Summer 2008

Object Detection in Poor Visibility Conditions Using Image Segmentation

Triveni Vuppalapati
Old Dominion University

OBJECT DETECTION IN POOR VISIBILITY CONDITIONS USING IMAGE SEGMENTATION

by

Triveni Vuppalapati

B.Tech., June 2006, Jawaharlal Nehru Technological University, India

A Thesis Submitted to the Faculty of Old Dominion University
In Partial Fulfillment of the
Requirement for the Degree of

MASTER OF SCIENCE

ELECTRICAL ENGINEERING

OLD DOMINION UNIVERSITY
August 2008

Approved by:

Zia-ur Rahman (Director)

Vijayan K. Asari (Member)

Jiang Li (Member)

ABSTRACT

OBJECT DETECTION IN POOR VISIBILITY CONDITIONS USING IMAGE SEGMENTATION

Triveni Vuppalapati
Old Dominion University, 2008
Director: Dr. Zia-Ur- Rahman

It can be dangerous for a pilot to attempt to land an aircraft safely in poor visibility conditions such as rain, fog, haze, snow and low light, but external imagery of a runway can be enhanced to provide increased situational awareness. Objects that are detected in a scene may or may not be a hazard, so interpretation is left to the pilot. In order to detect whether an object is a hazard to safe landing, it is necessary to determine whether the object is on the runway. For this purpose, two image segmentation algorithms: histogram based d-peak algorithm and local feature based quadtree algorithm have been implemented in this thesis. The purpose of image segmentation here is to separate the objects of interest from the background. Several algorithms and techniques have been developed for image segmentation, but there is no general solution. Several of these techniques may be combined to effectively solve an image segmentation problem. The d-peak algorithm is a histogram based method. Histogram Based methods are very efficient when compared with the other image segmentation techniques because they require only one pass through the pixels. In the quadtree segmentation algorithm, a full tree is formed by splitting a single parent node into four children or quadrants, and all descendants are recursively split by iteratively testing conditions for splitting the blocks until some minimum bound is reached. The parameters considered for splitting the parental blocks here are mean, standard deviation and the size of the block.

ACKNOWLEDGEMNTS

I would like to gratefully thank my advisor, Dr. Zia-ur Rahman for his support, advice and enthusiastic supervision. Without his constant guidance, it would not have been possible to complete this thesis. He explained things very patiently and supported me in numerous ways during this work.

I would like to thank Dr. Vijayan K. Asari and Dr. Jiang Li, members of my thesis advisory committee for their valuable time and suggestions.

Most of all I would like to thank my family members and friends for their support and love throughout the entire time.

Dedicated to Mom, Dad, Krishnaveni Garapati and Siva Garapati

For all of your love and support.

TABLE OF CONTENTS

CHAPTER PAGE

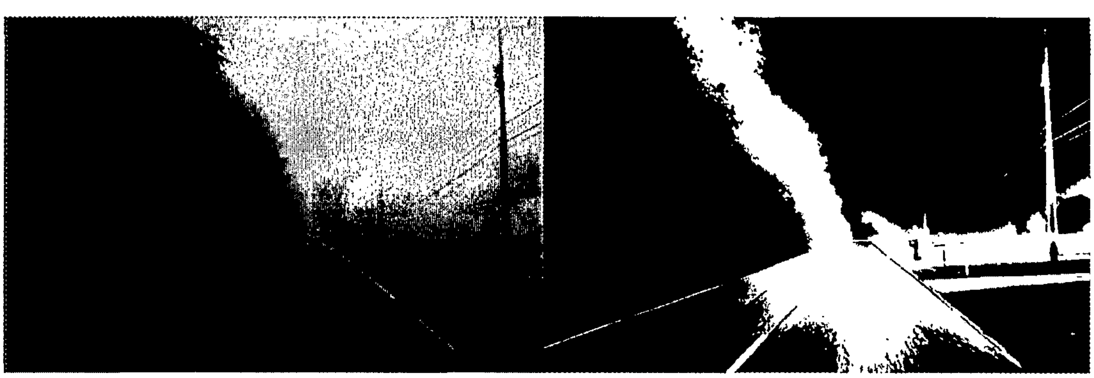
LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 General Introduction

The main objective of image segmentation is to distinguish objects of interest from the background [1]. Segmentation is the process of subdividing an image into its constituent regions or objects. The level to which this subdivision is carried out depends on the problem being solved; segmentation should stop when the objects of interest in an application have been isolated [2]. There is no single method for image segmentation that is suitable for all applications. The segmentation is based on the measurements taken from the image; these might be gray level values, color, texture, depth or motion. Image segmentation is the process of extracting and representing the information from an image to group pixels into regions of similarity [2]. Each of the pixels in a region may be similar with respect to some characteristic or computed property such as color, intensity or texture.



(a) Original image (b) Segmented image

Figure 1.1: Implementation of image segmentation.

IEEE style is used for a model in this thesis

Figure 1.1 shows the typical example of image segmentation in which the segmented image is a binary image formed by using a single threshold.

Applications of image segmentation include:

1. Detecting objects on a runway as a precursor to determining whether the object is a hazard or not,
2. Detecting objects in an image for object based measurements such as size and shape,
3. Detecting objects in a moving scene for object based video compression (MPEG4),
4. Detecting objects that are at different distances from a sensor using depth measurements from a laser range finder enabling path planning for mobile robots.

Additionally, image segmentation is used extensively in medical image applications. Some prominent applications are studying anatomical structure, computer guided surgery and diagnosis and treatment planning. Others are measuring tissue volumes and locating tumors and other pathologies.

1.2 Motivation

It can be difficult for a pilot to land an aircraft safely in poor visibility conditions such as rain, haze, fog, snow and low light. To increase the situational awareness of the pilot, many image processing techniques using approaches such as edge detection and segmentation have been proposed. In this thesis we have implemented and analyzed the results of two image segmentation algorithms, d-peak algorithm and quadtree algorithm, for detecting objects on a runway in poor visibility conditions. Detection of these objects is critical for a pilot to safely land an aircraft.

Segmentation schemes can be categorized as being one of three different types: In histogram based segmentation schemes, pixels are classified based on their gray levels [4]. In edge based schemes local discontinuities (edges) are detected and edge information is used to separate the images into regions [4]. Finally, in region based schemes, the process starts with a "seed" pixel (or group of pixels); this is an initialization point, typically chosen at random, that grows and splits the region until the whole original image is composed of homogenous regions.

1.2.1 Pixel Based Segmentation Using Mode Method

This is the most widely used image segmentation technique and is applicable to images with bimodal histograms, In bimodal histograms, one mode of the histogram corresponds to the gray- levels of the object, and the other mode corresponds to the gray-levels of the background. A fixed threshold value can be used to separate the background region from the objects.

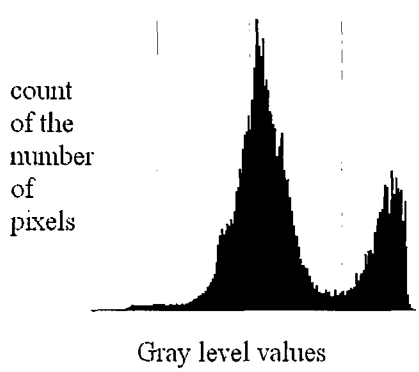


Figure 1.2: Bimodal histogram.

The threshold level is chosen to be the gray level in between the two modes using a number of different methods. The two most popular methods by which the threshold is chosen are:

1. Gaussian Filtering algorithm: This is the simplest segmentation method and is based on Bayes decision theory in pattern recognition. The gray level histogram of the image is computed; then, two component densities corresponding to the object and background are extracted from the mixture density associated with the histogram [8] [9].It is commonly assumed that both the background and the object densities are Gaussian.
2. Otsu's algorithm: In Otsu's method, threshold in a bimodal histogram is determined [8] [9] based on discriminent analysis in which thresholding is regarded as the partitioning of the pixels of an image into two classes, C_0 and C_1.

This is described in detail in chapter 2. The mode method works well for images with bimodal histograms where the modes are fairly distinct and are of nearly equal length. It does not work well if the gray level noise distribution is dependent on the gray level or is spatially correlated.

1.2.2 Edge based Segmentation schemes

Edge based segmentation schemes also consider local information but relative to the contents of the image and not based on an arbitrary grid. Each of the methods in this category involves finding the edges in an image and using that edge information to separate the image into regions. The most direct method used is the detect-and-link method [7] in which local discontinuities are first detected and then connected to form longer, and hopefully complete, boundaries. The disadvantage of this approach is that the edges detected are guaranteed to form closed boundaries and thus the subsequent thresholding scheme needs to merge regions that may not be uniform.

An improvement over this method is Yanowitz and Buckstein's adaptive thresholding method [7], in which an adaptive threshold is obtained by noting that the pixel intensities near the transitions between foreground and background (edge pixels) in a smoothed image, serve as the best local thresholds. They locate such pixels by checking for large gradients and interpolate the gray scale values of these pixels to form the thresholding surface.

A number of adaptive thresholding methods have been proposed that do not directly involve a thresholding surface. Many of these techniques approach thresholding as a special case of segmentation (with two region classes). Intensity gradient based thresholding [11] locates edge pixels in an image based upon the intensity gradient, classifies those pixels as foreground, and uses region growing techniques to classify the remaining pixels.

1.2.3 Region based segmentation schemes

Region based segmentation is a process of grouping pixels with similar characteristics, such as pixels with approximately equal gray levels, into regions. Traditionally, these are global hypothesis testing techniques. The process can start at the pixel level or at an intermediate level [2]. Generation and filtering of good seed regions of high confidence is essential [2]. Given initially poor or incorrect seed regions, these techniques usually do not provide any mechanism for detecting local gross errors in situations like those where an initial seed region spans two separate surfaces. These techniques can also fail if the definition of region uniformity is too strict i.e., when we insist on approximately constant brightness while in reality brightness may vary linearly.

1.3 Discussion

There is no single image segmentation technique that is applicable to all applications. The image segmentation technique adopted depends on the application.

