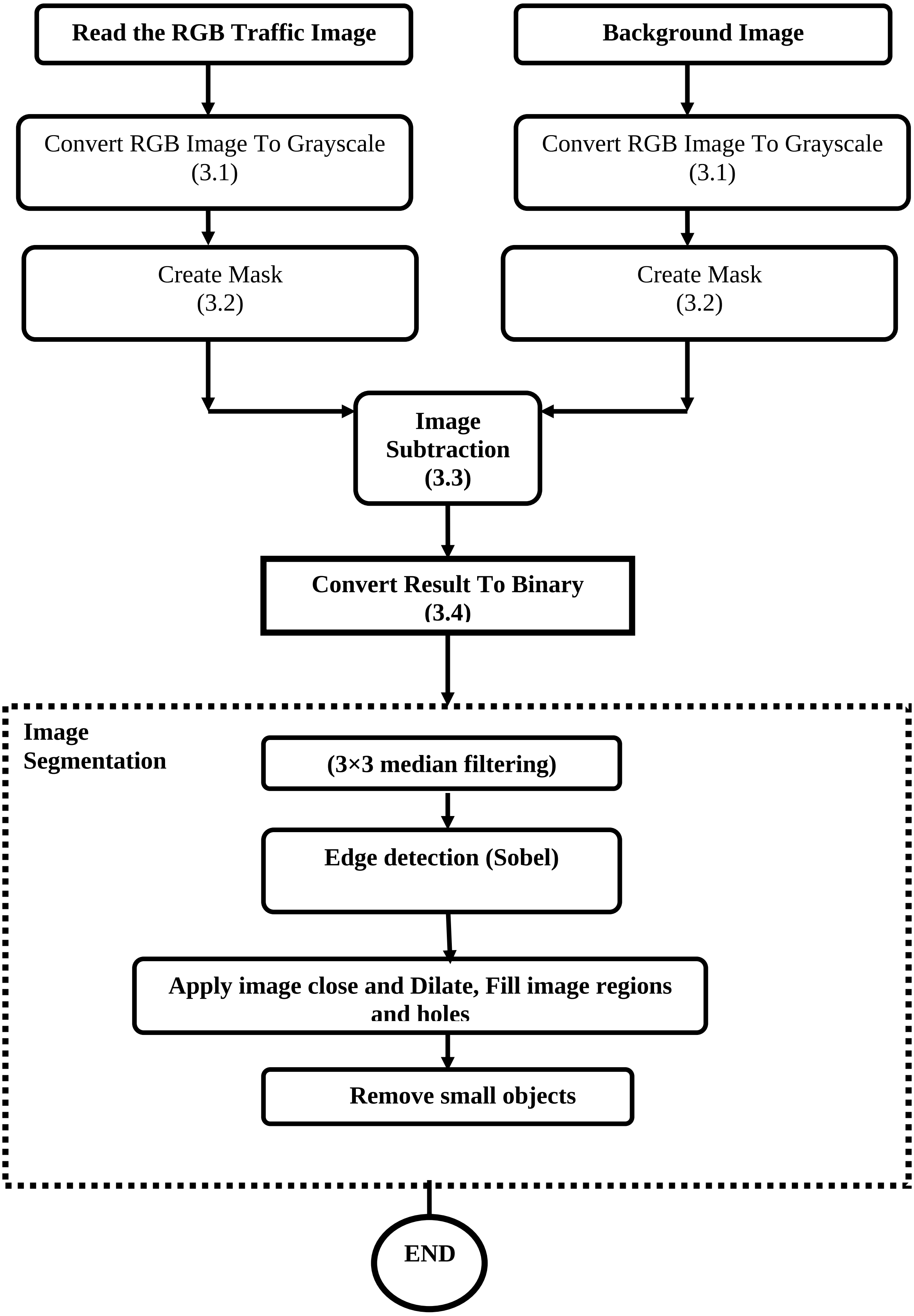


Figure 3.13 Preprocessing Phase

3.4.2 Module Counting Vehicles

The first step in detecting vehicles is segmenting the image to separate the vehicles from the background. There are various approaches to do this, with varying degrees of effectiveness. In this research, to make the segmentation method effective and useful, at first separate vehicles accurately from background, then count out number of objects (vehicles) in the image. To do so, the following condition will be considered (see Figure 3.14 which illustrates the flow graph of counting vehicles module):

- If segment-length (SL)=3 multiples of its width(SW) OR its size (hight×width) >15000 is considered an object (vehicle) and counted.

- If segment size between 22000 and 15000 (i.e. 22000 > size ≥ 15000), then it is considered as a long vehicle or two attached vehicles and counted as 2 objects.

- If segment size between 27000 and 22000 (i.e. 27000 > size ≥ 22000), then it is considered as a long vehicle or three attached vehicles and counted as 3 objects.

- If segment size ≥ 27000, then it is considered as a long vehicle or four attached vehicles and counted as 4 objects.

The important pint is that what is needed is the number of vehicles of the longest track not all objects on all tracks of road. To solve this point, find-out the center-point of each object, compare this point with the start and end points of the other vehicles to find out which vehicles are almost in the same level of this vehicle. In this case, count them all as one object.

As has been calculated the time required for vehicle traffic in the empty space on the street, where the expense of the empty space between any two consecutive terms. Figure 3.15 shows step by step the output of Vehicle counting implementation.

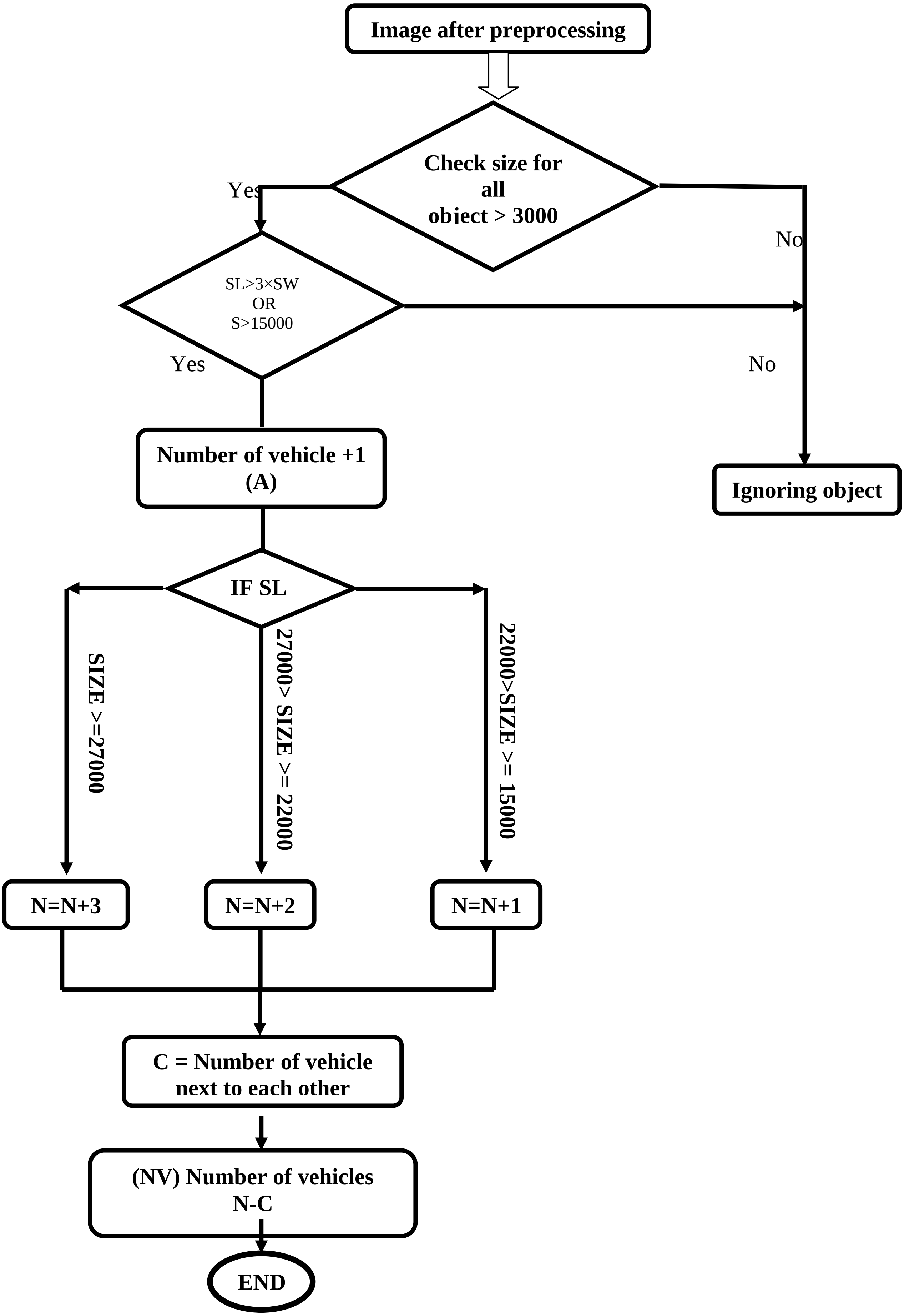


Figure 3.14 Model Counting Vehicle.

