IMPLEMENTATION OF CAR COUNTING SYSTEM USING MORPHOLOGICAL IMAGE PROCESSING

HNIN WIT YI OO

M.C.Tech. **JANUARY, 2019**

IMPLEMENTATION OF CAR COUNTING SYSTEM USING MORPHOLOGICAL IMAGE PROCESSING

By

HNIN WIT YI OO
B.C.Tech.(Hons:)

A dissertation submitted in partial fulfillment of the requirements for the degree of

Master of Computer Technology
(M.C.Tech.)

Universityof Computer Studies,Yangon

JANUARY, 2019

ACKNOWLEDGEMENTS

First of all, I wish to express my deep gratitude and sincere appreciation to all persons who contributed towards the success of the thesis.

I wish to express my sincere appreciation to **Dr. Mie Mie Thet Thwin**, the Rector of the University of Computer Studies, Yangon, for allowing me to develop this thesis.

I would like to express my thanks to **Dr. Khin Than Mya**, Professor and Head of Faculty of Computer Systems and Technologies, University of Computer Studies, Yangon, for helpful suggestions and administrative support.

Especially, I am most grateful to my supervisor, **Dr. Zin May Aye**, Professor of Cisco Network Lab and Faculty of Computer Systems and Technologies, University of Computer Studies, Yangon, for her fully guidance, patient supervision and technical assistance throughout the period of finishing thesis.

I would also like to acknowledge **Daw Khine Yin Mon**, Lecturer, Department of Language, University of Computer Studies, Yangon, for editing my thesis from language point of view.

I also thank to my teacher, **Dr. Aung Htein Maw**, Professor at University of Information Technology, Yangon, an external teacher of this thesis committee and I am gratefully indebted to his for his precious comments on this thesis.

I would also like to show my gratitude to my teacher, **Dr. Tweta Oo**, Lecturer and Dean of the Master Course of the University of Computer Studies, Yangon, for her suggestion and help in this work.

I must express my profound gratitude to **my family** and **all my teachers** who taught me throughout the master's degree course, for continually pushing me forward in my pursuits.

Finally, I also thank **my fellow classmates** who have been with me over the past few years, their support and assistance has been key in helping me.

ABSTRACT

Due to the increasing urban population and hence the number of cars has increased enormously and need of controlling the traffic on roads is vital. The resulting wastage of time and increase in pollution can be eliminated on a city-wide scale by the following system. Automatic detecting and counting vehicles in video on roads is a very challenging problem in computer vision with important practical applications such as to monitor activities at traffic and then predict the traffic flow which assists in regulating traffic. In this thesis, the system is developed for counting the number of cars running on the road to analyze traffic in any region. The system applied the video file that captured the cars running on roads with the help of a stationary camera. Detection of the moving object and removal of the noise is achieved by some morphological operations and the result shows the number of cars on roads is counted. This system uses MATLAB Program for the implementation.

CONTENTS

LIST OF FIGUREDS

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1. Introduction of the System

In recent year, as the result of the increase in vehicle traffic, many problems have appeared. For example, traffic accidents, traffic congestion, traffic induced air pollution and so on. Traffic congestion has been a significantly challenging problem. It has widely been realized that increases of preliminary transportation infrastructure e.g., more pavements, and widened road, have not been able to relieve city congestion. As a result, many transportation investigators have paid their attentions intelligent transportation system (ITS), such as predict the traffic flow on the basis of monitoring the activities at traffic intersections for detecting congestions. To better understand traffic flow, an increasing reliance on traffic is in a need for better vehicle detection at a wide area. Automatic detecting vehicles in video data are a very challenging problem in computer vision with important practical applications such as traffic analysis and security.

Vehicle detection and counting is important in computing traffic congestion on roads. This thesis mainly aims to develop methodology for automatic detection and its counting on the roads. Traffic management and information systems depend mainly on sensors for estimating the traffic parameters. In addition to vehicle counts, a much larger set of traffic parameters like vehicle classifications, lane changes, etc., can be computed. Vehicle detection and counting uses a single camera mounted usually on a pole or other tall structure, looking down on the traffic scene. Current approaches of monitoring traffic include manual counting of vehicles, or counting vehicles using magnetic loops on the roads. The main drawback of these approaches, besides the fact that they are expensive, is that these systems only count.

In this thesis, a system for detecting and counting vehicles in captured video from a stationary camera which uses segmentation with background subtraction and morphological operator to determine the foreground objects in a sequence of video frames. Detected objects will be counting which shows the areas depends on the size of the images and counting the vehicles in the domain of traffic monitoring over roads.

1.2.Objectives of the Thesis

The objectives of the thesis are:

- To detect multiple moving cars in a video sequence
- To review some morphological filters which are widely used in image processing
- To analyze mainly morphology and detection methods in the car counting system
- To implement the methodology for traffic analysis in counting number of cars on the roads from the offline captured video files

1.3. Motivation of the System

Vehicle detection and counting is important in computing traffic congestion and to keep track of vehicles that use state-aid streets and highways. Even in large metropolitan areas, there is a need for data about vehicles that use a particular street. A system like the one proposed here can provide important data for a particular design scenario. Magnetic loop detectors are currently used to count vehicles which pass over them, but vision-based video monitoring systems offer many more advantages.

Current approaches of monitoring traffic include manual counting of vehicles, or counting vehicles using magnetic loops on the road. The main drawback of these approaches, besides the fact that they are expensive, is that these systems only count. This system is implemented on differencing method, background subtraction and segmentation method in order to benefit vehicle detection and counting from video sequence to tracking and counting vehicles for high quality videos. Video analysis provides quick practical information resulting in increased safety and traffic flow. For example, objects are defined as vehicles moving on roads. Moreover, cameras are much less disruptive to install than loop detectors.

1.4 Organization of Thesis

In this chapter, the introduction, objectives and organization of the thesis are presented.

The Chapter 2 presents about background theory and literature review about morphological image processing techniques.

In Chapter 3, Car Counting Approach, a theoretical framework of segmentation to count the car objects in counting approach. This chapter also describes the image detection techniques.

The system flow diagram of this system is the former part of Chapter 4. The rest of this chapter is the implementation of car counting approach and experimental results of the system.

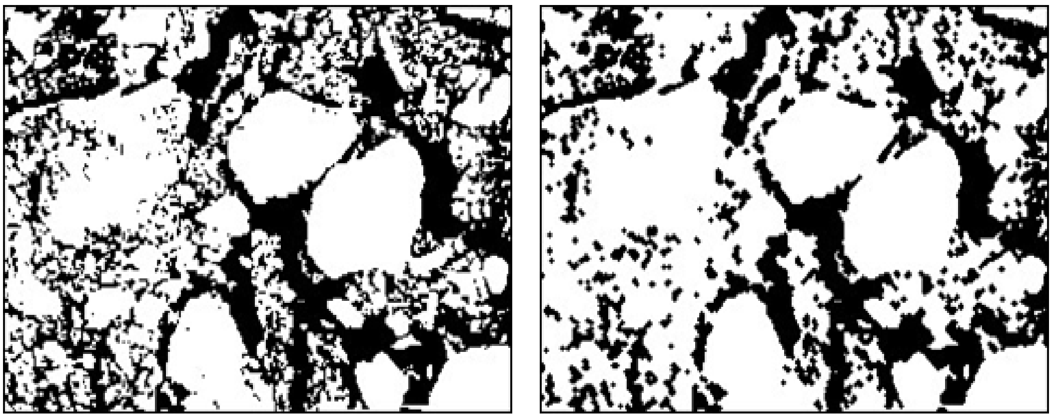
This thesis concludes with Chapter 5 which presents the conclusion, the advantages, limitations of the thesis and the further extension this system.

CHAPTER 2
BACKGROUND THEORY OF MATHEMATICAL MORPHOLOGY

2.1. Mathematical Morphology

In analysis of image processing, morphology which is based on the image structures that may be handled by set theory which offers a number of useful tools. This approach has become popular in recent years. This serves as a self-contained approach to handle images in this thesis. The basic procedures are accessible in software packages of the image analysis software packages and can be used to efficient parallel hardware implementations. Figure 2.1 express an operation based on morphology as an example. Figure 2.1(a) shows thresholded subset image of soil with pixel value 20. Figure 2.1(b) is the morphological closing operation result. Initially, it is seem to be adjust another filter that has smoothed the image, then the closing operation is often employed for this purpose.

This chapter expands the morphological ideas mainly for binary images. There are many applications which use gray scale threshold, whereas noise is not a dominant feature. On basis of morphological operations, morphological filters assembled for shape analysis are more suitable than the standard linear filters because the second distort the underneath geometric form of the image for a time.



(a) (b)

Figure 2.1 (a) soil image subset (b) closing operation result of (a) using approximation a disc of radius 1.0

There are some of the salient points regarding the morphological approach are as follows [24]:

1. As the same time as to maintain the important geometric characteristics' stability, the operations provide for the geometric content of an image systematic alteration.
2. A fine build-up morphological algebra exists and can be taken up for characterization and optimization.
3. It is likely to show digital algorithms by means of tiny class of primal morphological operations.
4. There are strict represented theorems via one can get the morphological filters expression in terms of the primitive morphological operations.

The operators change the original into another image through the structuring element. The shape and size of the image that are similar to the structuring elements preserved, during the other features are concealed. And so, operations describe the data of image, preserved with shape, characteristic and remove irrelevances. For many applications, the morphological operations can be for many purposes [25], including [27], including edge detection [27], segmentation [14], [17], [16], enhancement [29] of the image and so on.

Defining of morphology on the setting of Euclidean is Euclidean morphology and on the setting of digital is named digital morphology. In general, the relationship between the above twos is similar to the relationship amid continuous signal processing and digital signal processing. In spite of being the actual implementation of morphological operators in the digital setting, the model of the Euclidean is necessary to the development of an understanding and perception of the operators function in theory and application.

2.2. Binary Morphological Transformation

The basic theory of binary mathematical morphology is "set theory". Morphology offers a combined and influential approach to various image processing problems. In binary images, the set of all white pixels is a complete morphological description of the image and the members of the 2-D integer Z^2whose coordinates are (x, y) coordinates of black and white, depending on convention pixel in the image. In addition, the basic concepts of the set reflection and translation are widely used in morphology. The reflection of set B is defined as:

$$B^{\wedge} = \{\omega \mid \omega = - b, \quad \text{for } b \in B\} \tag{2.1}$$

If B is the set of pixels (2-D points) representing an object in image, then B^ is the set of points in B in which (x, y) coordinates have been replaced by (-x, -y). Figure 2.2 (a) and (b) show a sample set and its reflection.

The translation of a set B by point z = ($z_1 + z_2$), is denoted by $(B)_z$, is defined as

$$(B)_z = \{c \mid c = b + z, \quad \text{for } b \in B\} \tag{2.2}$$

If B is the set of pixels representing an object in an image, then $(B)_z$ is the set of points in B in which (x, y) coordinates have been replaced by(x + z_1 , y + z_2). Figure 2.2 (c) shows this concept using the set B from Figure 2.2 (a).

