UNIVERSITY OF MINNESOTA

This is to certify that I have examined this copy of a master's thesis by

Sumalatha Kuthadi

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Name of Faculty Advisor

Signature of Faculty Advisor

Date

GRADUATE SCHOOL

DETECTION OF OBJECTS FROM HIGH-RESOLUTION SATELLITE IMAGES

A thesis
submitted to the faculty of the graduate school
of the University of Minnesota
by

Sumalatha Kuthadi

In the partial fulfillment of the requirements
For the degree of
Master of Science

May 2005

Department of Computer Science
University of Minnesota Duluth
Duluth, Minnesota 55812
U.S.A.

Abstract

In the past three decades satellite imagery has been used successfully for weather, geographical and geological applications. With the advance of technology, more sophisticated sensors provide higher resolutions, and with faster computer systems, the use of satellite imagery has opened the fields of exploration and application.

In the present thesis we apply image-processing techniques to high-resolution satellite images to extract some of the detectable features in this type of images. Segmentation techniques based on thresholding are use to extract highways and vehicles from images containing roadways scenes. Color properties are used to extract vegetation areas from cities and fields scenes.

Results of this work could be use to help transportation agencies in the study of traffic density and trends across large geographic areas. The detection of green areas could help scientist to study deforestation and changes in vegetation.

Acknowledgements

I would like to thank Dr. Rocio Alba-Flores, Dr. Douglas Dunham and Dr. Gary Shute for their direction, assistance and guidance. In particular, I thank Dr. Rocio Alba-Flores for her recommendations and suggestions, which have been invaluable for the project and also for giving me financial aid.

I am also very thankful to Dr. Fernando Rios-Gutierrez for giving me invaluable feedback on my thesis. I would like to thank Prashant Jain for his cooperation when we worked as a team.

I am also very thankful to faculty and staff of Department of Computer Science at University of Minnesota Duluth. I would like to thank all my friends, especially my roommates for encouraging me.

Finally, I would like to thank my parents and family members for their priceless love, encouragement and inspiration.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

1 Introduction

Satellite data has been successfully used since the 1970's. The principal applications have been for weather prediction, to monitor global environment conditions, and geographical and geological applications. Satellite images provide detail information about earth's surface. The benefits of satellite imagery include covering large areas; frequent re-visits to almost any part of the globe, regardless of its remoteness; the ability to collect data unhindered by local air traffic [1].

Previously, satellite images were very expensive and they were used for military applications only, such as threat monitoring and assessment. As several commercial satellites have been launched, such as IKONOS, QUICK BIRD, IRS etc., they provide global, accurate, high-resolution images to individuals, organizations and governments. In this way, satellite images have become less expensive and the areas of application have increased enormously.

Satellite images of different spatial resolutions are commercially available. Images with high-resolution data with ground pixel sizes of less than 5m provide detail information about the Earth's surface and small objects, such as buildings, streets, and trees can be displayed in great details. High-resolution images are useful for applications such as transportation network mapping, disaster preparedness, urban planning, precision farming, and telecommunications. On the other hand, low-resolution satellite images, with ground resolution greater than 10m, are useful for applications like environmental assessment, regional mapping, forestry management, widespread disaster assessment, and urban monitoring.

Satellite images provide a perfect view of a desired region. They can also be used to determine elevation (of mountains, hills and even buildings) and topographical features, which make it easier to get a sense of what the land is like. Satellite images are also useful in agricultural fields where farmers can monitor the health of their crops. Scientists can look and study environmental changes that could affect the globe in the future. City planners can study, monitor, and plan the development of new housing communities with precision. In transportation, satellite data could be used for traffic study, and help in the planning on new roadways. In case of disasters, such as earthquakes or fires, government agencies could use satellite data to plan evacuation routes.

The main aim of this research is to use high-resolution satellite images to:

a) Detect, classify and count vehicles

b) Detect highways

c) Detect vegetation

To achieve this goal, we developed and analyzed different image segmentation algorithms for the detection of vehicles, highways, and vegetation. We used the MATLAB Image Processing Tool Kit for developing the algorithms.

The order of this thesis is as follows. In Chapter 2, background and literature review is presented. Chapter 3 deals with ‘Detection, Classification and Counting of Vehicles’. In Chapter 4 the ‘Detection of Highways’ is explained. Chapter 5 describes the techniques used for the ‘Detection of the Vegetation’. For the first two parts ‘Detection of Vehicles’ and ‘Detection of Highways’ intensity images were used. For the ‘Detection of the Vegetation’, the RGB color space was used.

CHAPTER 2: BACKGROUND ON IMAGE PROCESSING

2.1 Introduction

This chapter starts with the definition of digital image and some of the basic types of images, followed by a brief description of satellite images. Then the concepts of image segmentation and thresholding techniques are introduced, which are the main techniques used to achieve the goals of this research. The MATLAB Image Processing toolbox is used for the development of the algorithms and a brief description of the use of this toolkit is given too.

2.2 Digital Images and Basic Types

An image can be defined as a two-dimensional function, ***f(x,y)*** where ***x*** and ***y*** are the spatial coordinates and the amplitude value ***f*** represents the intensity (or color) of the image at that point (pixel). When ***x, y***, and ***f*** are discrete values, we have a digital image. Digital images are composed of ***pixels*** arranged in a rectangular array with a certain height (rows) and width (columns). Each pixel may consist of one or more bits of information (8 bits being the most common), representing the intensity of the image at that point (or color information encoded as RGB triples). The following are the four basic types of digital images:

- RGB (True Color) Images
- Intensity (Gray Scale) Images
- Binary Images
- Indexed Images

2.2.1 RGB Images

It is possible to construct (almost) all visible colors by combining the three primary colors Red, Green and Blue, because the human eye has only three different color receptors, each of them sensible to one of the three colors. Different combinations in the stimulation of the receptors enable the human eye to distinguish approximately *350,000* colors. A RGB color image is a multi-spectral image with one band for each color red, green and blue, thus producing a weighted combination of the three primary colors for each pixel [2]. Figure 2.1 shows a RGB image.

Figure 2.1: RGB Image

2.2.2 Intensity Images

A grayscale (or gray level) image is an image in which the only colors are shades of gray. In `gray' color the red, green and blue components all have equal intensity in RGB space, and so it is only necessary to specify a single intensity value for each pixel, as opposed to the three intensities needed to specify each pixel in a full color image. Often, the grayscale intensity is stored as an 8-bit integer giving 256 possible different shades of grey from black to white [2]. Figure 2.2 shows a gray scale image.

Figure 2.2: Gray Scale Image

2.2.3 Binary Images

Binary images are images whose pixels have only two possible intensity values. They are normally displayed as black and white. 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 the 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 that is to be thresholded, this *polarity* might be inverted, in such case the object is displayed with 0 and the background is with a non-zero value [2]. Figure 2.3 shows a binary image.

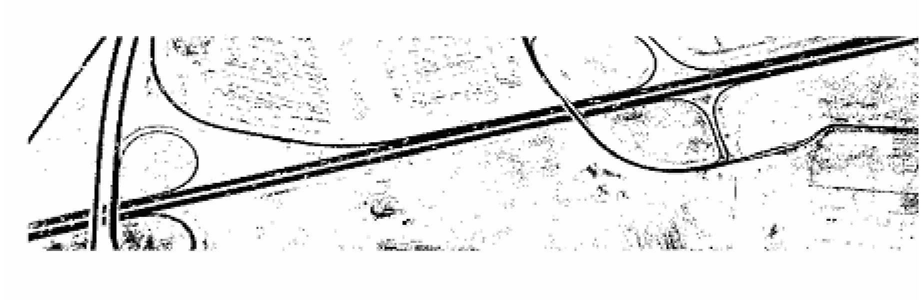

Figure 2.3: Binary Image

2.2.4 Indexed Images

Indexed images are visually similar to RGB images but the way of representing them is different. An indexed image consists of a data matrix, `X`, and a color map matrix, `map`. The color map matrix `is` an m-by-3 array containing values in the range [0, 1]. Each row of `map` specifies the red, green, and blue components of a single color. An indexed image uses direct mapping of pixel values to color map values. The color of each image pixel is determined by using the corresponding value of `X` as an index into `map`. The value 1 points to the first row in `map`, the value 2 points to the second row, and so on [2].

2.3 Image Segmentation

Satellite images are available in both intensity and RGB types. In our research, RGB images are used for detection of vegetation and intensity images are used for detection of vehicles and highways. Image Segmentation is the technique, which is used for the detection of vehicles, highways and vegetation from the satellite images.

Objects in an image are detected by extracting the foreground objects. That is, image being divided as foreground and background. Dividing an image into its constituent regions or objects is known as ***Image Segmentation***. The level to which subdivision is carried depends on the problem being solved. In our case, segmentation is achieved when the objects of interest are isolated.

Image segmentation can be done using common properties of the pixels. One of these properties is the pixel intensity [3]. Many images can be characterized as containing the objects of interest with a similar uniform intensity placed against a background of different intensities. For these cases, image segmentation can be done by using the thresholding techniques, which uses the intensity property of the pixels.

2.4 Thresholding Techniques

2.4.1 Threshold Technique for Vehicles

Thresholding is one of the powerful methods used for image segmentation. It is useful in discriminating foreground from the background. In many practical applications, in an intensity image that is considered for vehicle detection, foreground and background have different range of gray levels. Such images usually possess a bimodal histogram [3]. Figure 2.4 shows a bimodal histogram.

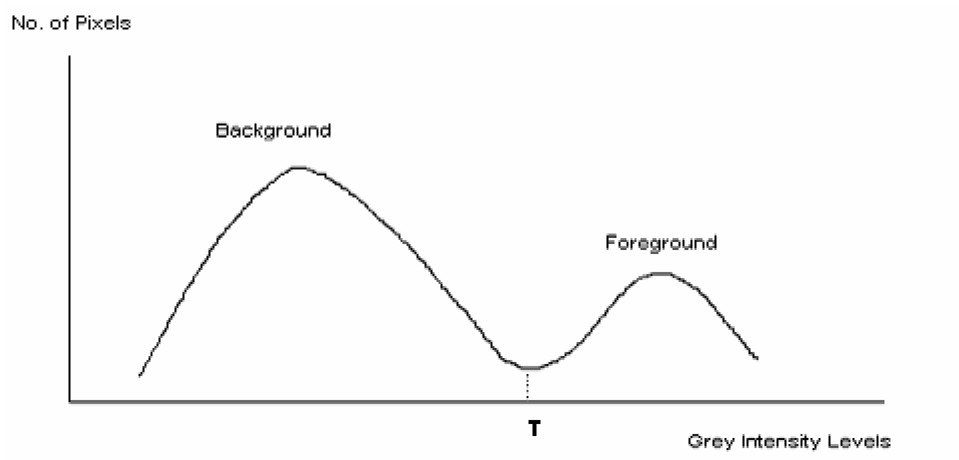

Figure 2.4: Bimodal Histogram

Pixels with intensities less than **T** belong to background and the pixels with intensities greater than **T** belong to objects. To extract the objects, the intensity image f (x, y) is converted to a binary image g (x, y) using **T**. For a pixel at a point (x, y), if its intensity f (x, y) > T, then it is considered as an Object Pixel (1) else Background Pixel (0). A thresholded image is defined as

$$g(x, y) = 1 \quad \text{if } f(x, y) > T$$
$$= 0 \quad \text{if } f(x, y) < T$$

Thus, pixels labeled 1 correspond to objects and the pixels labeled 0 correspond to the background. Chapter 3 provides detail information of calculating the threshold T for vehicle detection, the problems that occur while calculating the T, and the way the vehicles are detected with examples.

2.4.2 Threshold Technique for Highways

On a gray scale from 0 to 255: 0 represents black color and 255 represents white. As we move from 0 to 255, the brightness of gray color increases. We can think of gray shade as a mixture of white and black colors. Black color (0) has no content of white color and white color (255) has no content of black color. As we move from black to white, content of white color increases and black color decreases. The objects present in an intensity image can be divided into three categories based on their shade on gray scale. The three categories are:

- Dark Gray Shade Objects
- Medium Gray Shade Objects
- Bright Gray Shade Objects

The category of the highways depends on the pavement material, atmospheric conditions and surrounding objects. In reality, highways are dark in color. If the atmospheric conditions are favorable and the quality of the satellite image is good, then the highways in the image are dark in color. Otherwise, they have average gray shades. In cases like flyovers, bridges, highways are bright. Figures from 2.5 to 2.7 show examples of dark, medium and bright gray shade highways.

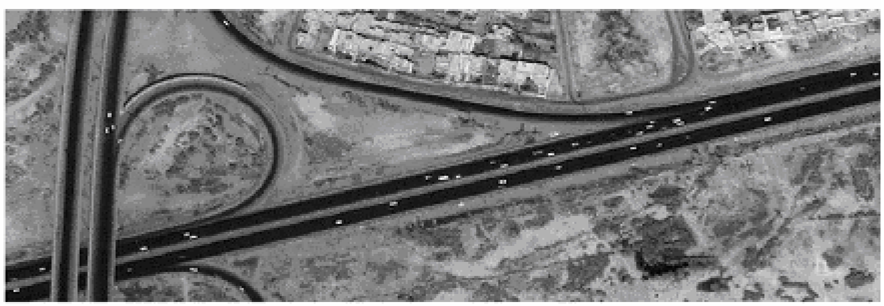

Figure 2.5: Dark Gray Shade Highway

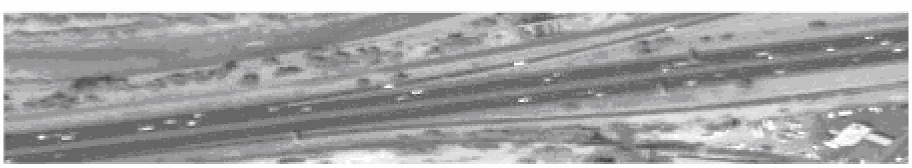

Figure 2.6: Medium Gray Shade Highway

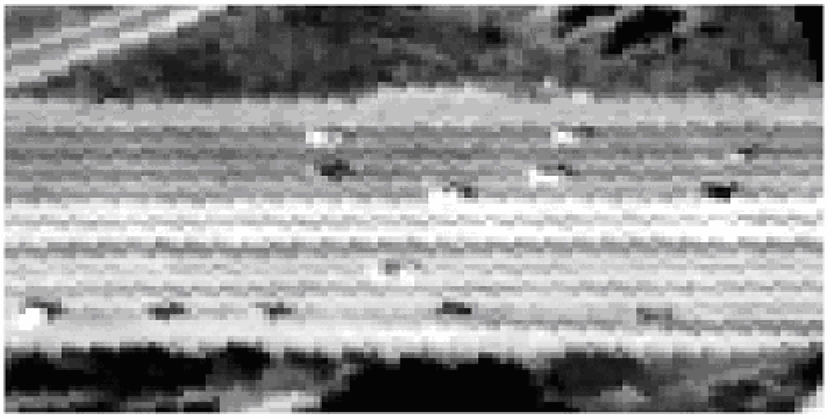

Figure 2.7: Bright Gray Shade Highway

In Figure 2.7, highway is brighter than the surroundings, where as in other two images 2.5 and 2.6, surroundings are brighter than the highways. Since we are interested in detecting highways, which are usually not bright, we are focusing on detecting highways, which are in dark color or in medium shades of gray. Chapter 4 presents the information regarding calculating the threshold T for highway detection and the way T is used for the detection of highways with examples.

2.4.3 Threshold Technique for Vegetation

All trees, plants have some shade of green. We know that each color is a mixture of basic colors Red, Green and Blue. Therefore the RGB combinations of green color are used for vegetation detection. In MATLAB, RGB image is stored in an m-by-n-by-3 data array that defines red, green, and blue color components for each individual pixel. Two shades of green, dark and light, are considered while detecting the vegetation. Figure 2.8 shows a true color test image, which is an ‘Ikonos 1-m Resolution Satellite image’. Chapter 5 provides detail information about the conditions that are satisfied by the R, G and B components of shades of green color and how these conditions are used for the detection of trees.

Figure 2.8: RGB Image of Vegetation

2.5 MATLAB Image Processing Toolkit

MATLAB is a high-level technical computing language and interactive environment for algorithm development, data visualization, data analysis, and numerical computation. Using MATLAB, you can solve technical computing problems faster than with traditional programming languages, such as C, C++, and Fortran [2].