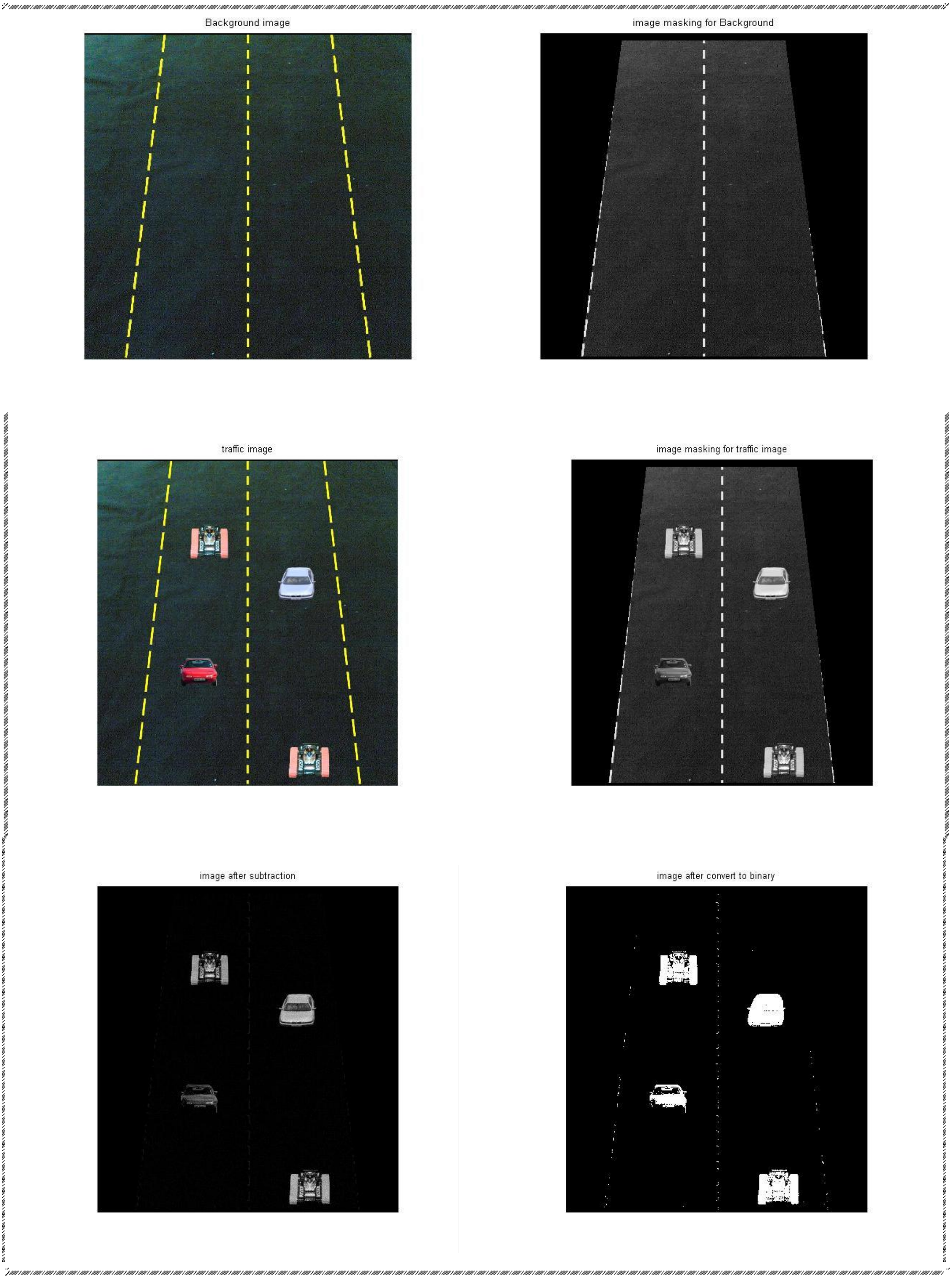
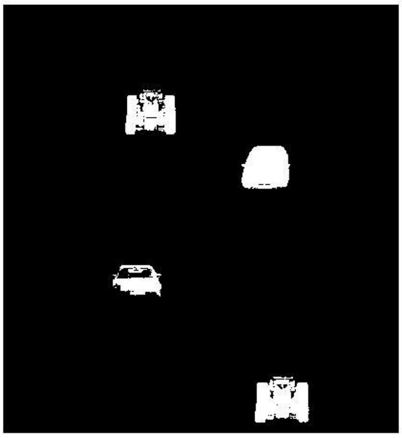
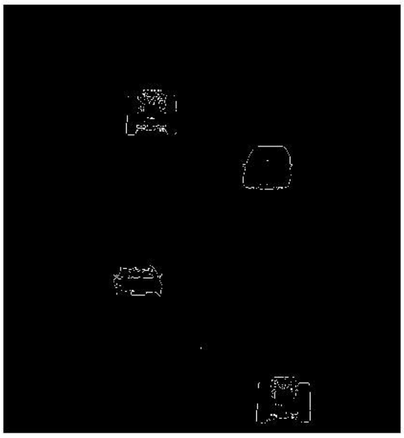

Background image
image masking for Background
traffic image
image masking for traffic image
image after subtraction
image after convert to binary

image after fillter 3*3



apply sobel edge detection



after closed

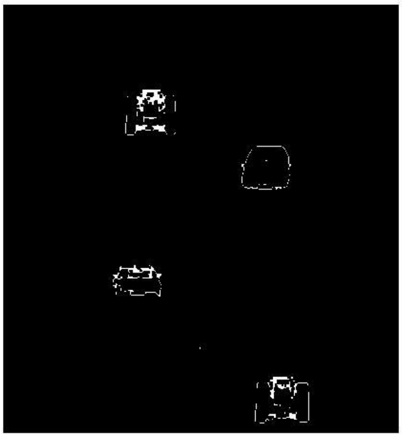


image after Dilate

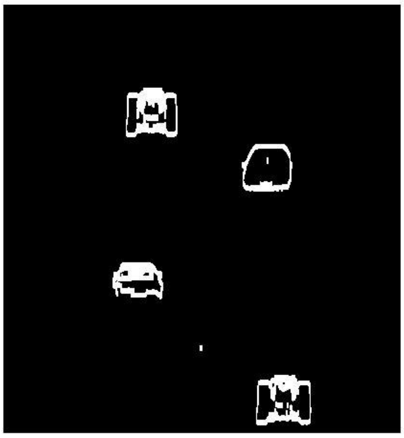


image after Fill image regions and holes

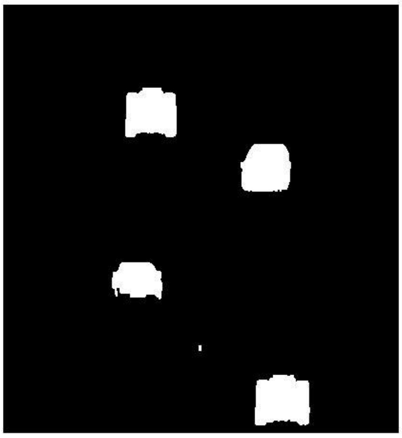
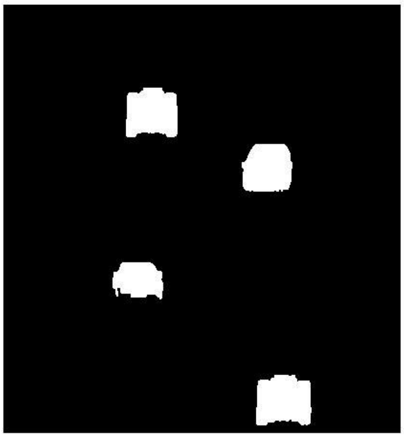


image remove small object less than 3000



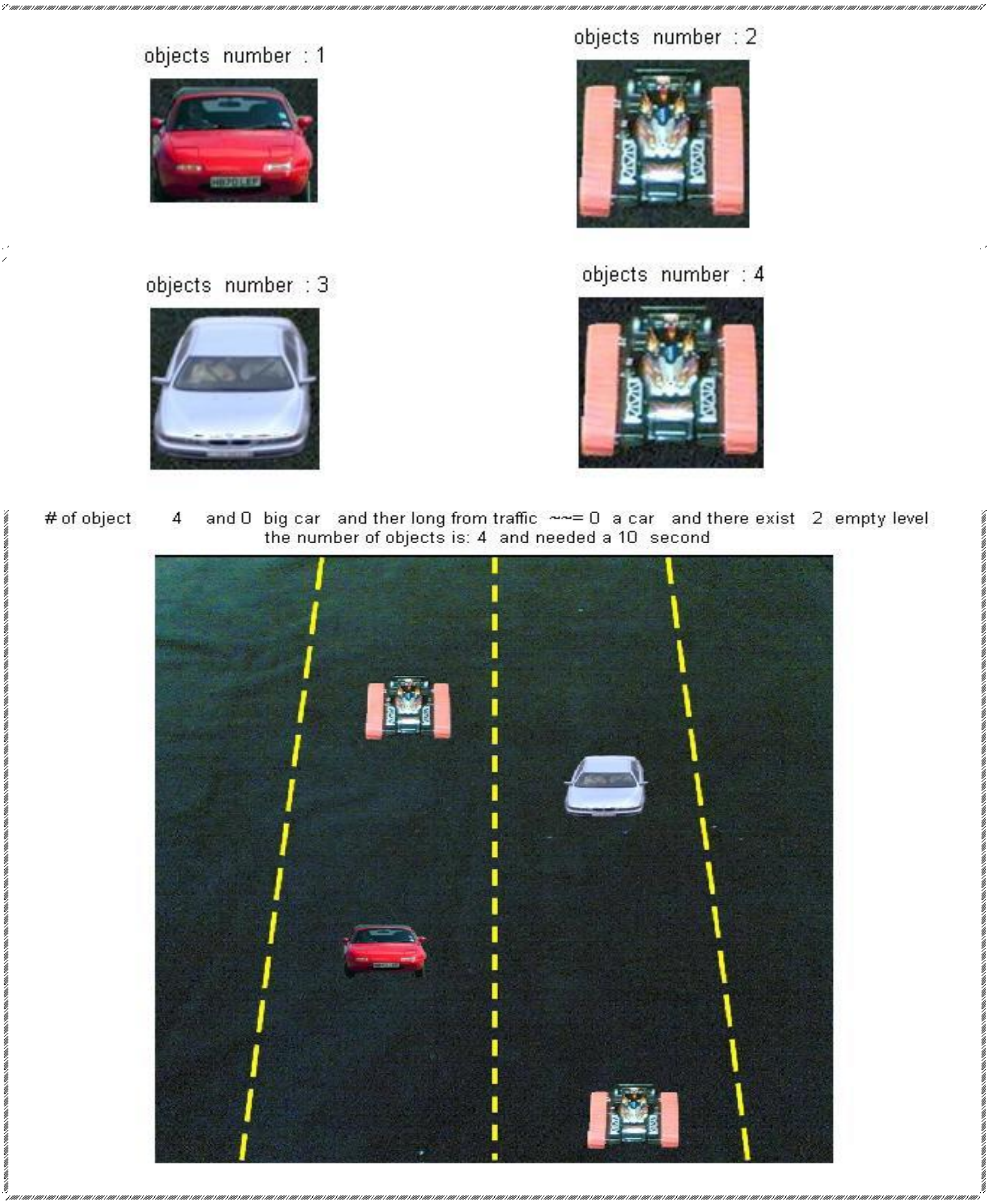


Figure 3.15 Implementation Outcome

3.4.3 Module Queue Length

Traffic queuing length could be found by dividing the road into regions depending on width of the lanes. In this work, the region is divided into two parts (right and left). At first, image is scanned and edge detection is performed. Then dived the resulted image into two lanes (parts). Each part is scanned top-down and bottom-up to calculate the queue length, this can be down by (see Figure 3.16).