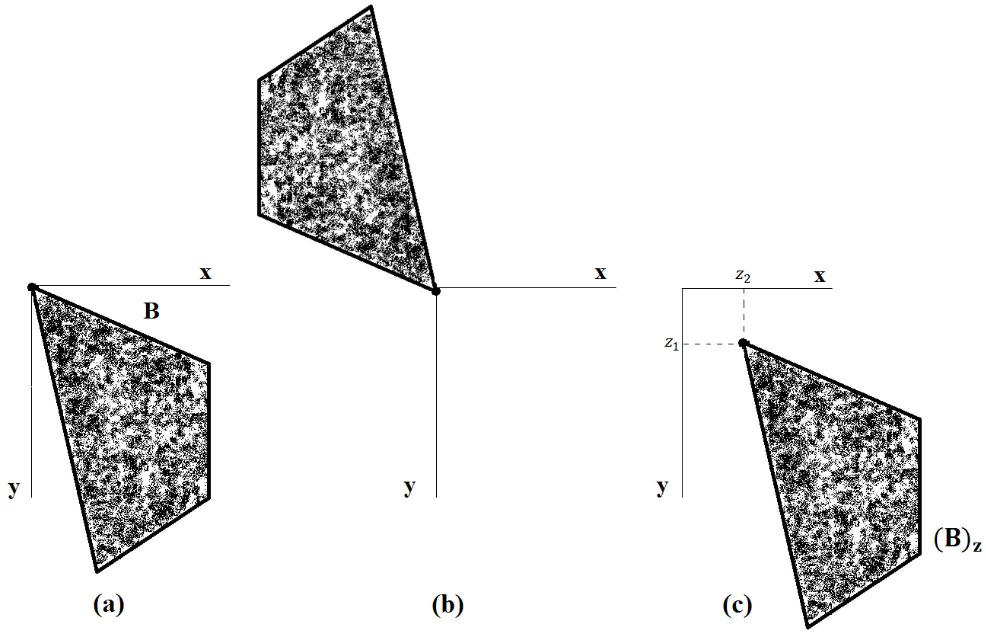


Figure 2.2 (a) a set of B, (b) reflection of set B and (c) translation of set B by z

2.2.1. Binary Dilation

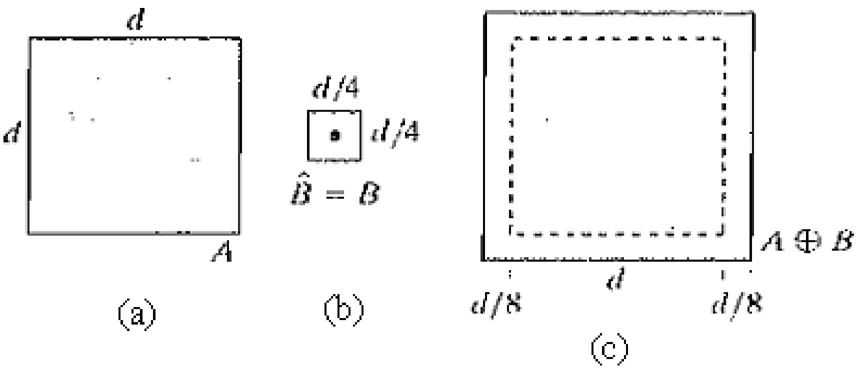
With A and B are sets in Z^2, the dilation of a binary image A by structure element B, is denoted by A⊕B, is defined as

$$A \oplus B = \{z \mid (\hat{B})_z \cap A \neq \emptyset\} \quad (2.3)$$

This is based on reflecting B about its origin and shifting that reflection by z in Figure 2.2. The dilation of A by B is the set of all displacements z, so that $\hat{B}$ and A overlapped by least one element. Based on the interpretation, equation (2.3) can be written as

$$A \oplus B = \{z \mid [(\hat{B})_z \cap A] \subseteq A\} \quad (2.4)$$

Where, B is a structuring element and A is the set of image objects to be dilated. Unlike erosion, dilation expands the objects in binary images by the shape of structuring element.



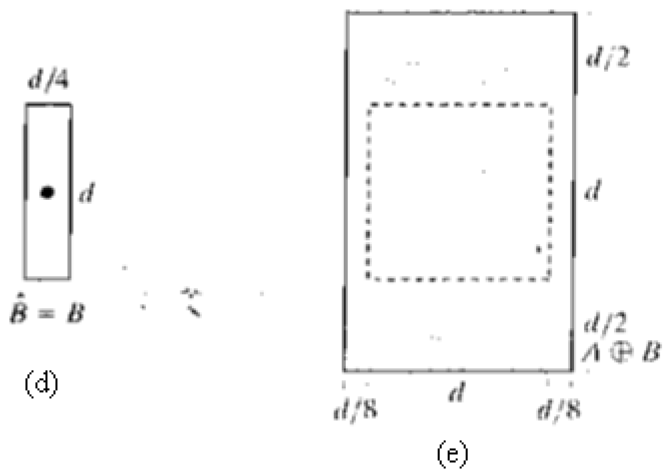


Figure 2.3 (a) Set A. (b) Square structuring element

(the dot line denoted as the origin)

(c) Dilation of A by B. (d) Elongated structuring element.

(e) Dilation of A using (d).

The dotted border in (c) and (e) is the boundary of set A

Figure 2.3(a) expresses the set A. Figure 2.3(b) shows the structuring element in which ($\hat{B}$) = B because the SE is symmetric about its origin. The dotted line in Figure 2.3(c) shows the original set for the reference and the solid line describes the limit beyond which any further displacements of the origin of $\hat{B}$ by z that cause the intersection of $\hat{B}$ and A to be empty. So, all the points on and inside that boundary constitute the dilation of A by B. Figure 2.3(d) represents a structuring element designed to achieve more dilation vertically and horizontally and Figure 2.3(e) expresses the dilation achieved with this element.

2.2.2. Binary Erosion

With A and B are sets in Z^2, the erosion of a binary image A by structure element B, is denoted by A Θ B, is defined as

$$A \ominus B = \{z \mid (\hat{B})_z \cap A \subseteq A\} \quad (2.5)$$

Where the erosion of A by B is the set of all points z such that of B, translated by z is contained in A. In the following equation, set B is assumed that the structuring element and that has to be contained in A is equivalent to B and can express the erosion as the following equivalent form

$$A \ominus B = \{z \mid (\hat{B})_z \cap A^c \neq \emptyset\} \tag{2.6}$$

Where, A^c is complement of A and $\emptyset$ is empty set. Figure (2.4) shows as an example of erosion. The elements of A and B are shown in shaded and the background is in white. The solid boundary in Figure 2.4(c) is the limit beyond the further displacements of the origin of B that cause the structuring element to terminate being completely contained in A. Therefore, the location points of the origin of B within the boundary, constitutes the erosion of A by B. The boundary set of A is described dashed in Figure 2.4(c) and (e) as a reference. Figure 2.4(d) expresses an elongated structuring element and Figure 2.4(e) shows the erosion of A by that element.

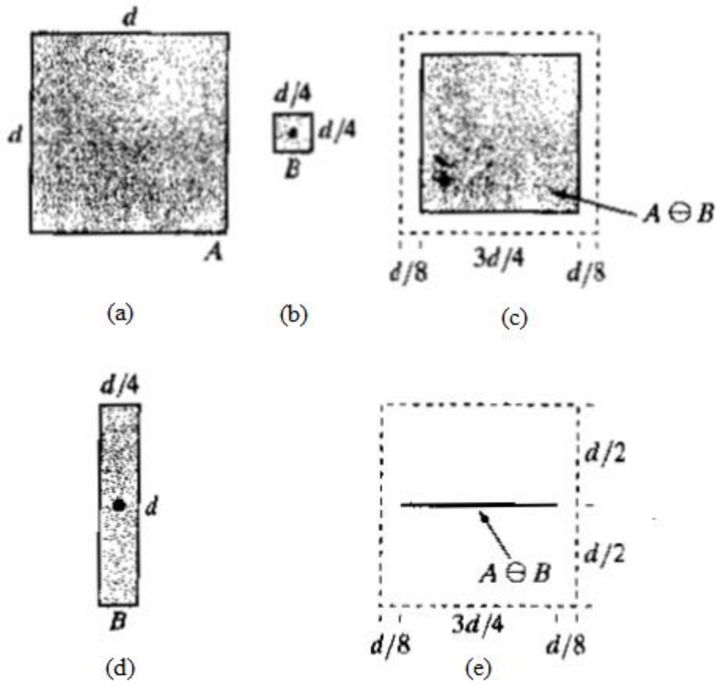


Figure 2.4 (a) Set A. (b) Square structuring element, B (c) Erosion of A by B, shown shaded. (d) Elongated structuring element.(e) Erosion of A by B using (d). The dotted border in (c) and (e) is the boundary of set A as reference

The erosion of the original image by the structuring element can be described intuitively by template translation. The erosion reduces in size of objects in a binary image as a filtering operation, “line filter” where the image deals with the smaller structuring element.

2.2.3. Binary Opening

Opening of a binary image A by structure element B, denoted by A ○ B, is defined as

$$A \circ B = (A \ominus B) \oplus B \qquad (2.7)$$

According to the definition, the erosion of original image is first occurred and then dilation takes place. The definition (2.8) gives a strict set-theoretic characterization of “fitting” property. The opening of A by B is obtained by taking the union of all the translations of B that fits into A:

$$A \circ B = \bigcup \{ B + x \mid B + x \subset A \} \qquad (2.8)$$

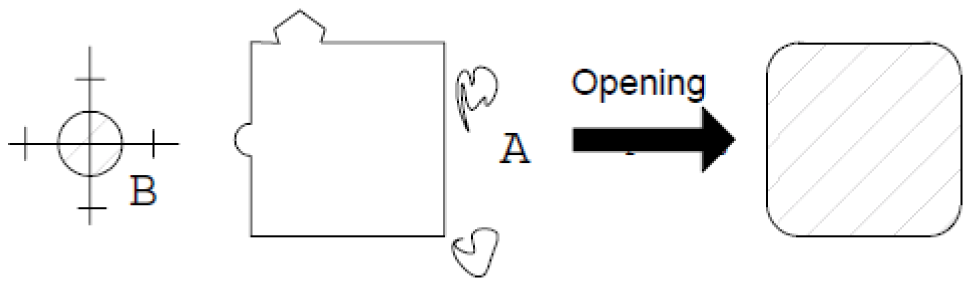


Figure 2.5 Morphological Opening

Figure 2.5 expresses the smoothed original image and removal of spot-like noise that is why the disk can't fit into them. The smoothing effect of the object boundary is highly depends on the shape of the structure element.

2.2.4. Binary Closing

Closing operation of a binary image A by structuring element B, is denoted by A ● B, and is defined as:

$$A \bullet B = (A \oplus B) \ominus B \tag{2.9}$$

The closing operation takes place where the original image includes all points satisfying the condition that anytime the point can be covered by a translation of the structure element; there exist some between the translated structure element and the original image.

From equation (2.8), an element of A ● B is z if and only if (B + y) ∩ A ≠ ϕ, for any translate (B + y) containing z. As stated in equation (2.9), firstly dilation of the image is done and then eroding it. Therefore, as a replacement for eliminating the small peaks, it fills the holes, as shown in Figure 2.6.

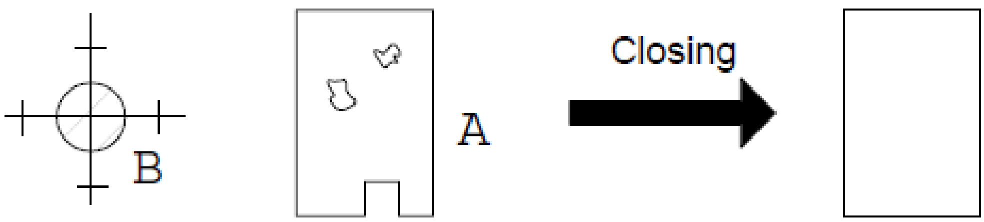


Figure 2.6 Morphological Closing

2.3. Holes Filling

Holes may be defined as background region surrounded by a connected border of foreground pixels. The flood-fill operation is commonly used to fill holes in images. In a binary image or gray scale image, the foreground objects represent spheres. In the processing, these objects should appear as disks, but instead are donut shaped because of reflections in the original photograph. Before doing any further processing of the image, it is need to first fill in the "donut holes"**.**

Set as an array, X_0 of 0s is the same size of A, except at the location in X_0 corresponding to the given point in each hole, which set to 1. Then the following procedure fills the holes with 1s:

$$X_k = (X_{k-1} \oplus B) \cap A^c \qquad k = 1,2,3, \ldots \ldots \ldots. \qquad (2.10)$$

Where, B is the symmetric structuring element. This algorithm ends at iteration step k if $X_k = X_{k-1}$. The set of X_k contains all filled holes. The union set of X_k and A contains all filled holes and their boundaries.