This toolbox provides a complete suite of digital image processing and analysis tools. It uses the broad set of algorithms provided or modifies these algorithms to develop customized tools to solve unique image processing problems.

The Image Processing Toolbox is a collection of functions that supports a wide range of image processing operations, including:

- Spatial image transformations
- Morphological operations
- Neighborhood and block operations
- Linear filtering and filter design
- Transforms
- Image analysis and enhancement
- Image registration
- Deblurring

Chapter 3: Detection of Vehicles from High-Resolution Satellite Images

3.0 Introduction

As information technologies and advances in electronics continue to revolutionize all aspects of our modern-day world, from our homes and offices to our schools and even our recreation, they are also being applied to transportation network. Transportation infrastructure of highways, streets and bridges and the number of vehicles, including cars, buses, trucks and trains are increasing rapidly. It is important to use the existing facilities intelligently to provide better transportation facilities. Therefore, it is required to automate the management of traffic flow by intelligent traffic control and traffic guidance systems.

Traffic-related data play an important role in urban and spatial planning, e.g., for road planning and for estimation of air and noise pollution. Therefore, an algorithm that automatically detects and counts vehicles from satellite images would effectively support traffic-related applications [4].

Different techniques like installing cameras at fixed locations, weigh-in motion sensors on the pavements are being used for vehicle detection. Though they are well developed systems, they have disadvantages like cameras cannot observe the spatial progression and movement of traffic beyond the field of view [5], sensors can be affected by weather and traffic stress, maintenance of sensors may reduce the pavement lifetime etc. Using satellite images for vehicle detection can eliminate these problems. They can cover wide areas and images of an area can be taken frequently. Using satellite images also have problems like, atmospheric conditions that affect the visibility of vehicles in the images; daytime images provide better information than at night, etc.

Spatial resolution is an important aspect to be considered while using the satellite images for vehicle detection. Satellite images are available in different resolutions. Images with high spatial resolution provide detail information of an area. Many commercial satellites like IKONOS, QUICKBIRD, and IRS, have been launched in recent years. These satellites are capable of providing different spatial resolutions [6][7][8]. Very high spatial resolutions are in the order of 1-meter.

The aim of this research is to develop efficient algorithms for automatic detection, classification and counting of the vehicles from high-resolution satellite images. The images used for this project are 1-m panchromatic images from the IKONOS satellite. Two different Image Segmentation algorithms, which are based on Multiple Thresholds and Otsu Threshold, are developed for vehicle detection. These algorithms were tested on several images and the results were analyzed to determine which algorithm gives better results under which conditions.

In Section 3.1 we provide a literature review referring to several authors that have been working in similar projects. Section 3.2 gives a detailed explanation of the two algorithms used for the vehicle detection. It also explains how the classification and counting of the vehicles is performed. Section 3.3 presents results of both algorithms on several test images. In section 3.4 conclusions about the results are given.

3.1 Literature Review

In the literature several authors have proposed different approaches for car detection, most of them using aerial images. For example in [4], [9] the authors proposed an approach to detect and count cars using aerial images from urban scenes. They proposed a 3D car model for the detection and counting of vehicles. In [10] vertical view aerial images of urban areas are used, and the approach to detect the cars is based on a generic car model (shape of the boundary of the car, and boundary of front windshield), then the car model is used to predict if the shapes detected in the images are cars or not. In [11] a vehicle is modeled as a rectangle of a range of sizes, and convolution with edge masks are used to extract the four sides of the rectangular boundary. In [12] a model is created by example images of cars and their statistics are recorded in vectors; then by computing the features vector from image regions and testing them against the statistics of car models, the vehicles are detected. In [13] the authors extract and group image features to construct structures similar to a car model.

The goal of this work was to develop efficient algorithms for the detection, classification, and counting of vehicles on highways using high-resolution satellite images. Because our focus is on highway scenes, and not complex urban areas, our approach is much simpler, it is based on gray value intensities and thresholding, and it does not use any car models. The data used for this project are 1-m panchromatic images from the IKONOS satellite and image-processing techniques are used for the implementation of the algorithm. The algorithm is divided in two parts. The first part detects bright vehicles and utilizes two methods, Multiple Thresholds and Otsu Threshold. The second part detects the dark vehicles based on the Otsu method. The MATLAB™ software was used for the implementation of the algorithms. The test images were obtained from the World Wide Web sites listened in references [6][7][8].

3.2 Algorithm to Detect, Classify and Count Vehicles

As we mentioned in Section 2.4.1, detection of vehicles in an image is achieved by extracting the foreground objects. Dividing an image into foreground and background is called ***Image Segmentation***. Image segmentation is achieved by using the common properties of the pixels. In this research work, intensity property of pixels is used in developing image segmentation algorithms.

Image segmentation algorithms generally are based on one of the two basic properties of intensity values: discontinuity and similarity. In the first criteria, the image is partitioned when there is an abrupt change in the intensities of neighboring pixels, such as edges in an image. In the second criteria, image is partitioned into regions that are similar according to some predefined conditions [3]. In this project we used the second criteria i.e., similarity in developing the Image Segmentation algorithms.

3.2.1 Thresholding Technique

Thresholding is one of the powerful methods used for image segmentation. It is useful in discriminating foreground (objects of interest) from the background. In many cases intensity images have different range of grey levels for the foreground and background [3]. Such images usually possess a bimodal histogram as shown in figure 3.2.1.

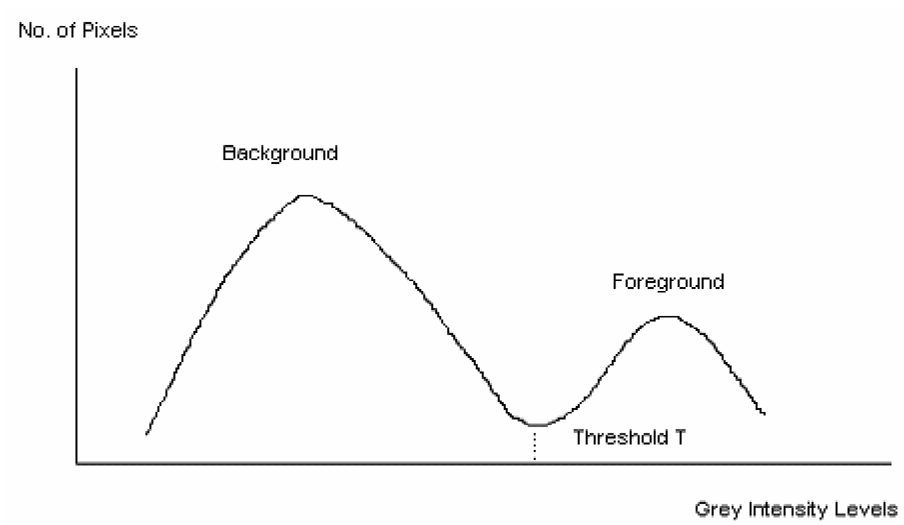

Figure 3.2.1: Bimodal Histogram

Pixels with intensities less than **T** (threshold value) belong to background and the pixels with intensities greater than **T** belong to objects (or vice versa). To extract the objects, the intensity image f(x,y) is converted to a binary image g (x, y) by using **T**. For a pixel at a point (x,y), if its intensity f (x, y) > T, then it is considered as an Object Pixel else Background Pixel. A thresholded image is defined as

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

Thus, pixels labeled 1 correspond to objects and the pixels labeled 0 correspond to the background.

3.2.2 Multiple Thresholding

The ideal case is a bimodal shape histogram, but such histograms are usually unavailable in real applications. In general, an intensity image have to be divided into several sub-ranges to perform thresholding, but these ranges usually overlap one with another and this makes thresholding difficult.

In this project the images that are used for vehicle detection present a histogram of the type shown in figure 3.2.2. From this figure we can see that two peaks are clearly defined. The peak on the left (dark side) represents the background. The peak on the right (bright side) represents pixels of the foreground (vehicles). Notice that the histogram presents an overlapping region between the background and foreground pixels. This overlapping means that some of the objects or regions of background like lane markers have similar intensities as foreground objects (vehicles). Therefore to detect vehicles only, multiple thresholds are used. Sections 3.2.3 and 3.2.4 describe the procedures followed to compute three different threshold values, and the way that they are used to detect the vehicles.

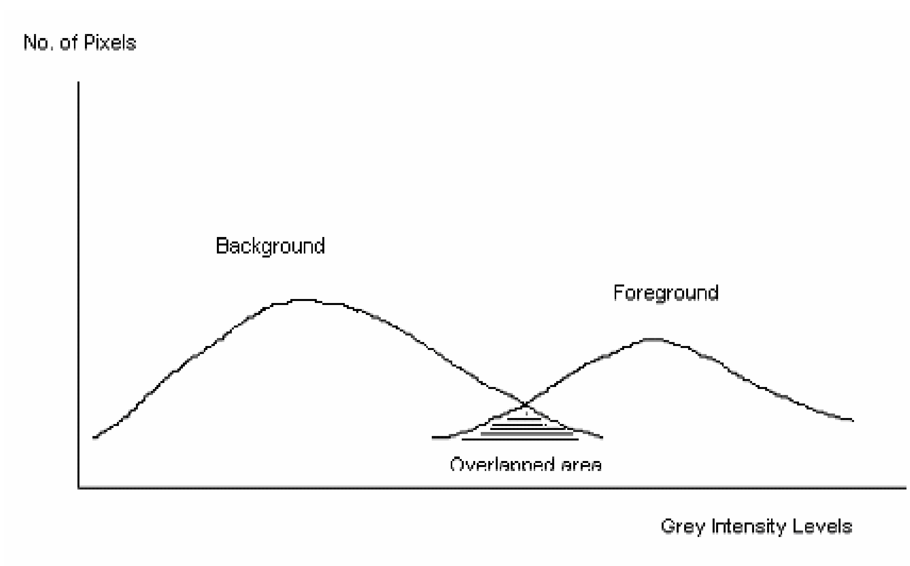

Figure 3.2.2: Bimodal Histogram with an overlapped area

To achieve our goal of detecting all the vehicles on the test images, we have divided the algorithm in two parts:

- Detection of vehicles, which are bright with respect to the background.
- Detection of vehicles, which are dark with respect to the background.

Figure 3.2.3 gives an example of dark and bright vehicles in a test image. In the following sections both parts are explained in detail.

Figure 3.2.3: Bright and Dark Vehicles

3.2.3 Detection of the Bright Vehicles

Two different methods are proposed to identify the bright vehicles:

- **Multiple Thresholds**
- **Clustering by Otsu Method**

Multiple Thresholds

By analyzing several of the sample images in the database, we can say that the intensity values of bright vehicles are greater than the intensities of the background, and sometimes there is a region where they overlap. Because of the overlapping, some objects or regions on roads, such as lane markers and road dividers may have intensity values similar to the intensity of some of the bright vehicles. Also, each bright vehicle may not have same range of intensity because of atmospheric conditions and colors of vehicles. Therefore, to identify only the vehicles and to avoid the detection of irrelevant objects like lane markers, three different thresholds **T1, T2,** and **T3** are used.

Computation of the Threshold Values

In MATLAB, an intensity image can be stored in a two dimensional matrix. Let this matrix be M1. Each element of matrix M1 corresponds to the intensity of each pixel of the image. A one-column matrix, M2 is formed by using M1. Each row of M2 contains the **maximum intensity** of the corresponding row of intensities of M1. We know that in an intensity image, on a highway, bright vehicles have maximum intensity levels than any other objects. Therefore we are considering the **maximum intensity** in each row of M2 to calculate the three thresholds T1, T2, and T3.

M2 is calculated by using M1

```
M2:    for each row i in M1
           M2[i] = maximum_intensity[M1, i]
```

T1, T2 and T3 are calculated by using M2

T1: $T1 = \text{Mean}[M2]$

T2: $T2 = \text{Minimum}[M2]$

T3: $T3 = \text{Mean}[T1, T2]$

Thresholds T1, T2, and T3 are used to convert the test image to three different binary images Image1, Image2, and Image3. For a pixel at coordinates (x, y), if its intensity I (x, y) > T, then it is considered as an Object Pixel (1) else Background Pixel (0). A binary image is defined as:

$$\text{Image}(x, y) = 1 \quad \text{if } f(x, y) > T$$
$$= 0 \quad \text{if } f(x, y) < T$$

Figure 3.2.4 shows a highway in Oklahoma. To detect the vehicles by using the proposed algorithm, the user selects first a region of interest. This region of interest has a rectangular shape, which is selected using the mouse, and should contain only one-way on the highway. If the highway under study is not vertical or horizontal on the image, the user should rotate the image first to make easier the selection of the region under study. Also, if the selected region of the highway contains two-ways, factors such as road dividers, sign boards, etc, will affect the result of the algorithm. Therefore to obtain better results, the region under study is selected in such a way that it contains only one-way highway. Selection of road shoulders and ramps should be avoided, and the regions under study should contain only the highway with the vehicles. Figure 3.2.5 shows an example of the selection of a region under study. This region is the one highlighted in Figure 3.2.4. The selected one-way has ten bright vehicles and no dark vehicles.

Figure 3.2.4: Highway in Oklahoma

Figure 3.2.5: One-way Selected

By applying the Multiple Threshold algorithm on the test case, 199.67, 149.00 and 174.34 are obtained as thresholds. Figure 3.2.6 shows the three binary images Image1, Image2 and Image3 obtained by using these thresholds.

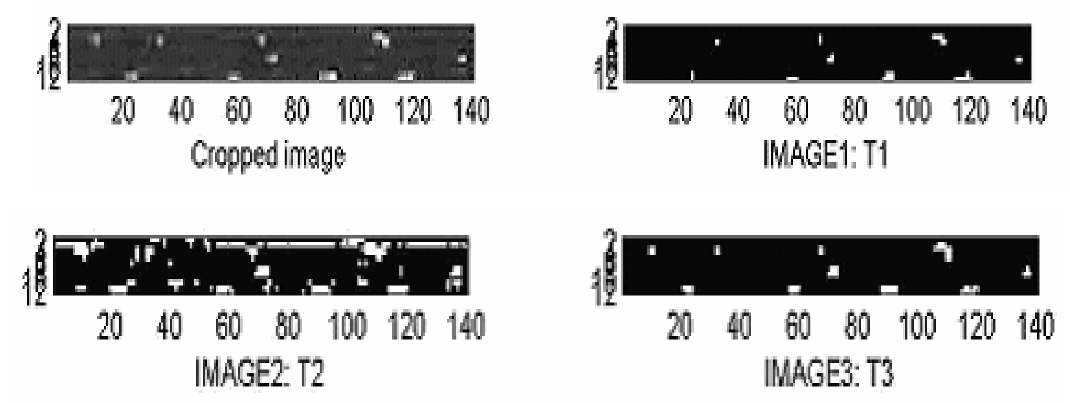

Figure 3.2.6: Binary Images obtained by using T1, T2 and T3

Reducing the detection of irrelevant objects such as lane markers and road dividers

From this example (Figure 3.2.6) we can see that each threshold detects some or all vehicles in the region under study, but lane markers and road dividers are detected by some of the threshold values too (due to the overlapping of the intensities of foreground and background). To avoid or reduce the detection of irrelevant objects such as lane markers and road dividers, logical operations are performed among the binary images. For the case of detecting bright vehicles, best results are obtained by taking common objects from pairs of the binary images generated by the thresholds T1, T2, and T3. This means, common objects from the pairs: (Image1, Image2), (Image2, Image3), and (Image1, Image3) are taken.

We can observe from Figure 3.2.6 that Image1 contains some of the vehicles, Image2 contains all vehicles and also lane markers, road dividers, etc., and Image3 contains all the vehicles present in the test image. To detect all the bright vehicles, and to reduce the detection of irrelevant objects, the LOGICAL AND operation is performed to extract the common objects among the binary images. By applying the AND operation on the three combinations of binary images, three new binary images: New_Image1, New_Image2, and New_Image3, are formed which are shown in Figure 3.2.7.

New_Image1 = bitwiseAND [Image1, Image2]
New_Image2 = bitwiseAND [Image2, Image3]
New_Image3 = bitwiseAND [Image1, Image3]

Figure 3.2.7: Common Objects taken from pairs of binary images

Detecting all the vehicles on the test image

The resultant images New_Image1, New_Image2, and New_Image3 that are obtained after eliminating the irrelevant objects will contain only vehicles. Each of these binary images may contain some or all vehicles. These three images are added to obtain a single image with all the vehicles. The addition of images is achieved by using the LOGICAL OR operation.