- Find the first pixel in first object (by scanning bottom-up and register the first white dot), and last pixel in last object (by scanning image top-down and register the first white dot).
- Find the distance between first pixel and last pixel of the object which represents the queue length (as will be shown later).
- Compute how many vehicles in the road by dividing the queue length by the assumed vehicle length (the assumed vehicle length is find out by calculating different objects sizes in the given images after segmentation) taking in consideration different image resolutions will affect the assumed object size.

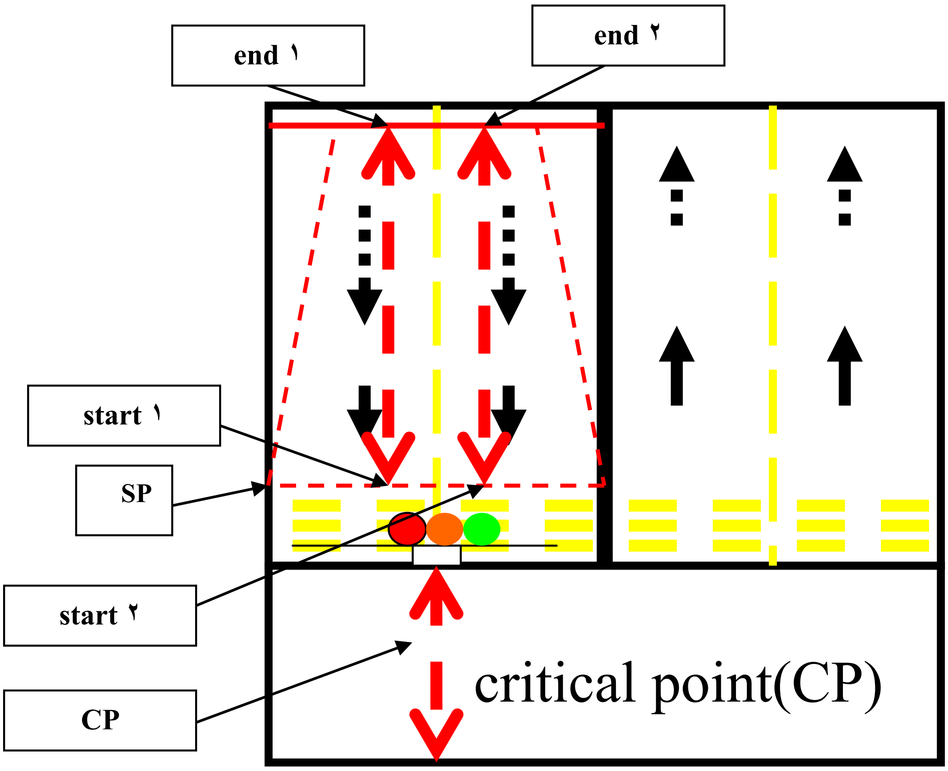


Figure 3.16 Road Map

The length of the queue has been calculated in 6 ways (for the street has two lanes). Firstly, the distance between two points of the XY-plane can be found using the distance formula (3.6) which calculates the distance between (x1, y1) and (x2, y2).

$$D= \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad ....(3.6). \quad [HOW 05] .$$

As we take the point any helicopters $X2 = X_1$ to be the Equation Y_2-Y_1. Beginning at the first column (Start1), the start of the second column (Start2), the end of the first column (End1), the end of the second column (End2), Beginning of the street (SP), after the beginning of the queue from the beginning of the street (SR). The empty space between the queues is (EL), and the safety zone (CP). For categories queues, there are six cases, these are:

1 - The first line is longer than the second line from start to finish.

End1<End2 && Start1>Start2
D=|End2-Start2|
SR= |SP-End2| ----------------(3.7)

2- the second longest queue of the first line from start to finish.

End1>End2 && Start1<Start2
D=|End1-Start1|
SR=|SP-End1| ----------------(3.8)

٣ - the first line is shorter than the second face, and the second longest of his face.

Start2<End1 && End1<End2 && Start1<Start2
D=|End2-Start1|
SR=|SP-End2| ----------------(3.9)

4 - The second line is shorter than the second hand, and the first longest hand.

End1>End2 && Start1<End2 && Start1>Start2
D=|End1-Start2|
SR=|SP-End1| ----------------(3.10)

5 - The first column does not participate with the second.

End1<Start2
D=(End1-Start1)+(End2-Start2)
EL=|Start2-End1|
SR=|SP-End2| ----------------(3.11)

٦ - the second column does not participate with the first.

End2 < Start1
D=(End1-Start1)+(End2-Start2)
EL=|Start1-End2| ----------------(3.12)
SR=|SP-End1|

After obtaining the queue length of the two lanes, the queue lengths are divided by the assumed car length (in this research, the average length of the car almost 100 p), and thus the Equation :

The number of cars = length of the line \ 100 ----------------(3.13)

In Equation 3.11 and 3.12, we note that there is a blank space between queues this needs to calculate the distance to give them time to cross the vehicle through. Figure 3.17 shows low density road at which there is space between the first and the last vehicle on lane.

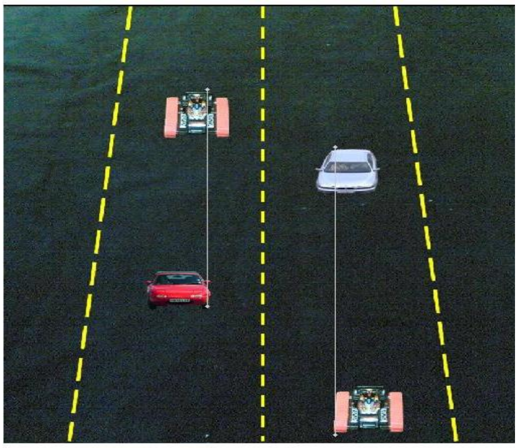
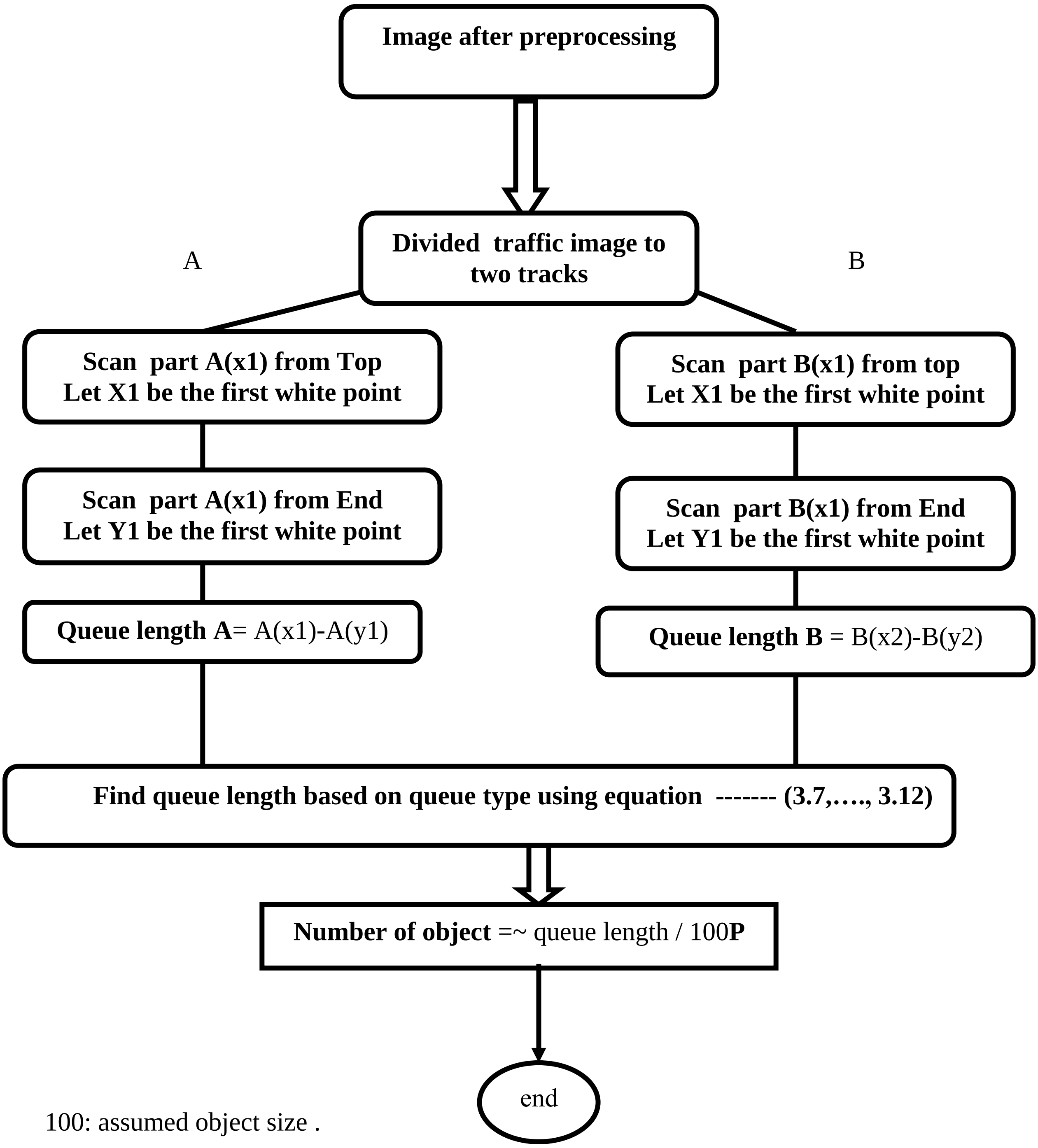


Figure 3.17 Low Density Road

Figure 3.18 shows the flow graph of the Queue Length Module.