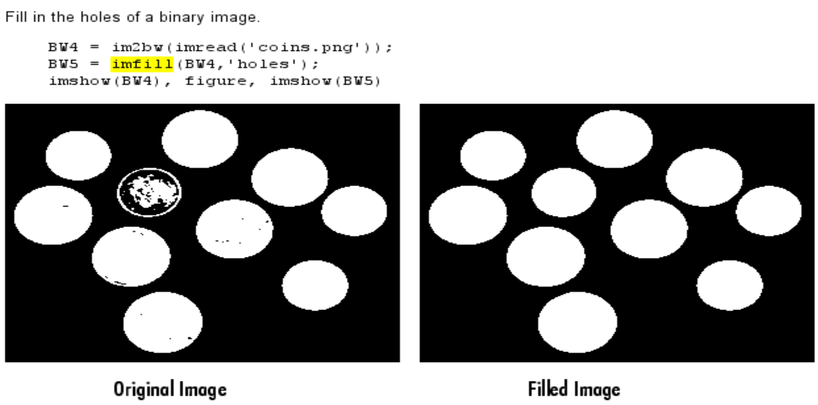


Figure 2.7 Sample of filling holes

Figure 2.7 shows the result of filling all the spheres. An image may result from the thresholding into two levels a sense containing spheres. The dark spots inside the sphere can be the result of reflection. The white dot inside one region is the points for the holes filling operation.

2.4. Structuring Element

Transforming of the given image with the purpose of drawing conclusions on how this shape fits or misses the shapes in the image is known as a structuring element. In morphological operations, the structuring element is archetypal used [31]. The choice of a structuring element for a particular morphological operation influences the information one can obtain. There are two characteristics mainly:

- **Shape**. Ball, line, curved or a ring, etc. which are the shape structuring element.
- **Size**. One of the structuring elements can be a 3×3 square or a 21×21 square.

There are two types of structuring elements which are flat and non-flat [31]. Flat structuring element is binary 2-D neighborhood or multidimensional, where true pixels are computed and the false pixels are not. The center pixel of the structuring element, the origin, shows the pixel in the image being processed is shown in Figure 2.8.

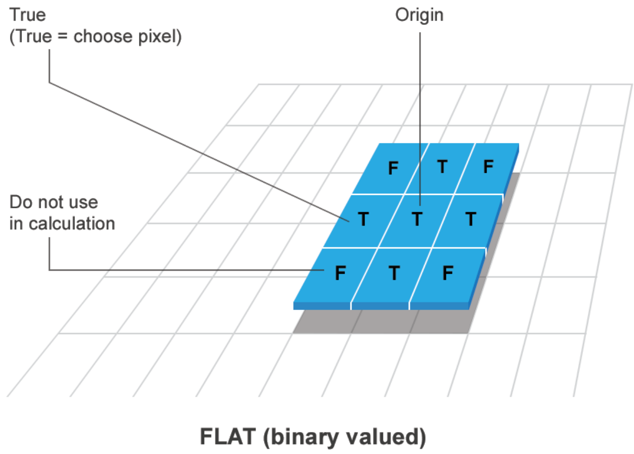


Figure 2.8 Flat Structuring Elements

Non-flat is a double matrix type that identifies the pixel in the processed image and defines the neighborhood used in the processing of the pixel. It contains finite values used as additive offsets in the morphological computation. The center pixel is the origin which expresses the pixel in the image that is being processed. Pixels in the neighborhood with the value –Inf are not in the computation are shown in Figure2.9.

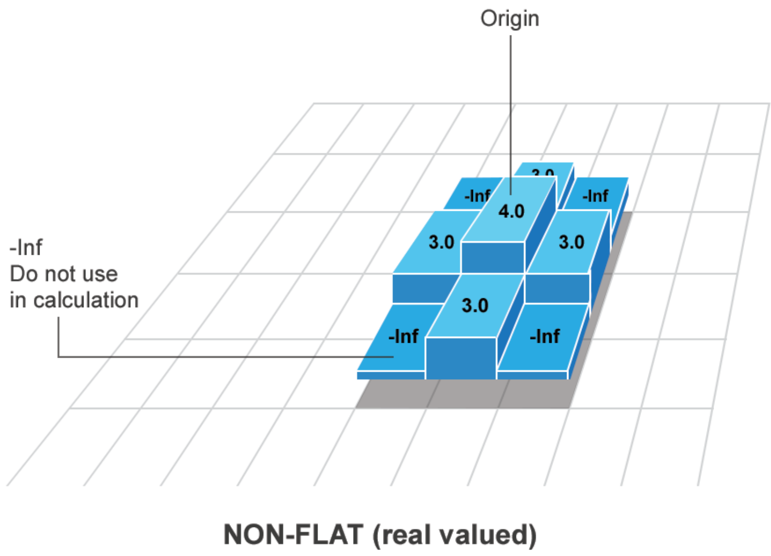


Figure 2.9 Non-Flat Structuring Elements

2.4.1. Mathematical Particulars and Examples of Structuring Element

The binary images are subsets of Euclidean space R^d or the integer grid Z^d, for some dimension d in mathematical morphology. The followings are some examples of widely used structuring elements which are denoted by B

- $E=R^2$; B is an open disk of radius r, centered at the origin.
- $E=Z^2$; B is a 3x3 square, B = {(-1, -1), (-1,0), (-1,1), (0, -1), (0,0), (0,1), (1, -1), (1,0), (1,1)}.
- $E=Z^2$; B is the cross given by: B= {(-1, 0), (0, -1), (0, 0), (0, 1), (1, 0)}.

A structuring element can represent as a set of pixels on a grid assuming the values 1s or otherwise 0s [32].The morphological functions use the following statement to get the coordinates of the origin of structuring element of any size and dimension:

origin = floor((size (n hood) +1) /2) (2.11)

Where, `n hood` is the neighborhood that is defined the structuring element. The pixels with values of 1's classify the neighborhood of structuring element. For example, the following statement creates the decomposition of 4-by4 diamond shaped structuring element containing 41 as shown in Figure 2.10.

Example:

sel = strel('diamond',4);

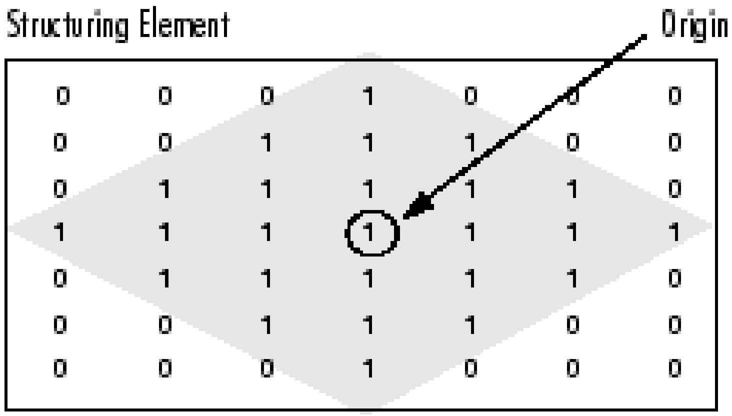
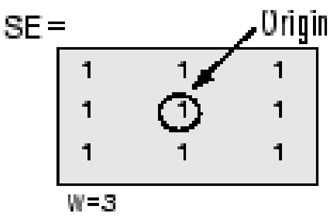
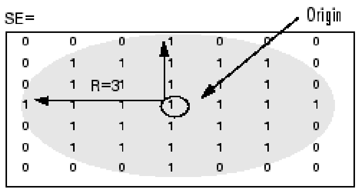


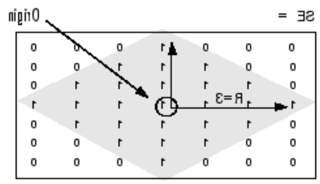
Figure 2.10 Flat STREL object containing 41 neighbors. Decomposition: 3 STREL objects containing a total of 13 neighbors



Square (3 × 3) element
SE = 3



Disk (3 × 3) element
SE = 3



Diamond (3 × 3) element
SE = 3

1	1	1
1	1	1
1	1	1

Square (3 × 3) element
SE = 3

0	1	1	1	0
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
0	1	1	1	0

Disk (5 × 5) element
SE = 5

0	0	0	1	0	0	0
0	0	1	1	1	0	0
0	1	1	1	1	1	0
1	1	1	1	1	1	1
0	1	1	1	1	1	0
0	0	1	1	1	0	0
0	0	0	1	0	0	0

Diamond (7 × 7) element
SE = 7

Figure 2.11 Some Shapes of Structuring element

CHAPTER 3

CAR COUNTING SCHEME

3.1. Overview of the System

The proposed image car counting system consists of five steps: background registration, foreground detection, image segmentation, extracted object detection and counting. The first four steps are detecting object functions and the last is counting the number of cars. The overview of the system is described in Figure 3.1.

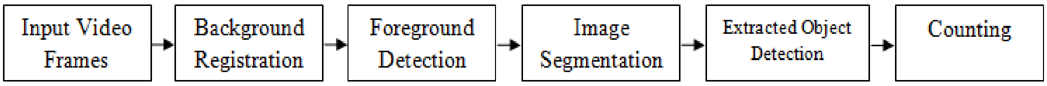


Figure 3.1 Overview of the Car Counting System

The system uses an existing video sequence. First, the input video file is fetched, and then read after that it is converted to frames. The first frame is taken as the reference frame. The subsequent frames are considered as the input frames. Secondly, the difference between frames is computed and compared pixels having the same values in the frame difference are eliminated. In the next stage, the post processing is executed on the image obtained in the previous stage and the final stage is counting the number of cars.

3.1.1. Input Video

The captured input video file can be loaded from any source such as digital camera, memory stick and hard disk. Examples of input videos are shown in Figure 3.2. In this system, Audio Video Interleaved (.avi) file format is used. Matlab application uses the following file formats which vary by platform:

Table 3.1 Parameters and Setting Result of Sample Video Files Taken for Testing

No.	Type of Platforms	File Formats Setting
1	Any Type of Platforms	Audio Video Interleaved (.avi) Motion JPEG 2000 (.mj2)
2	All Windows	MPEG 1(.mpg) Windows Media Video (.wmv, .asf, .asx) Any format supported by Microsoft Direct Show
3	Window 7 or later	MPEG 4,including H.24 encoded video(.mp4, m4v) Apple Quick Time Movie (.mov) Any format supported by Microsoft Media Foundation
4	Macintosh	Most format supported by Quick Time Player including: MPEG 1(.mpg) Apple Quick Time Movie (.mov) MPEG 4,including H.24 encoded video(.mp4, m4v) 3GPP,3GPP2, AVCHD, DV
5	Linux	Any format supported by installed plug-ins for GStreamer 0.10, and including Ogg Theora (.ogg)

Syntax Example:
videoObj = VideoReader(‘myExampleFile.avi');
get(videoObj);
Example: 'myExampleFile.avi'

Example: '../dir/videos/myExampleFile.avi’