From Figure 3.2.7 we can observe that New_Image1 contains some of the vehicles, New_Image2 also contains some of the vehicles and New_Image3 contains all the vehicles. Figure 3.2.8 shows the final result, that is, all the bright vehicles detected by the Multiple Thresholds Vehicle Detection Algorithm.

Final_Result = bitwiseOR (New_Image1, New_Image2, New_Image3)

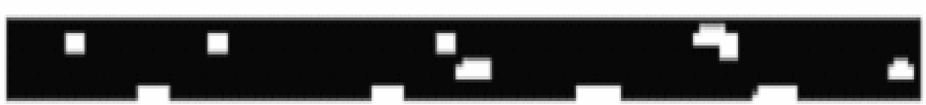

Figure 3.2.8: Bright Vehicles Detected By Multiple Threshold Technique

Otsu Threshold

The Otsu threshold [14] uses class separability and maximize the between-class variance to find an optimal threshold value **k*.** This threshold value is used to extract objects from their background. MATLAB has a built-in function that evaluates the Otsu threshold **k*.** Applying directly the Otsu threshold to the test image, will detect the bright vehicles, but also some of the lane markers and road dividers that are present on the highways. To reduce the problem of lane markers and road dividers, a preprocess step is applied first. The preprocess step involves the application of a sliding neighborhood operation to the test image. The sliding neighborhood operation consists of assigning to each pixel of the test image, the maximum intensity of its neighborhood (this is a rectangular area of 3-by-3 pixels, being the center pixel the one that is being processed by the operation).

Sliding Neighborhood Operation:

```
f = inline('max(x(:))');
slide_image = nlfilter (test_image, [3 3] ,f);
```

nlfilter(testimage, [3 3], f) applies the function ‘f’ to each 3-by-3 sliding block of the test image.

Sliding Neighborhood Operation: A sliding neighborhood operation is an operation that is performed a pixel at a time, with the value of any given pixel in the output image being determined by the application of a function to the values of the corresponding input pixel's neighborhood. A pixel's neighborhood is some set of pixels, defined by their locations relative to that pixel, which is called the center pixel. The neighborhood is a rectangular block, and as you move from one element to the next in an image matrix, the neighborhood block slides in the same direction [2].

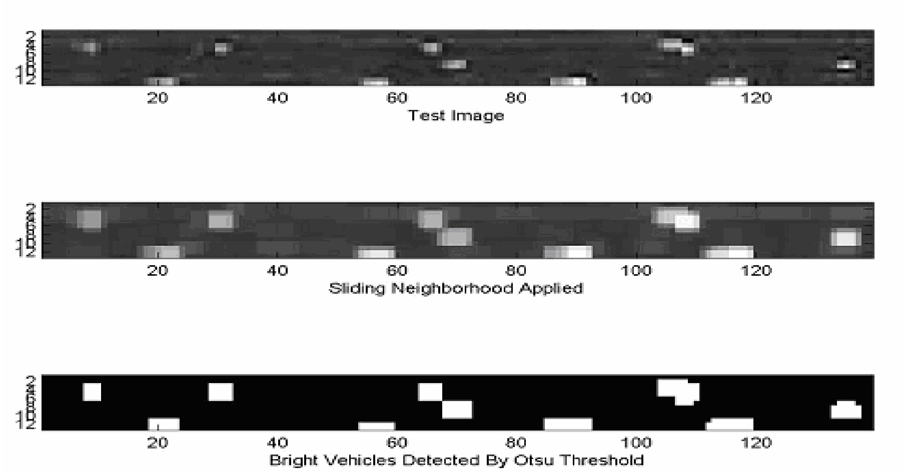

Figure 3.2.9 Detection of the Bright Vehicles Using the Otsu Threshold

By applying this sliding neighborhood operation to the test images, the bright pixels corresponding to large objects, such as vehicles, become brighter, but the bright pixels corresponding to irrelevant objects such as lane markers stay about the same bright. This preprocessing step will help to highlight the vehicles, and dim some of the irrelevant objects such as the lane markers. After applying the sliding neighborhood operation, the Otsu Threshold is computed, and a binary image is generated. Figure

3.2.9 shows a test image, the resulting image after the sliding neighborhood operation is applied, and the binary image generated after the computation of the Otsu Threshold. Note that the Otsu Threshold detected all the bright vehicles.

3.2.4 Detection of Dark Vehicles

For the detection of dark vehicles, the Otsu Threshold is used. Before applying the Otsu Threshold, a sliding neighborhood operation is applied to the test image. As we want to detect dark vehicles, each pixel is assigned with the **minimum** intensity of its neighboring pixel in a rectangular neighborhood of a 3-by-3 matrix. As a result, dark vehicles become darker when compared to the background. After applying the sliding neighborhood operation, the Otsu Threshold is used to convert the test image to a binary image.

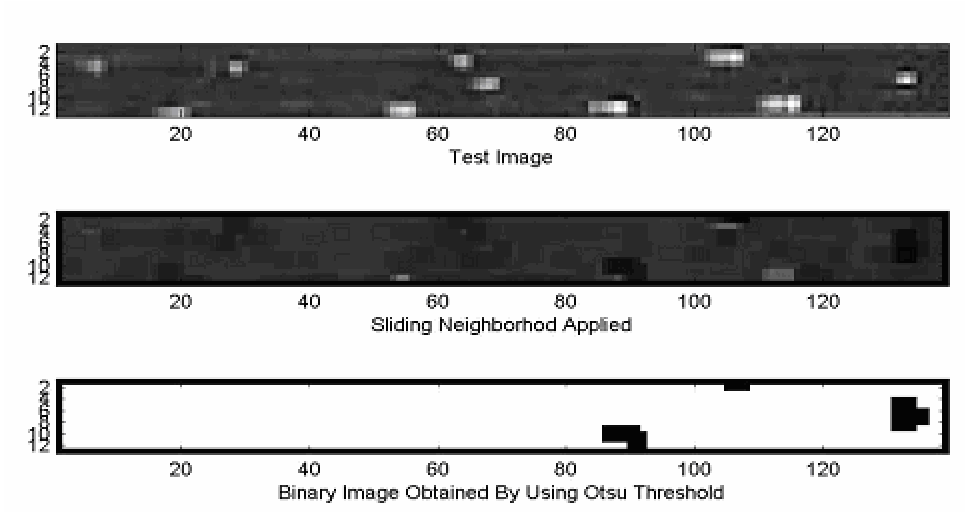

Figure 3.2.10: Dark vehicles Detected by Otsu Threshold

Continuing with the example of Figure 3.2.4 b), Figure 3.2.10 shows a) the test image, b) the image obtained after applying the sliding neighborhood operation, and c) the binary image obtained after using the Otsu Threshold. Note that the binary image displays all the dark vehicles as black patches. To display the dark vehicles as white patches, the complement of the binary image is taken. Figure 3.2.11 shows the complement of the binary image.

Figure 3.2.11 Negative of figure 3.2.10.c)

Looking carefully at the original test image (Figures 3.2.4 and 3.2.19a) we can notice that the image does not contain dark vehicles. The white patches in the above binary image (Figure 3.2.11) represent the shadows of vehicles in the test image. To avoid considering vehicle's shadows as dark vehicles, the results of algorithms Detection of Bright Vehicles and Detection of Dark Vehicles are added. This is carried out by performing a LOGICAL OR operation on the two binary images: the first one obtained by the bright vehicles detection algorithm (Otsu or Multiple Thresholds technique) and the second one obtained by the dark vehicles detection algorithm (Otsu Threshold technique). As a result of the addition, shadows of the bright vehicles, which are very close to the bright vehicles, are combined as single vehicles. This process is presented in Figure 3.2.12, where the first image shows the binary image resulting after applying the bright vehicles detection algorithm (Otsu Threshold); the second image shows the binary image obtained after applying the dark vehicles detection algorithm; and the third image shows both images combined to give the final result.

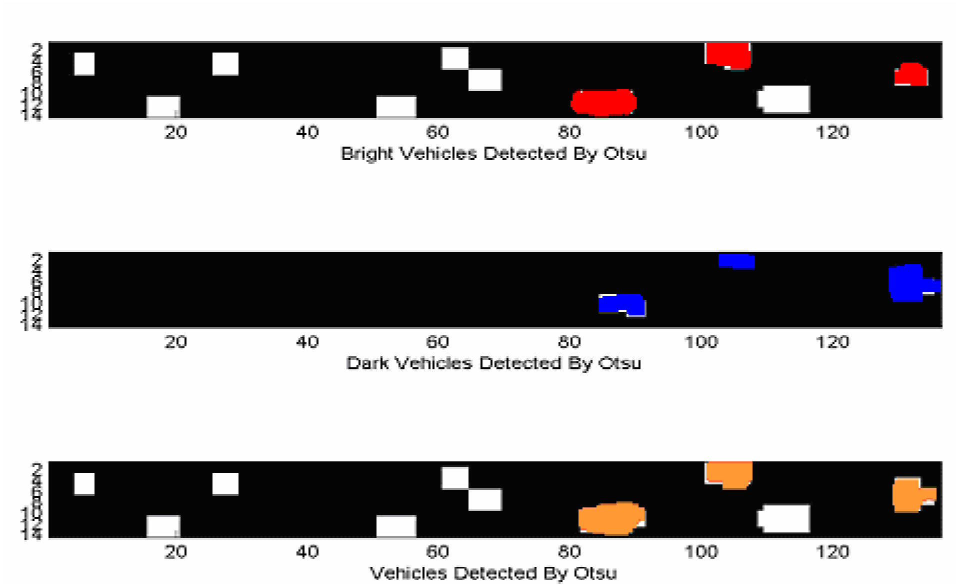

Figure 3.2.12: Application of the vehicle detection algorithm to the test image in Figure 3.2.10

a) Bright vehicles detected by "Bright vehicle detection algorithm (Otsu Threshold)"

b) Dark vehicles detected by "Dark vehicle detection algorithm"

c) Bright and Dark vehicles

In Figure 3.2.12 a), the vehicles colored in red have shadows, which are colored in blue in Figure 3.2.12 b). In Figure 3.2.12 c), the orange patches are the vehicles combined with their shadows. If the test image has dark and bright vehicles, then adding the two binary images (the one with the bright vehicles and the one with the dark vehicles), would not affect the final detection of the vehicles as long as the detected vehicles in each binary image are not very close to each other.

In cases where dark and bright vehicles are placed very close, it is possible that a bright vehicle could be combined with a dark vehicle and counted as a single vehicle, giving an error in the result. This is also true when two or more bright or dark vehicles are very close to each other, then the final result would combine this group of very close vehicles as a single vehicle, causing an error in the final count.

3.2.5 Classification of Vehicles as Cars and Trucks

To classify the vehicles, as cars or trucks, three parameters: width, height, and area, of the detected vehicles are considered. First the average of each of these parameters is computed. This is carried out by taking into account all the detected vehicles on the test image. Then, the three parameters for each of the detected vehicles are compared with the average values. Any vehicle with width, height, and area greater than the average values is considered as a truck otherwise it is a car. The algorithm for the classification is as follows:

```
Mean_Area = Mean [All Vehicles Areas]
Mean_Width = Mean [All Vehicles Widths]
Mean_Height = Mean [All Vehicles Heights]

Car_Count = 0;
Truck_Count = 0;

for i = 1 to no_of_vehicles
    If (Vehicle[i].Area > Mean_Area)
        If (Vehicle[i].Width > Mean_Width)
            If (Vehicle[i]. Height > Mean_Height)
                Truck_Count = Truck_Count + 1;
    else
                Car_Count = Car_Count + 1;
```

Special Case: At a given time, it is possible that a road may contain only cars (no trucks). In this case, there is a possibility that some cars may be classified as trucks.

3.2.6 Counting Of Vehicles

To compute the final count of vehicles in the test image (number of cars and number of trucks), the MATLAB function: BWLABEL is used. This function returns the number of connected objects in a binary image.

3.2.7 Special Cases that will affect the Results

To test our algorithm, we applied it to several of the images in the database. From the results, we can say that the algorithm gave excellent results in the detection and classification of vehicles. However, as in all image processing algorithms, the performance strongly depends on the particular application, and there will be cases where the algorithm will not give the desired results. In order to obtain acceptable results in the application of our algorithm, the images under study should satisfy the following conditions.

- The region of study should include only one-way highway segments, and road shoulders and ramps should be avoided.
- The presence of bushes and trees decrease the accuracy of the results since those objects could be detected as vehicles.
- For the cases where heavy traffic is present, the algorithm will cluster together the vehicles that are very close to each other, causing an error in the final counting of the vehicles.

3.3 Results and Analyses

In this section we show the results obtained by applying the proposed algorithms to some of the images in our database. The bright vehicles are detected by applying both techniques: Multiple Thresholds and Otsu and the results are compared. The dark vehicles are detected using the Otsu Threshold.

3.3.1 Image: Phoenix

Figure 3.3.1: Highway in Phoenix

Figure 3.3.1 shows a highway in Phoenix, Arizona. The area highlighted with the red rectangle is the region under study. Figure 3.3.2 shows the selected region (one-way highway). This selected region contains 9 bright vehicles and 5 dark vehicles.

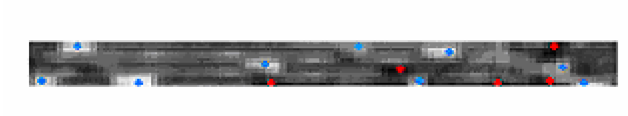

Figure 3.3.2: Selected Region (one-way highway)

Bright Vehicles detected by Multiple Thresholds

Figure 3.3.3 shows the three binary images obtained by applying the different thresholds T1, T2 and T3. These three threshold values were computed using the Multiple Thresholds method. From Figure 3.3.3 we can observe that T1 and T3 detected all bright vehicles except the vehicles highlighted in red. Notice that T2 has detected all the bright vehicles. In this example, none of the thresholds detected irrelevant objects (lane markers).

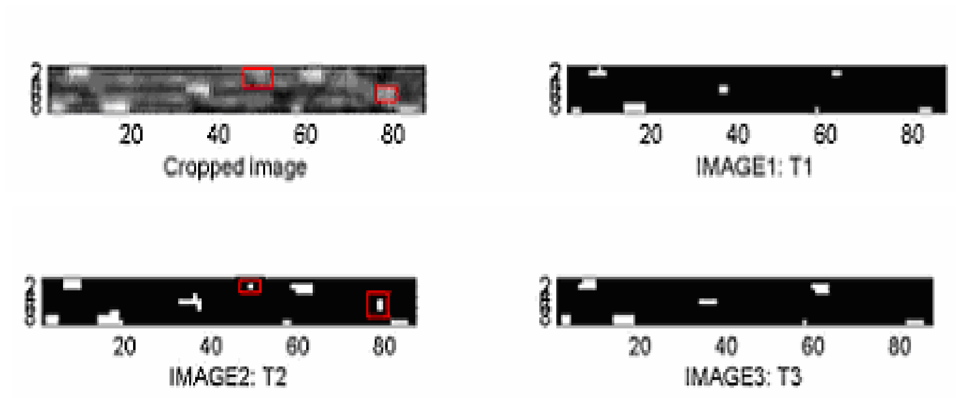

Figure 3.3.3: Binary Images Obtained by Using T1, T2 and T3

To avoid the detection of lane markers and other unwanted objects, only common objects are taken from each combination of these three binary images. Figure 3.3.4 shows the common objects of each combination. During the process of eliminating the lane markers and road dividers, there is a possibility of losing some of the vehicles. From figures 3.3.3 and 3.3.4, we can observe that while trying to eliminate the lane marker, we lost two bright vehicles (marked in red).

Figure 3.3.4: Common Objects

In Figure 3.3.4, the three binary images have all the bright vehicles that are present in the test image, except two, the vehicles highlighted with a red box in figure 3.3.3. Notice that no lane markers or road dividers are present in any of the three images. To obtain all the bright vehicles, the three binary images (New_Image1, New_Image2, New_Image3) are added. Figure 3.3.5 shows the bright vehicles detected by the Multiple Thresholds method.