100: assumed object size .

Figure 3.18 Queue Module

3.4.4 Timing

After number of vehicles computed, the estimated time for each technique (depending on the computed number of vehicles) is calculated then compared with the actual time from control unit.

To calculate the time for the first approach (counting vehicle), number of cars (NV), for each vehicle (MT=2) seconds were added that represents the time needed by each car to move to the next point (forward), one second to prepare for movement, and one second to reach the next point. In addition to the time needed by car to reach the start point (SR), and the distance between vehicles (EL) which represents empty levels are also calculated were the time need to cross ft is 1 second. Thus the Equation of the time needed to get out cars is show in Equation (3.14).

$$NV\_time = (NV) * (MT) + (SR) + (El) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3.14)$$

As for the second approach (queue length), time calculation depends on the finding number of vehicles (NV) using Equation (3.13). Now, to find out the estimated time, the same principles of the first approach is used.

Chapter Four

Experimental Results

4.1 Introduction

This chapter covers the experimental results retrieved through the application of the proposed technique and presents the analysis of the results. The ability of each of the proposed techniques to detect vehicles is a vital issue, because this leads to detect exactly the vehicles appears in the images, and eventually to realize the best estimated time for green traffic-light for each situation where it should not exceed the exact time needed no more no less. The estimated time calculation (for car starting-up, moving from the current point to the next, and passing the traffic region) based on experiment and consultation of the traffic engineer office in Greater Amman Municipality.

In this research, the following factors should be calculated to find out the estimated total time needed for green traffic-light:

- Number of Small Vehicles. (NV).
- Number of Large Vehicles (BV) is calculated taking in consideration that the large vehicle is counted as two small vehicles since it needs twice the time needed for small vehicle to move or start- up. In this way the problem concerning two contiguous (attached) small vehicles.
- Number of vehicles on the Same Level (SL) should be specified since the vehicles on the same level need approximately the same amount of time needed for moving as one vehicle for that counts all vehicles on the same level as one vehicle.
- The distance of the first vehicle from the Start Point (SR). The distance is set to one vehicle length and since it is open area, it was given half the time needed for vehicle movement.
- The time needed to move to the next point is set to two seconds (MT).

After calculating the above factors, the total number of vehicles (NVfinal) that affect the estimated time is calculated using Equation (4.1), while the estimated time calculated using Equation (4.2).

As has been calculated the time required for vehicle traffic in the empty space on the street, where the expense of the empty space between any two consecutive terms. (Blank account-level (EL)), and then estimate the distance of time where you give each unit a distance of one second to get to them.

NVfinal= NV+BV-SL ……………..(4.1)

Time =(NVfinal×MT)+SR +EL ……………..(4.2)

For queuing approach, the queue length is calculated as mentioned in chapter three. The expected total number of vehicles is calculated using Equation (4.3). Based on the observation of the used pictures, vehicle_length is assumed to be 100. the estimated time is calculated using Equation (4.2) at which Objects_in_queue is used instead of NVfinal, as in Equation (4.4).

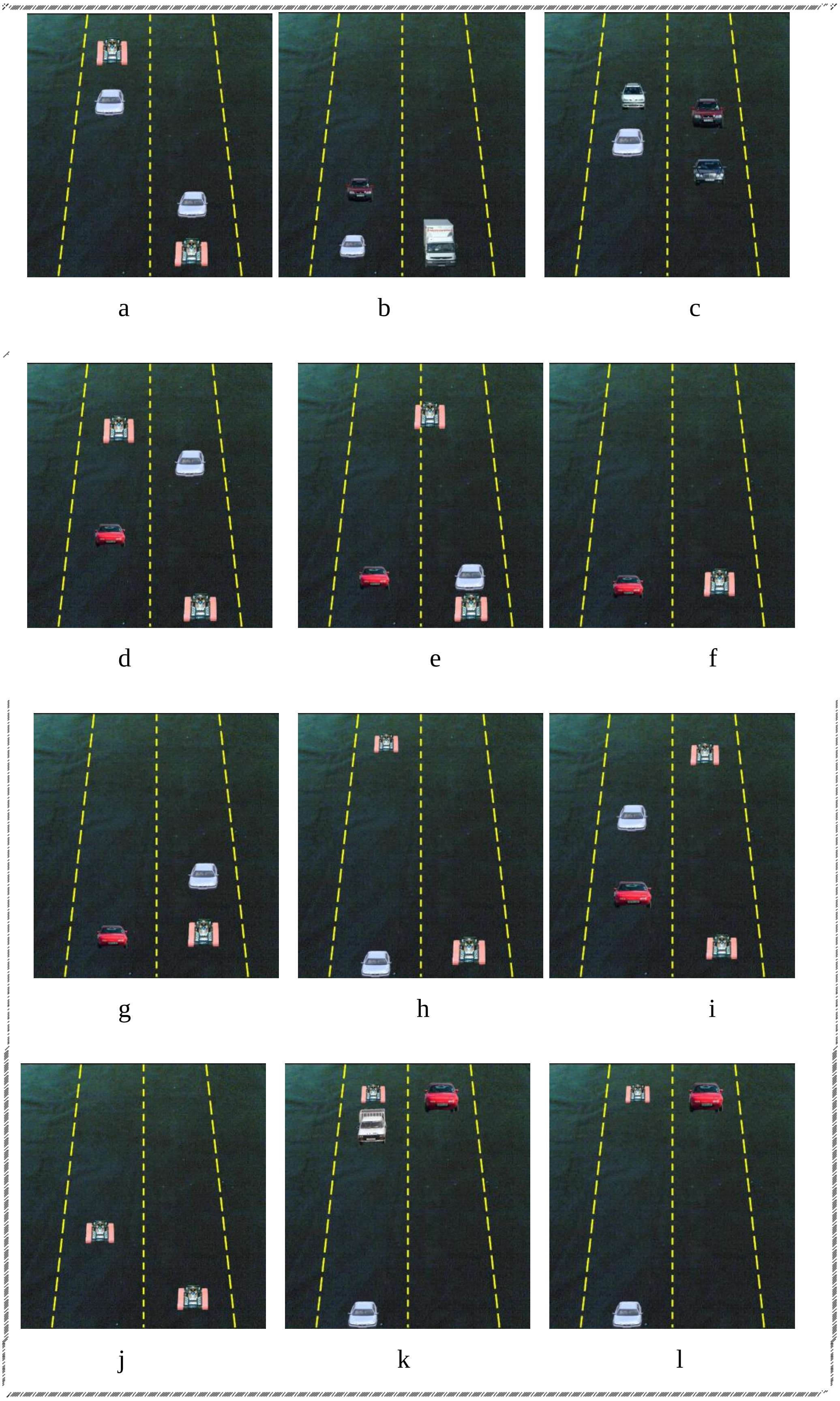
Objects_in_queue= (queue_length)/(vehicle_length) ……….. (4.3)

Time =(Objects_in_queue ×MT)+SR + EL …………………..(4.4)

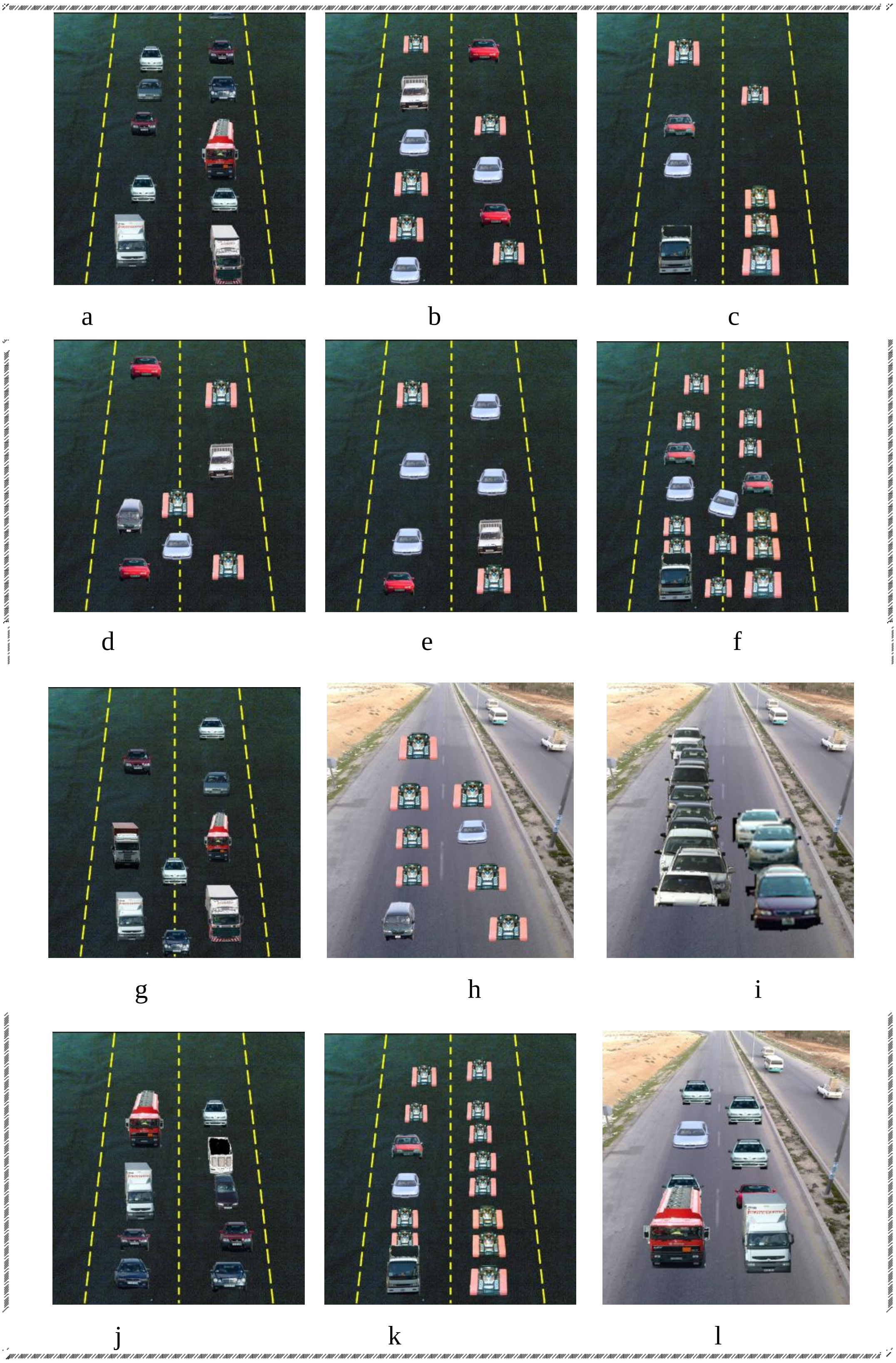
4.2 Image Data

To evaluate the performance of the two proposed approaches, thirty-two pictures are taken with different sizes but same resolution. These pictures are partitioned into two data-sets, according to the number of cars in the picture, *few cars with a lot of space as in Figures (4.1 a-l), Crowded pictures as shown in Figures(4.2 a-l).*

.