DataType: char

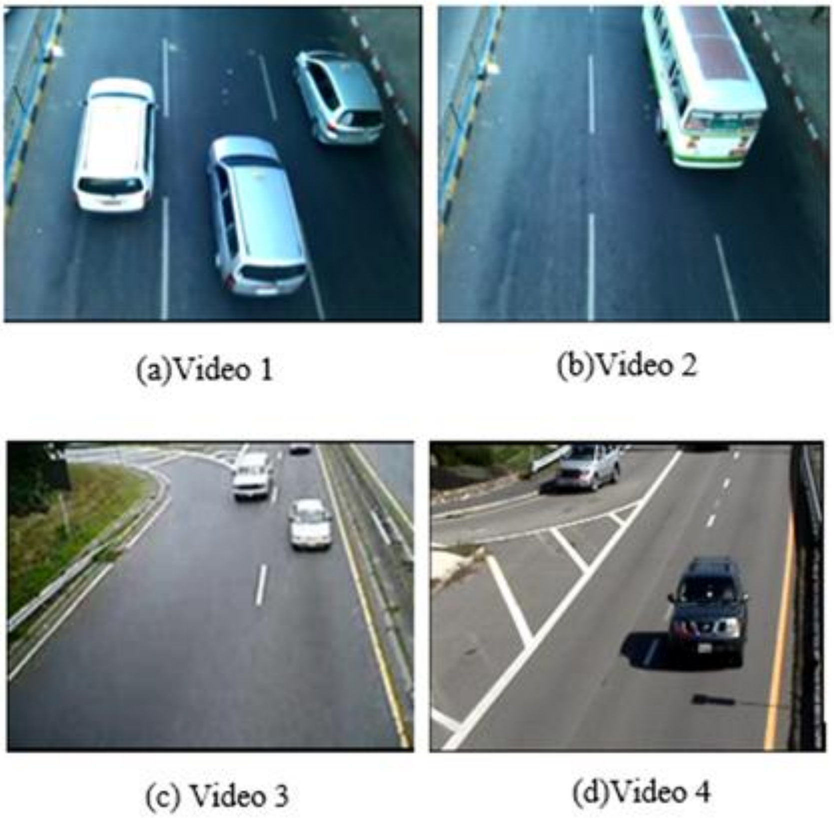


Figure 3.2 (a), (b), (c) and (d) shows The Sample Video Files Taken for Testing

The testing videos are taken from the Suelay Pagoda Road, Lan Thit Road (Maw Tin) and the rests are taken from the internet as references. The following Table 3.2 shows the parameters and setting results of sample video files taken for testing.

Table 3.2 Parameters and Setting Result of Sample Video Files Taken for Testing

Parameters	Setting Results of video 1	Setting Results of video 2	Setting Results of video 3	Setting Results of video 4
No Of Frames	3120	2052	120	531
Parameters	Setting Results of video 1	Setting Results of video 2	Setting Results of video 3	Setting Results of video
Duration (minute)	02:01	01:07	00:08	00:17
Frame Width (pixels)	640	1280	160	856
Frame Height (pixels)	480	720	120	480
Frame Rate (frame/sec)	25	30	15	29
File Format	AVI	AVI	AVI	AVI
Bit Per Pixels	24	24	24	24

3.1.2. Registration of Background

In general detecting approach, to take out the regions from the taken video file, background modeling technique is being used. That involves subtracting every image from the background scenes. The first frame is assumed as an initial background and thresholding the resultant difference image to determine the foreground image. A vehicle is a group of pixels that move in a coherent manner, either as a lighter region over a darker background. Often the vehicle may be of the same color with the background, or may be some portion of it may be resembled the background, due to which detecting the vehicle becomes difficult. This leads to an erroneous vehicle count.

3.2. Detection of Foreground

Foreground detection is a technique in which any image's foreground is extracted for further processing like object tracking, human poses estimation etc. In general, humans, cars, texts etc. may be foregrounds. After the preprocessing state, de-noising, post-processing like object localization is required. This approach is widely used for detecting moving objects in videos from static cameras. The background model in this thesis is that of detecting the moving objects from the difference between the current frame and the reference frame.

3.2.1. Frame Differencing

A motion detection algorithm begins with the segmentation part where foreground or moving objects are segmented from the background. There is a simple way to implement this is to take an image as background and take the frames obtained at time t, denoted by I (t) to compare with the background image denoted by B. In mathematical format, it is written as:

$$P\,[F\,(t)] = P\,[I\,(t)] - P\,[B] \tag{3.1}$$

At time t, the background is assumed to be the frame. The difference image will show a few intensity for the pixel locations which have changed in the two frames [23]. It is need to put a threshold on the differencing to improve the subtraction. This means that the differences of image pixels' intensities are filtered on the basis value of threshold [4]. The accuracy of this thesis is dependent on the speed of movement in the scene so faster movement may require higher threshold.

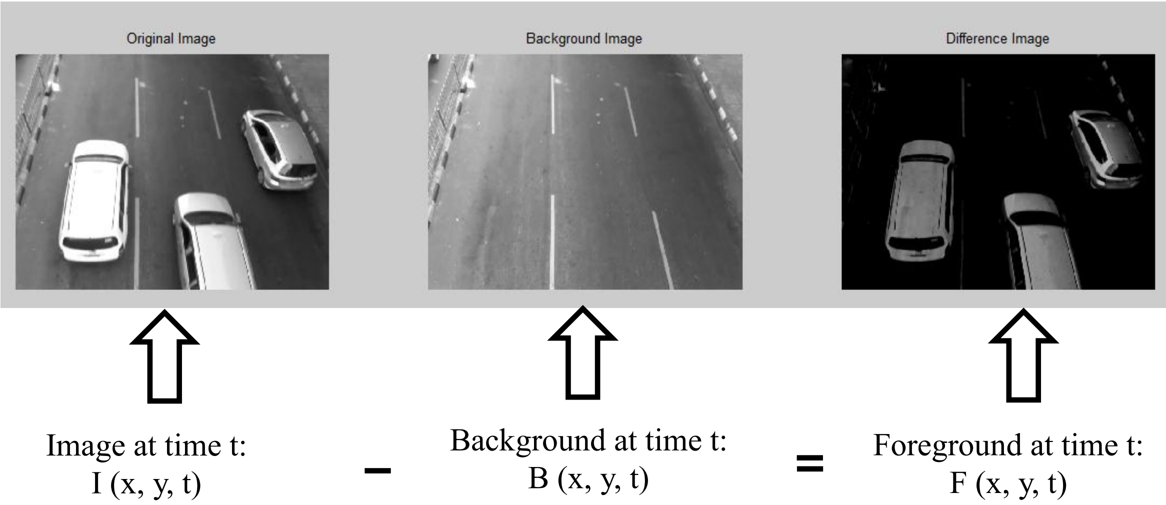


Figure 3.3 Foreground Object Using Frame Differencing

3.2.2. Mean Filter

In this section, we will calculate the image containing only the background, a series of preceding images are averaged. It begins with the background image at the instant t,

$$B\ (x, y, t) = \frac{1}{N} \sum_{i=1}^{N} V\ (x, y, t - i) \tag{3.2}$$

Where, N is the number of preceding images taken for averaging. That average is referred to average of corresponding pixels in the given images. N depends on the number image per second in the video and the amount of movement in the video [5]. After calculating the background B(x, y, t), we have to subtract it from the image V(x, y, t) at time t=t and then threshold it. Thus, the foreground is

$$|V\ (x, y, t) – B\ (x, y, t)\ | > Th \tag{3.3}$$

Where, Th is threshold value. We can also use median instead of mean in the above calculation of B(x, y, t). [21].

3.3. Image Segmentation

In computer vision, segmentation is the process of transforming from low-level a gray scale or color image into high-level image description in terms of features, objects, and scenes. The result depends on reliability; however, an accurate partitioning of an image is a challenge. In this step, the region of interest may contain unknown detected objects.

3.3.1. Thresholding

In segmentation section, thresholding is the simplest non-contextual technique. A gray scale or color image transforms into a binary image with a single threshold. Assume that f(x, y) is composed with light objects on dark background. To extract the objects from the background, choose threshold, T. The points anywhere f(x, y) > T is object point, otherwise, background point. In symbol, the segmented image g(x, y) is:

$$g(x, y) = \begin{cases} 0 \text{ if } f(x,y) \leq T \\ 1 \text{ if } f(x,y) > T \end{cases} \quad (3.21)$$

By using two types of light objects on dark background, T_1, T_2, a range of grey levels related to region 1 can be defined:

$$g(x, y) = \begin{cases} 0 & \text{if } f(x,y) < T1 \\ 0 & \text{if } f(x,y) > T2 \\ 1 & \text{if } T1 \leq f(x,y) \leq T2 \end{cases} \quad (3.22)$$

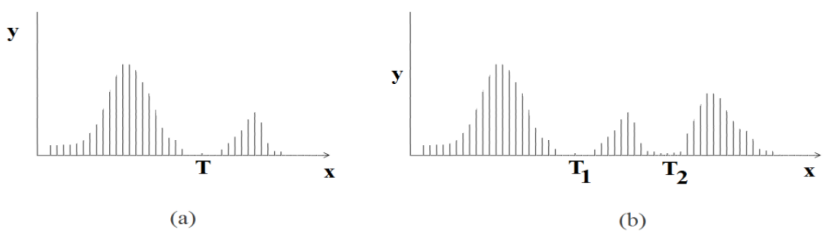


Figure 3.4 (a) by single threshold histogram (b) by double thresholds histogram

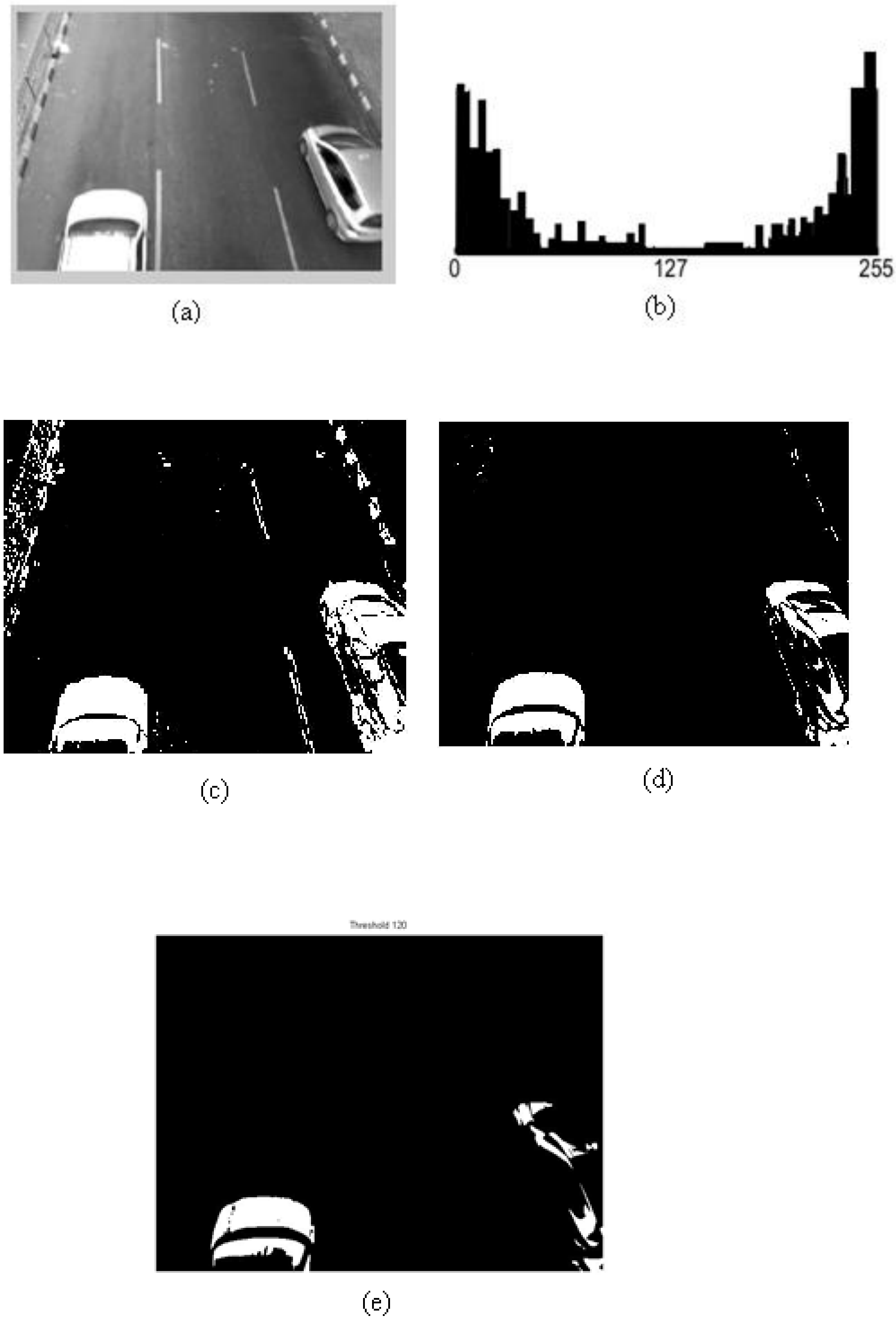


Figure 3.5 (a) Gray scale image of Cars (b) Its gray level histogram (c) Binary regions for threshold = 20 (d) Binary regions for threshold = 60 (e) Binary regions for threshold = 120

The above Figure (3.5) shows the results of binary image at various threshold values.