1. The mode method is applicable to images with bimodal histograms where the modes are fairly distinct and of nearly equal length [8][9].This method does not work well if the gray level noise distribution is dependent on the gray level or is spatially correlated.
2. In edge detection methods, the quality of the segmentation depends on the quality of the edge detector; edge based schemes work best on images in which edges are easily detectable i.e., images that have good local contrast. They do not work well with images in which well defined edges occur due to noise [7].
3. Region based schemes work well for images with obvious homogeneity criteria. These schemes tend to be less sensitive to noise since homogeneity is typically determined statistically. The disadvantages of these methods is that an initial split level must be chosen well and if the condition for an initial split level is chosen in such a way that the image is not split in the first stage, then the region based scheme cannot be used for any application.

1.4 Proposed Research

In the following sections, we describe the various elements of our proposed research.

1.4.1. Histogram based Segmentation techniques

Histogram based methods are very efficient when compared to other image segmentation techniques because they require only one pass through the image pixels. In this technique, the histogram of the image is first computed. In the next step, the histogram is analyzed to determine either a single threshold- if segmentation is required to produce a binary image- or several thresholds based upon local grouping of gray level values. In this thesis a histogram based method called the d-peak algorithm is being implemented. The d-peak algorithm searches for clusters of gray level peaks in the histogram. All the pixels belonging to a particular range of gray level values are grouped together and represented by a single value. In this way, pixels across the image that have similar gray level values are represented by the same value leading to well defined clusters or segments.

1.4.2 Dithering

After applying the d-peak algorithm to images we may observe false edges, lines and regions because of false contouring. In the d-peak algorithm, large ranges of values are assigned to a single value, i.e. the image is quantized using the peak values.

Quantization error corresponds to a difference between the quantized value of a pixel after quantization and its value before quantization:

$$E[m,n] = I[m,n] - Q[I[m,n]] \tag{1.1}$$

Where $E[m,n]$ the error due to quantization is, $I[m,n]$ is the image pixel at location $[m,n]$ and $Q[]$ is the quantization operator. If the quantization level is less than about 32, there is a distinct difference in the representation of neighboring gray levels. This gives rise to the appearance of edges between neighboring gray levels. These edges are called false contours.

To mitigate this false contouring effect visually, we adopt a technique called dithering. Dithering is a process of intentionally adding noise to randomize the quantization error and visually improve the contouring effect. In the process of quantization, the intensity values of the pixel are rounded to the nearest gray level value. Dithering algorithms attempt to spread the quantization error in such a way that the eye notices the effects of false contouring to be less. We have different techniques in dithering:

1. Rectangular probability density function (RPDF) is equivalent to a roll of a die. Any number has same random probability of surfacing [10].
2. Triangular probability density function (TPDF) uniform probability is equivalent to a roll of two dice. This is nothing but the sum of two independent samples of RPDF [10].
3. Gaussian PDF is equivalent to a roll of large number of dice. The relationships of probabilities of results follow a bell-shaped or Gaussian curve [10].

4. Colored dithering is sometimes mentioned as dithering, that has been filtered to be different from white noise. Some of the dithering algorithms use noise that has more energy in the high frequencies so as to lower the energy in the critical band.
5. Noise shaping is not actually dithering; rather it is a feedback process that has dither within it.

In this thesis we have used Gaussian dithering; Gaussian noise is added to the quantized gray level values to spread the quantization error.

1.5 Quadtree Segmentation

Quadtree segmentation is a recursive operation. It starts out with an M x N input image. First, the input image is checked for homogenous regions using a predefined condition such as the brightness and the contrast of the image. If the condition is satisfied, the image is divided into four blocks or quadrants each of size M/2 x N/2. Each block is treated as a "new" image and tested for homogeneity using the condition described earlier. If the "new" image satisfies the conditions, it is further subdivided into quadrants. This process of testing and decomposing the input image is stopped when the resulting quadrants are too small. The size of the block to which this decomposition is carried out is also defined in the condition. This procedure is applied to all of the image quadrants. The main idea is that the regions where the objects are present in the image should have different brightness and contrast when compared to the other regions in the image. Thus, regions where objects are present in the scene should have blocks divided into very small regions.

The image contrast and the quality of the image are degraded by poor visibility conditions such as fog and haze. This degradation of images restricts the imaging applications to good visibility conditions. Hence, the foggy images are enhanced, and then the d-peak algorithm and quadtree algorithm are applied to the enhanced images.

1.6 Thesis Outline

Chapter 2 presents some of the image segmentation techniques like clustering and pixel based image segmentation, edge detection methods and region growing methods. Because of the degradation in images due to poor visibility conditions such as fog and haze, some of the image segmentation techniques are not directly applicable since the histograms of such images do not have the desired structure needed for segmentation. In order to apply segmentation, we first improve the overall brightness and contrast by using image enhancement techniques. We describe basic global enhancement techniques such as histogram equalization. Additionally, we also describe locally adaptive techniques such as the Retinex. For our thesis, we used the MSR that is part of the PhotoFlair[1]. The image segmentation techniques, d-peak algorithm and quadtree segmentation are discussed in detail. To mitigate the effects of false contouring and quantization error after the implementation of the d-peak algorithm, dithering is adopted. Gaussian dithering, implemented in this thesis, is explained in detail.

---

[1] PhotoFlair is the registered trademark of TruView Imaging Company which also developed the PhotoFlair software.

Chapter 3 presents the theory and the method of the d-peak algorithm and dithering and quadtree segmentation techniques, proposed image segmentation techniques.

Chapter 4 reports the results obtained with the proposed segmentation techniques. In this chapter several features of the algorithm will be discussed with respect to experimental results.

Chapter 5 includes conclusions and directions for future work.

CHAPTER 2

LITERATURE SURVEY

2.1 Introduction

Image segmentation is the process of partitioning a digital image into multiple regions to change the representation of the image into something that is more meaningful and easier to analyze [4] [5]. A region is a set of pixels that are similar to each other with respect to some characteristic or computed property such as color intensity or texture. Some practical applications [12] of image segmentation are:

1. Locating objects on a runway,
2. Locating tumors and other pathologies,
3. Measuring tissue volumes,
4. Performing computer-guided surgery,
5. Diagnosing,
6. Planning medical treatment,
7. Studying anatomical structure,
8. Recognizing face,
9. Recognizing fingerprint ,

10. Implementing automatic traffic controlling systems.

Several general purpose algorithms and techniques have been proposed and developed [12] for image segmentation. Since there is no general solution for an image segmentation problem, these techniques have to be combined with the knowledge of the domain in order to effectively solve an image segmentation problem in the domain of the problem.

Researchers categorize segmentation schemes into pixel based, edge based and region based and histogram based [12]. Histogram based segmentation schemes classify pixels based on their gray levels. Edge based schemes first detect local discontinuities (edges) and use that information to separate the images into regions. Regions based schemes [2] start with a seed pixel (or group of pixels) and then grow or split the seed until the original image is composed of homogenous regions. Finally, histogram based methods use histogram analysis to group pixels that have similar intensity values.

2.1.1 Pixel Based Segmentation schemes Using Mode Method

This is the most widely used image segmentation technique. This is applicable to images with bimodal histograms, in which one mode of the histogram corresponds to the gray levels of the object and the other mode corresponds to the gray levels of the background. A fixed threshold can be used to separate the background from the objects [8]. The threshold level is chosen to be the gray level in between the two modes using any number of different methods. The two most popular methods by which the threshold is chosen are the Gaussian Filtering algorithm and Otsu's algorithm.

Gaussian Filtering algorithm

This is the simplest segmentation method and is based on Baye's decision theory in pattern recognition. The gray level histogram of the image is computed; then, two component densities corresponding to background and the object are extracted from the mixture density associated with the histogram. It is commonly assumed that both the background and object densities are Gaussian [8] [9].

The following algorithm is substantially taken from [8, pp.257-270].

Algorithm

1. Compute the mean (μ) and the standard deviation of the histogram (σ) of the histogram.

$$\mu = \frac{1}{N}\sum_{i=1}^{255} F(i)i \tag{2.1}$$

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{255} F(i)(i-\mu)^2} \tag{2.2}$$

Where F(i) is the histogram value for gray level i (out of L gray levels), and N is the number of points in the window.

2. Find a least squares fit of

$$\mathrm{f}\,(\mathrm{i}) = \left(\frac{P_1}{\sigma_1} e^{-(i-\mu_1)^2 / 2\sigma_1^2}\right) + \left(\frac{P_2}{\sigma_2} e^{-(i-\mu_2)^2 / 2\sigma_2^2}\right) \tag{2.3}$$

to the histogram F(i) by adjusting the parameters $P_1, \mu_1, \sigma_1,\ P_2, \mu_2, \sigma_2,$ as follows:

(i) Smooth the histogram by taking a local weighted average:

$$F(i)=(F(i-2)+2F(i-1)+3F(i)+2F(i+1)+F(i+2))/9 \qquad (2.4)$$

On the smoothed histogram find the lowest value v= (lowest value) and use it to divide the histogram into two parts (for the original histogram F(i)).

$$N_1 = \sum_{i=1}^{v} F(i) \qquad N_2 = \sum_{i=1}^{v} F(i) \qquad (2.5)$$

$$\mu_1 = \frac{1}{N_1}\sum_{i=1}^{v} F(i)i \qquad \mu_2 = \frac{1}{N_2}\sum_{i=v+1}^{L} F(i)i \qquad (2.6)$$

$$\sigma_1 = \sqrt{\frac{1}{N_1}\sum_{i=1}^{v} F(i)(1-\mu_1)^2} \qquad \sigma_2 = \sqrt{\frac{1}{N_2}\sum_{i=v+1}^{L} F(i)(1-\mu_2)^2} \qquad (2.7)$$

(ii) Use a hill climbing method to minimize

$$\sum_{i=1}^{L}[f(i)-F(i)]^2 \qquad (2.8)$$

(a) Calculate val =|f(i)-F(i)| for i=lowest value v chosen in step (i)

(b) Calculate left_val=|f(i-1)-F(i-1)|

(c) Calculate right_val = |f(i+1)-F(i+1)|

(d) If left_val$\leq$ val, set the estimate for deepest valley at i-1;else if right_val$\leq$ val, set the estimate for deepest valley at i+1; else deepest valley found at i.

(e) If the deepest value was changed in step (d), reestimate $P_1, \mu_1, \sigma_1, P_2, \mu_2, \sigma_2$, using equations (2.5)-(2.7) and the new value of v. Repeat steps (a-d).