Figure 3.3.5: Bright Vehicles Detected By Multiple Threshold Technique

Notice that from the 9 bright vehicles present in the test image, 7 are detected using the Multiple Thresholds technique. From this example we can see that the Multiple Thresholds technique can detect only those vehicles that are much brighter than the background. Vehicles that have intensities similar to lane markers are not detected by the Multiple Thresholds technique.

Bright Vehicles Detected by Otsu Threshold

Figure 3.3.6 shows the test image, image after sliding neighborhood operation and the final binary image obtained by using Otsu threshold. By applying sliding neighborhood operation to test image, vehicles are made very bright. In second image, vehicles are brighter when compared to vehicles in test image. Third image shows the vehicles detected by the Otsu Threshold. Otsu Threshold has detected all the bright vehicles that are present in the test image. It has detected the vehicles (marked red) that were not detected by the Multiple Threshold. As Otsu detects any pixel that is brighter than the background, all the vehicles were detected.

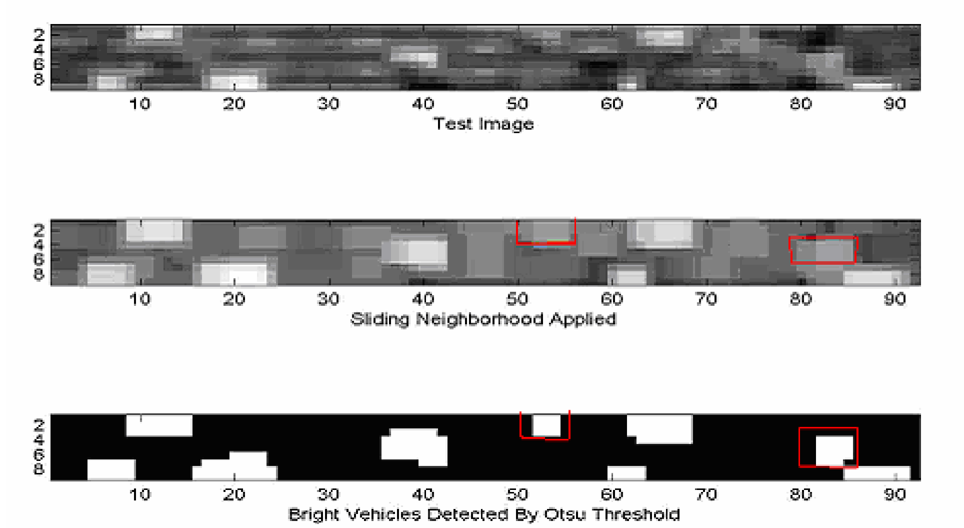

Figure 3.3.6: Test Image, Image with Sliding Neighborhood and Final Result

Dark Vehicles Detected By Otsu Threshold

Figure 3.3.7 shows the results obtained after applying the Otsu Threshold to detect the dark vehicles. From the figure we can observe that all the dark vehicles were detected. As it was explained in section 3.2.4, a preprocessing step is applied first, that is, a Sliding Neighborhood operation that applies a minimum intensity function to a 3 x 3 neighbor of each pixel.

Figure 3.3.7: Dark Vehicles Detected by Otsu Threshold

Detecting All the Vehicles

To obtain the final image showing all the vehicles, the binary image containing the bright vehicles, and the binary image containing the dark vehicles are added. Figure 3.3.8b) shows the result obtained by applying the Multiple Threshold technique to detect the bright vehicles and the Otsu Threshold to detect the dark vehicles. Figure 3.3.8 c) presents the result obtained by applying the Otsu Threshold to detect both types of vehicles, bright and dark. Comparing Figures 3.3.8b) and 3.3.8c) we can observe that Multiple Thresholds Technique could detect only 7 vehicles out of 9 vehicles. Two of the vehicles (highlighted by red boxes) are not detected because they have intensities very similar to the background intensities. From Figure 3.3.8c) it can be seen that the Otsu Threshold technique has detected all the vehicles. Note also that the Otsu threshold is clustering three different vehicles (red rectangle in Figure 3.3.8c) and this will give an error in the final count.

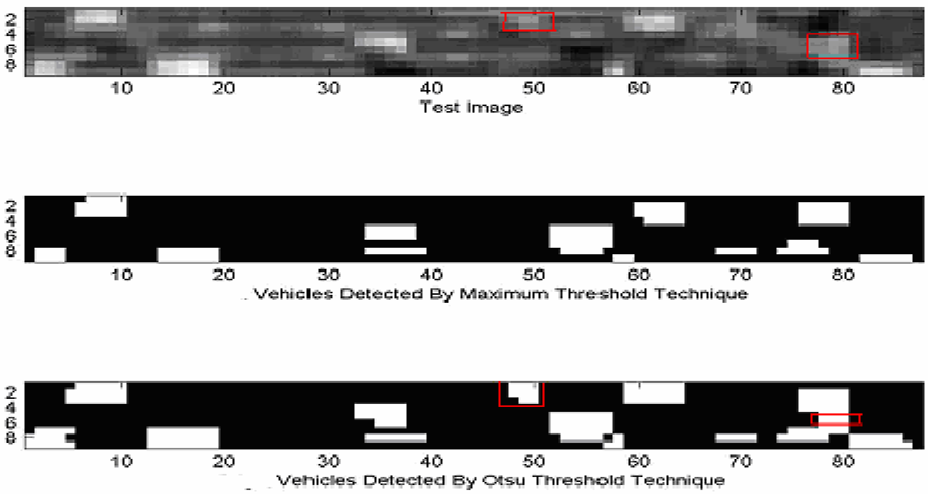

Figure 3.3.8: Detecting all the Vehicles. a) Test Image b) Bright vehicles by Multiple Threshold and dark vehicles by Otsu c) Bright and dark vehicles by Otsu

Results

Two main algorithms were developed for the detection and classification of vehicles. The first algorithm uses Multiple Thresholds for the detection of the bright vehicles, and Otsu method for the detection of the dark vehicles. The second algorithm uses the Otsu method for the detection of both, bright and dark vehicles. Both algorithms were tested on several of the images in the database. To measure the performance of the algorithms, the results obtained from each algorithm were compared with a manual count of the vehicles; this is, by visually inspecting the region under study.

From the results, we can say that the algorithms gave very good results in the detection and classification of vehicles. However, as in all image processing algorithms, the performance strongly depends on the particular application, and there will be cases where the algorithm will not render the desired results.

From the performed tests we found out that in order to obtain acceptable results (an error of less than 10% in the detection and counting of the vehicles) in the application of the algorithms, the regions under study should satisfy certain conditions. For example, the region under study should include only one-way highway segments, and road shoulders and ramps should be avoided. The presence of bushes and trees decrease the accuracy of the results since those objects could be detected as vehicles. For the cases where heavy traffic is present, the algorithms would cluster vehicles that are very close to each other, causing an error in the final count of the vehicles.

Count and Classification

Below we are presenting two tables that show manual and automatic count of bright and dark vehicles detected by both the methods. Manual Count table gives the manual count of the vehicles from test image and final resultant images obtained from Multiple Threshold and Otsu Threshold techniques. Automatic Count table gives the count of the vehicles obtained by using the algorithm Counting of Vehicles defined in section 3.2.6

Manual Count

	Actual Count	Multiple Threshold	Otsu Threshold
Cars	7	7	7
Trucks	7	5	7
Total	14	12	14

Automatic Count

	Multiple Threshold	Otsu Threshold
Cars	7	4
Trucks	5	7
Total	12	11

Multiple Thresholds Technique has classified and counted the detected vehicles properly. Otsu Threshold has detected all the vehicles present in the test image. Since the size of the bright vehicles detected by the Otsu Threshold is greater than the Multiple Thresholds, while combining binary images of dark and bright vehicles to get a final binary image with all the vehicles, some of the white and dark vehicles were combined as single vehicles. Therefore, the automatic count of trucks, cars, and total count were altered for the Otsu Threshold.

3.3.2 Image: Highway I10

Figure 3.3.9 shows highway I10. Test image contains 8 bright vehicles and 0 dark vehicles. Figure 3.3.10 shows the bright vehicles and dark vehicles detected by the Otsu Threshold. This algorithm detected all bright vehicles. As there are no dark vehicles and shadows in the test image, image of dark vehicles detected by Otsu threshold is empty. Since there are no dark vehicles and shadows, total vehicles detected are equal to the total bright vehicles detected.

Figure 3.3.9: Test Image I10

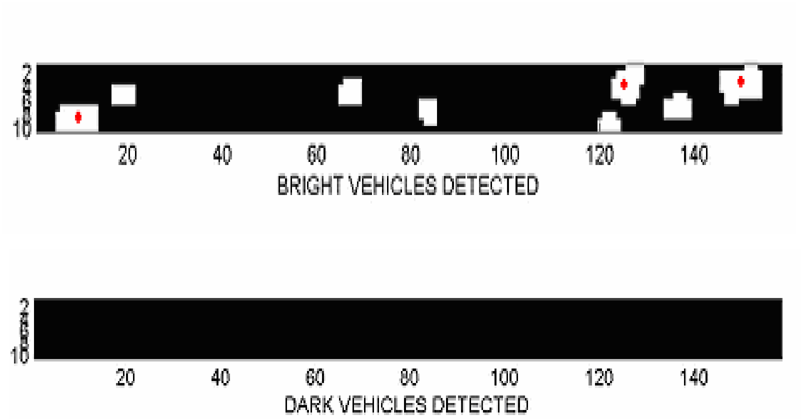

Figure 3.3.10: Bright and Dark Vehicles Detected by Otsu Threshold

The thresholds T1, T2 and T3 obtained by the Multiple Thresholds algorithm are 206.33, 114 and 160.166 respectively. Figure 3.3.11 shows the bright vehicles detected by Multiple Thresholds algorithm and the dark vehicles detected by the Otsu Threshold. In this case also all 8 bright vehicles got detected. Since there are no dark vehicles and shadows, total number of vehicles is equal to the number of bright vehicles detected.

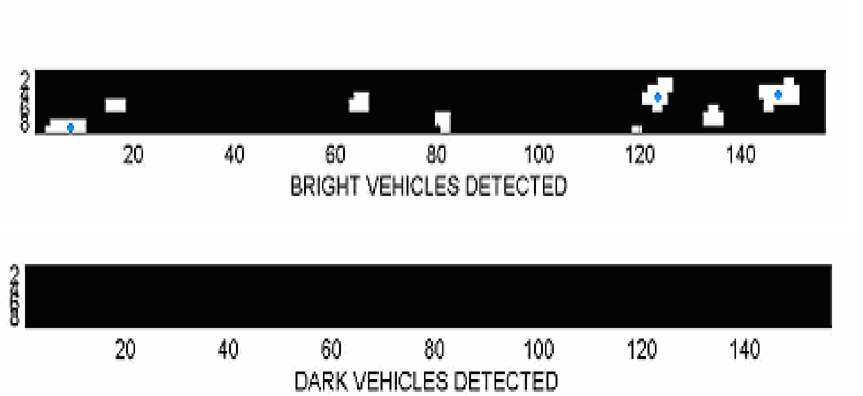

Figure 3.3.11: Bright Vehicles Detected by Multiple Thresholds and Dark Vehicles detected by Otsu Threshold

Figure 3.3.12 shows a comparison between the test image, vehicles detected by Otsu Threshold and vehicles detected by the MultipleThresholds.

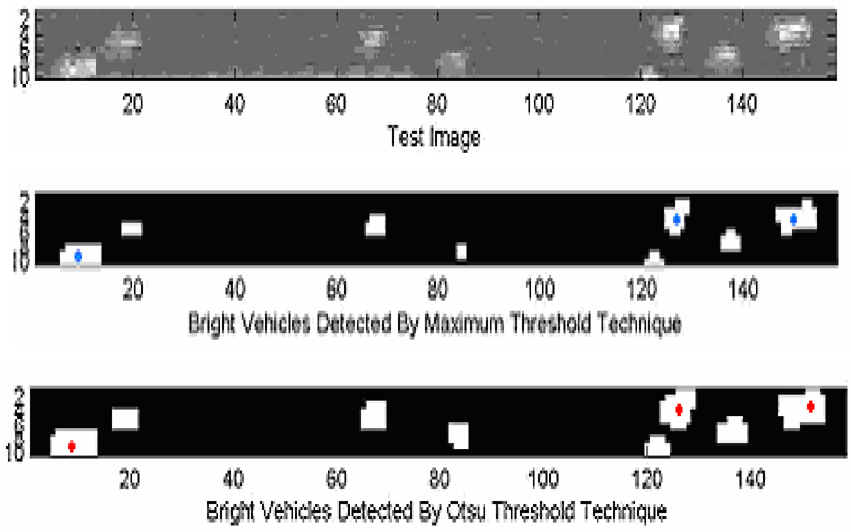

Figure 3.3.12: Comparison of Vehicles Detected by Multiple and Otsu Threshold

Count and Classification

Manual Count

	Actual Count	Multiple Threshold	Otsu Threshold
Cars	5	5	5
Trucks	3	3	3
Total	8	8	8

Automatic Count

	Multiple Threshold	Otsu Threshold
Cars	5	5
Trucks	3	3
Total	8	8

In Figure 3.3.12, the vehicles marked with colors are the trucks. Red marked vehicles are the trucks detected by Otsu Threshold and blue marked vehicles are the trucks detected by Multiple Thresholds. In this particular example the results of the manual and automatic calculation were the same. Since there are no shadows, no dark vehicles, no lane markers and as all the vehicles are placed far from each other, very accurate results are obtained.

3.4 Conclusions

- Multiple Thresholds does not detect those vehicles that have intensity values similar to the lane markers.
- Because of the sliding neighborhood operation, the size of the brighter vehicles detected by the Otsu threshold is greater than the size of the brighter vehicles detected by Multiple Thresholds.
- If the vehicles are very close to each other, then there are possibilities of getting them identified as a single vehicle.
- While detecting the vehicles, dark and bright vehicles are identified in two different binary images. These two binary images are combined to merge a vehicle with its shadow. If a bright vehicle has no shadow and a dark vehicle is close to this vehicle, then there are possibilities of getting dark and bright vehicles combined as a single vehicle. This will alter the result of the count of the vehicles.

3.5 Limitations

- Better results are obtained for only one ways.
- Not able to crop polygon shaped roads.

Chapter 4: Detection of Highways from High-Resolution Satellite Images

4.1 Introduction

As transportation infrastructure of highways, streets and bridges and the number of vehicles are increasing rapidly, serious problems like traffic congestion and safety issues are being faced by the transportation agencies. Looking for solution to some of these problems, transportation agencies are trying to incorporate advance technologies such as real-time traffic management systems, which incorporate detectors on roads, traffic cameras, and traffic lights to gather information and build expert systems for decisions.

Aerial images and satellite images are also becoming to be considered for traffic applications. For example, satellite images are capable of covering large areas; they can frequently revisit to almost any part of the globe, regardless of its remoteness; they are capable to collect data unhindered by local air traffic [1]. Detail information about the highways of any large area can be obtained by satellite images. As satellite images have many advantages, we are using satellite images in our research.

The goals of this part of the research are:

- Implement an algorithm to detect Highways without Vehicles
- Implement an algorithm to detect Highways with Vehicles

Thresholding techniques are used for both goals. Section 4.2 gives an introduction about the type of highways that the algorithm will detect, and explains the algorithm ‘Detection of Highways without Vehicles’. Section 4.3 explains the algorithm ‘Detection of Highways with Vehicles’. Both algorithms are implemented using Matlab, and tested on several images. The results with analyses are given in Section 4.3. In Section 4.4 three examples are provided, and the encountered problems are explained.

4.2 Detection of Highways without Vehicles

In this research, intensity images are used for the detection of highways from satellite images. Threshold Techniques are used for the highways detection. Detailed information about Image Segmentation and Threshold Techniques were given in Section 3.2.1.

By analyzing the histograms of several of the scenes in the database containing highways and roads, we divided the histogram in four main regions. Figure 4.2.1b shows these regions. Region I starts from the lowest intensity values to midway of average intensity. For most of the cases, the intensity values in this region, covers dark objects of the image such as dark vehicles, shadows, etc. The Region II covers intensities that go from approximately midway of the average intensity to average intensity, and typically covers dark gray shades objects like highways, trees, etc. The Region III goes from average intensity to midway of highest intensities and covers bright gray objects such as buildings, lane markers, rivers, etc. Region IV goes approximately from midway of highest intensity to highest intensity and generally covers bright objects like bright vehicles, road dividers etc. Figure 4.2.1 shows a typical highway scene and its histogram.