3.4. Extracted Object Detection

The motion of the car is irregular, there are some noise regions both in the foreground object and background region. In addition, the boundaries of the cars are also not sleek and so a post processing technique is used on the foreground image. In this thesis filtering is applied on the mathematical morphology operators: adjusting the image intensity values to the color map, filling holes if necessary and open blobs of area greater than x (x may be a constant value that may vary on the input image intensity). The final output of the detection stage is a binary image of car detected.

3.5. Car Counting

The detected image forms the input image for counting. The image is scanned from top to bottom for detecting the presence of vehicle. The counting processes consists of the measuring blobs properties for each connected component (object) in the detected image and by tracking as the direction of blobs pass a virtual line. Blobs that completely passes over the lines increment the counter and gives the number of cars present in the image sequence.

CHAPTER 4
IMPLEMENTATION AND EXPERIMENTAL RESULTS

4.1 System Flow Diagram

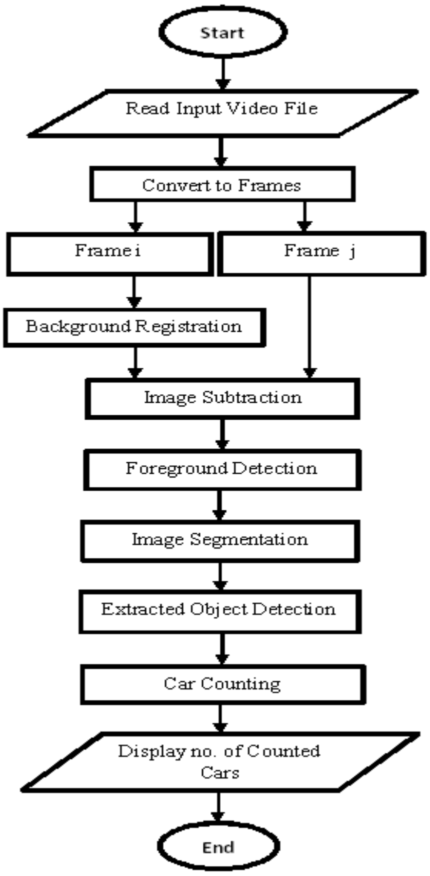


Figure 4.1 System Flow Diagram of the Car Counting

Above Figure 4.1, as shown the video file as an input and read then convert into frames. These converted frames are changed into gray scale and then into binary images before they are differences from foreground frame and background frame. At frame differencing processing state, it executed on the image using image subtraction and the background is eliminated then only the foreground vehicle get and counting the detected vehicle.

4.2 Implementation of the System

This system is implemented with MATLAB programming to detect the car and count in the offline video scene. This is an interactive system whose basic data element is an array that does not require dimensioning. It is a tool used for formulating solutions to many technical computing problems, especially those involving matrix representation.

Step 1: Read the input video file.

Step 2: Convert the video file to frames.

Step 3: Register the first frame as background and convert it to gray scale format using double precision. Figure 4.4 shows the samples of tested backgrounds.

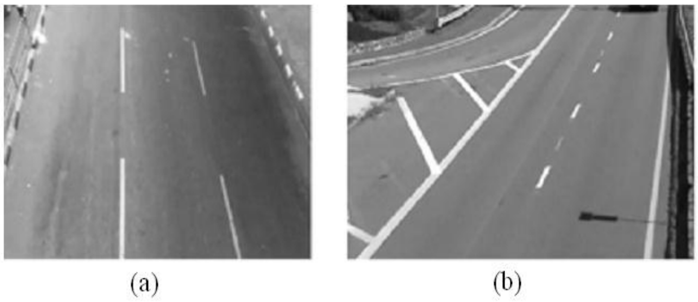


Figure 4.2 (a), (b) Background Registered Images

Step 4: Find the width and height of the image.

Step 5: Set the threshold value a (‘a’ may be varied depends on the usage of video file resolution).

Step 6: Read the next images and convert them to gray scale format.

Step 7: Find the difference between frames based on the threshold value.

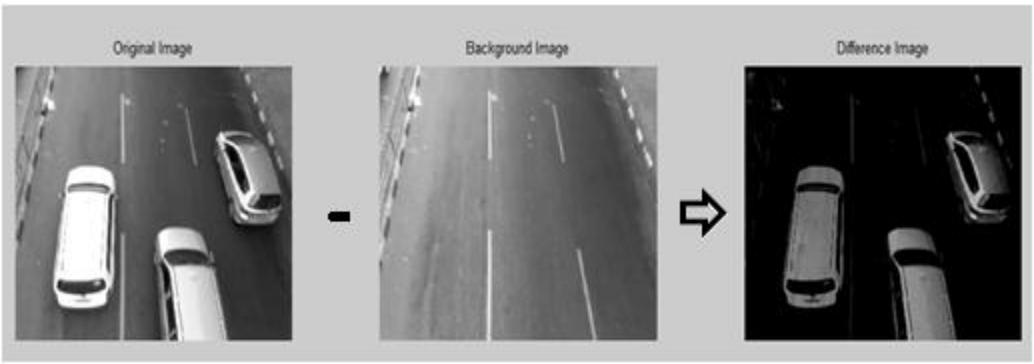


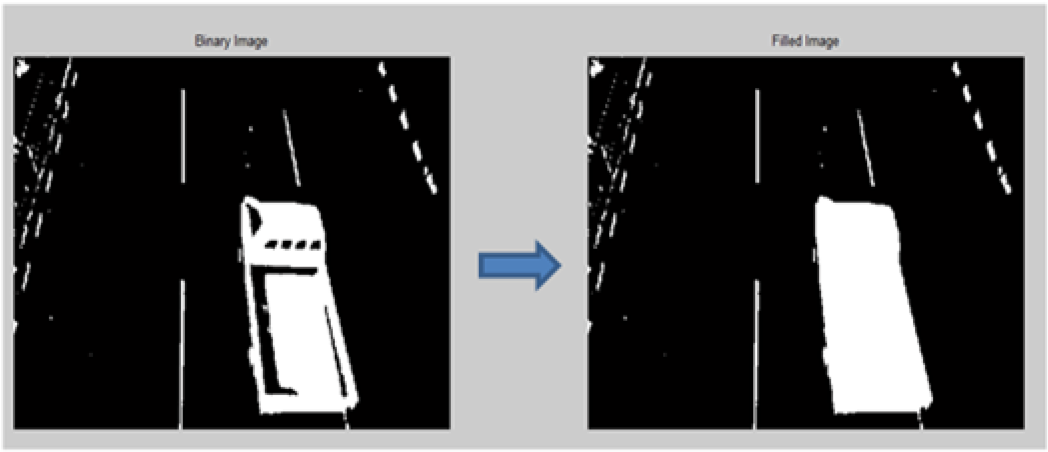
Figure 4.3 Image Differencing Frame Images by using threshold values

Step 8: If frame difference is greater than the threshold value then assigns that image to a variable else assign zero if no difference is found. Figure 4.3 shows the result of differencing frames.

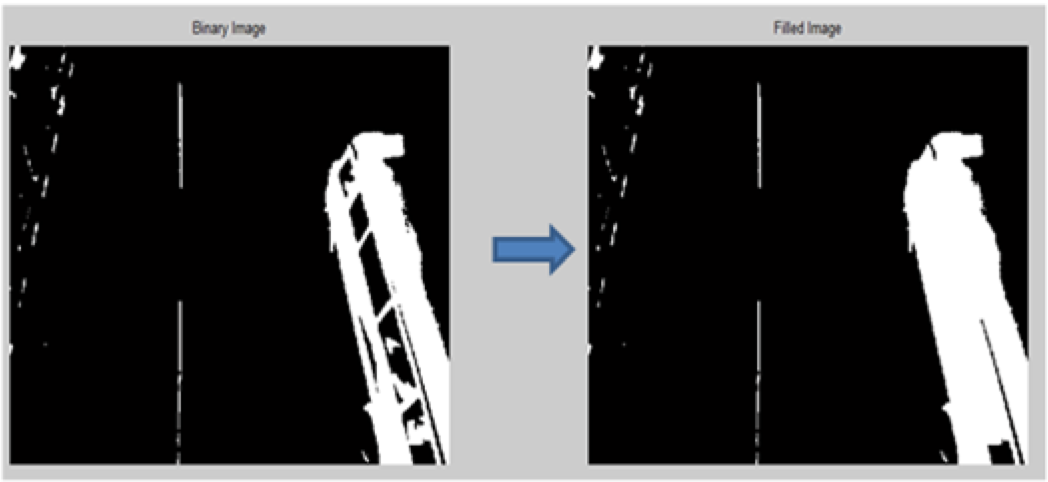
Step 9: Increase the contrast of the output image using function imadjust().

Step 10: Find a gray threshold value using function greythresh() and then convert to binary.

Step 11: Fill any holes to the blobs, so that regionprops can be used to estimate the area enclosed by each of the boundaries by using function imfill() as shown in Figure 4.6 (a) and (b).



(a)



(b)

Figure 4.4 (a) and (b) Filled Image Frames by using morphology filling method

Step 12: Remove the pixels which do not belong to the objects of interest in Figure 4.7.