3. After the parameters of the mixture density have been estimated, a pixel with gray level *i is* assigned to the object if

$$\frac{P_1}{\sigma_1} e^{-(i-\mu_1)^2 / 2\sigma_1^2} > \frac{P_2}{\sigma_2} e^{-(i-\mu_2)^2 / 2\sigma_2^2} \tag{2.9}$$

The threshold value t is then defined as

$$\frac{P_1}{\sigma_1} e^{-(t-\mu_1)^2 / 2\sigma_1^2} = \frac{P_2}{\sigma_2} e^{-(t-\mu_2)^2 / 2\sigma_2^2} \tag{2.10}$$

And satisfies

$$\left(\frac{1}{\sigma_1^2} - \frac{1}{\sigma_2^2}\right) + 2\left(\frac{\mu_2}{\sigma_2^2} - \frac{\mu_1}{\sigma_1^2}\right) t + \left(\frac{\mu_1^2}{\sigma_1^2} - \frac{\mu_2^2}{\sigma_2^2}\right) + 2\ln\frac{P_2\sigma_1}{P_1\sigma_2} = 0 \tag{2.11}$$

Otsu's algorithm

Otsu's algorithm is a method of determining a threshold in a bimodal histogram and is based on discriminent analysis in which thresholding is regarded as the partitioning of the pixels in an image into classes C_0 and C_1 at gray level T. The following algorithm is substantially taken from [9, pp.62-66].

Algorithm

n_i = number of pixels at level i, i =1,…,L

$$N = \text{Total number of pixels} = \sum_{i=1}^{L} n_i$$

The gray level histogram is normalized and regarded as a probability distribution:

$$p_i = n_i/N \qquad (2.12)$$

$$p_i \geq 0 \qquad (2.13)$$

$$\sum_{i=1}^{L} p_i = 1 \qquad (2.14)$$

2. Divide pixels into two classes C_0 and C_1 by a threshold at level k.

3. Calculate the probabilities of class occurrence:

$$w_0 = P_r(C_0)\sum_{i=1}^{k} P_i = w(k) \qquad (2.15)$$

$$w_1 = P_r(C_1)\sum_{i=k+1}^{L} p_i = 1 - w(k) \qquad (2.16)$$

4. Calculate the mean class levels

$$\mu_0 = \sum_{i=1}^{k} i\,\mathrm{P_r}(i \mid C_0) = (\sum_{i=1}^{k} ip_i\,)/w_0 = \mu(k)/w(k) \qquad (2.17)$$

$$\mu_1 = \sum_{i=k+1}^{L} i\,\mathrm{P_r}(i \mid C_1) = (\sum_{i=k+1}^{L} ip_i\,)/w_1 = (\mu_T - \mu(k))/(1 - w(k)) \qquad (2.18)$$

Where $w(k)$ and $\mu(k)$ are the zeroth and first order moments of the histogram up to the kth level, and $\mu_T = \sum_{i=1}^{L} ip_i = \mu[L]$

5. Calculate the class variances:

$$\sigma_0^2 = \sum_{i=1}^{k} (i-\mu_0)^2 \, \mathrm{P_r}(i \mid C_0) = (\sum_{i=1}^{k} (i-\mu_0)^2 p_i)/w_0 \tag{2.19}$$

$$\sigma_1^2 = \sum_{i=k+1}^{L} (i-\mu_1)^2 \, \mathrm{P_r}(i \mid C_1) = (\sum_{i=k+1}^{L} (i-\mu_1)^2 p_i)/w_1 \tag{2.20}$$

6. In order to evaluate the "goodness" of threshold k, we can use the following discriminent criterion measures (or measures of class seperability):

$$\lambda = \frac{\sigma_B^2}{\sigma_w^2} \quad k = \frac{\sigma_T^2}{\sigma_w^2} \quad \eta = \frac{\sigma_B^2}{\sigma_T^2} \tag{2.21}$$

Where

$$\sigma_w^2 = w_0 \sigma_0^2 + w_1 \sigma_1^2 \tag{2.22}$$

is within class variance

$$\sigma_B^2 = w_0(\mu_0 - \mu_T)^2 + w_1(\mu_1 - \mu_T)^2 = w_0 w_1 (\mu_1 - \mu_0)^2 \tag{2.23}$$

is the between class variance, and

$$\sigma_T^2 = \sum_{i=1}^{L} (i-\mu_T)^2 p_i \tag{2.24}$$

is the total variance.

It is noticed that λ, k and η are equivalent to one another for a given k because

$$\sigma_w^2 + \sigma_B^2 = \sigma_T^2 \qquad \text{(2-25)}$$

The problem is now reduced to an optimization problem to search for a threshold k that maximizes one of the object functions (the criterion measures). It is noticed that σ_w^2 and are functions of threshold level k, whereas σ_B^2 is independent of k. Further, σ_w^2 is based on second-order statistics while σ_B^2 is based on first order statistics. Thus, we use η since it is the simplest measure with respect to k:

$$\eta = \frac{\sigma_B^2}{\sigma_T^2} \qquad \text{(2.26)}$$

Since σ_T^2 is independent of k, we can maximize η by maximizing σ_B^2 (k):

$$\sigma_B^2(\mathrm{k}) = \frac{[\mu_T w(k) - \mu(k)]^2}{w(k)[1 - w(k)]} \qquad \text{(2.27)}$$

Thus the threshold is chosen to be that k which maximizes σ_B^2 (k).

The threshold determination methods discussed above work well [20] if the object size is large enough to make a distinct mode in the histogram, the gray level noise distribution (intensity noise) is independent of the gray level, and the noise is spatially uncorrelated. These methods fail [20] when it is difficult to detect the valley bottom, as in images with extremely unequal peaks or in those images having broad and flat valleys in the histogram. Since peaks tend to become wider and lower with an increasing amount of intensity noise, some sharpening of the peaks and valleys can be accomplished by applying noise reduction preprocessing procedures.

2.1.2 Edge detection methods

Edge detection plays a very important role in image processing. Edges and region boundaries are closely related [12] since there is often a sharp adjustment in intensity at the region boundaries. Hence, the edge detection techniques are used as the base of another segmentation technique. To segment an object from an image closed region boundaries are needed [12], but the edges detected by edge detection techniques are discontinuous. These discontinuities are bridged if the distance between the two edges is within some predefined threshold. The most commonly used method is the detect and link method in which local disconuities are first detected and then linked to form longer, and hopefully complete, boundaries. But the disadvantage here is that the edges that are detected are guaranteed to form closed boundaries; thus, the subsequent thresholding scheme needs to merge regions that may not be uniform.

An improvement over this method is Yanowitz and Buckstein's adaptive thresholding method. The following description is substantially taken from [13].

Yanowitz and Buckstein's Adaptive Thresholding Algorithm

1. Smooth the image, replacing every pixel by the average gray-level values of some small neighborhood around it.

2. Derive the gray- level gradient magnitude image from the smoothed original. In discrete images, the gradient is actually computed as intensity difference over a small distance:

$$G(i, j) = \min \left(I(i, j) - I(i+\delta_i, j+\delta_j) \right) \tag{2.32}$$

$\delta_i = -1, 1, \quad \delta_j = -1, 1$

where I is the image being examined and G is the resulting image consisting of differences.

3. Thin the gradient image, keeping only points in the image that have local gradient maxima. This should produce a one pixel edge.

4. Sample the smoothed image at these maximal gradient (or edge) points. These points are assumed to be correct.

5. Interpolate the sample gray levels over the image. The result is a threshold surface with a possibly different threshold value for each pixel.

6. Using the obtained threshold surface, segment the image. If the original pixel value is greater than or equal to the threshold value at that location, set the threshold value to 1 (or white). Otherwise, set the value to 0 (or black). Thus, objects will be set to white and background to black.

2.1.3 Region growing methods

Region growing [16] is a procedure in which pixels or sub-regions are grouped into larger regions based on predefined criteria. This method first takes a set of seed points and from these, regions are grown by appending to each seed those neighboring pixels that have similar properties to the seed. Properties such as the difference between the pixel's intensity value and the region's mean can be used [21] as a measure of similarity. Pixels having the smallest difference between the intensity value and the region mean are allocated to the respective regions. This process is continued until all the pixels are allocated to a region.

The problem with this technique [16] is in how the starting points or seeds are selected. These starting points are typically selected based on the nature of the problem. When a priori information is not available, the same set of properties is computed at every pixel and is ultimately used to assign pixels to regions during the growing process. If clusters of values are shown as the results of these computations, the pixels whose properties place them near the centroid of these clusters can be used as seeds.

The selection of similarity criteria [21] depends not only on the nature of the problem but also on the type of image data available. For example, the analysis of land-use satellite imagery depends heavily on the use of color. This problem would be more difficult and may be impossible without the inherent information available in color images. In the monochrome images, region analysis must be carried out with a set of descriptors based on gray levels and spatial properties (such as moments or texture), but the descriptors alone can yield misleading results if connectivity or adjacency information is not used in the region growing process. If we consider random arrangement of pixels with only three distinct gray level values, grouping of pixels with the same gray level to form a region without paying attention to connectivity may result in segmentation that is meaningless.

Another problem [16] in region growing is the formulation of a stopping rule; the stopping rule controls the growth of a region. Growing should stop when no more pixels satisfy the criteria for inclusion in that region. Criteria such as gray level, texture and color do not take into account the "history" of region growth because the criteria considered are local in nature. Criteria such as concept of size, likeness between a candidate pixel and the pixels grown so far increase the power of a region growing algorithm.

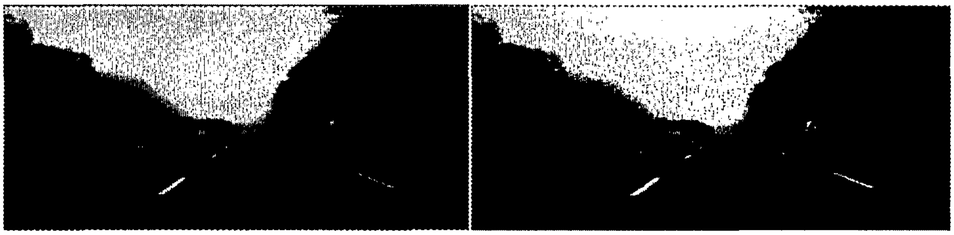
In this thesis, we implemented two image segmentation methods:

1. The d-peak algorithm which is histogram based, and
2. The quadtree algorithm which uses local characteristics and properties for regional segmentation.