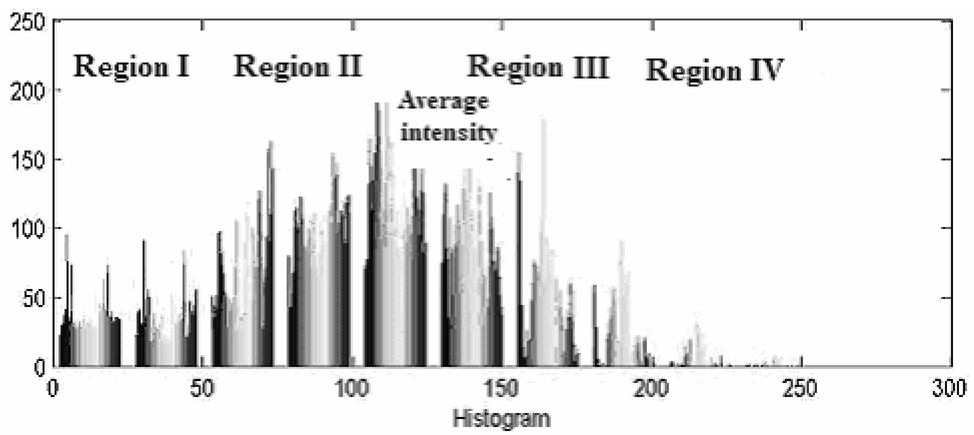

Fig. 4.2.1 a) Typical scene shwowin higways and roads in a metropolitan area
b) Its histogram and the four regions

4.2.1 Calculating Threshold T

From the histogram in fig 4.2.1 b), we can observe that Region II includes intensities in the range of 50 to100, and these values generally belong to highways (intensity range of the highway is checked manually). Since highways have intensity values less than the average intensity of the image being analyzed, we decided to use the average intensity of each row of our image under study (M1) for creating a second matrix (M2). M2 is a one-column matrix, and each element is the average intensity value of the corresponding row of intensities of M1. The threshold value T is selected as the minimum value of M2.

```
M2 is calculated by using M1:
 for each row i in M1
               M2[i] = average_intensity[M1, i]

T is calculated using M2:
T = minimum[ M2 ]
```

T is used to convert the test image M1 to binary image B1. A pixel at any position (x y), if f(x y) < T, then it is considered as Highway(0) else it is turned white(1). Binary image B1 obtained by T is defined as:

```
[ row column page ] = size [M1];
M2 = mean [M1];
T = minimum [M2];
 for i = 1:row,
    for j = 1:column,
            if M1[ i, j ] < T
                 B1[ i, j ] = 0;
            end
            else
                 B1[ i, j ] = 255;
            end
    end
 end
```

By using the algorithm to find the threshold T, for the test image shown in Figure 4.2.1 a), we obtained T = 70.7. Figure 4.2.2 shows the resulted binary image B1.

Figure 4.2.2: Binary Image B1 Obtained By Using T

4.2.2 Morphological Operations

Since we are making any pixel less than T as 0 to detect highways, there are chances of detecting dark vehicles, shadows, and trees, which have intensity values less than T (i.e. range similar to that of highways). In the Figure 4.2.2, it can be observed that highways are dark in color. It can also be observed that there are some isolated black pixels, which correspond to dark vehicles, shadows, and trees. To minimize the detection of shadows, dark vehicles, and trees, we apply four morphological operations: Clean, Majority, Fill, and Holes to the binary image B1.

Clean: Remove isolated pixels (1's surrounded by 0's).
Majority: Set a pixel to 1 if five or more pixels in its 3-by-3 neighborhood are 1's.
Fill: Fill isolated interior pixels (0's surrounded by 1's).
Holes: Fills the black holes in the binary image.
These morphological operations are inbuilt functions of MATLAB Image Processing Toolkit. Therefore we are not going into implementation details of these operations

As a result of these four morphological operations, most of the isolated dark pixels will be eliminated. In the particular case where large areas are covered by trees and vegetation (with intensity values in the range of highways), the possibilities of being eliminated are minimal. This is due to the fact that these large areas are not isolated set of pixels and the morphological operations, which are restricted to a 3-by-3 block of pixels, will not eliminate them. If we try to increase the size of the block, there are possibilities of missing the highways. Figure 4.2.3 shows the binary image B2 obtained after applying the morphological operations.

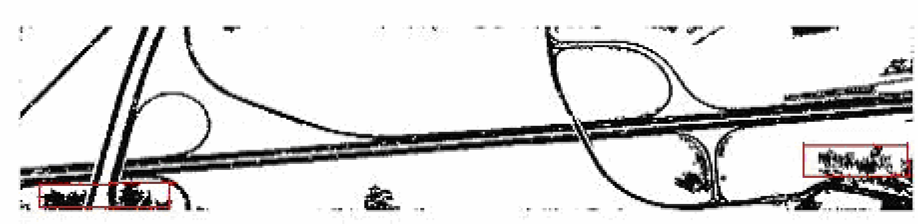

Figure 4.2.3: Binary Image B2 after Morphological Operations

From Figure 4.2.3 we can observe that in the binary image B2, most of the isolated pixels were eliminated. Notice that in this figure the red highlighted rectangle shows some vegetation that could not be eliminated by the morphological operations.

4.2.3 Intensity Image of Highways without Vehicles

To obtain the final intensity image (Final_Image) showing only highways, pixels with value **0** in B2 are considered. If a pixel has value **0** in B2, then the intensity of the corresponding pixel in the original test image M1 is copied to Final_image. In this way, only the detected highways will be copied from M1 to Final_Image using B2. Figure 4.2.4 shows the final image obtained by the algorithm 'Detection of Highways'. The Final_Image is obtained from B2 and M1 as:

```
[row column page] = size [B2];
for i = 1:row,
  for j = 1:column,
    if B2 [i,j ] ==  0
      Final_Image [i,j] = M1 [i,j];
    else
      Final_Image [i,j] = 0;
    end
  end
end
```

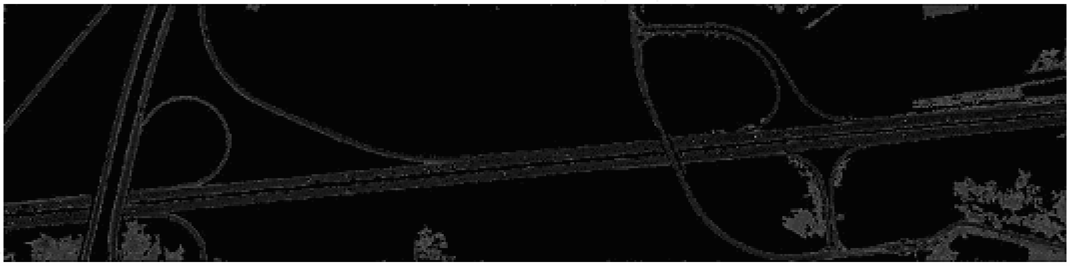

Figure 4.2.4: Highway Detected

4.3 Detecting Highways with Vehicles

In the previous section, no vehicles present on the highways in the original test image M1 were detected in the Final_Image. In this thesis, we were also interested in detecting the highways with vehicles. To achieve this, the complement of the binary image B2, which was obtained while detecting highways without vehicles, is used. Let the complement of binary image B2 be B3. Figure 4.3.1 shows the binary images B2 and B3.

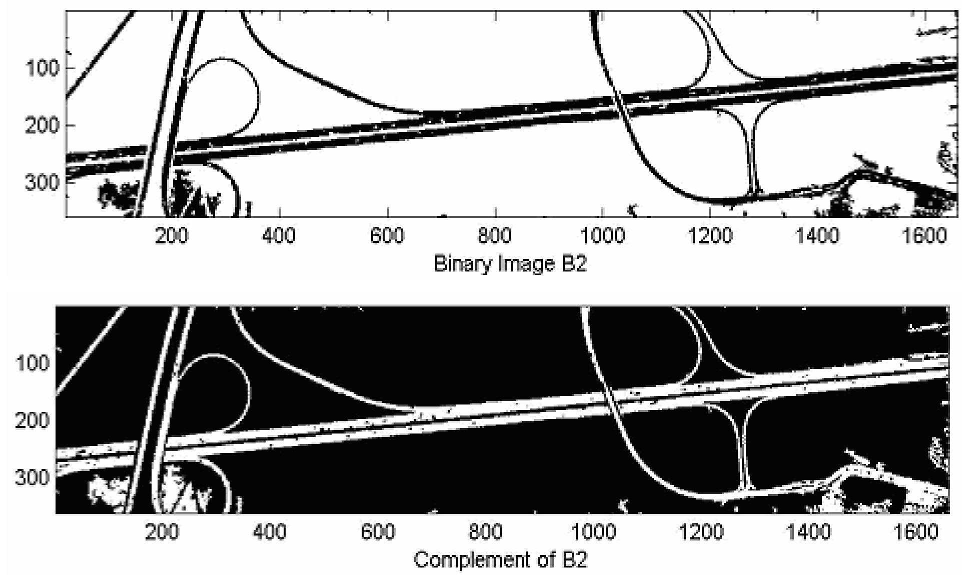

Figure 4.3.1: Binary Image B2 and its Complement B3

In B3, we can observe that the vehicles on the highway are dark holes. To detect the vehicles, these holes are filled with white pixels. In this way, we get a binary image B4, which contains only highways and vehicles in white color and remaining parts of the image in black color. Figure 4.3.2 shows the binary image B4 obtained by filling dark holes in B3.

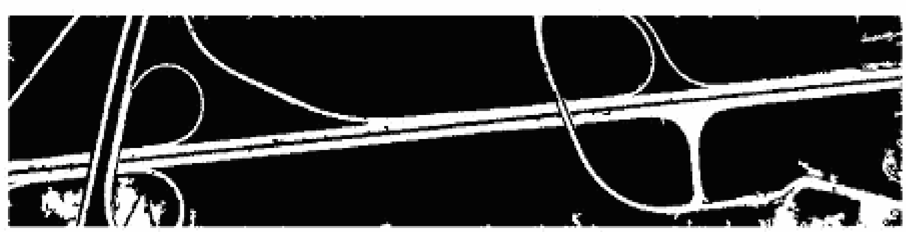

Figure 4.3.2: Binary Image B4

4.3.1 Intensity Image of Highways with Vehicles

To obtain the final intensity image Final_Image, highways with vehicles, only pixels with value **1** in B4 are considered. If a pixel has value 1 (white) in B4, then the intensity of the corresponding pixel in the original test image M1 is copied to Final_Image. In this way, the detected highways with vehicles will be copied from M1 to Final_Image using B4. Figure 4.3.3 shows the final image obtained by the algorithm 'Detection of Highways with Vehicles'.

The algorithm is as follows:

```
[row column page] = size [B4];
for i = 1:row,
  for j = 1:column,
    if B4 [ i, j ] ==  1
      Final_Image [ i, j] = M1 [ i, j];
    else
      Final_Image [ i, j] = 0;
    end
  end
end
```

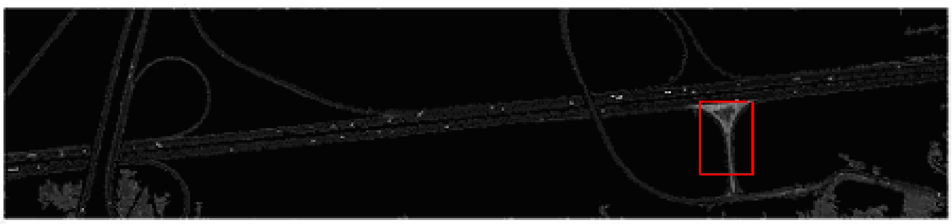

Figure 4.3.3: Highway with Vehicles

In Figure 4.3.3, the region highlighted with the red rectangle is not part of the highway, and it is not a vehicle either. The highlighted area is a black loop formed by the highway in image B3. While filling the black holes with white color to detect vehicles, the loop was also filled. Since the loop was a white hole in image B4, it was detected in the final image.

4.4 Results and Analysis

4.4.1 Image: Highway in France

Figure 4.4.1 shows a highway in France. The threshold T obtained by using algorithm 'Calculating Threshold T', which is given in Section 4.2.1, is 99.15. Figure 4.4.2 shows the test image and the Highway without Vehicles detected.

Figure 4.4.1: Highway in France

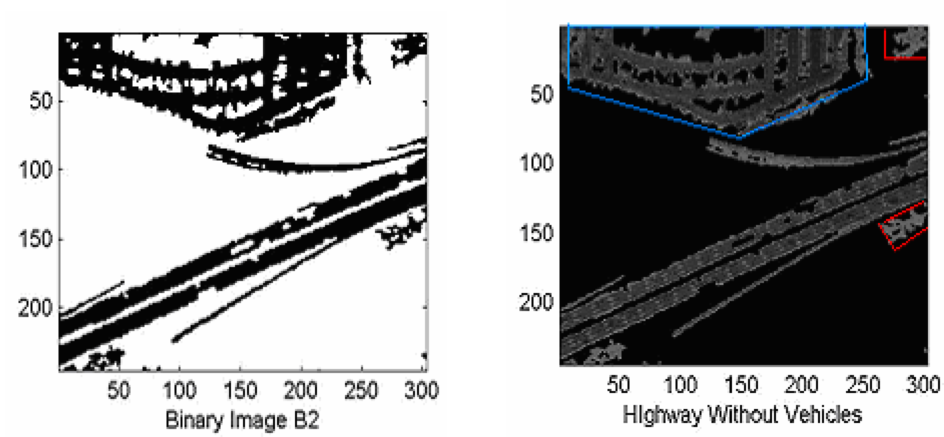

Figure 4.4.2: Binary Image B2 and Highways without Vehicles

In this example, we obtained very good results. All the highways were detected but small parts of barren land (part highlighted with red rectangles in Figure 4.4.2) were also detected. Barren land is detected because it had same intensity range as the highway. One important point to notice is the area highlighted by blue lines. This highlighted area represents a parking lot. Therefore the algorithm 'Detection of Highways without Vehicles' is not only capable of detecting highways but also parking lots.

Figure 4.4.3 shows the binary image B4 and the highways with vehicles. The detection of the highways is accurate but not all the vehicles on the highway were detected. The vehicles marked with red rectangles in the binary image B4 were not detected. The areas highlighted by blue color in Figure 4.4.3 b) correspond to the vehicles marked in red in figure 4.4.3 a). These marked vehicles were not detected because to detect the vehicles we filled the black holes in B4, but the red marked vehicles in B4 are not holes, rather they are connected with regions that are not highways. So, these regions are not filled with white color and are not detected as vehicles. In this example also, the parking lot with the vehicles on it was detected.

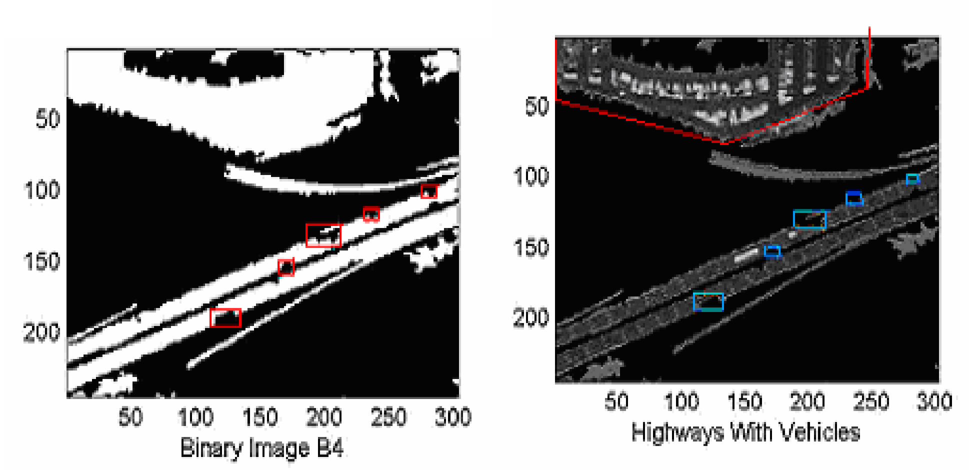

Figure 4.4.3: Binary Image B4 and Highways with Vehicles

4.4.2 Image: Highway in Baghdad

Figure 4.4.4 shows a highway in Baghdad. The threshold value T is calculated by the algorithm 'Calculating Threshold T' and it is 111.4. Figure 4.4.5 shows the binary image B2 obtained by using T and the resulting image showing the detected highways, obtained after applying the algorithm 'Detection of Highways without Vehicles'.