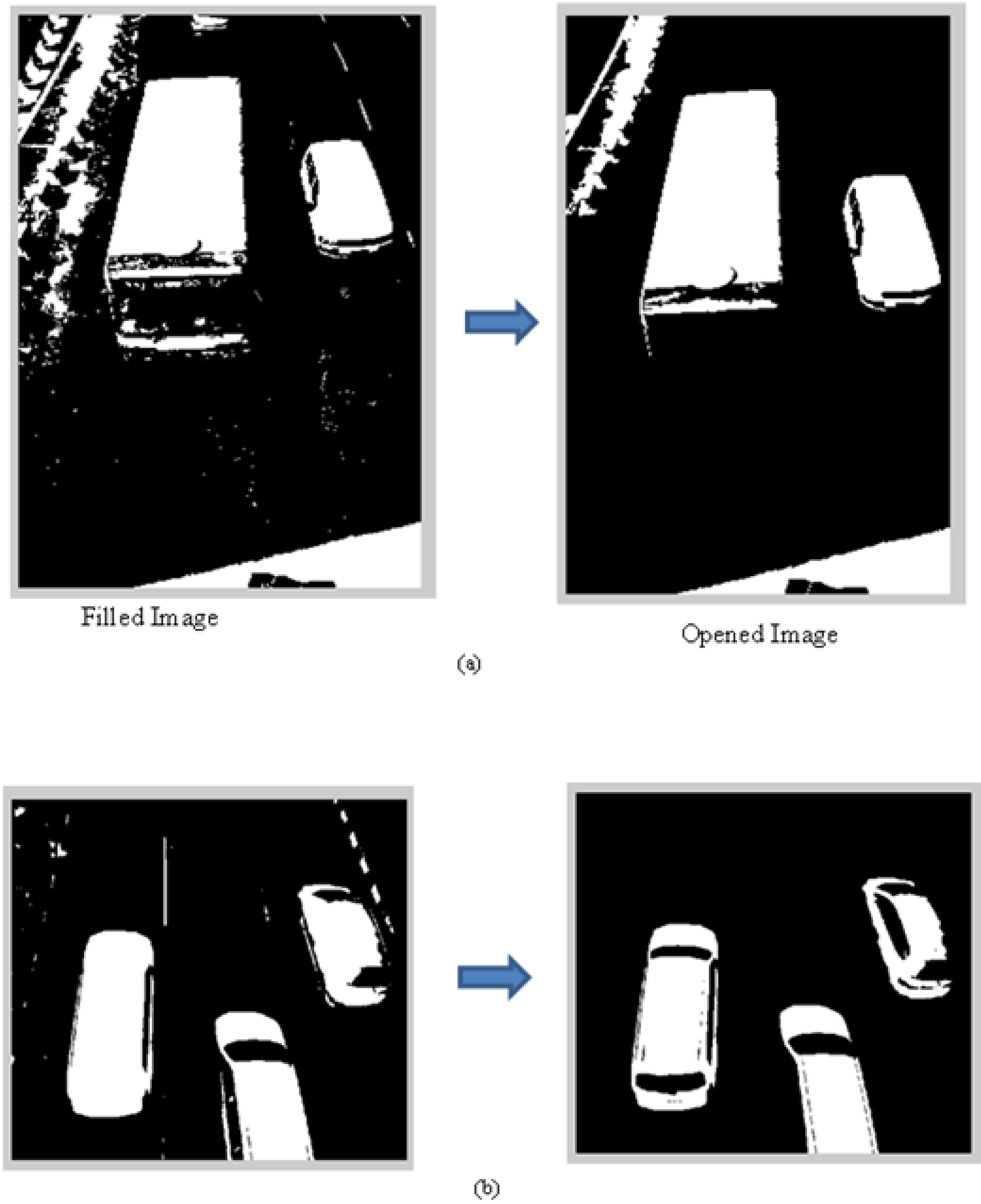


Figure 4.5 (a) and (b): Removing small structures

Step 13: Locate the cars' objects in each binary image using blob analysis. Then draw a reference line for counting cars passing through it to track the number of cars in the region of interest.

Step 14: Count the number of cars by calculating Area and Centroid of the detected objects. If the Centroid is nearly equal to the line position, the system increases the counting and tag the car with red dot.

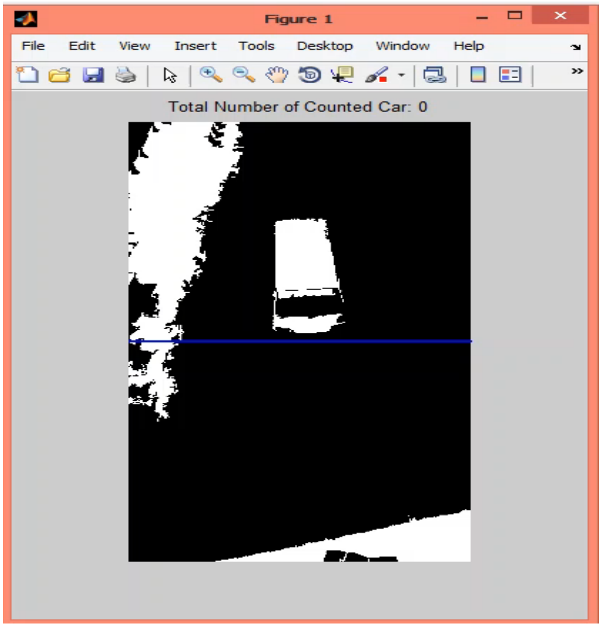


Figure 4.6 Before the Object (Detected Car) Passing Through the Line, Count 0,

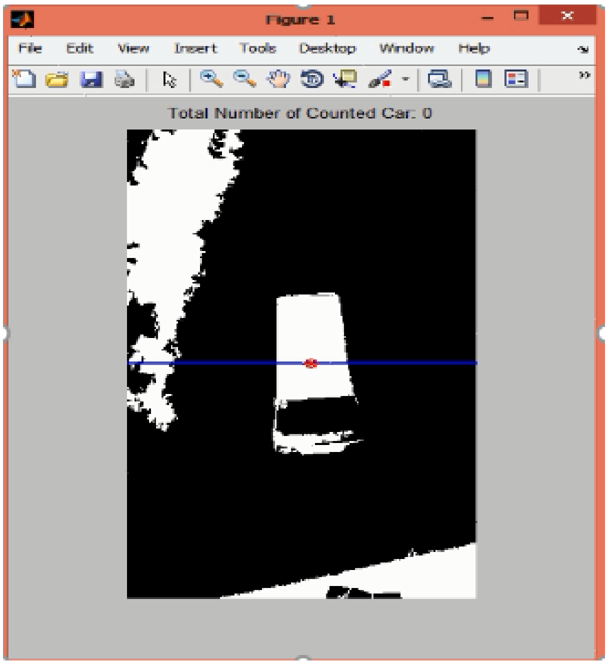


Figure 4.7 Tags with Red Dot to Object While It Is Passing Through the Line

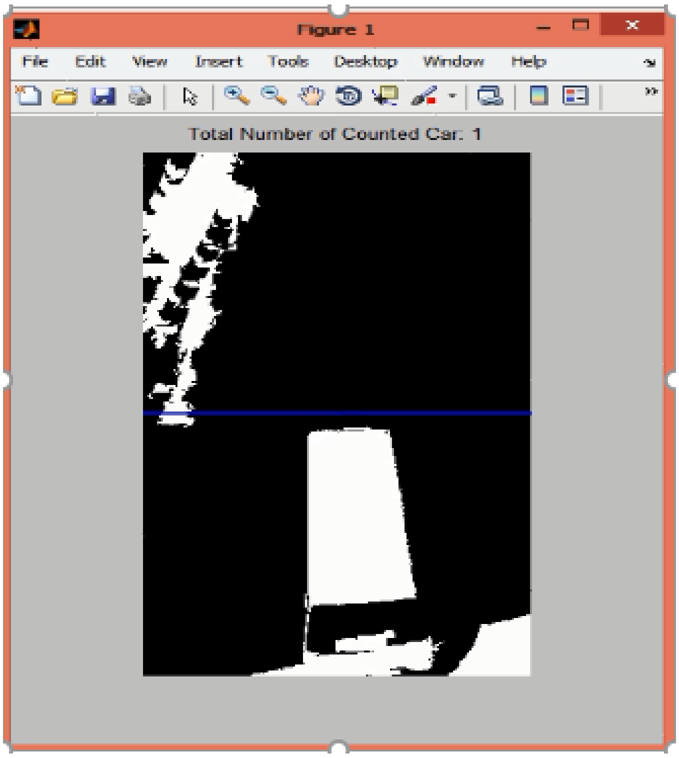


Figure 4.8 Increases Counting on Moving Car after Passing the Line

In Figure 4.7, 4.8 and 4.9 the testing of this implemented system, while the vehicle is passing the line before and after passing through the detected line.

4.3 Experimental Results

In this system, the experimental results are tested with the offline video image sequences on different roads. All the videos chosen for vehicle counting have been taken during the day time. As a part of preprocessing in the proposed method, moving objects are converted the color video frames to gray scale images and extracted using frame difference and binary detection of objects. Techniques are used to count the number of vehicles passing through the line on the roads.

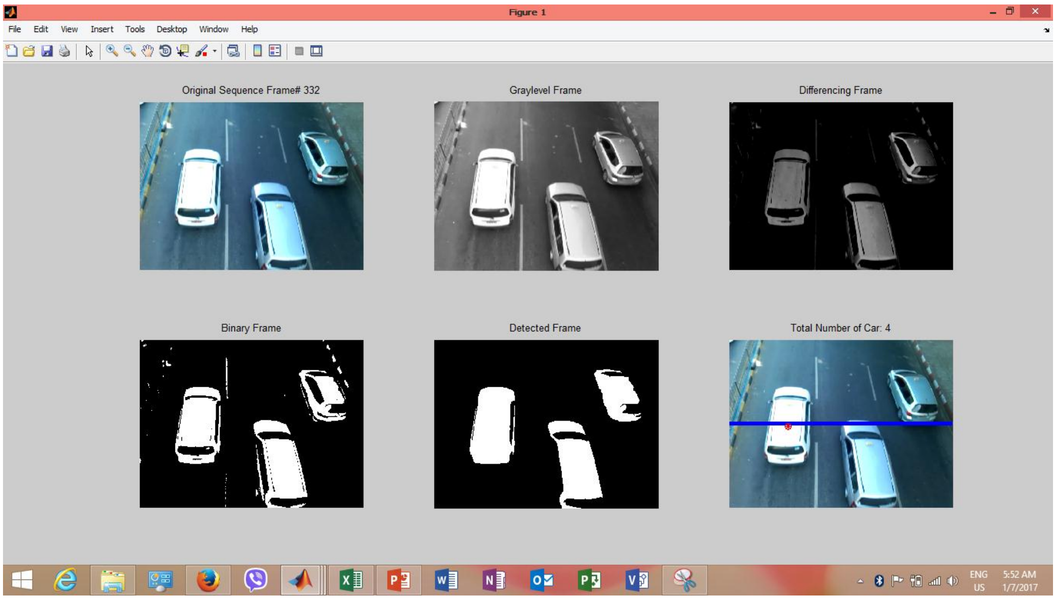


Figure 4.9 Testing Results of Video 1

Figure 4.9 show the result of different testing case is done on video 1 at different frame location to detect the vehicle. In moving car video, the frame sequence used is 332.

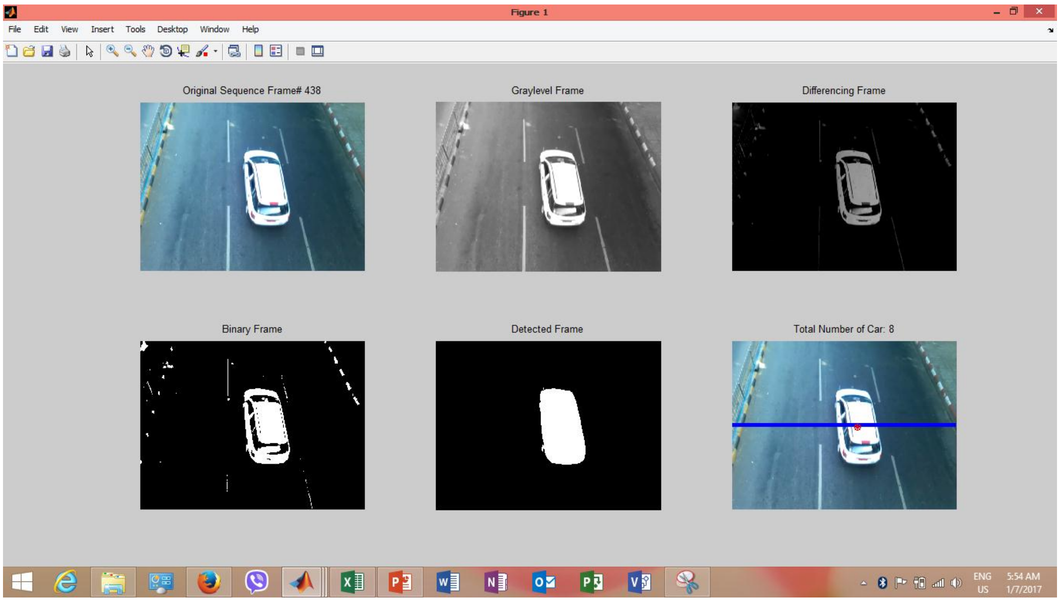


Figure 4.10 Testing Result of Video 1

Figure 4.10 show the result of different testing case is done on video 1 at different frame location to detect the vehicle. In moving car video, the frame sequence used is 438.