In daylight viewing conditions [17] the contrast of an image is often significantly degraded by atmospheric aerosols such as fog and haze. This degradation of images by atmospheric aerosols restricts imaging applications to good visibility conditions. Therefore, the foggy images are enhanced using PhotoFlair. PhotoFlair [18] is a revolutionary new digital image or photo enhancement software. PhotoFlair uses the MSRCR image enhancement algorithm [25]. The MSR's automatic image enhancement is particularly good at correcting images degraded by atmospheric aerosols such as fog and haze. This also restores rich colors in the image [26]. Retinex based algorithms are explained in more detail later in the chapter. The proposed d-peak algorithm and quad tree segmentation are applied to retinexed images. Because of the quantization error values in the d-peak algorithm, false edges, lines and false regions are observed. To mitigate these effects, dithering is applied to the quantized images. Effects of implementing this dithering are also discussed in detail.

2.1.4 Histogram Based methods

Histogram based methods are very efficient when compared to other image segmentation techniques because they require only one pass through the pixels. In this technique, first the histogram of the given image is computed; then the peaks and valleys in the histogram are used to locate the clusters in the image. The peak seeking method is recursively applied to clusters in the image in order to divide them into larger clusters. In this thesis, we implemented a histogram based method that uses the d-peak algorithm for the purpose of image segmentation. The d-peak algorithm is a histogram based method. In this method pixels with similar characteristics are represented by the same gray-level value, and the objects of interest may fall into one of these regions. In order to speed up the processing, we apply the algorithms to an intensity only image. In other words, all color images are converted to grayscale before segmentation. Figure 2.1 shows a color image and its intensity only representation.



(a) Color image (b) Intensity only image

Figure 2.1: Intensity only representation of a color image.

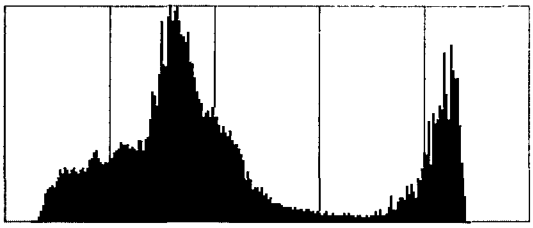


Figure 2.2: Histogram of the intensity only image.

Figure 2.2 shows the histogram of the intensity only image. Since this histogram is strongly bimodal, the histogram -based method can be used to convert it into a binary image as shown in figure 2.3.

Figure 2.3: Binary Image.

However, the object of interest in the image, the car, is also classified as background using this threshold, so we need more than a single threshold. This is can be done with the d-peak algorithm.

The image contrast and the quality of the image is degraded by the poor visibility conditions such as fog and haze. This degradation of images by poor visibility conditions restricts [16] the imaging applications to good visibility conditions. Hence the foggy images are enhanced before the d-peak algorithm is applied. Image enhancement is a process in which the interpretability or perception of information in images for human viewers is improved. It is a process by which better input can be provided for other automated image processing techniques.

2.2 Image Enhancement

Image Enhancement techniques can be divided into two broad categories: spatial domain and frequency domain. Spatial domain refers to the image plane, and in this category approaches are based on direct manipulation of pixels in an image. In frequency domain, processing techniques are based on modifying the Fourier transform of an image.

2.2.1 Spatial Domain Image Enhancement Techniques

In the spatial domain method, the value of a pixel with coordinates (x, y) in the enhanced image g(x, y) is the result of performing some kind of operation on the pixels in the neighborhood of (x, y) in the input image(x, y). Neighborhoods can be of any shape, but usually they are rectangular.

$$g(x, y) = T\,[f(x, y)] \tag{2.33}$$

where f(x, y) is the neighborhood in the input image at location (x, y) is the output image and T is an operator on f(x, y).

2.2.1.1 Grayscale Manipulation

The simplest form of operation is when the operator T only acts on a 1x1 pixel neighborhood in the input image; that is, g(x, y) only depends on the value of f(x,y).This is called gray scale transformation or mapping. The simplest case in grayscale manipulation is thresholding [23] where the intensity profile is replaced by a step function active at a chosen threshold value. In this case, any pixel with a gray level below the threshold in the image gets mapped to zero in the output image. Other pixels are mapped to 255.

2.2.1.2 Histogram Equalization

Histogram equalization (HE) is another image enhancement technique. Suppose we have an image that is predominantly dark. Its histogram would be skewed towards the lower end of the grayscale, and all the image detail is compressed into the dark end of the histogram. If we could expand or 'stretch out' the gray levels at the dark end then the image would become much clearer. Histogram equalization (HE) involves finding a gray scale transformation [23] that creates an output image with uniform histogram. The gray levels are normalized such that they lie between 0 and 1. The transformation, T, maps gray values, r, in the input image f(x, y) to gray values s=T(r) in the transformed image g(x,y).It is assumed that T is single valued and monotonically increasing.

$$0 \leq T(r) \leq 1 \text{ for } 0 \leq r \leq 1 \tag{2.34}$$

The transformation from s to r is given by

$$r = T^{-1}(s) \tag{2.35}$$

when the histogram of the input image is taken and is normalized so that the area under the histogram is 1, We have a probability distribution for gray levels in the input image $p_r(r)$.

If we transform the input image to get s=T(r) the probability distribution $P_s(s)$ is given by

$$p_s(s) = p_r(r)\frac{dr}{ds} \tag{2.36}$$

where $r=T^{-1}(s)$

Consider the transformation

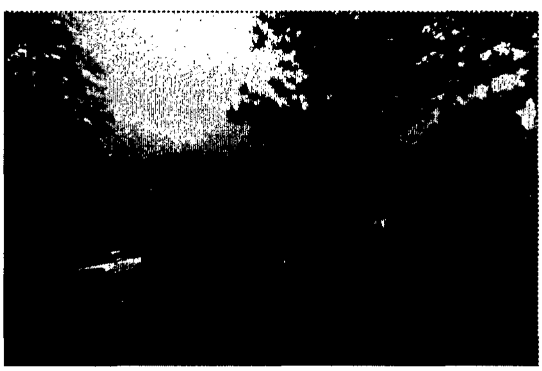
$$s = T(r) = \int_0^r p_r(w)dw \tag{2.37}$$

This is the cumulative distribution function of r. If the definition of T is used we see that the derivative of s with respect to r is

$$\frac{ds}{dr} = p_r(r) \quad (2.38)$$

Substituting this back into the expression for p_s, we get

$$p_s(s) = p_r(r) \times \frac{1}{p_r(r)} \qquad (2.39)$$



(a) Original Image

(b) Histogram Equalized Image

Figure 2.4: Implementation of Histogram Equalization-I.

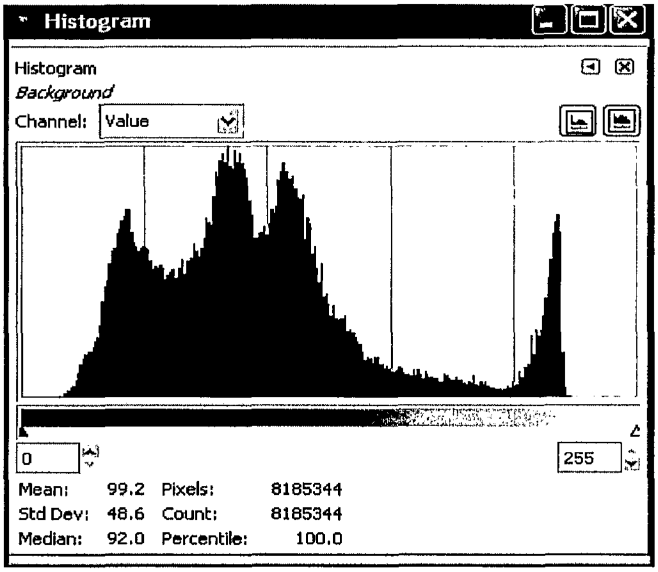


Figure 2.5: Histogram of original Image

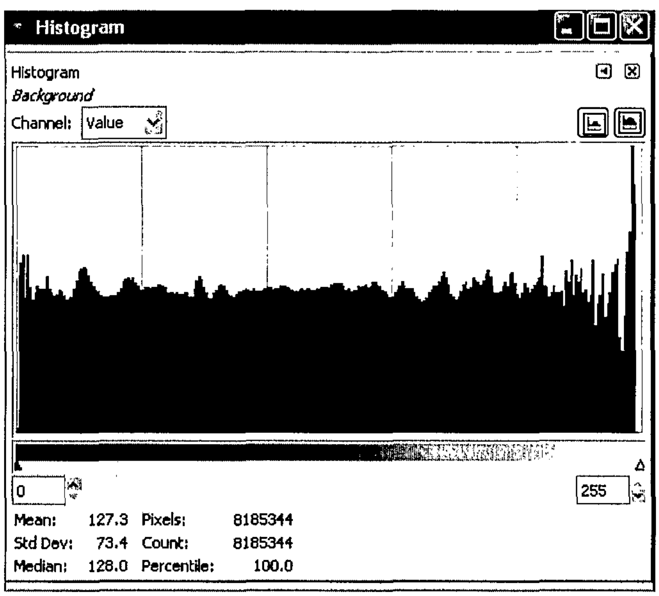


Figure 2.6: Histogram of the image

after histogram equalization

Figure 2.5 (a) above shows a histogram that is relatively narrow in Figure 2.5 (b) the gray levels are spread out and always reach white. This process increases the dynamic range of gray levels and consequently produces an increase in image contrast. This technique works well for scenes that have unimodal or weakly bi-modal histograms (i.e., very dark or very bright), but not so well for those images with strong bi-modal histograms (i.e., scenes that contain very dark and very bright images). Hence we use the image enhancement technique "retinex".

2.3 Retinex and Retinex based Algorithms

Multiscale Retinex is a method of bridging the gap [15] between the image of the scene and its (human) observation. Human perception has the ability to encompass wide dynamic ranges in a scene. It can perceive details across a large range of lightness and spectral variations. Additionally, human perception is color constant. However, recorded film and electronic cameras are typically unable to either encompass the full dynamic range or produce imagery that is color constant [29]. Since the dynamic range compression for perception of the recorded images is weaker than that for the scene itself, even with wide dynamic range imaging systems, the recorded images will not be seen [15] as real observation.