Figures 4.1 Low Density



Figures 4.2 High Density

4.3 Experimental Results: Analysis & Comparison

The experimental results analysis and comparison are needed to answer the questions:

- *which of the two approaches is more accurate in specifying number of cars on road?*
- *What is the advantages and disadvantages of each approach.*

Using two different sets of pictures, crowded and non-crowded sets, each of the two sets will be analyzed separately and the behavior of the two approaches is evaluated by comparing the number of cars predicted by each approach compared with the actual number of cars in each picture with each set.

The actual time was calculated as follows(after dividing the road into levels as shown in figure 4.3):

1 - Calculate the number of vehicles. (NV).

2 - Calculate the number of large vehicles. (BV).

3 - Calculate the number of vehicles that are on one level. (SL).

4 - Calculating the blank. (EL).

5 - Calculate the final number of vehicles. (NV_final).

NV_final = NV + BV-SL

6 - Account the distance from the starting point. (SR).

7 - Calculate the final time (Time). Time = (NV_final × 2) + EL + SR

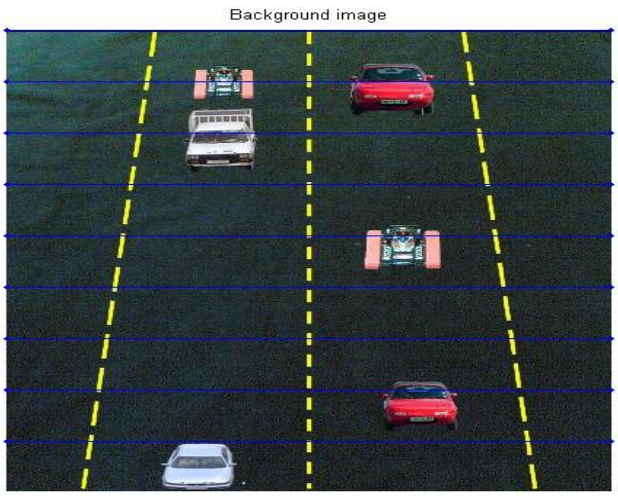


Figure 4.3 Image is Divided Into Levels

Table 4.1 shows the results of the comparison points (Number of vehicle, Estimated time, Empty level, Cars beside other, Distance, number of Big-cars) for actual, vehicle counting, and queuing system for pictures that represents low-density.

Table 4.1 Low Density

	Actual							Counting vehicles							Queue							
	No. of vehicle	Big car	Distance	Beside other	No of vehicle (4.1)	Empty level	E.Time (4.2)	No. of vehicle	Big car	Distance	Beside other	No of vehicle (4.1)	Empty level	E.Time (4.2)	Queue 1	Time Q1	Queue 2	Time Q2	Distance	Empty level	All queue (4.3)	Time all (4.4)
Image (1)	4	0	0	0	4	2	10	4	0	0	0	4	3	11	3	6	3	6	0	2	5	12
Image (2)	3	1	0	1	3	0	6	3	1	0	1	3	0	6	3	6	2	4	0	0	3	6
Image (3)	4	0	3	0	4	0	11	4	0	3	0	4	0	11	3	6	3	6	3	0	3	9
Image (4)	4	0	0	0	4	3	11	4	0	0	0	4	2	10	4	8	6	12	0	0	7	14
Image (5)	4	0	0	1	3	4	10	3	1	0	1	3	5	11	6	12	8	16	0	0	8	16
Image (6)	2	0	1	1	1	0	3	2	0	0	1	1	0	2	1	2	1	2	0	0	1	2
Image (7)	3	0	1	1	2	1	6	3	0	0	1	2	0	4	1	2	3	6	0	0	3	6
Image (8)	3	0	0	1	2	6	10	3	0	0	1	2	7	11	8	16	1	2	0	0	8	16
Image (9)	4	0	0	0	4	2	10	4	0	0	0	4	0	8	4	8	7	14	0	0	7	14
Image (10)	2	0	0	0	2	1	5	2	0	0	0	2	0	4	1	2	1	2	0	1	2	5
Image (11)	4	0	0	1	3	5	11	4	0	0	1	3	6	12	8	16	1	2	0	0	8	16
Image (12)	3	0	0	1	2	6	10	3	0	0	1	2	7	11	8	16	1	2	0	0	8	16

Note: Although number of vehicles in both actual and counting vehicles are equal, but the estimated time is deferent, this is due in actual the number of gaps between cars is calculated depending on number of empty levels, while in counting vehicle, we find to the distance between contiguous cars and divided by assume car size for that the estimated time will vary. For **Example**: In photo No. 1, the number of vehicles in the actual is 4 and the estimated time is 10 sec, but although in vehicle counting number of vehicles is the same (4) but the estimated time is 11 since number of empty levels in the actual is 2, while in the counting-vehicles is 3. This gives the time in the actual 10 and segmentation 11.

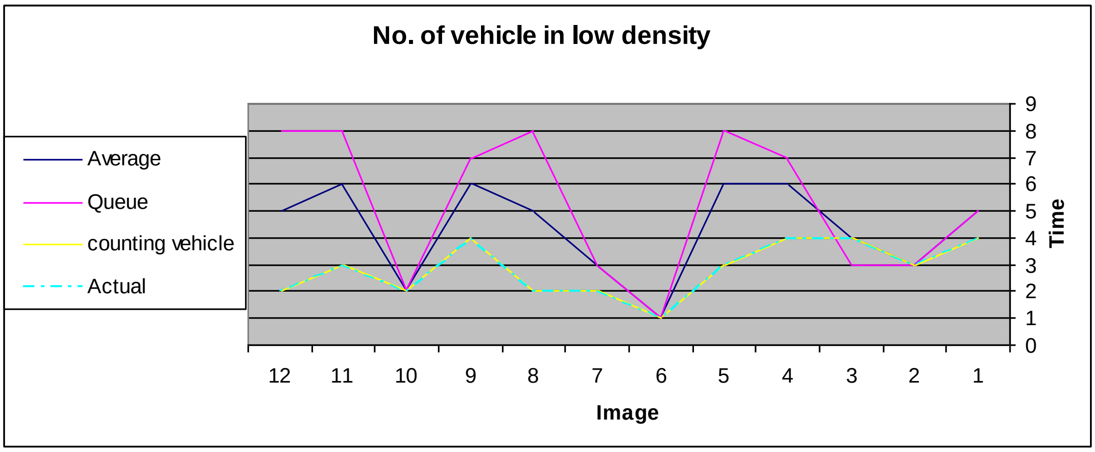


Chart 4.1 Number of Vehicle in Low Density

Chart 4.1 represents the number of vehicles in images with low densities. Where we note here that the number of vehicles detected in the counting vehicle is almost the same of number in the actual, while in queue, number of vehicles is relatively larger because the queue count vehicles based on the deference between first and last vehicle divided by vehicle size, regardless if the vehicles exist or not. Thus, the calculation time on this basis lead to increase in error rate. The following Chart shows the estimated time for low density image for the two suggested methods with their average compared with the actual time needed. As shown in Chart 4.2, it is clearly seen that vehicle counting is more suitable to be used with low density than queuing.

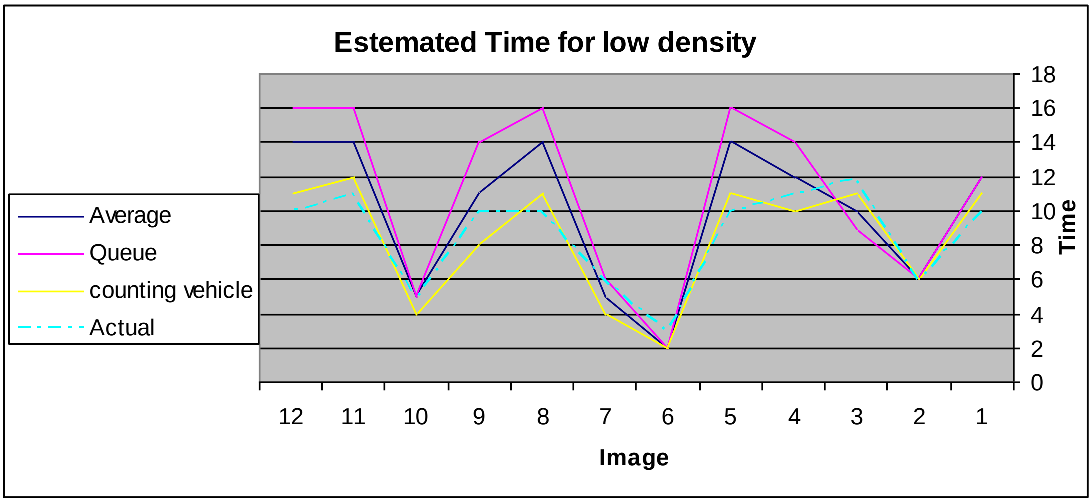


Chart 4.2 Estimated Time in Low Density Pictures

Table 4.2 shows the results of the comparison points (Number of vehicle, Estimated time, Empty level, Cars beside other, Distance, number of Big-cars) for actual, vehicle counting, and queuing system for pictures that represents high-density pictures.