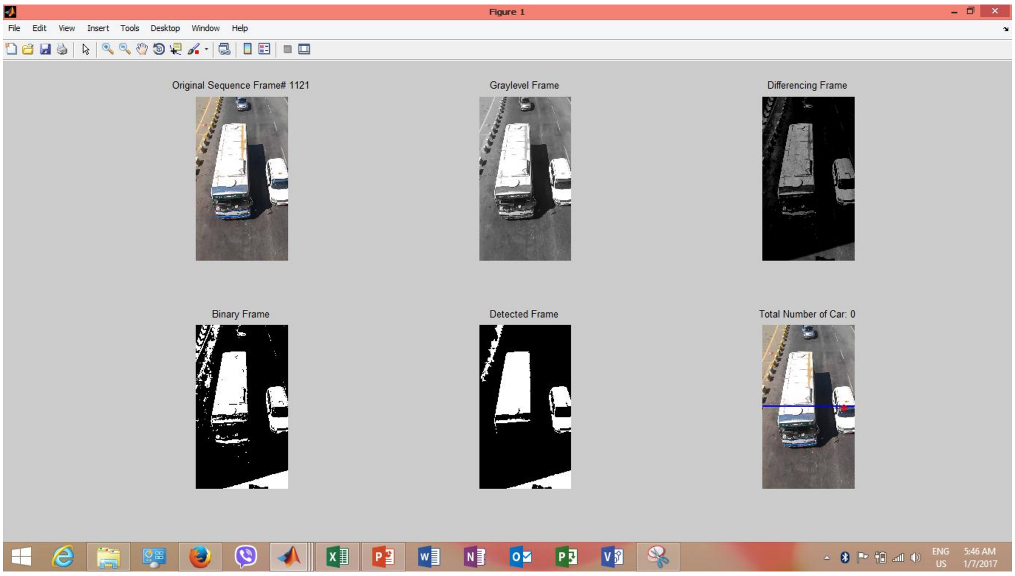


Figure 4.11 Testing Results of Video 2

Figure 4.11 show the result of different testing case is done on video 2 at different location to detect the vehicle. In moving car video the frame sequence used is 1121.

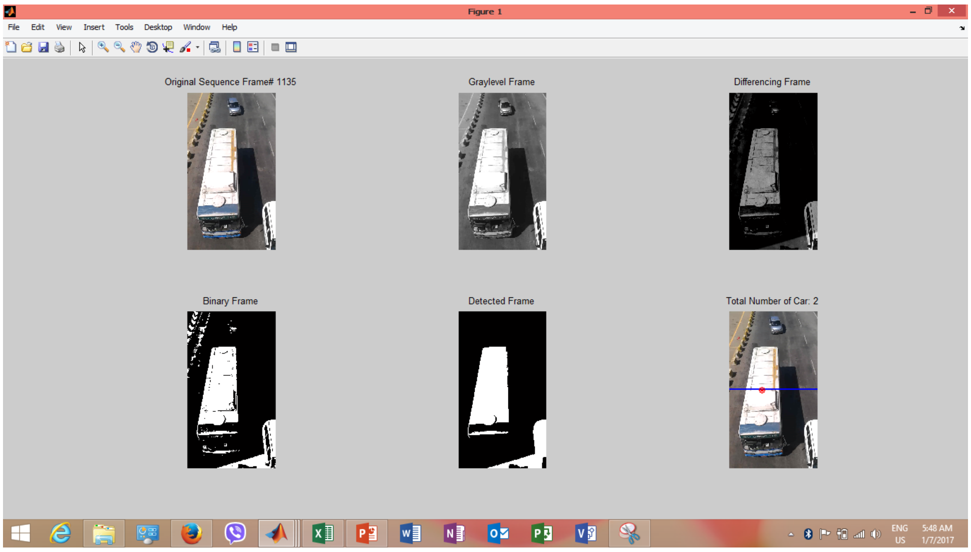


Figure 4.12 Testing Results of Video 2

Figure 4.12 show the result of different testing case is done on video 2 at different frame location to detect the vehicle. In moving car video the frame sequence used is 1135.

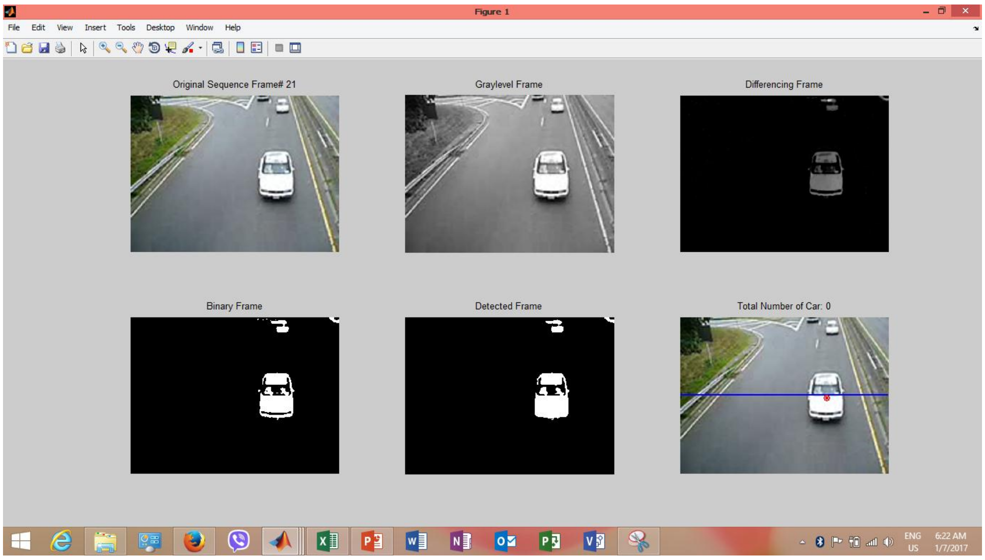


Figure 4.13 Testing Results of Video 3

Figure 4.13 show the result of different testing case is done on video 3 at different frame location to detect the vehicle. In moving car video the frame sequence used is 21.

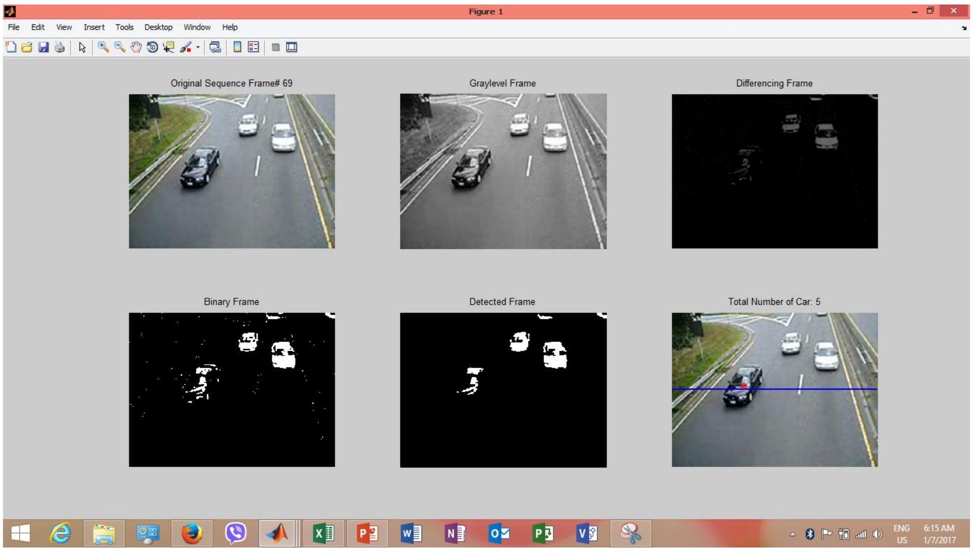


Figure 4.14 Testing Results of Video 3

Figure 4.14 show the result of different testing case is done on video 3 at different frame location to detect the vehicle. In moving car video the frame sequence used is 69.

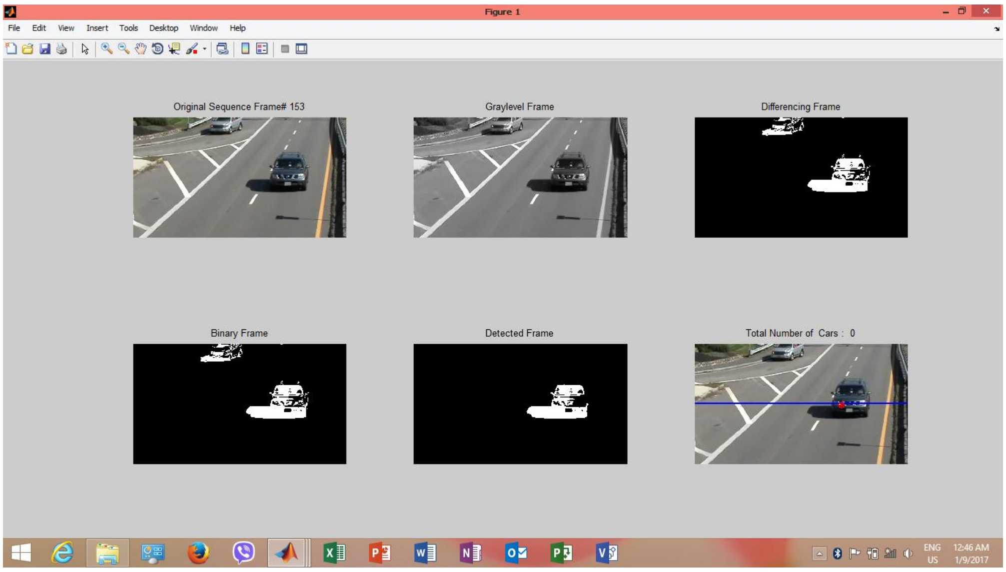


Figure 4.15 Testing Results of Video 4

Figure 4.15 show the result of different testing case is done on video 4 at different frame location to detect the vehicle. In moving car video the frame sequence used is 353.

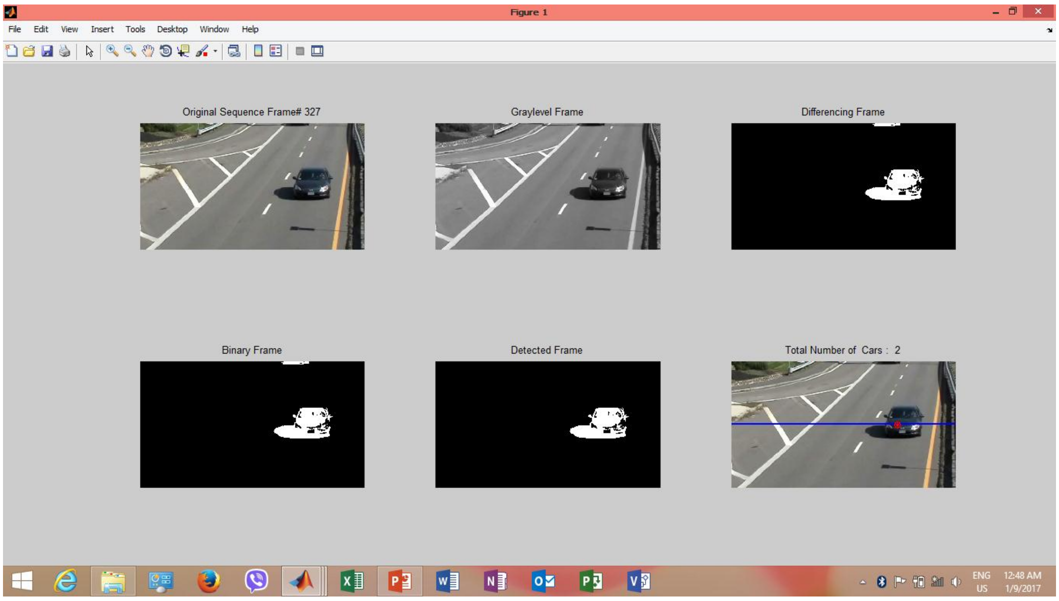


Figure 4.16 Testing Results of Video 4

Figure 4.9 show the result of different testing case is done on video 4 at different frame location to detect the vehicle. In moving car video the frame sequence used is 327.