Edwin Land [17] presented the last version of his retinex as a model of lightness and color constancy. Hulbert [18] [19] studied lightness theories and showed that there is no mathematical solution to the problem of removing lighting variations. Moore [23][24] encountered scene context dependent limitations when he tried to implement a version of the retinex in analog VLSI for real time dynamic compression and hence failed to achieve a generalized implementation. Jobson et al. [14] defined a technique called single scale retinex (SSR) that is an implementation of center/surround retinex. Depending on the scale, the SSR can either provide dynamic range compression (small scale) or good tonal rendition (large scale). Superposition of weighted different scale single scale retinex is a choice to balance these two effects. This is called Multiscale scale retinex (MSR). The MSR forces regions of slowly or unchanging intensities to become gray. This problem can be solved by introducing a color correction factor that is applied to the output of the MSR [27]. This version of MSR is called the Multiscale retinex with color restoration (MSRCR) [28].

2.3.1 Single scale retinex (SSR)

The single scale retinex [15] is given by

$$R_i(x, y) = \log I_i(x, y) - \log [F(x, y) * I_i(x, y)] \tag{2.40}$$

where $I_i(x, y)$ is image distribution in the ith spectral band, $R_i(x, y)$ is the corresponding retinex output, and $F(x, y)$ is the normalized Gaussian surround function.

$$F(x,y) = Ke^{-(x^2+y^2)/C^2} \tag{2.41}$$

K is determined so that

$$\iint F(x,y)dxdy = 1 \tag{2.42}$$

and C is the Gaussian surround space constant.

The image distribution is the product of a scene's reflectance and illumination.

$$I_i(x, y) = S_i(x, y)\, r_i(x, y) \tag{2.43}$$

where $S_i(x, y)$ is the spatial distribution of illumination and $r_i(x, y)$ is the distribution of a scene's reflectance. The convolution with surround function works as averaging in the neighborhood. So that

$$R_i(x, y) = \log \frac{S_i(x, y) r_i(x, y)}{\overline{S_i(x, y) r_i(x, y)}} \tag{2.44}$$

Generally, the illumination has low spatial variation, which means

$$S_i(x, y) \approx \overline{S_i(x, y)} = S_0 \tag{2.45}$$

It is apparent that by using this result color constancy, independence from spatial distribution of illumination is achieved.

$$R_i(x, y) \approx \log \frac{r_i(x, y)}{\overline{r_i(x, y)}} \tag{2.46}$$

Various surround functions can be used for the surround operation. Edwin land proposed an inverse square spatial function

$$F(x, y) = {}^{1}/_{r^2} \tag{2.47}$$

Moore suggested the exponential formula with absolute parameter,

$$F(x, y) = \exp(-|r|/c) \tag{2.48}$$

And Hulbert suggested a Gaussian surround

$$F(x, y) = \exp(-r^2/c^2) \tag{2.49}$$

Jobson et al [14][15] stated that Gaussian has a property of being more regional and offers good dynamic range compression over a large range of space constant. The space constant is related to the visual angle in direct observation. The value of the space constant cannot be theoretically modeled and determined. Basically there is a tradeoff between dynamic range compression and color rendition.

2.3.2 Multiscale retinex (MSR)

Because of the tradeoff between dynamic range compression and color rendition, we have to choose a good value of c in the formula of F(*x*, *y*) in SSR. If we want both dynamic range compression and color rendition, Multiscale retinex, which is a combination of weighted different scales of SSR, is a good solution [16].

$$R_{MSRi} = \sum_{n=1}^{N} w_n R_{ni} \tag{2.50}$$

where N is the number of scales and R_{ni} is the ith component of the nth scale. The obvious question about MSR is the number of scales needed, scale values and weight values. Generally fixed scales of 15, 80 and 250 can be used, or scales of fixed portion of image size can be used. The weights can be adjusted to weigh more than dynamic range compression or color rendition.

2.3.3 Color restoration method for MSR (MSRCR)

Generally, the effect of MSR processing on images with regional or global gray world violations is a "graying out" of the image. This may be either in global or specific regions [15], but this desaturation of color may be severe in some cases. More rarely, the gray world violations can simply produce an unexpected color distortion. Hence, a color restoration scheme which provides good color rendition for images that contain gray world violations is considered [16]. Weights for the three color channels depending on the relative intensity of the three channels in the original images are introduced [16].

$$C_i(x,y)=f[I_i'(x,y)] \tag{2.51}$$

The relative intensity of three channels is given by [9]

$$I_i'(x,y) = I_i(x,y) / \sum_{i=1}^{S} I_i(x,y) \tag{2.52}$$

where i is the ith color band and S is the number of spectral channels. Generally, S=3 using the red-green-blue color space. The color restoration function should be monotonic. Several linear and non linear functions were tried on a range of text images. Jobson et al [15] found that the best overall color restoration was

$$C_i(x,y) = \beta \log[\alpha I_i'(x,y)] \tag{2.53}$$

This color restoration method can be described as

$$R_{MSRCRi}(x,y) = C_i(x,y) R_{MSRi}(x,y) \tag{2.54}$$

2.3.4 MSR with gain/offset

Generally, the range of the output values of MSR processing will be out of the display domain. The output needs to be shifted and expanded to the display domain. Fixed values of gain and offset are used [26].

(a) Foggy Image (b) Retinexed Image

Figure 2.7: Implementation of image enhancement.

2.4 Quadtree Segmentation

Quadtree segmentation is a recursive operation. It starts with the input image and divides it [22] into four quadrants of equal size if the image satisfies some predefined criterion. Each quadrant is recursively examined [22] with this criterion, and further subdivisions and analysis occur until either all of the quadrants meet the criteria or a minimum image quadrant size is reached. The criteria used is that the mean and standard deviation of the quadrant should lie in a particular range, i.e., between two particular values. The values, mean and standard deviation, are chosen such that the quadtree is able to detect objects in poor visibility conditions.

2.5 Summary

In this chapter, the theory and principles of some of the basic and advanced image segmentation techniques were discussed. Pixel based segmentation schemes using mode method, edge detection methods, and region growing methods were described in detail. The proposed algorithms, the d-peak algorithm and quadtree algorithm, were discussed briefly. The algorithms were applied to foggy and enhanced images. The images were enhanced before the algorithms were applied. Different enhancement techniques, gray scale manipulation and Histogram equalization were explained in detail. "Retinex" and "Multiscale Retinex", image enhancement techniques were explained in detail.

CHAPTER 3

PROPOSED ALGORITHMS FOR IMAGE SEGMENTATION

Two image segmentation algorithms, the d-peak algorithm and quadtree, have been implemented in this thesis.

The following block diagram clearly shows the steps in detecting the objects in poor visibility conditions using the image segmentation algorithms proposed in this thesis.

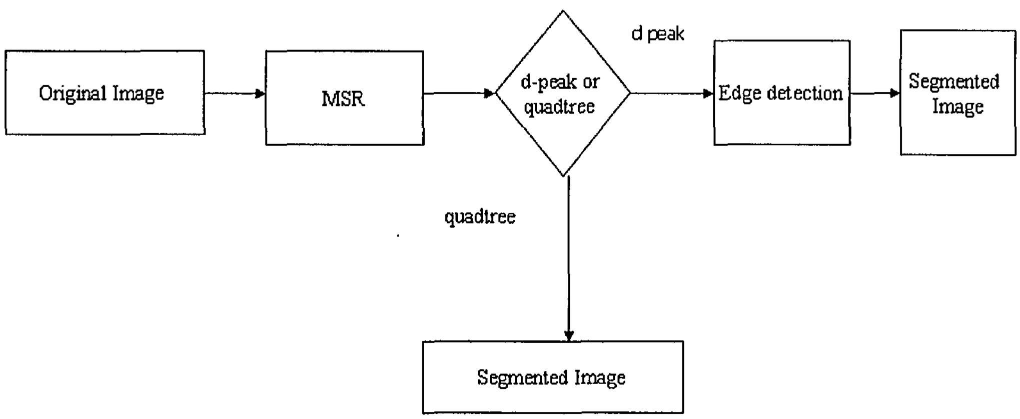
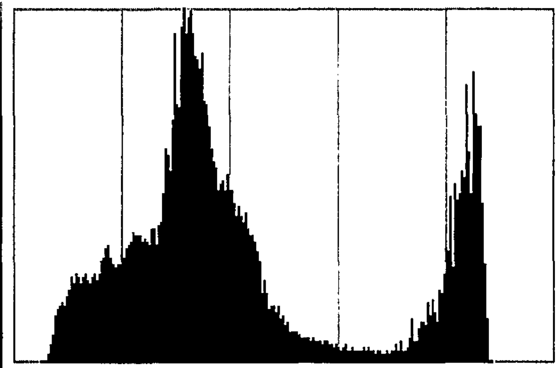


Figure 3.1: Block diagram of object detection in poor visibility conditions using the proposed algorithms.

The original is enhanced using the PhotoFlair which uses the concept of Multiscale Retinex, and then the image is segmented using the d-peak algorithm and quadtree algorithm. If the image is segmented using the d-peak algorithm, the d-peak algorithm itself is not sufficient to interpret the images; hence, edge detection is employed for images segmented using the d-peak algorithm.

The d-peak algorithm is a histogram based method. In this method, pixels with similar gray level values are grouped into regions, and the objects of interest may fall into one of these regions.



(a) Foggy image (b) Histogram of Figure 3.2 (a)

Figure 3.2: Image and its histogram representation.

In the histogram, the x-axis represents the gray level value, and the y-axis, the number of pixels at that particular gray level.