Figure 4.4.4: Highway in Baghdad

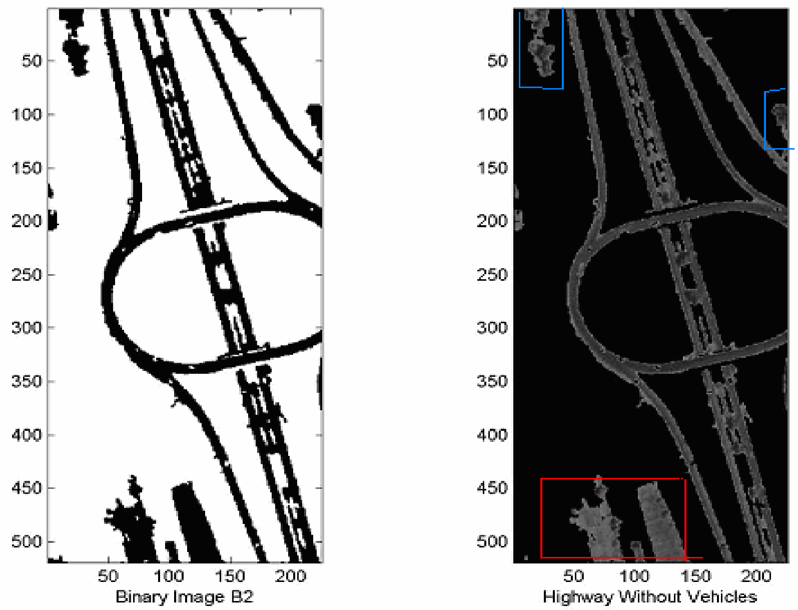

Figure 4.4.5: Binary Image B2 and Highways without Vehicles

From Figure 4.4.5 we can say that all the highways in the test image were detected. The region highlighted by the red rectangle corresponds to barren land and the areas enclosed by the blue rectangles are trees and vegetation. The reason for not being able to avoid the detection of trees and barren land was explained in the previous section (4.4.1)

Figure 4.4.6 shows the highways with vehicles detected by the algorithm 'Detection of Highways with Vehicles'. In this example all the highways and vehicles were detected. The bright region highlighted by red color in Figure 4.4.6 is not a highway or a vehicle. Notice that this region was not detected while detecting highways without vehicles (see Figure 4.4.5). The reason for detecting the highlighted region, while detecting the highways with vehicles, was give in Section 4.3.1.

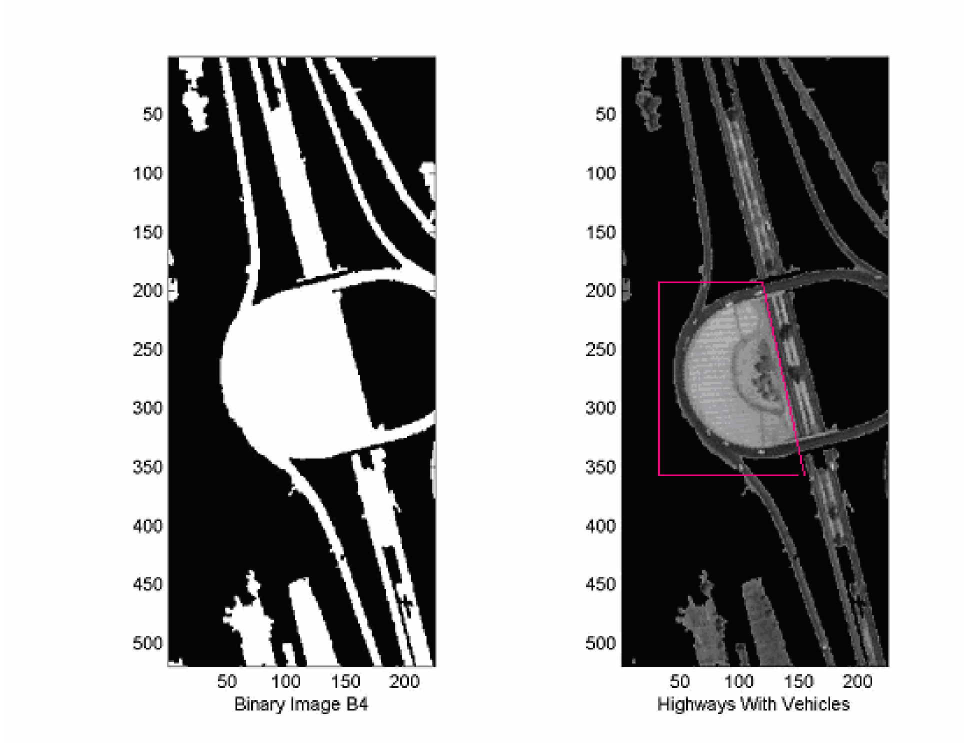

Figure 4.4.6: Binary Image B4 and Highways with Vehicles

4.4.3 Image: I10_1 Highway

Figure 4.4.7 shows highway I10_1. Threshold T calculated by the algorithm 'Calculating Threshold T' is 131.46.

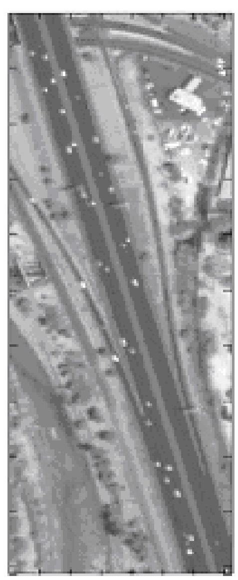

Figure 4.4.7: Highway I10_1

Figure 4.4.8 shows binary image B2 obtained by using T and highway detected by the algorithm 'Detection of Highways without Vehicles'. Irrelevant things like barren land (part highlighted by red color) were also detected.

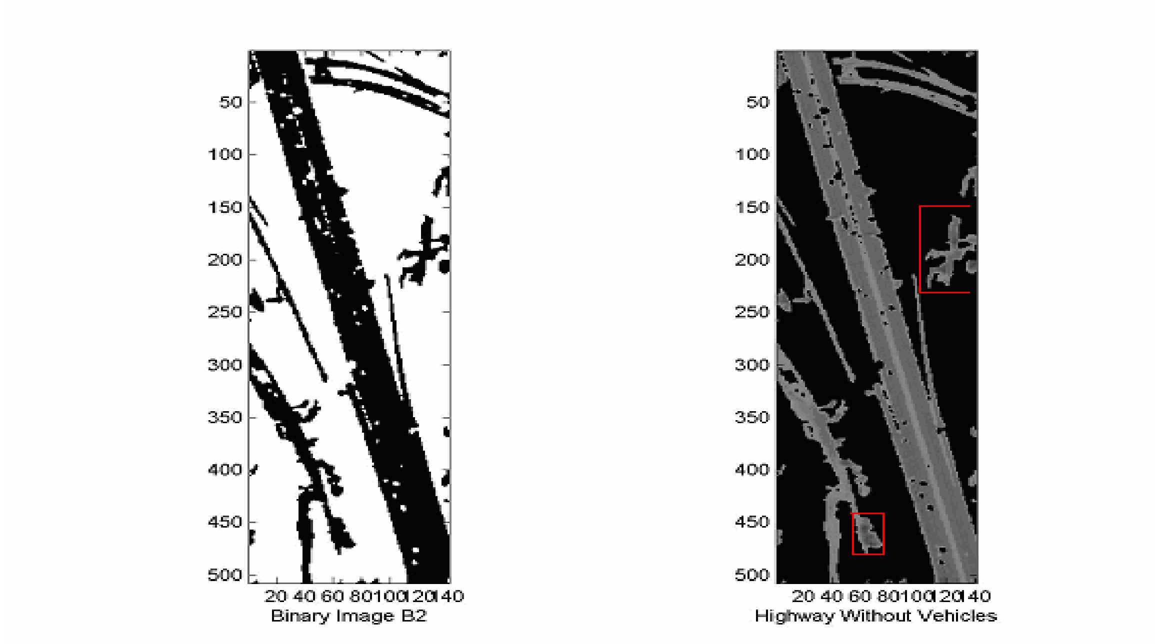

Figure 4.4.8: Binary Image B2 and Highways without Vehicles

Figure 4.4.9 shows the binary image B4, and the highways with vehicles detected by the algorithm 'Detection of Highways with Vehicles'. Accurate results are obtained with respect to highways. But some of the vehicles (parts marked red in Figure 4.4.9) are not detected, as they are not black holes in the binary image B2.

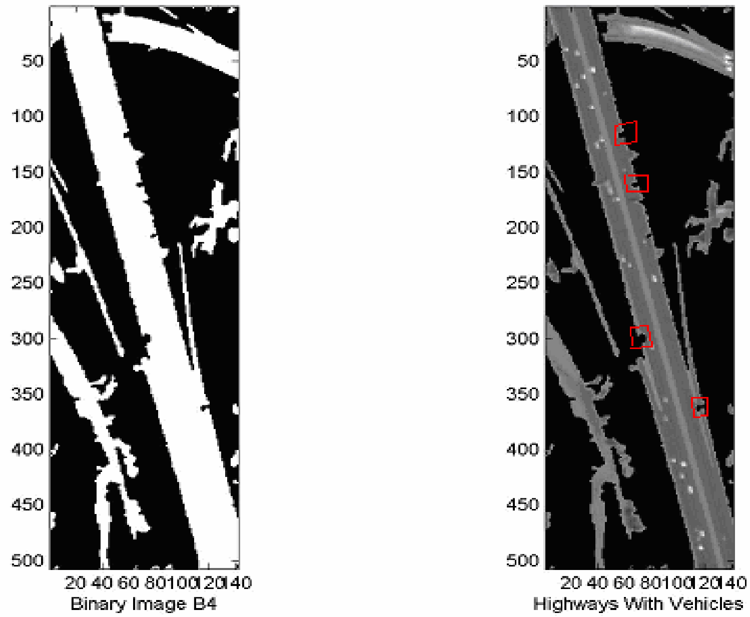

Figure 4.4.9: Binary Image B4 and Highways with Vehicles

4.5 Conclusion

Accurate results were obtained with respect to the detection of highways without vehicles, while detecting highways with vehicles, did not render accurate results. In most of the cases, few vehicles are not detected because they were not proper black holes in binary image B2. To detect the vehicles, we white filled the black holes in the binary image B3. As the undetected vehicles were not black holes in B3, they are not detected. The undetected vehicles are the vehicles that are very close to the edges of the highway. Therefore better results are obtained in the cases where the vehicles are well inside the highway.

Chapter 5: Detection of Vegetation from High-Resolution Satellite Images

5.1 Introduction

Automatic detection and extraction of vegetation from satellite images can be done by using the common properties of the vegetation. Green color, texture and shape are some of the common properties [17]. Detail information about the texture and shape is difficult to obtain from satellite images. This information can be obtained in detail from aerial images. Information about the color property can be obtained from both aerial and satellite images. As we are interested in using the satellite images, which have many advantages over aerial images, we used the color property i.e. green color for vegetation detection. The proposed methods for vegetation detection from satellite images can also be used for vegetation detection from true color aerial images. Figure 5.1 shows a true color test image, which is an 'Ikonos 1-m Resolution Satellite image'.

Figure 5.1: RGB Image from the IKONOS satellite

5.2 Proposed Algorithms for the Detection of Vegetation

The algorithm that we propose for the detection of green area is composed of two parts. In the first part, the light green shades of the image are detected. In the second part, the dark green shades are detected. The algorithms use the RGB color model, which consists of three component images, one for each primary color. Each image uses 8-bits, this is the triplet of values (R,G,B) have a depth of 24 bits. Pixel values in each of the three components (R, G, B) can have values 0 - 255.

5.2.1 Detection of Light Green Vegetation

The RGB value of light green has green component greater than blue and red components. Color of a pixel in original image, Image1, whose green component is greater than blue and red components, is copied to a new image, Image2. In this way, only light green pixels will be copied to Image2. Pixels that do not satisfy the above property are not copied from Image1 to Image2 and their positions in Image2 are made black. Figure 5.2 shows the light green plants detected from a test image.

The 'Detection of Light Green Vegetation' algorithm is:

```
[row column page] = size(Image1)
for i = row
   for j = column
         if (Image1 (i, j, 2) > Image1(i, j, 1) & Image1(i, j, 2) > Image1(i, j, 3))
            Image2 (I, j, 1) = Image1 (I, j, 1);
            Image2 (I, j, 2) = Image1 (I, j, 2);
            Image2 (I, j, 3) = Image1 (I, j, 3);
        else
            Image2 (I, j, 1) = 0;
            Image2 (I, j, 2) = 0;
            Image2 (I, j, 3) = 0;
         end
    end
end
```

Figure 5.2: Applying the Light Green Algorithm to the image in figure 5.1

By using this algorithm, light green can be detected. But, some shades of grey also satisfy the condition that is satisfied by light green i.e. green component is greater than the blue and red component.

Therefore, irrelevant things like clouds could be detected while detecting the light green vegetation as shown in Figure 5.2.

5.2.2 Detection of Dark Green Vegetation

The RGB value of dark green has red component of less value than both the green and blue components. Also, the green component is approximately 20 units less than the blue component. To identify the dark green pixels, the green component is increased by 20 and the light green algorithm is applied. Figure 5.3 shows the result of applying the dark green and the light green algorithms to the image in Figure 5.1.

The 'Detection of Dark Green Vegetation' algorithm is:

```
[row column page] = size(Image1)
for i = row
   for j = column
        Green = Image1 (i, j, 2) + 20;
        if (Green > Image1(i, j, 1) & Green > Image1(i, j, 3))
           Image2 (i, j, 1) = Image1 (i, j, 1);
           Image2 (i, j, 2) = Image1 (i, j, 2);
           Image2 (i, j, 3) = Image1 (i, j, 3);
      else
           Image2 (i, j, 1) = 0;
           Image2 (i, j, 2) = 0;
           Image2 (i, j, 3) = 0;
    end
 end
```

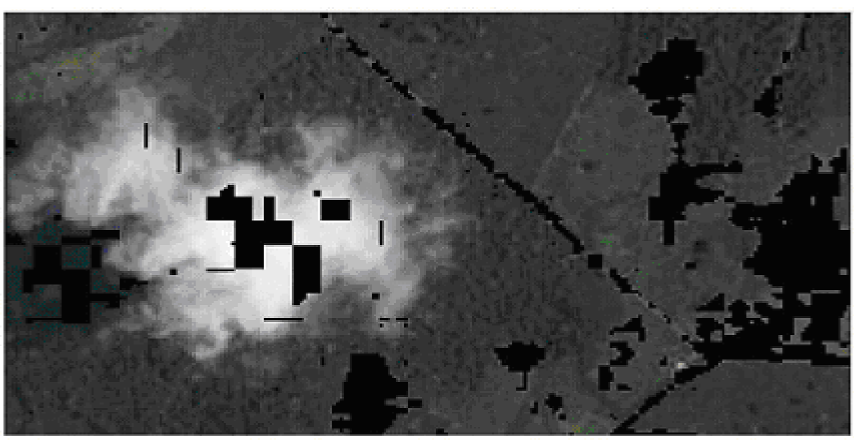

Figure 5.3: Dark and Light Green

By using this algorithm, dark green can be detected. Notice that in this case also some shades of grey satisfy the condition that is satisfied by dark green, this is, the green component value is less than the blue component. Therefore, objects such as clouds could also be detected while detecting the dark green vegetation as shown in Figure 5.3.