Table 4.1 Overall Counting Results

Input Video	Frame Rate	No. of Frames	Actual No. of Cars	Detected No. of Cars
Video 1	**25**	**1865**	**30**	**27**
Video 2	**30**	**2052**	**20**	**16**
Video 3	**15**	**120**	**10**	**9**
Video 4	**29**	**531**	**4**	**4**

By using the detection and counting method on car counting system by using the original captured video files, the various results are obtained as shown in the Table 4.1.

Table 4.2 Counting Results with Difference Frame Rates

Input Video	Frame Rate	No. of Frames	Actual No. of Cars	Detected No. of Cars
Video 1	**12**	**868**	**30**	**16**
Video 1	**20**	**1447**	**30**	**23**
Video 1	**25**	**1865**	**30**	**27**
Video 1	**30**	**2168**	**30**	**31**

The above Table 4.2 gives different results are done on the testing video 1 with different frame rates.

Table 4.3 Counting Results with Difference Frame Rates

Input Video	Frame Rate	No. of Frames	Actual No. of Cars	Detected No. of Cars
Video 2	12	995	20	7
Video 2	20	2052	20	16
Video 2	25	2465	20	21
Video 2	30	2873	20	27

The above Table 4.3 express different results are done on the testing video 2 with different frame rates.

Table 4.4 Counting Results with Difference Frame Rates

Input Video	Frame Rate	No. of Frames	Actual No. of Cars	Detected No. of Cars
Video 3	12	46	10	2
Video 3	20	86	10	4
Video 3	25	120	10	9
Video 3	30	531	10	11

The above Table 4.4 describes different results are done on the testing video 3 with different frame rates.

Table 4.5 Counting Results with Difference Frame Rates

Input Video	Frame Rate	No. of Frames	Actual No. of Cars	Detected No. of Cars
Video 4	12	65	4	1
Video 4	20	175	4	1
Video 4	25	410	4	2
Video 4	30	531	4	4

The above Table 4.5 express different results are done on the testing video 3 with different frame rates. As a result from Table 4.2, 4.3, 4.4, 4.5, the number of detected car can get nearly like actual number of car with approximately equal frame rates of the original video files.

Table 4.6 Counting Results with Threshold Values

Input Video	Threshold Value	Frame Rate	No. of Frames	Actual No. of Cars	Detected No. of Cars
Video 1	35	25	1865	30	24
Video 1	40	25	1865	30	26
Video 1	60	25	1865	30	27
Video 1	80	25	1865	30	20
Video 1	>100	25	1865	30	11

The above Table 4.6 express different results are done on the testing video 3 with different threshold values.

Table 4.7 Counting Results with Threshold Values

Input Video	Threshold Value	Frame Rate	No. of Frames	Actual No. of Cars	Detected No. of Cars
Video 2	35	30	2052	20	13
Video 2	40	30	2052	20	14
Video 2	60	30	2052	20	16
Video 2	80	30	2052	20	17
Video 2	>100	30	2052	20	6

The above Table 4.7 express different results are done on the testing video 3 with different threshold values.

Table 4.8 Counting Results with Threshold Values

Input Video	Threshold Value	Frame Rate	No. of Frames	Actual No. of Cars	Detected No. of Cars
Video 4	35	25	120	10	7
Video 4	40	25	120	10	9
Video 4	60	25	120	10	9
Video 4	80	25	120	10	9
Video 4	>100	25	120	10	2

The above Table 4.8 express different results are done on the testing video 3 with different threshold values.

Table 4.9 Counting Results with Threshold Values

Input Video	Threshold Value	Frame Rate	No. of Frames	Actual No. of Cars	Detected No. of Cars
Video 4	35	29	531	4	6
Video 4	40	29	531	4	7
Video 4	60	29	531	4	4
Video 4	80	29	531	4	5
Video 4	>100	29	531	4	0

The above Table 4.9 express different results are done on the testing video 3 with different threshold values. As a result from the Table 4.6, 4.7, 4.8, 4.9 the number of detected car can get as the same as the actual number of car based on the threshold values using on the variant video files' frames.

CHAPTER 5
CONCLUSION

5.1. Conclusion

In this thesis, a method is proposed for detecting and counting of moving cars in captured video by combining the basic background subtraction technique and segmentation using morphological operator. This is done by using the camera images of cars captured on the road and videos taken are converted to the image sequences. Each image is processed separately and the number of cars has been counted. The objective of the study is to gain more knowledge with morphological image operations for traffic analysis in counting the number of cars on the roads from the captured video files through image processing techniques in MATLAB Simulink environment.

5.2. Advantages of the System

A simple way of counting objects from the image sequences is not need towithout use any hardware components. Background subtraction approach uses the current frame and the reference image. Difference between them is greater than the threshold is considered as moving cars.

5.3. Limitations of the System

The input video file must be (.avi) file while using **mmreader** function of MATLAB. In this system, most of the detected cars are white in captured videos. So, the black car detection is disappeared on some frames. Therefore, the threshold value must adjust according to the loaded video file resolution.

5.4. Further Extension

In future work, the detection and counting of moving vehicles can be extended to real time live video feeds. Apart from the detection and extraction, process of recognition can also be done. By using recognition technique, the vehicle can be classified. Recognition technique would require an additional database to match with the given vehicle and the system is designed for further devised to alarming system.

REFERENCES

[1] Chen S. C., Shyu M. L.and Zhang C., "An Unsupervised Segmentation Framework for Texture Image Queries", In the 25th IEEE Computer Society International Computer Software and Applications Conference (COMPSAC), Chicago, Illinos, USA, Oct 2000.

[2] Chen S. C., Shyu M. L. and Zhang C., "An Intelligent Framework for Spatio-Temporal Vehicle Tracking", 4th International IEEE Conference on Intelligent Transportation Systems, Oakland, California, USA, Aug. 2001.

[3] Gupte S., Masoud O., Martin R. F. K. and Papanikolopoulos N. P., "Detection and Classification of Vehicles", In IEEE Transactions on Intelligent Transportation Systems, vol. 3, no. 1, March, 2002, pp. 37– 47.

[4] *N. Lu; J. Wang; Q. Wu; L. Yang (February 2012). An improved Motion Detection method for real time Surveillance.*

[5] *Y. Benezeth; B. Emile; H. Laurent; C. Rosenberger (December 2008).* Review and evaluation of commonly-implemented background subtraction algorithms *(PDF). International Conference on Pattern Recognition. pp. 1–4.*

[6] Toufiq P., Ahmed Egammal and Anurag Mittal, "A Framework for Feature Selection for Background Subtraction", In Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), 2006.

[7] P. Kaewtra Kulpong and R. Bowden, "An Improved Adaptive Background Mixture Model for Real-time Tracking with Shadow Detection", In Proceedings of the 2nd European Workshop on Advanced Video-Based Surveillance Systems, Sept. 2001.

[8] Mrs.P.M. Daigavane and Dr.P.R.Bajaj, "Real Time Vehicle Detection and Counting Method for Unsupervised Traffic Video on Highways" IJCSNS International Journal for Computer Science and Network Security, Vol 10, No 8, August 2010.

[9] Megha C. Narhe and Dr.M.S.Nagmode, “Vehicle Counting using Video Image Processing” IJCAT International Journal of Computing and Technology, Volume 1, Issue 7, August 2014.

[10] S. Y Elhabian, K. M. El-Sayed, “Moving object detection in special domain using background removal techniques state of the art”, Recent patents on computer science, vol-1, pp 32-34, April, 2008.

[11] S. Cheung and C. Kamath, “Robust Techniques for Background Subtraction in Urban Traffic Video”, In Video Communications and Image Processing, SPIE Electronic Imaging, San Jose, January, 2004.

[12] N. Kanhere, S. Pundlik and S. Birchfield, “Vehicle Segmentation and Tracking from a Low-Angle Off-Axis Camera”, In IEEE Conference on Computer Vision and Pattern Recognition”, San Diego, June, 2005.

[13] E. Atkociunas, "Image Processing in Road Traffic Analysis", Vilnius University, 2005.

[14] John M. Gauch. "Image Segmentation and Analysis via Multiscale Gradient Watershed Hierarchies". IEEE. Trans. on Image Processing, 8(1):69 - 79, January 1999.

[15] David G. Lowe “Distinctive Image Features from Scale-Invariant Key points” January 5, 2004.

[16] Il Dong Yun Sang Ho Park and Sang Uk Lee. "Color Image Segmentation Based on 3D Clustering: Morphological Approach". Pattern Recognition, 31(8):1061 - 1076, 1998.

[17] Chuang Gu and Ming-Chieh Lee. "Semantic Video Object Segmentation and Tracking Using Mathematical Morphology and Perspective Motion Model". IEEE. Proceeding, pages 514 - 517, 1997.

[18] Jun Yee Ng, Yong Haur Tay “Image-based Vehicle Classification System”11th Asia-

Pacific ITS Forum & Exhibition, June 2011.

[19] Saroj K.Meher, M.N.Murty “Efficient Method of Moving Shadow Detection and Vehicle Classification”International Journal of Electronics and Communication (AEU), 665-670, 2013.

[20] *M. Piccardi (October 2004).* Background subtraction techniques: a review *(PDF). IEEE International Conference on Systems, Man and Cybernetics. 4. pp. 3099–3104.* doi*:*10.1109/icsmc.2004.1400815.

[21] *B. Tamersoy (September 29, 2009).* "Background Subtraction – Lecture Notes" *(PDF). University of Texas at Austin.*

[22] R. Chellappa “Vehicle detection and tracking using acoustic and video sensors”, 2004.

[23] *B. Patel; N. Patel (March 2012). Motion Detection based on multi-frame video under surveillance systems. Vol. 12.*

[24] Charles R. Giardina and Edward R. Dougherty. “Morphological Methods in Image and Signal Processing”. Prentice-Hall, Inc., 1988.

[25] Petros Maragos. "Tutorial on Advances in Morphological Image Processing and Analysis". Optical Engineering, 26(7):623 - 632, July 1987.

[26] Malay K. Kundu Bhabatosh Chanda and Y. Vani Padmaja. "A Multiscale Morphologic Edge Detector". Pattern Recognition, 31(10):1469{1478, 1998.

[27] Xudong Song and Yrjo Neuvo. "Robust Edge Detector based on morphological fillters". Pattern Recognition Letters, 14:889{894, 1993.

[28] Philippe Salembier. "Morphological Multiscale Sgementation for Image Coding". Signal Processing, 38:359 - 386, 1994.

[29] Mary L. Comer and Edward J. Delp. "An Empirical Study of Morphological Operators in Color Image Enhancement". Proceedings of the SPIE Conference on Image Processing Algorithms and Techniques III, 1657: 314 - 325, 1992.

[30] B. Coifman "A Real-Time Computer Vision System for Vehicle Tracking and Traffic Surveillance" University of California, 1998.

[31] https://www.mathworks.com/help/images/structuring-elements.html

[32] *C. Stauffer; W. Grimson (August 1999).* Adaptive background mixture models for real-time tracking. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition.* ***2****. pp. 246–252*

PUBLICATION

[1] Hnin Wit Yi Oo and Zin May Aye, "Implementation of Car Counting System Using Morphological Image Processing", Parallel and Soft Computing Conference, University of Computer Studies, Yangon, Myanmar, 2017