3.1 Proposed Algorithm

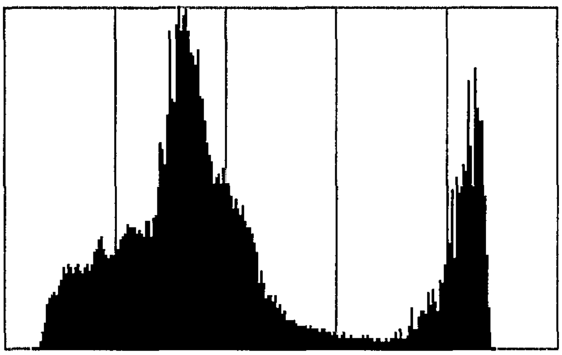
d-peak Algorithm steps:

1. Let C_i be the number of pixels at gray level i. C_i is compared with C_{i-1} andC_{i+1},

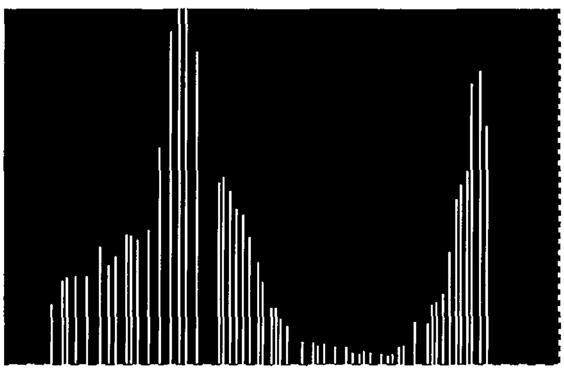
C =1,.., L-1.

2. If $C_i > C_{i-1}$ and $C_i > C_{i+1}$, there is a potential peak at i. Only potential peaks are considered further. Figure 3.2(b) shows the histogram after the application of the first two steps. Only potential peaks remain after the processing.

After the implementation of first two steps, the histogram for the above image is as follows:



(a) Histogram of an image



(b) Histogram after first two steps of d-peak algorithm

Figure 3.3: Implementation of first two steps of d-peak algorithm.

3. For the peaks that remain, find the gray level k, corresponding to the maximum peak.

k=arg (max(C_i)),i=0,L-1.

If C_k is the maximum value then the following condition is applied.

If $C_i > p\ C_k$ retain peak

Else discard peak

4. Repeat step 2 and step 3 on the modified histogram. Now we have only a few gray levels, which satisfies step 2, step 3 and step 4.

5. If there are a total of 'm' peaks at gray level $g_1, g_2, g_3, \ldots, g_m,$ all the pixels whose values fall between g_i and g_{i-1} $i=0,..,m+1$ are assigned the value $\frac{1}{2}(g_i + g_{i-1})$ where $g_0 = 0$ and $g_{m+1}=255$.

The following figure shows a diagram that clearly explains the d-peak algorithm steps:

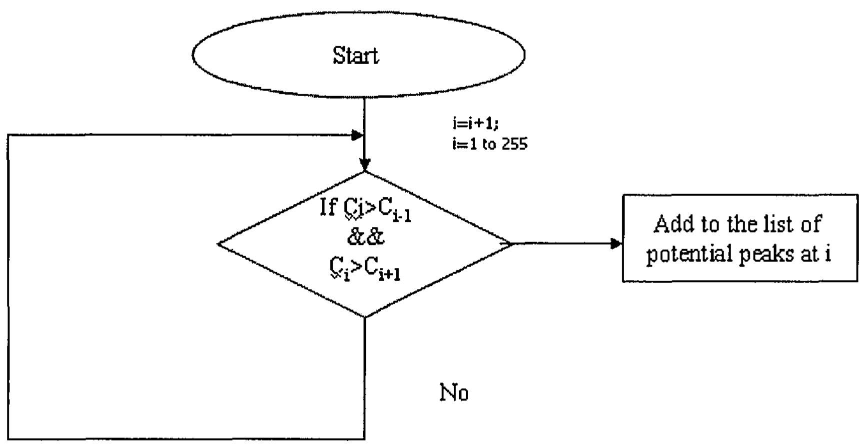


Figure 3.4: Flow Chart showing d-peak algorithm steps.

After the potential peaks are found:

1. Find the maximum peak,

2. Keep all those pixels that are above $pC_{max,}$.

3. Repeat this until all the potential peaks have been examined,

4. If there are 'm' peaks all the pixels whose values fall between g_i and g_{i-1} $i=0,..,m+1$ are assigned the value $\frac{1}{2}(g_i + g_{i-1})$ where $g_0 = 0$ and $g_{m+1}=255$.

Thus, a large range of values is compressed to a single value. Thus, pixels with similar characteristics are grouped together and the objects of interest may fall into any one of these regions therefore detecting the objects of interest on the runway. In order to speed up the processing, we apply the algorithm to an intensity only image. In other words, all color images are converted into gray scale before segmentation. Figure 3.5 shows a color image and its intensity only representation. The value of P in step 3 is variable, P= {0.10, 0.20, 0.30, 0.40}.

The threshold value is chosen to be P=0.10 for foggy images. This is because when the algorithm is applied to foggy images with different threshold values, it is observed that the objects of interest are separated from the background when P=0.10. The object of interest in Figure 3.5 is a car and is marked by a rectangle. Figure 3.5 shows the conversion from a color to intensity only image.

(a) Color Image

(b) Intensity only image

Figure 3.5: Representation of intensity only image of a color image.

Figures 3.6-3.13 show examples of d-peak segmentation for different values of P.

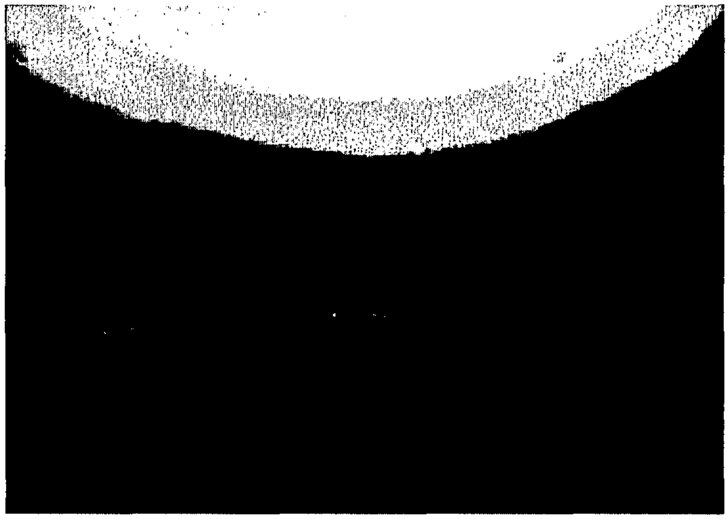


Figure 3.6: Segmented image if P=0.40.

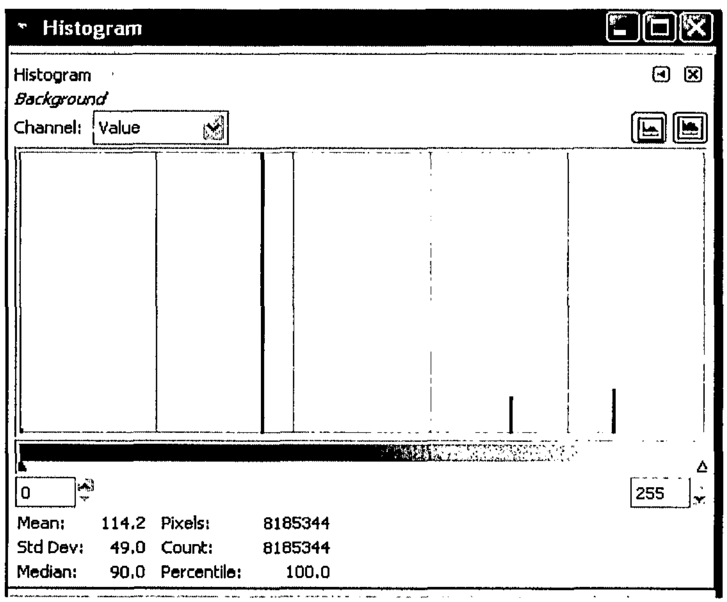


Figure 3.7: Histogram of figure 3.6.

When P=0.40 the number of peaks for the above image is observed to be 3. This is a general trend: larger values of P lead to fewer peaks. It is observed that the object of interest in the image, the car is also classified as background in Figure 3.6.

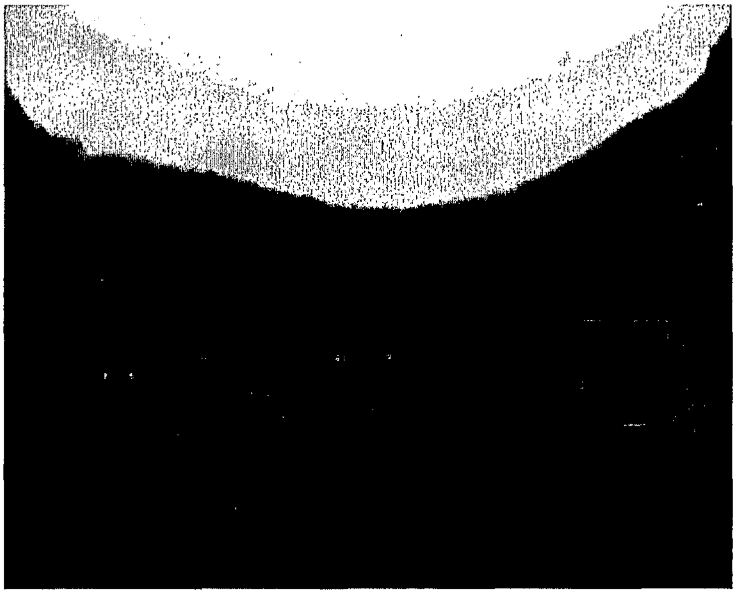


Figure 3.8: Segmented image if P=0.30.