5.2.3 Preventing the Detection of Shades of Grey

Intensities of R, G, and B components of green color are approximately less than 100. On the other hand, intensities of R, G, and B components of grey shades that satisfy also the conditions of light and dark green are greater than 150. Therefore pixels with R,G and B components greater than 150 are not considered. Figure 5.4 shows the dark and light green vegetation without any clouds. The algorithm 'Preventing the Detection of Shades of Grey' is:

```
 [row column page] = size(Image2)
for i = row
   for j = column
          if Image2 (i, j, 1) > 150
               if Image2 (i, j, 2) > 150
                   if Image2 (i, j, 3) > 150
                         Image2 (i, j, 1) = 0.0;
                         Image2 (i, j, 2) = 0.0;
                         Image2 (i, j, 3) = 0.0;
                    end
               end
           end
       end
  end
```

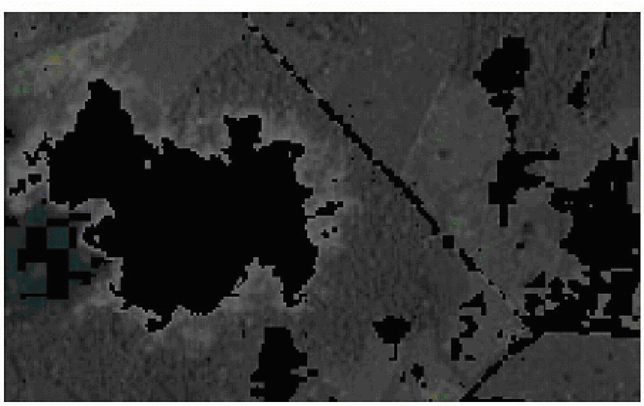

IMAGE WITH ONLY DARK AND LIGHT GREEN

Figure 5.4: Vegetation Detected

5.2.4 Comparing the Test Image and Final Image

In Figure 5.5 we can observe that after applying the three algorithms (light green, dark green, and preventing shades of grey) only vegetation was detected. Irrelevant objects like clouds and barren land are made black in the final image.

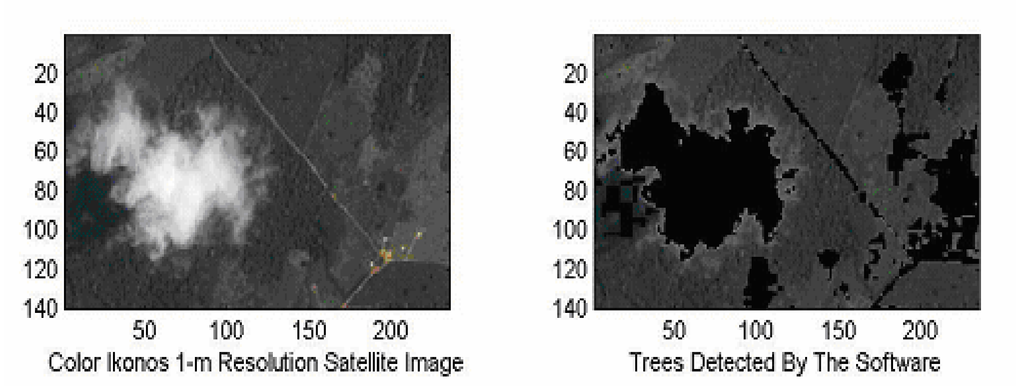

Figure 5.5: Test Image and Final Image

5.3 RESULTS

Results of the application of the algorithms for vegetation detection on various test images are discussed in this section. Test images and the final images are presented for visual comparison.

5.3.1 Image: TajMahal, India

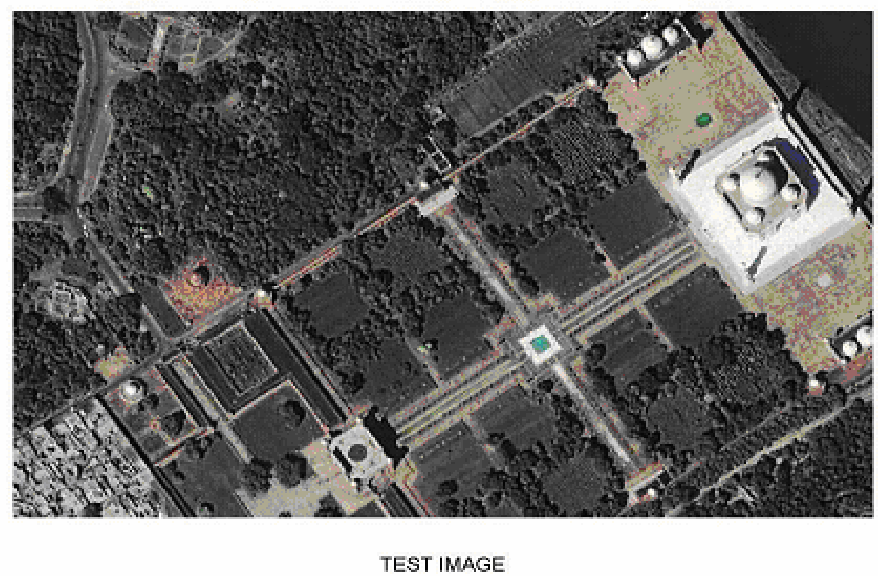

Figure 5.3.1: Test Image

Figure 5.3.1 is a one-meter resolution satellite image of TajMahal, India which was collected by Space Imaging's IKONOS satellite [6]. In this case, the test image has dark and light green vegetation, a white monument, and roads and houses. By applying the algorithm ‘Detection of Light Green Vegetation’, we could detect all light green vegetation from the test image. Figure 5.3.2 shows the light green vegetation extracted from the test image.

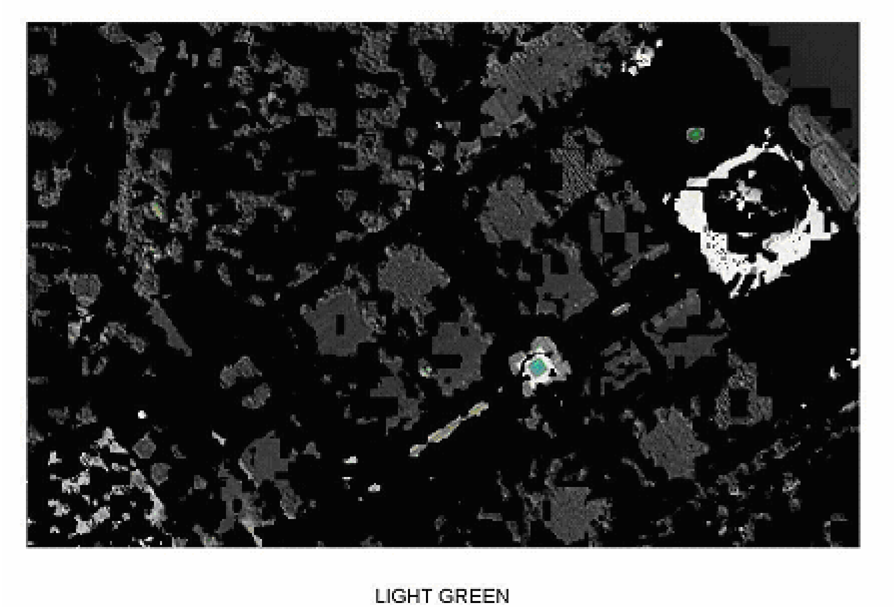

Figure 5.3.2: Light Green Shade

Figure 5.3.3 shows the dark green vegetation that is detected by using the algorithm ‘Detection of Dark Green Vegetation’. Notice, that the light green vegetation is also shown here.

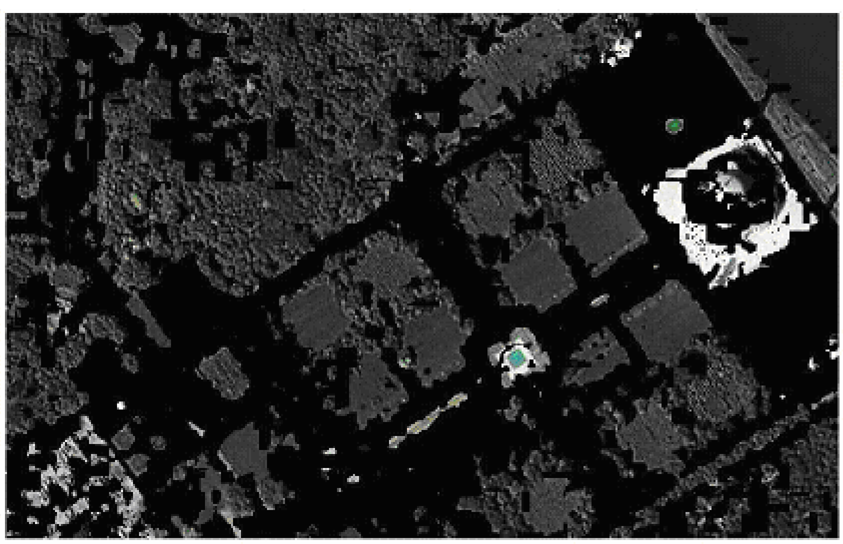

Figure 5.3.3: Dark Green Shade

From figures 5.3.2 and 5.3.3, we can observe that dark and light green vegetation was detected along with some grey parts that were present in the test image. Grey parts were detected because grey color and green color have some similar properties that were used in the algorithms 'Detection of Light Green Vegetation' and 'Detection of Dark Green Vegetation'. Figure 5.3.4 shows the image that is obtained after applying the algorithm 'Preventing the Detection of Shades of Grey'. This image contains only light and dark green vegetation.

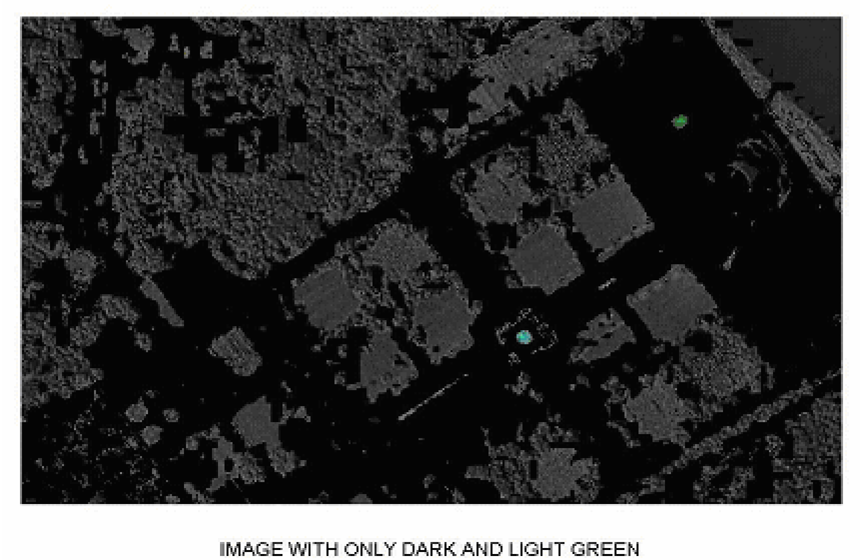

Figure 5.3.4: Vegetation Detected

Figure 5.3.5 shows the test image and the final image for visual comparison. From figure 5.3.5 we can say that in this case we obtained 100% accurate results. Only green vegetation was detected. Irrelevant things like roads, houses, land, etc. were not detected.

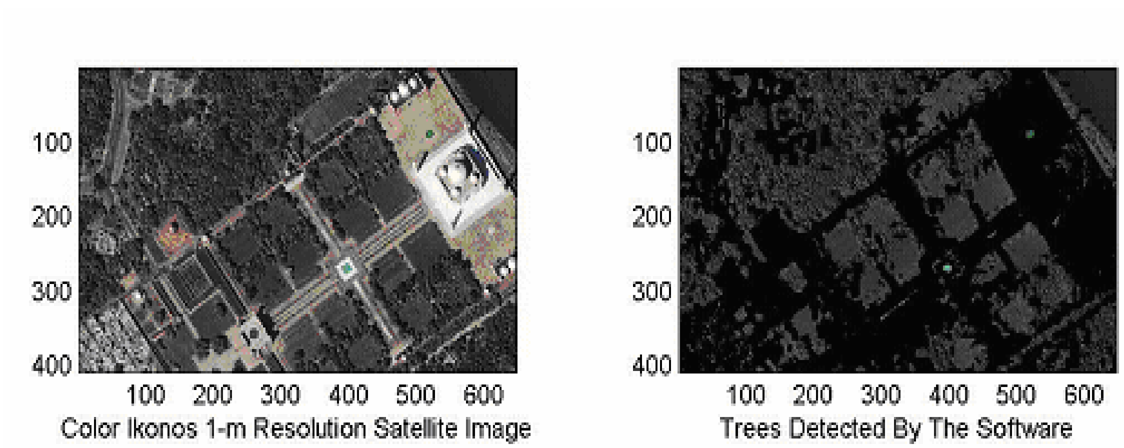

Figure 5.3.5: Test Image and Final Image

5.3.2 Image: Atsukeshi Lake, Japan

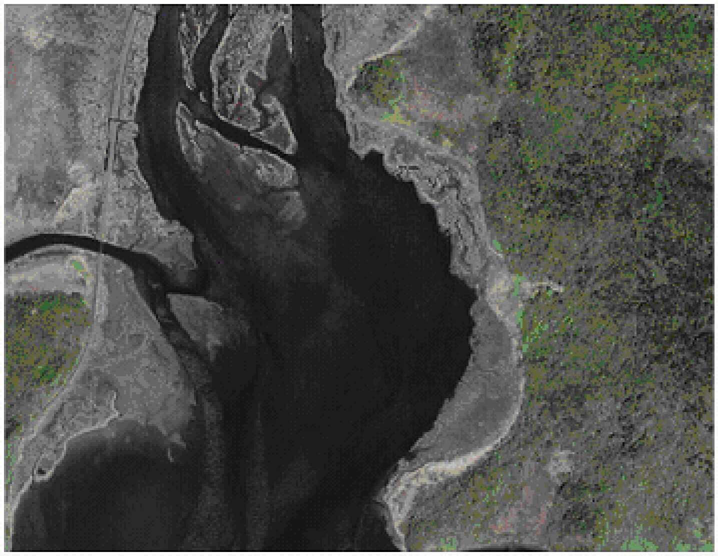

Figure 5.3.6: Test Image

Figure 5.3.6 is a one-meter resolution satellite image of Atsukeshi Lake, Japan, which was also collected by Space Imaging's IKONOS satellite [6]. In this case, the test image has a lake, barren land, rocks, and light green vegetation. Figure 5.3.7 shows the light green vegetation extracted from the test image after applying the algorithm 'Detection of Light Green Vegetation'.

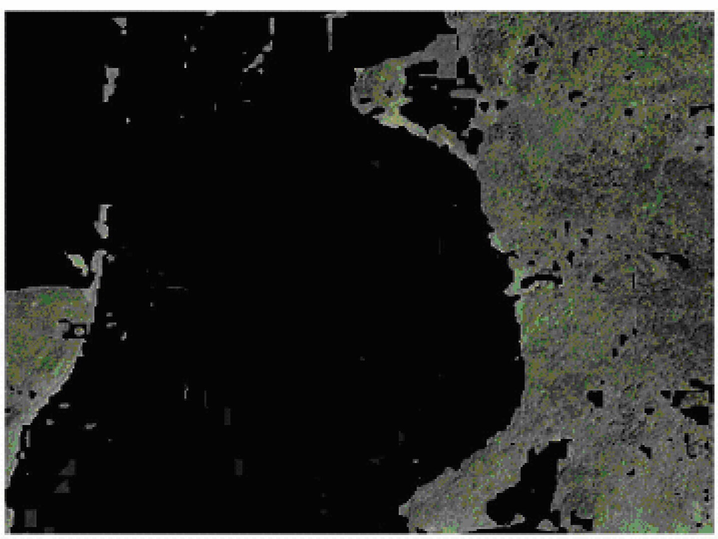

Figure 5.3.7: Light Green Shade

Figure 5.3.8 shows the image that was obtained after applying the algorithm 'Detection of Dark Green Vegetation'. Figure 5.3.7 and 5.3.8 are similar because there is no dark green vegetation present in the test image.