Table 4.2 High Density

	Actual							Segmentation							Queue							
	No. of vehicle	Big car	Distance	Beside other	No of vehicle (4.1)	Empty level	E.Time (4.2)	No. of vehicle	Big car	Distance	Beside other	No of vehicle (4.1)	Empty level	E.Time (4.2)	Queue 1	Time Q1	Queue 2	Time Q2	Distance	Empty level	All queue (4.3)	Time all (4.4)
Image (1)	10	3	0	5	8	0	16	10	3	0	5	8	0	16	7	14	8	16	0	0	8	16
Image (2)	11	0	0	3	8	0	16	11	0	0	2	9	0	18	8	16	8	16	0	0	8	16
Image (3)	8	1	0	2	7	0	14	8	1	0	1	8	0	16	8	16	6	12	0	0	8	16
Image (4)	8	0	0	2	6	0	12	8	0	0	2	6	0	12	8	16	7	14	0	0	8	16
Image (5)	8	0	0	4	4	0	8	8	0	0	3	5	0	10	7	14	7	14	0	0	7	14
Image (6)	17	1	0	9	8	0	16	15	3	0	8	10	0	20	8	16	8	16	0	0	8	16

Image (7)	9	4	0	2	9	0	22	9	3	0	3	9	0	18	7	14	8	16	0	0	8	16
Image (8)	9	0	0	4	5	0	10	9	0	0	4	5	0	10	7	14	5	10	0	0	7	14
Image (9)	11	1	0	3	8	0	17	6	5	1	2	9	0	19	6	12	3	6	1	0	6	13
Image (10)	8	2	0	4	6	0	12	8	3	0	4	7	0	14	7	14	6	12	0	0	7	14
Image (11)	15	1	0	7	9	0	18	12	3	0	4	11	0	22	8	16	8	16	0	0	8	16
Image (12)	8	2	1	3	7	0	15	6	5	1	2	9	0	19	5	10	6	12	1	0	6	13

From Charts (4.3 and 4.4), it is clearly seen that is queuing technique behaves better than vehicle counting for that it is more suitable to be used with high-density situations.

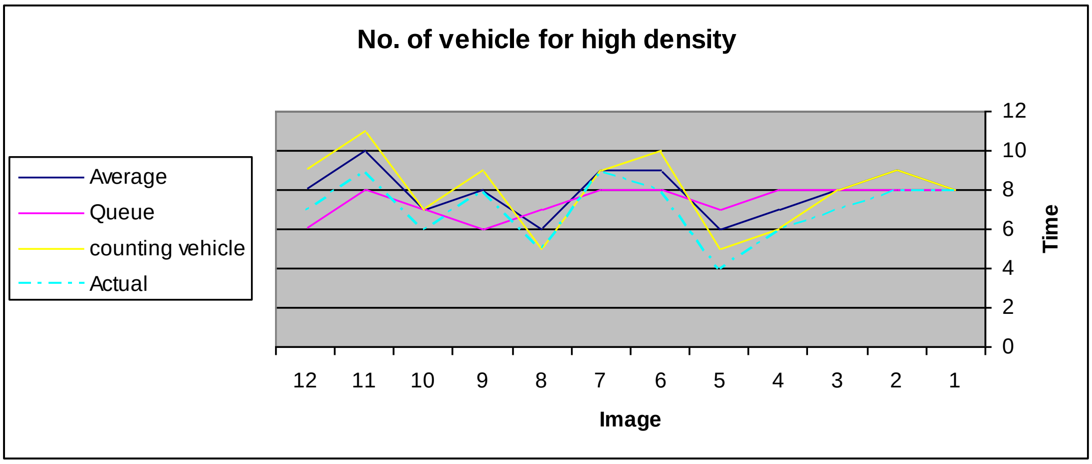


Chart 4.3 Number of Vehicle in High Density

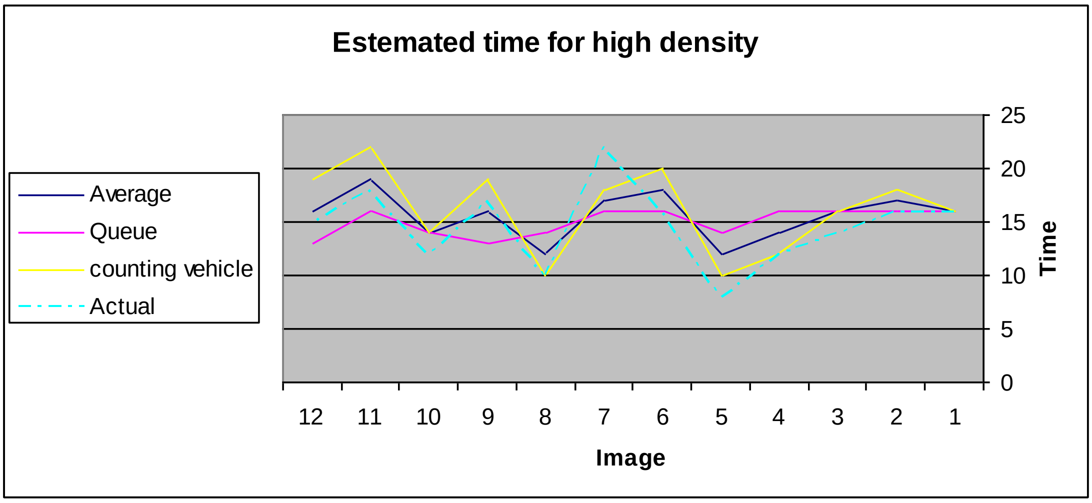


Chart 4.4 Estimated Time For High Density

Table 4.3 shows the results of the comparison points (Number of vehicle, Estimated time) between vehicle counting, and queuing system with respect to actual. It also shows the mean square error with respect to actual for all type of pictures high and low density pictures.

Table 4.3 All Pictures With Mean Square Error (**MSE**).

	Actual		**Segment**		**Queue**		**Average**	
No of image	# objects	Time	#objects	Time	#object	Time	# object	Time
Image (1)	4	10	4	11	5	12	5	12
Image (2)	3	6	3	6	3	6	3	6
Image (3)	4	11	4	11	3	9	4	10
Image (4)	4	11	4	10	7	14	6	12
Image (5)	3	10	3	11	8	16	6	14
Image (6)	1	3	1	2	1	2	1	2
Image (7)	2	6	2	4	3	6	3	5
Image (8)	2	10	2	11	8	16	5	14
Image (9)	4	10	4	8	7	14	6	11
Image (10)	2	5	2	4	2	5	2	5

Image (11)	3	11	3	12	8	16	6	14
Image (12)	2	10	2	11	8	16	5	14
Image (1)	8	16	8	16	8	16	8	16
Image (2)	8	16	9	18	8	16	9	17
Image (3)	7	14	8	16	8	16	8	16
Image (4)	6	12	6	12	8	16	7	14
Image (5)	4	8	5	10	7	14	6	12
Image (6)	8	16	10	20	8	16	9	18
Image (7)	9	22	9	18	8	16	9	17
Image (8)	5	10	5	10	7	14	6	12
Image (9)	8	17	9	19	6	13	8	16
Image (10)	6	12	7	14	7	14	7	14
Image (11)	9	18	11	22	8	16	10	19
Image (12)	7	15	9	19	6	13	8	16
MSE (ACTUAL & AVERAGE) time			MSE (ACTUAL & SEGMENT)		MSE (ACTUAL & QUEUE)		MSE (ACTUAL & AVERAGE)	
			10.000		**17..4069**		**11.4455**	
MSE (ACTUAL & AVERAGE) object			MSE (ACTUAL & SEGMENT)		MSE (ACTUAL & QUEUE)		MSE (ACTUAL & AVERAGE)	
			4.1231		**13**		**7.6158**	

Charts (4.5 and 4.6) and the MSE measurements for both techniques shows that counting vehicles shows better estimation of time, with respect to actual, than queuing system.