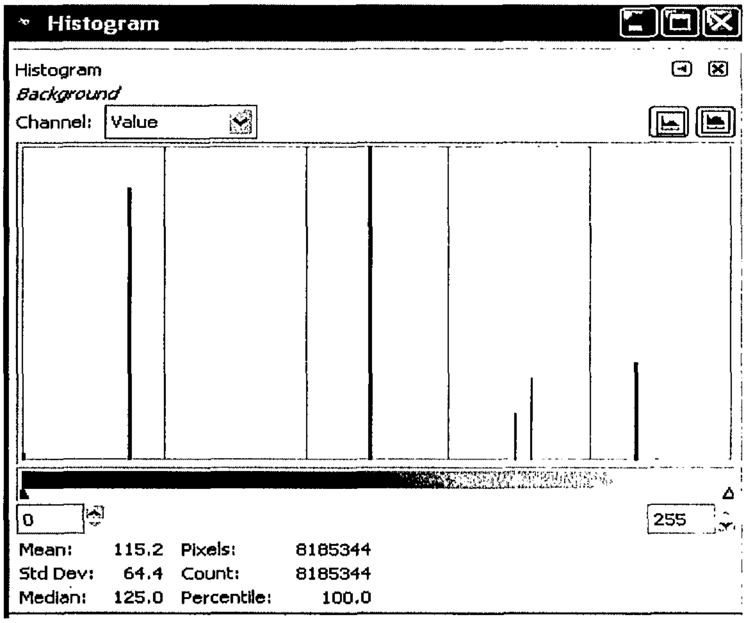


Figure 3.9: Histogram of figure 3.8.

Figure 3.8 shows the results of the d-peak segmentation when P=0.30.Though the car is "visible" now, it is still difficult to recognize it. The number of peaks increases to 5 for this value of P.

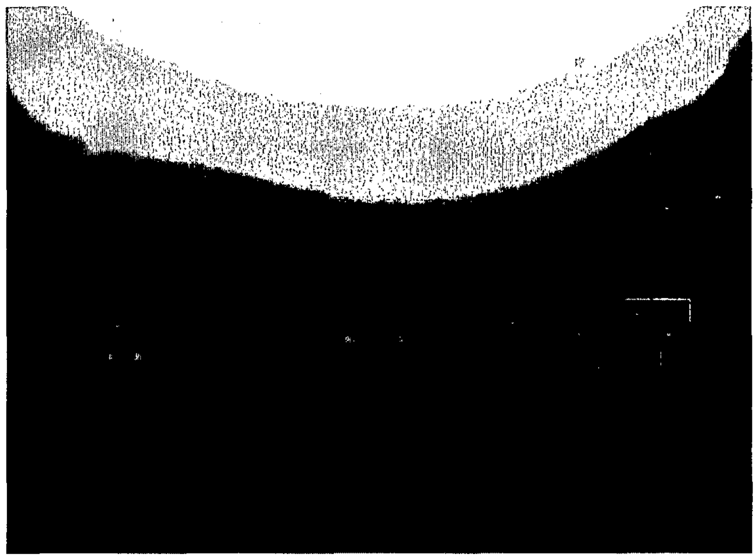


Figure 3.10: Segmented image if P= 0.20.

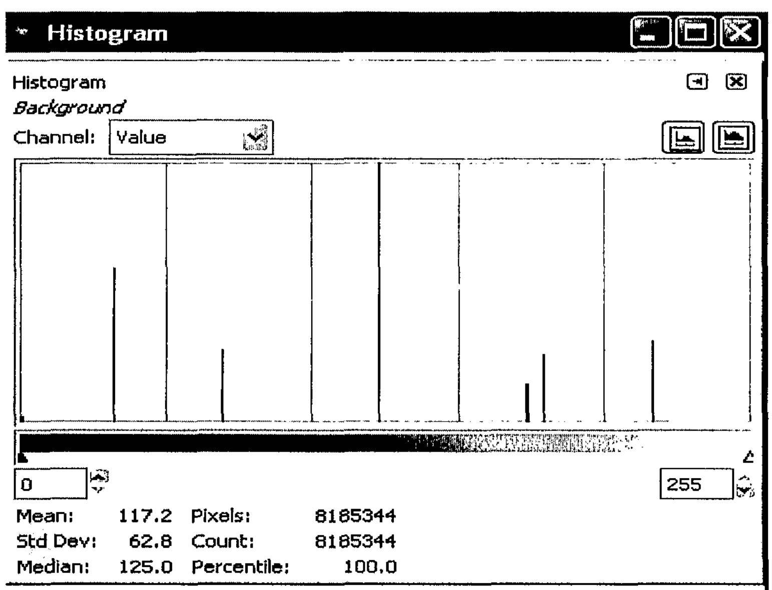


Figure 3.11: Histogram of figure 3.10.

Figure 3.10 shows the result for d-peak segmentation with P=0.20. It is observed that the object of interest in the image, the car, is separated from the background, and it is far better when compared to the image in Figure 3.6 and Figure 3.8. Finally, when P=0.10, the number of peaks remains at 6, so there is no change in the segmentation as the value of P changes from 0.2 to 0.1.

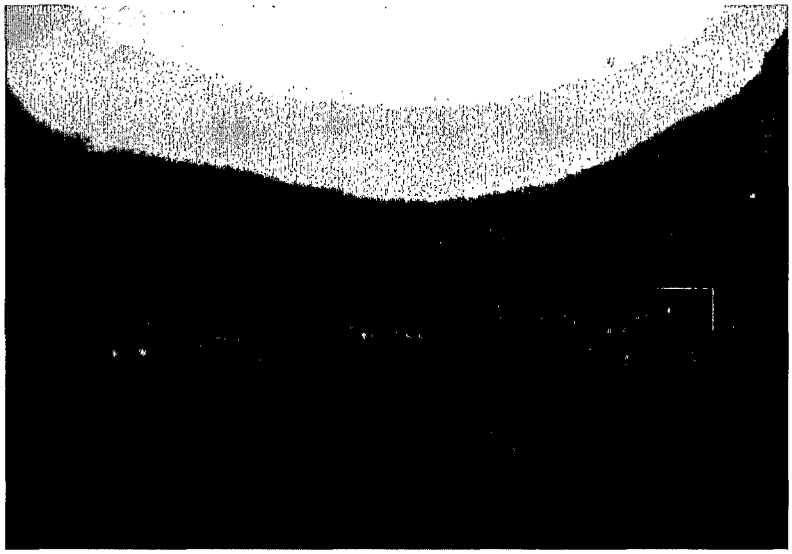


Figure 3.12: Segmented image if P= 0.10.

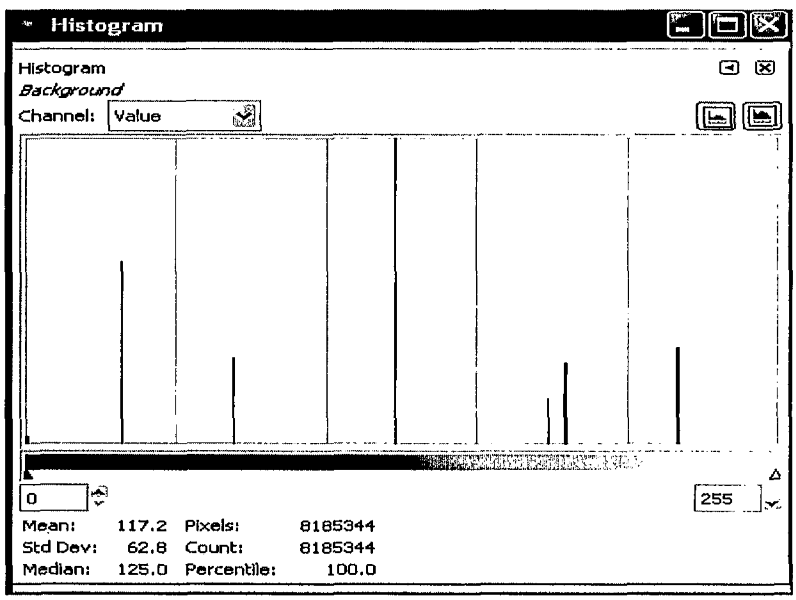


Figure 3.13: Histogram of figure 3.12.

We choose P=0.10 to segment foggy images. This is because the separation of objects from the background is much better when compared to the separation of objects when P=0.30 or 0.40. The algorithm was tested on a number of images, and for all the images the results were better at P=0.10. The image contrast and the quality of the image are degraded by the poor visibility conditions such as fog and haze. This degradation of images by poor visibility conditions restricts the imaging applications to good visibility conditions. Hence, foggy images are enhanced and the d-peak algorithm is applied to the enhanced images with different values of P. Figure 3.14 shows the MSR enhanced version of Figure 3.5(a). The MSR processing brings out a lot more detail than was evident in the original image. Additionally, the processed image has better contrast and brightness. Figures 3.16-3.25 show the results of applying the d-peak algorithm to the image shown in Figure 3.14.

Figure 3.14: Retinexed Foggy Image.

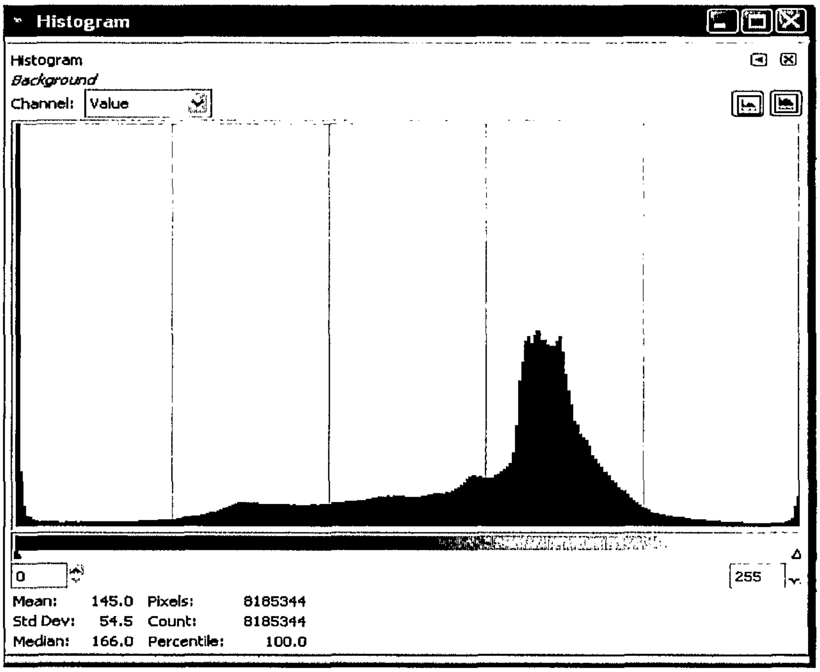


Figure 3.15: Histogram of figure 3.14.

Figure 3.16: Retinexed image with P =0.40.