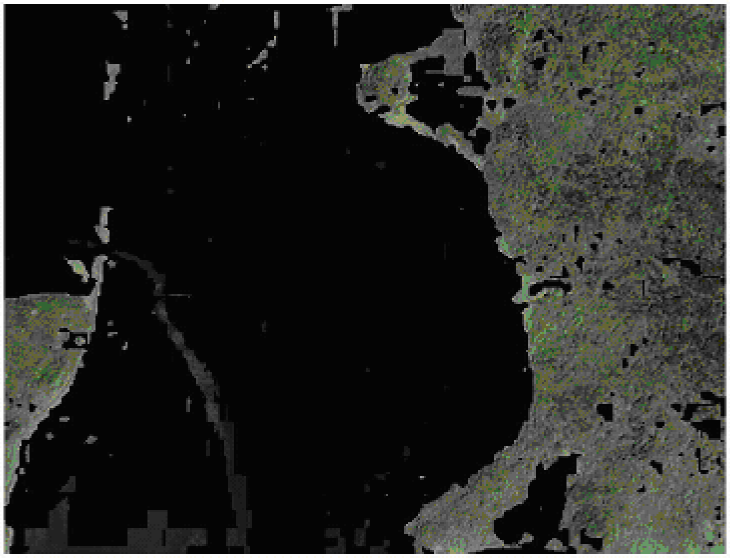

Figure 5.3.8: Dark Green Shade

Figure 5.3.9 shows the image that was obtained after applying the algorithm ‘Preventing the Detection of Shades of Grey’. This image is similar to images in figure 5.3.6 and 5.3.7 because this test image has no grey parts.

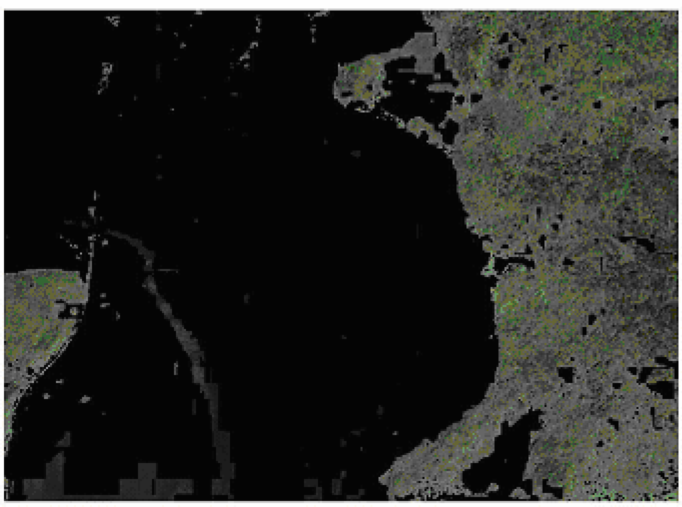

Figure 5.3.9: Vegetation Detected

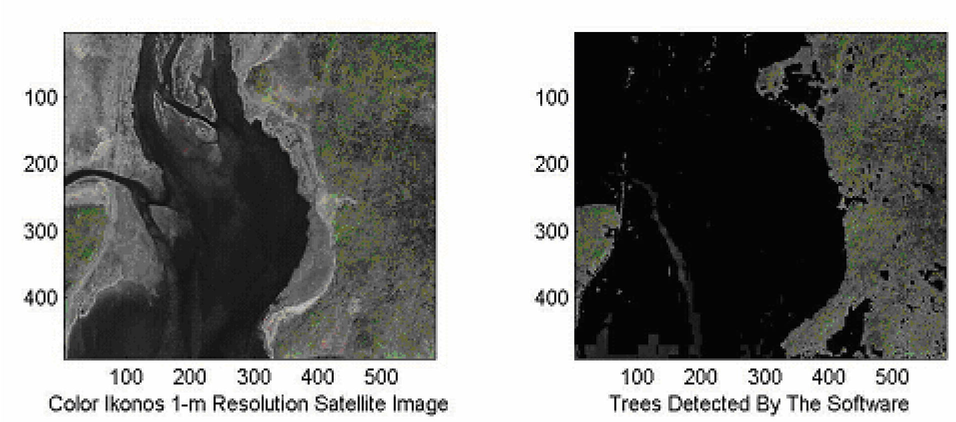

Figure 5.3.10: Test Image and Final Image

Figure 5.3.10 provides a visual comparison of the test image and the final image. From figure 5.3.10 we can say that, in this case 100% accurate results were obtained. Notice that all the vegetation was detected and no irrelevant things like lake, land, rocks etc were detected.

5.3.3 Image: Kyoto, Japan

Figure 5.3.11: Test Image

Figure 5.3.11 shows a one-meter resolution satellite image of Kyoto, Japan, which was also collected by Space Imaging's IKONOS satellite [6]. In this case, the test image has roads, buildings, vehicles and vegetation. Figure 5.3.12 shows the light green vegetation extracted from the test image after applying the algorithm ‘Detection of Light Green Vegetation’.

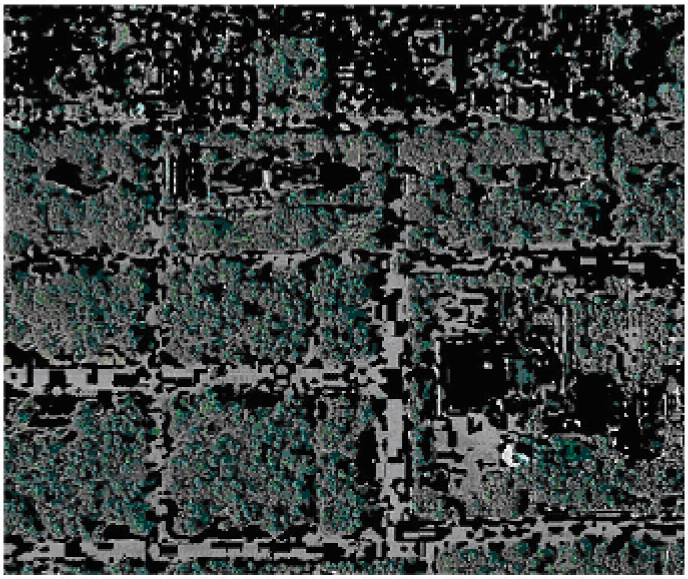

Figure 5.3.12: Light Green Shade

In Figure 5.3.12, we can observe that all light green vegetation was detected. Also, some parts of roads were detected because they are in shades of grey color. Figure 5.3.13 shows the dark green vegetation detected by the algorithm ‘Detection of Dark Green Vegetation’.

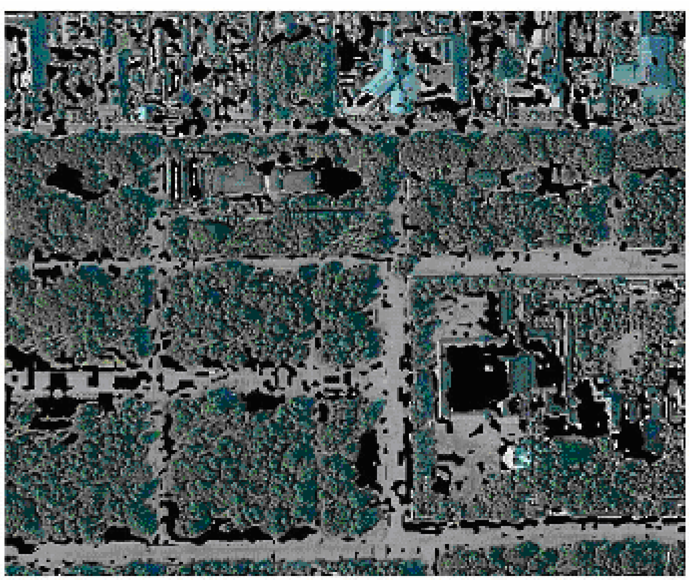

Figure 5.3.13: Dark Green Shade

In figure 5.3.13, we can see that all dark green vegetation was detected. Some parts of roads were also detected as they are in shades of grey. One important thing to notice is that some buildings, which are in green color were also detected. Green buildings were detected because they satisfy the color property that is satisfied by the dark and light green vegetation. By using the proposed algorithms of vegetation detection, we cannot avoid the detection of green objects, which are not vegetation. Figure 5.3.14 shows the image that is obtained after applying the algorithm 'Preventing the Detection of Shades of Grey'. This image has only dark and light green vegetation and buildings that have green color.

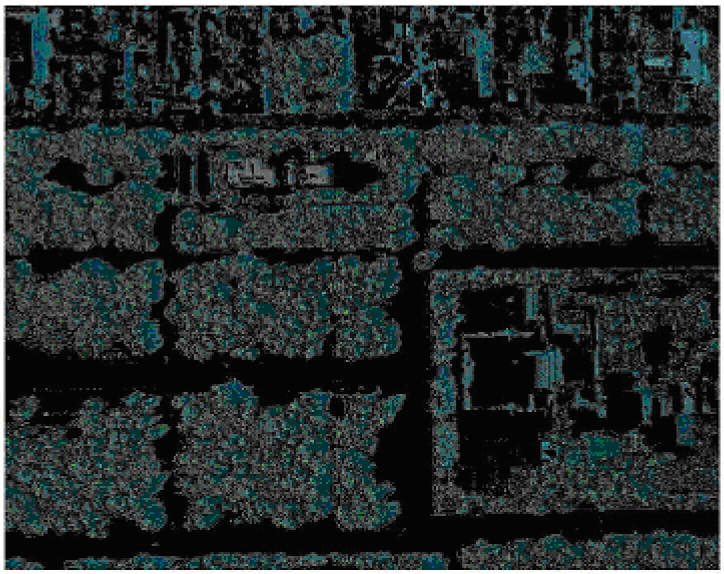

Figure 5.3.14: Vegetation Detected

Figure 5.3.15 shows the test image and the final image. From this image we can say that we could not obtain accurate results. Even though we could detect all the vegetation, we also detected some buildings that are in green color.

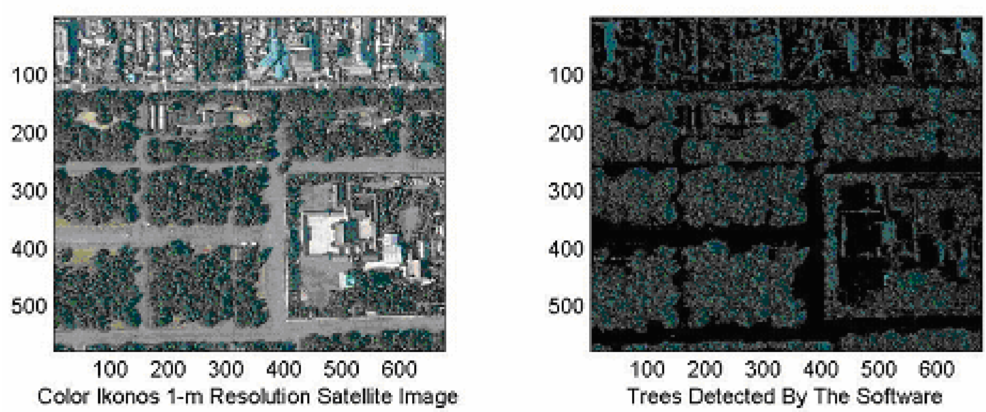

Figure 5.3.15: Vegetation Detected

5.4 Conclusions

From the examples presented in Section 5.3, we could say that entire vegetation with different shades of green is detected. Only green fields were detected even if the image has fields in different colors. Better results are obtained if the image contains nothing in shades of green other than vegetation. Any object, which is in green color, will be detected by the proposed algorithms because we are using the color property of vegetation. So, if the image contains things like green buildings, tennis courts etc, they will be also detected as shown in example 5.3.3. This problem could have been eliminated if model property of the vegetation were used. However, the overall results of the proposed method for the detection of vegetation are very satisfactory.

CHAPTER 6: CONCLUSIONS AND FUTURE WORK

6.0 Detection of Vehicles from High-Resolution Satellite Images

By using the proposed algorithms, better results are obtained only for one-ways. The region under study should include only one-way highway segments, and road shoulders and ramps should be avoided. The presence of bushes and trees decrease the accuracy of the results since those objects could be detected as vehicles. For the cases where heavy traffic is present, the algorithms would cluster vehicles that are very close to each other, causing an error in the final count of the vehicles.

As the intensity property of the vehicles is being used to detect the dark vehicles, shadows of the vehicles are also being detected as vehicles.

As future work to try to improve the results, some statistical approaches, such as the Bayesian classifier [16], could be used to evaluate the threshold values and see if more robust results can be obtained. Also, multi-spectral imagery could be combined with the panchromatic information, and apply techniques base on color properties to detect the vehicles and try to avoid irrelevant objects. To make the detection and classification fully automatic, without requiring from the user to rotate the image first, and then to select a region of study (only a highway segment), techniques based on pattern recognition should be investigated.

6.1 Detection of Highways from High-Resolution Satellite Images

Detection of Highways with Vehicles algorithm doesn't detect the vehicles that are close to the edges of the highway. This problem can be eliminated if model property of vehicles is used instead of the intensity property. As barren land and closely placed trees have similar intensity ranges as the highways, they are wrongly detected as highways. Surroundings of highways usually contain bushes, trees, and barren lands. Therefore, with the proposed algorithm it is not possible to avoid the detection of these objects, if they are present in the images. For those cases, different techniques based on pattern recognition should be investigated.

6.2 Detection of Vegetation from High-Resolution Satellite Images

As color property is being used to detect the vegetation any green object in the image will be detected by the proposed algorithm. Better results are obtained if the image contains nothing in green color other than vegetation. By using the model property of the vegetation this problem can be eliminated.

7 References

[1] Holland, D., and Marshall, P., “Using High-Resolution Satellite Imagery In a Well Mapped Country”, *Proceedings of ISPRS-EARSeL Joint Workshop on High Resolution Mapping from Space*, Hannover, October 2003.

[2] The MathWorks, Inc., Image Processing Toolbox, URL: http://www.mathworks.com/access/helpdesk/help/toolbox/images/

[3] Gonzalez, R., and Woods, R., *Digital Image Processing*, second edition, Prentice Hall, 2002.

[4] Schlosser, C., Reitberger, J., and Hinz, S., "Automatic Car Detection in High Resolution Urban Scenes Based on an Adaptive 3D-Model", *Proceedings of the IEEE/ISPRS joint Workshop on Remote Sensing and Data Fusion over Urban Areas, Berlin, 2003*, IEEE, Piscataway, pp. 167 - 171.

[5] Hickman, A., et al., “Methods of Analyzing Traffic Imagery Collected from Aerial Platforms”, *IEEE Transactions on Intelligent Transportation Systems*, Vol. 4, No.2, June 2003, pp. 99-107

[6] Space Imaging Inc., Gallery [Online] http://www.spaceimaging.com/gallery/, accessed May 2003

[7] INTEC Americas Corp, Gallery [Online] http://www.intecamericas.com/english.htm, accessed May 2003

[8] ORBIMAGE [Online] http://www.orbimage.com/prods/index.htm, accessed May 2003

[9] Hinz, S., "Detection and Counting of Cars in Aerial Images", Proceedings of the International Conference on Image Processing, Vol. 3, Barcelona, Sep. 2003, pp. III-997-1000

[10] Zhao, T., and Nevatia, R., "Car Detection in Low Resolution Aerial Image", Proceeding of the 8th IEEE International Conference on Computer Vision, Vol.1, Vancoucer, Canada, July 2001, pp. 710-717

[11] Moon, H., Chellappa, R., Rosenfeld, A., " Performance Analysis or a Simple Vehicle Detection Algorithm", Proceedings of the 3th ARL Federal Laboratory Symposium, Feb. 1999

[12] Schneiderman, H., and Kanade, T., "A statistical Method for 3D Object Detection Applied to Faces and Cars", Proceedings of the Computer Vision and Pattern Recognition, Vol.1, Hilton Head, SC, June 2000, pp. 746-751

[13] Liu, G., Gong, L., and Haralick, R., "Vehicle Detection in Aerial imagery and Performance Evaluation" ILS Technical Report, Intelligent Systems Laboratory, University of Washington, Seattle, WA, 1999

[14] Otsu, Nobuyuki, “A threshold selection method from gray-level histograms”, *IEEE transactions on systems, man, and cybernetics*, vol. SMC9, no.1, January 1979, pp.62-66

[15] Shen, H., and Johnson, C., ”Semi-Automatic Image Segmentation: A Bimodal Thresholding Approach”, Jan 3, 1995, Technical report, Department of Computer Science, University of Utah, Salt Lake City.

[16] L. Shapiro, G. Stockman, *Compute Vision*, Prentice Hall, 2001, p. 114.

[17] Roy Hay and Patrick Synge, The Color Dictionary of Flowers and Plants, Crown Publishers, 1992