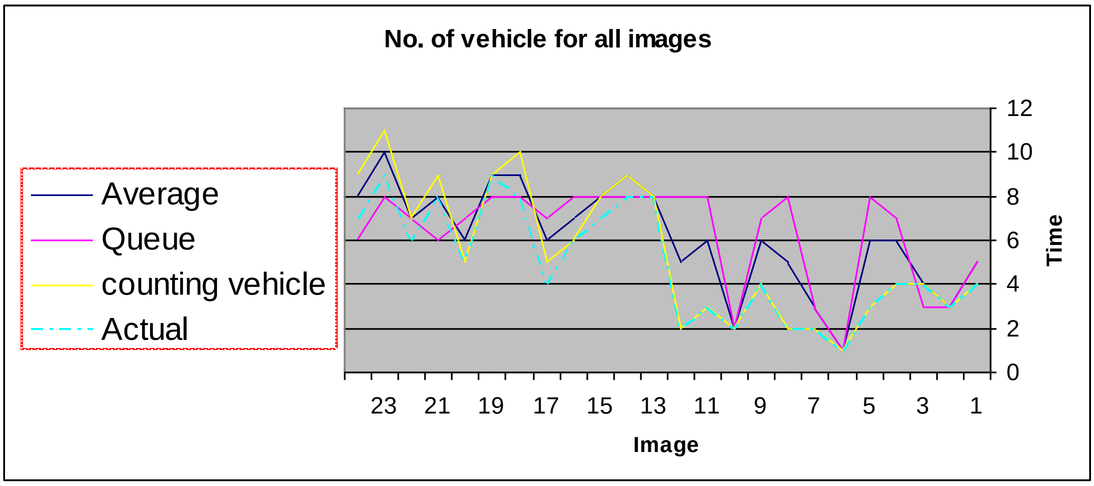


Chart 4.5 Number of Vehicles For All Pictures

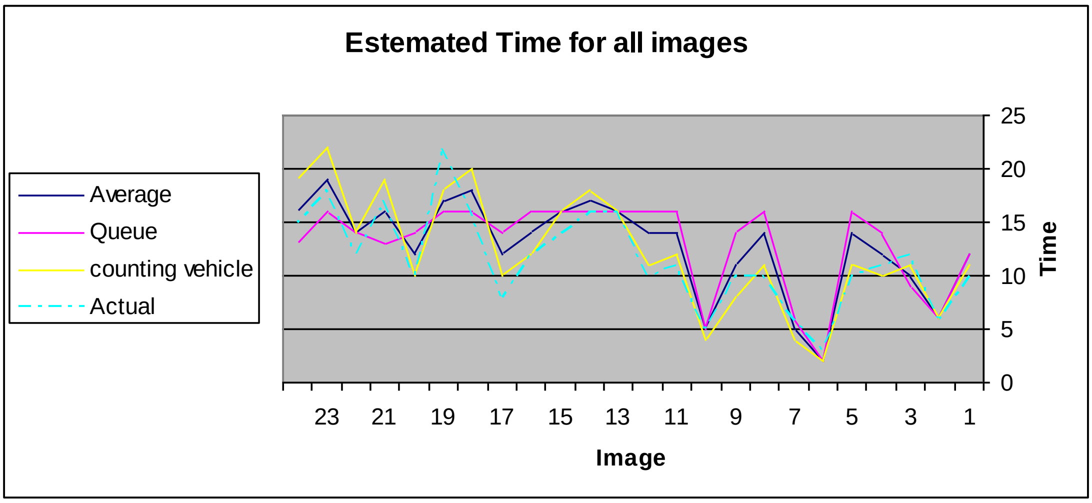


Chart 4.6 Estimated Time For All Pictures

Finally, from execution time point of view, the time needed to calculate the estimated time for a particular picture is less than the time needed by queuing system.

Chapter Five

Conclusions

And

Future Works

5.1 Conclusions

From the experimental results illustrated in the tables and Chart of chapter four, one can deduce:

1. To convert images to binary form, it was found that (by trail and error) the best threshold value is 20 since it gives the best result among 15 and 25 and solves shadows problem.

2. From time execution point of view, the segmentation approach is better since: the queue length approach needs approximately 11.0849 second (8.936 sec for preprocessing and 2.438 sec for queue algorithm), while counting vehicle approach needs approximately 9.174 seconds (8.936 sec for preprocessing and 0.238 sec for segmentation algorithm). The average approximately needs 11.612 second (8.936 sec for preprocessing and 2.438 for queue length and 0.238 for segmentation).

3. From Estimated-time and Number-of-vehicles point of view: the queue length is better for high density situations and counting vehicles is better for low density since the queuing will assume the space between vehicles also cars.

4. Counting vehicles is recommended to be used since it gives best approximation results (number of vehicle, estimated time, and execution time) compared with actual results for all type of images (see minimum MSE with respect to actual situation).

5.2 Future Works

1. Shadow removal during day light.
2. The Weather and the Night Technology development to be able to work through the night a valuable distinction of light emitted from the vehicle Finder, and find solutions for the problem of lighting and to lesson its effects on the technique ,and Technology development to be able to work through weather changeability.
3. Connecting with satellites: Connectivity with satellite images are saluting the more vertical accuracy of the photos horizontal.
4. Extract object without using background image.
5. Develop the counting number of vehicles so it could work on more that one cross point on road (i.e. manage all traffic lights at a cross point).

References

1. [ALV 96] Alvaro Soto and Aldo Cipriano , "Image processing applied to real time measurement of traffic flow" System Theory, Proceedings of the Twenty-Eighth Southeastern Symposium on ,Vol 2 , Issue , Pp:312 – 316,(1996).

2. [ATK 05] E. Atkoči ūnas, R. Blake, A. Juozapavičius and M. Kazimianec, “Image processing in road traffic analysis”, Nonlinear Analysis: Modeling and Control, Vol. 10, No. 4, pp. 315 - 332.(2005) .

3. [BEY 97] Beymer, D., McLauchlan, P., Coifman, B., and Malik, J. "A Real-Time Computer Vision System for Measuring Traffic Parameters", Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Juan, Puerto Rico, Pp 495-501. (1997).

4. [CHA 96] Chachich, A, Pau, A, Barber, A, Kennedy, K, Olejniczak, E, Hackney, J, Sun, Q, Mireles, E, "Traffic Sensor Using a Color Vision Method", Transportation Sensors and Controls: Collision Avoidance, Traffic Management, and ITS, SPIE Proc. Vol 2902, Pp 156-165. (1996).

5. [CON 96] Condos, F "Traffic Surveillance and Detection Systems for Intelligent Transportation Systems Applications", Proc. ITE Annual Conference, Minneapolis, MN, Pp 354-356. (1996).

6. [DAL 03] Dalaff, C.; Ruhé, M.; Reulke, R.; Schischmanow, A.; Schlotzhauer, G:, "OIS - An optical information system for road traffic measurement and management", Joint Workshop of ISPRS Working Groups IV/3, IV/6 and IV/7, 8. - 9, Germany. (2003).

7. [FAT 95] M. Fathy and M.Y. Siyal,"Real-time image processing approach to measure traffic queue parameters ", IEE Proceedings - Vision, Image, and Signal Processing -- Vol 142, Issue 5, Pp. 297-303,(1995).

8. [FOR 03] D.A. Forsyth, J. Ponce. "Computer Vision. A Modern Approach", Prentice Hall, (2003).

9. [GIU 02] Giusto, D.D., Massidda, F., Perra, C.,"A fast algorithm for video segmentation and object tracking", Digital signal Processing, 14th International Conference on , Vol: 2 , Pp:697 – 700. (2002)

10. [GON 08] Gonzalez, R. C. and Woods, R. E. "Digital image processing" . New Jersey : Prentice Hall. (2008).

11. [GRE 05] Greta Cunningham,"Some traffic problems caused by traffic lights",Minnesota Public Radio , (2005). http://news.minnesota.publicradio.org/features/2005/04/22_cunninghamg_traffic/

12. [GUT 04] Guthe, S., & Strasser, W. "Advanced techniques for high-quality multi-resolution volume rendering. Graphics and Visualization", Vol 28, Pp 25-78.(2004).

13. [HAR 00] Haritaoglu, I., Harwood, D., Davis, L.S., "W4: real-time surveillance of people and their activities", Pattern Analysis and Machine Intelligence, IEEE Transactions on , Vol: 22 , Issue: 8 , Pp:809 – 830.(2000)

14. [HOW 05] Howard Anton, IrI Bivens, Stephen Davis ,"Calculus" Eighth Edetion,WELEY, ISBN, 0-471-48273-0.(2005).

15. [HSU 03] *Hsu, W.L.; Liao, H.Y.M.; Jeng, B.S.; Fan,* "Real-time vehicle tracking on highway" The Proceedings of the IEEE International Conference on Intelligent Transportation Systems, Pp:909- 914 vol.2, 12-15 . (2003).

16. [HUD 96] Huddart, K. W.(chair). " Eighth international conference on road traffic monitoring and control", London: IEE Conference Publication, no 422. (1996).

17. [INT 09] Internet : http://www.freefoto.com/index.jsp .

18. [JPL 97] JPL,"Traffic Surveillance and Detection Technology Development", Sensor Development Final Report, Jet Propulsion Laboratory Publication No. 97-10. (1997).

19. [KAR 90] Karmann, K, Brandt, A, "Moving object recognition using an adaptive background memory", Time-Varying Image Processing and Moving Object Recognition, 2, V Cappellini, editor, Elsevier, Amsterdam, The Netherlands, Vol 2,Pp 289-296, (1990).

20. [KIL 92] Kilger, M "A shadow handler in a video-based real-time traffic monitoring system", IEEE Workshop on Applications of Computer Vision, Palm Springs, CA, Pp 1060-1066. (1992).

21. [KOB 07] Kobus Barnard, "image_analysis",(2007), http://vision.cs.arizona.edu/nvs/research/image_analysis/edge.html

22. [LAW 05] Lawrence Y. Deng, Nick C. Tang, Dong-liang Lee, Chin Thin Wang and Ming Chih Lu,"Vision Based Adaptive Traffic Signal Control System development", Proceedings of the 19th International Conference on Advanced Information Networking and Applications (AINA’05) 1550-445X/05© IEEE.(2005).

23. [LIP 98] A. Lipton, H. Fujiyoshi, and R. Patil, “Moving target classification and tracking from real-time video,” in Proc. 4th IEEE Workshop on Applications of Computer Vision (WACV ’98) , Pp. 8–14,(1998).

24. [MAR 03] Markos Papageoggius, Christina Diakaki, Vaya Dinopoulou, Apoatolos Kotsialos, and Yibing Wang, “Review of Road Traffic Control Strategies”, PROCEEDINGS OF THE IEEE, Vol. 91, No. 12, (2003).

25. [MAR 97] Markus Ebbecke, Majdi Ben Hadj Ali, Andreas Dengel,"Real Time Object Detection, Tracking and Classification in Monocular Image Sequences of Road Traffic Scenes", Proceedings of the International Conference on Image Processing (ICIP '97), (1997)

26. [MAS 93] Masakatsu Higashikubo, Toshio Hinenoya and Kouhei Takeuchi. "Traffic Queue Length Measurement Using an Image Processing". Available: http:// www.utms.or.jp/japanese/inter/paper/paper02.pdf.

27. [MAS 98] Masaki I., “Machine-vision systems for intelligent transportation systems,” IEEE Intell. Syst., Vol. 13, no. 6, Pp. 24–31, (1998).

28. [MAT 02] J. Matthews. “An introduction to edge detection: The sobel edge detector,” Available at http://www.generation5.org/content/2002/im01.asp, (2002).

29. [MAT 09] Mathworks website : http:// www.mathworks.com/

30. [MIC 03] Michele Zanin, Stefano Messelodi, "An efficient vehicle queue detection system based on image processing".; Modena, C.M. Image Analysis and Processing,Proceedings. 12th International Conference ,Vol 2, Issue , 17-19 .Pp: 232 – 237.(2003).

31. [MIC 91] P. Michalopoulos, “Vehicle detection video through image processing: The Autoscope system,” IEEE Trans. Veh. Technol., Vol. 40, Pp. 21–29,(1991).

32. [MUR 99] Y. L. Murphey, H. Lu, S. Lakshmanan, R. Karlsen, G. Gerhart, and T. Meitzler, “Dyta: An intelligent system for moving target detection,” in Proc. 10thInt. Conf. Image Analysis and Processing,Venice, Italy, Pp. 1116–1121, (1999).

33. [REB 09] Rebecca Vincent and Dr. Olusegun Folorunso,"A Descriptive Algorithm for Sobel Image Edge Detection", Proceedings of Informing Science & IT Education Conference (InSITE) .(2009).

34. [REU 02] R. Reulke, A. Boerner, H. Hetzheim, A. Schischmanow, and H. Venus." A sensor web for roadtraffic observation. In Image and Vision Computing", NewZealand , Pp: 293–298, (2002).

35. [RIC 08] Richard Lipka, Pavel Herout, "Implementation of Traffic Lights in JUTS," uksim, pp. 422-427, Tenth International Conference on Computer Modelingand Simulation (uksim 2008), (2008).

36. [RIT 00] Rita Cucchiara, Member, IEEE, Massimo Piccardi, Member, IEEE, and Paola Mello,"Image Analysis and Rule-Based Reasoning for a Traffic Monitoring System",IEEE transactions on intelligent transportation systems, Vol. 1, No. 2, (2000).

37. [SEV 02] Seventh International Conference on Control, Automation, Robotics and Vision(ICARCV’02), ." Chin Contour Estimation Using Modified Canny Edge Detector".singapor, (2002).

38. [SHI 01] M.C. Shin, D. Goldgof, and K.W. Bowyer .“Comparison of Edge Detector Performance through Use in an Object Recognition Task”. Computer Vision and Image Understanding, Vol. 84, No. 1, Pp. 160-178, (2001).

39. [SHI 01] M.C. Shin, D. Goldgof, and K.W. Bowyer ."Comparison of Edge Detector Performance through Use in an Object Recognition Task". Computer Vision and Image Understanding, Vol. 84, No. 1, Pp. 160-178, (2001).

40. [SIE 97] Siemens AG, "MOTION—Traffic-actuated signal control for traffic computer system (Special Edition of Blickpunkt ANL)," report, (1997).

41. [SIV 05] J. Sivic, B. C. Russell, A. Efros, A. Zisserman, and W. T. Freeman. "Discovering object categories in image collections". MIT AI Lab Memo AIM 2005-005, MIT, (2005).

42. [SUL 92] Sullivan, G, "Visual interpretation of known objects in constrained scenes.", Phil. Trans. Roy. Soc (B), 337:Pp 361-370, (1992).

43. [VIO 01] P. Viola and M. Jones. "Rapid object detection using a boosted cascade of simple features". In Proc. CVPR, (2001).

44. [WIX 98] L. Wixson, K. Hanna, and D. Mishra, "Improved illumination assessment for vision-based traffic monitoring," in 1998 IEEE Workshop on Visual Surveillance, Pp. 34–41, (1998).

45. [YUK 95] Yukawa, S. & Kikuchi, M.. "Coupled–map modeling of one dimensional traffic flow". Journal of the Phyical Society of Japan. Vol 64, Pp.35–38, (1995).

46. [ZEN 96] Zeng, N, Crisman, J, "Evaluation of Color Categorization for Representing Vehicle Colors", Transportation Sensors and Controls: Collision Avoidance, Traffic Management, and ITS, SPIE Proc. Vol 2902, Pp.148-155. (1996).

47. [ZHA 98] Zhang, Zhengyou: "Determining the epipolar geometry and its uncertainty": A Review. International Journal of Computer Vision, 27(2), Boston: Kluwer Academic Publishers, Pp. 161-198, (1998).

48. [ZHU 96] Z. Zhu, B. Yang, G. Xu, and D. Shi, "A real-time vision system for automatic traffic monitoring based on 2D spatio-temporal images," in Proc. 3rd IEEE Workshop on Applications of Computer Vision (WACV '96), Pp. 162-167. (1998).

Appendix (A)

Canny edge detection algorithm:

The Canny Edge Detection Algorithm has the following steps:

· Smooth the image with a Gaussian filter: Let $I[i, j]$ denote the image; $G[i, j; \sigma]$ be a Gaussian smoothing filter where σ is the spread of the Gaussian and controls the degree of smoothing. The result of convolution of $I[i, j]$ with $G[i, j; \sigma]$ gives an array of smoothed data as:

$$S[i, j] = G[i, j; \sigma] * I[i, j]$$

· Compute the gradient magnitude and orientation using finite-difference approximations for the partial derivatives. Firstly, the gradient of the smoothed array $S[i, j]$ is used to produce the x and y partial derivatives $P[i, j]$ and $Q[i, j]$ respectively as:

$$P[i,j] \approx (S[i,j+1] - S[i,j] + S[i+1,j+1] - S[i+1,j])/2$$
$$Q[i,j] \approx (S[i,j] - S[i+1,j] + S[i,j+1] - S[i+1,j+1])/2$$

The x and y partial derivatives are computed with averaging the finite differences over the 2x2 square. From the standard formulas for rectangular-to-polar conversion, the magnitude and orientation of the gradient can be computed as:

$$M[i,j] = \sqrt{P[i,j]^2 + Q[i,j]^2}$$
$$\theta[i,j] = \arctan(Q[i,j], P[i,j])$$

Here the arctan(x,y) function takes two arguments and generates an angle.

· Apply non-maxima suppression to the gradient magnitude, Given the being the magnitude image array one can apply the thresholding operation in the gradient-based method and end up with ridges of edge pixel. But canny has a more sophisticated approach to the problem. In this approach an edge point is defined to be a point whose strength is locally maximum in the direction of the gradient. This is a stronger constraint to satisfy and is used to thin the ridges found by thresholding. This process, which results in one pixel wide ridges, is called "Nonmaxima Suppression". After nonmaxima suppression one ends up with an image $N[i, j] = nms(M[i, j], \zeta [i, j]$ which is zero everywhere except the local maxima points. At the local maxima points the value of is preserved.

· Use the double thresholding algorithm to detect and link edges: In spite of the smoothing performed as the first step in edge detection, the nonmaxima suppressed magnitude image $N[i, j]$ will contain many false edge fragments caused by noise and fine texture. The contrast of the false edge fragments is small. These false edge fragments in the nonmaxima-suppressed gradient magnitude should be reduced. One typical procedure is to apply a threshold to $N[i, j]$. All values below the threshold are set to zero. After the application of threshold to the nonmaxima-suppressed magnitude, an array E(i,j) containing the edges detected in the image $I[i, j]$ is obtained. However; in this method applying the proper threshold value is difficult and involves trial and error. Because of this difficulty, in the array E(i,j) there may still be some false edges if the threshold is too low or some edges may be missing if the threshold is too high. A more effective thresholding scheme uses two thresholds. To overcome the problem, two threshold values, τ_1 and τ_2 are applied to $N[i, j]$. Here $\tau_2 \approx 2\tau_1$. With these threshold values, two thresholded edge images $T_1[i, j]$ and $T_2[i, j]$ are produced. The image T_2 has gaps in the contours but contains fewer false edges. With the double thresholding algorithm the edges in T_2 are linked into contours. When it reaches the end of a contour, algorithm looks in T_1 at the locations of the 8-neighbours for edges that can be linked to the contour. This algorithm continues until the gap has been bridged to an edge in T_2. The algorithm performs edge linking as a by-product of thresholding and resolves some of the problems with choosing a threshold.

" التحكم بإشارةِ المرور من خلال اَستعمالُ طولَ طابورِ واكتشاف الأجسام"

إعداد الطالب

عُباده "محمد عدنان" بني عيسى

إشراف الدكتورة

فينوس سماوي

الخُلاصة

أن الأنشطة الاجتماعية والاقتصادية تنمو بأشكال مختلفة في هذه الأيام ، وبالتالي يصبح من الضروري ومن الملح وجود نظام مراقبة مرور متقدم يواكب تغيرات ظروف حركة المرور على الطرق.

نقترح في هذا البحث تقنيتين تستند كل منهما على صور أَخذتْ بآلة تصوير رقمية وَضعتْ في موقع ثابت في إشارة المرور بزاوية ومسافة معيّنة.

الصور الناتجة سوف يجرى عليها عدة عمليات (طرح الصورة من الخلفية، إزالة الضوضاء والأجسام الصغيرة) ثمّ نستعمل التقنيتين طول الطابور (يكتشف طول الطابور السيارات الموجودة في الطريق بالاعتماد على المسافة.

بـين النقطتـين النهـائيتين علـى خَطّـي الطريـق)، وتقنيـة تحديـد عـدد المركبـات. بالاعتماد علـى تجزئـه الصـورة ومن ثم عدّ القطاعات التـي تشبه فـي حجمها حجم المركبة الطبيعي).

لقياس طـول الطـابور أو عدد من المركبات ، لكشف الحواف تم استخدام طريقـة السـوبول (sobel) وفـي تجزئـة الصـورة (يعتمـد علـى أسـاس region growing)

وأخيرا استخدمنا معادلة لحساب الوقت التقريبي والوقت الفعلي لتحديد الوقت الـلازم لفتح للإشـارة الضـوئية الخضـراء. لقد تمت مقارنـة التقنيتـان المقترحتـان من وجهتـين مختلفتـين الأولـى وهـي الوقـت الـلازم لإعطـاء القرار للإشـارة الضـوئية أي الوقت اللازم التنفيذي للبرنامج. والثانية وهي الوقت المقدر من خلال التقنيتين الذي تحتاجه الإشارة الخضراء بالمقارنة مع الوقت الحقيقي المناسب لفتح الإشارة .

ونتيجـة لذلك وجد الباحـث أن استخدام تجزئـة الصـورة أفضـل فـي الاسـتخدام لأنه كان أفضل في إيجاد الوقت المناسب حيث كان اقرب إلى الوقت الفعلي . وكما انـه تم دراسـة ذلـك علـى أنـواع الصـور (ذات الكثافـة العاليـة و ذات الكثافـة القليلـة) حيث تبـين أن اسـتخدام تقنيـة طـول الطـابور مناسـبة فـي حالـة الصـور ذات الكثافـة العالية ولكن تجزئة الصور تناسب كلا النوعين .