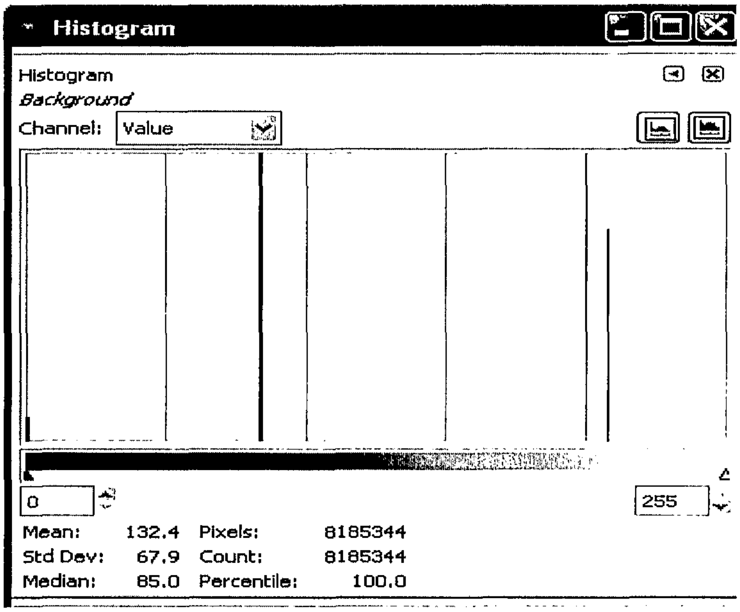


Figure 3.17: Histogram of figure 3.16.

Figure 3.18: Retinexed image with P=0.30.

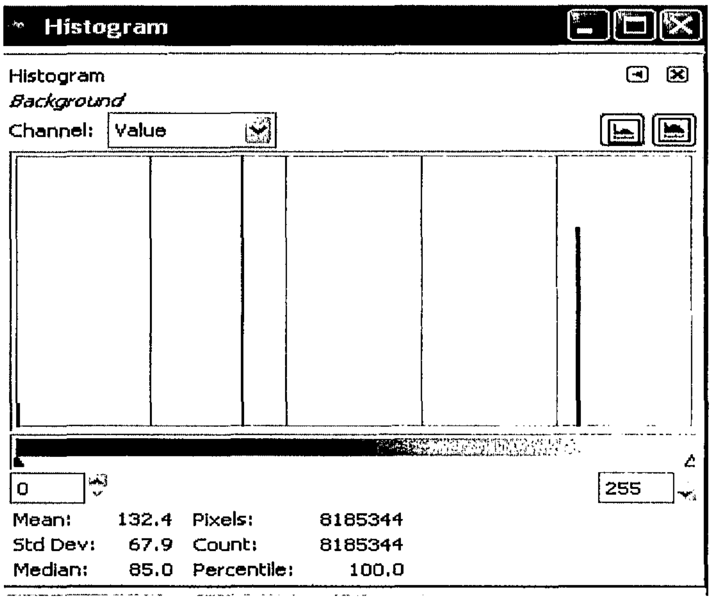


Figure 3.19: Histogram of figure 3.18.

Figure 3.20: Retinexed image with P =0.20.

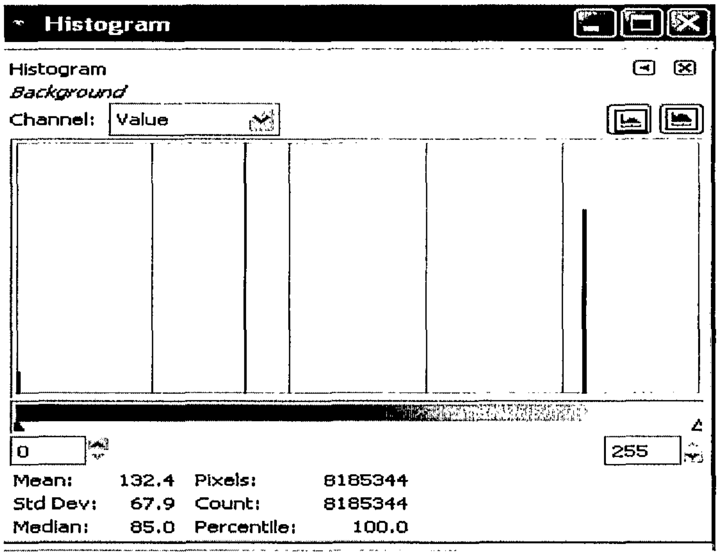


Figure 3.21: Histogram of figure 3.20.

Figure 3.22: Retinexed image with P =0.10.

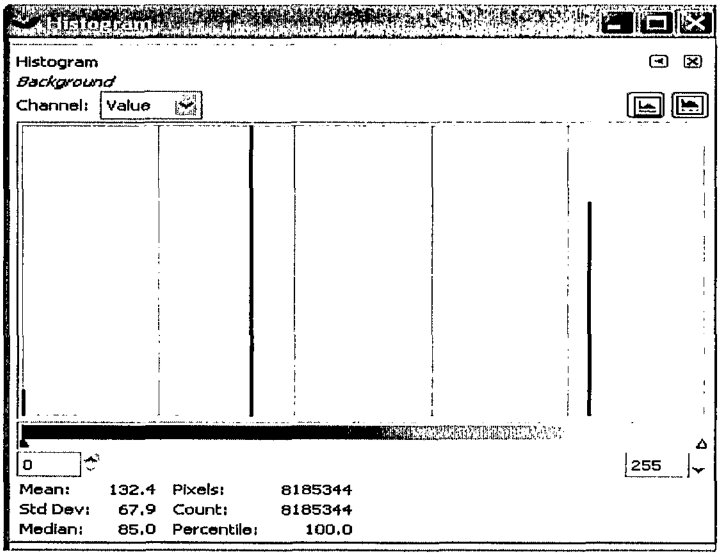


Figure 3.23: Histogram of figure 3.22.

As P changes from 0.40 to 0.10, there is no impact on the performance of the d-peak algorithm. This is because the MSR has a tendency to concentrate pixel values towards middle gray, i.e. i=128.

Figure 3.24: Retinexed image with P=0.05.

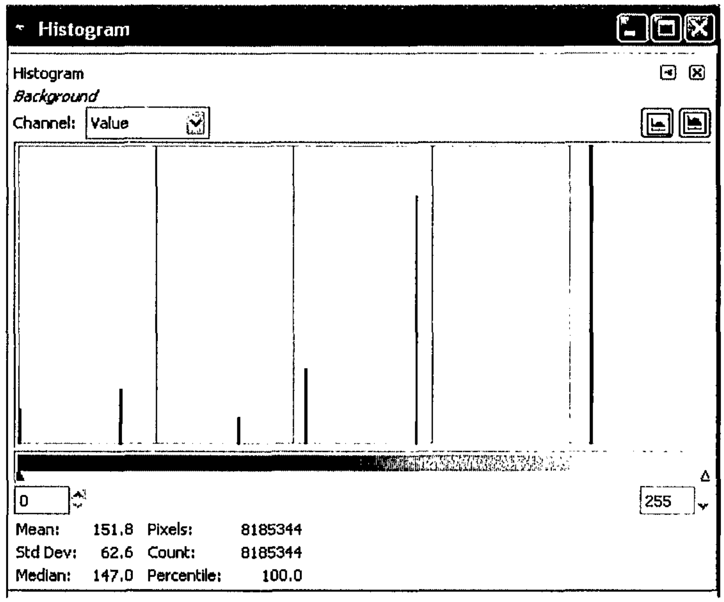


Figure 3.25: Histogram of figure 3.24.

Finally, we tested the d-peak algorithm with P=0.05. For this case, the object of interest, the car, is clearly separated from the background, so the threshold value for the retinexed images is chosen to be P=0.05. Further testing corroborates this result. While the d-peak algorithm produces a quantized version of the original image, it keeps most of the information at a great cost savings in terms of bits needed to represent the data.

3.2 Dithering

In the above images obtained after applying the d-peak algorithm we observe false edges, lines and regions due to false contouring. Since a large range of values is compressed to a single value, significant differences exist between the neighboring representations. The quantization error value of a pixel corresponds to a difference between its quantized value and its original value. As the number of gray levels used to represent an image decreases, the number of bits per pixel used to represent an image also decreases. Because of the quantization error and decrease in the number of gray levels, we observe false contouring in the above images. To mitigate the false contouring effect visually, we employ a technique called dithering. Dithering is the process of intentionally adding noise to randomize quantization error and mitigate the false contouring effect visually. In quantization, the values in the pixel are simply rounded to the nearest level whereas the dithering algorithms attempt to spread the quantization errors around in such a way that the eye notices them less. We described various techniques in dithering in chapter 2. We employ Gaussian dithering, where white Gaussian noise is added to the quantized gray level values.

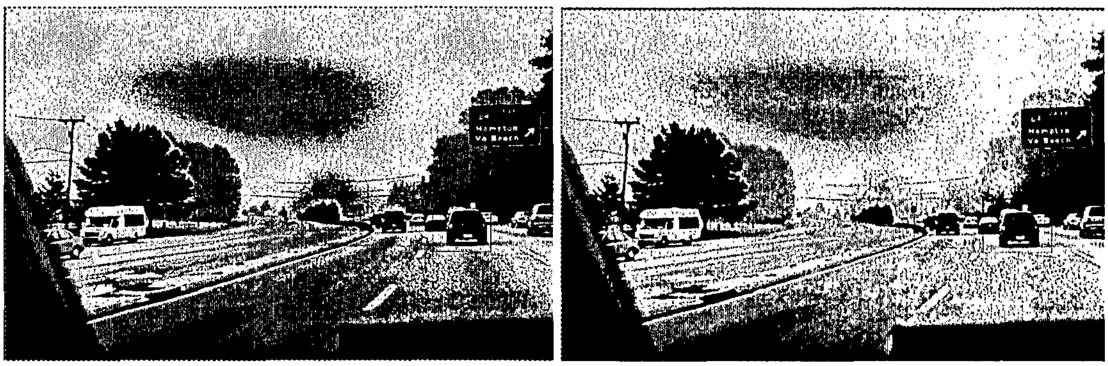
In Figure 3.25, it is observed that there are six gray level values in the histogram of an image; Gaussian noise is added to the quantized pixel values of the image. Gaussian noise is generated by the Box-Muller method. Box-Muller transform is a method of generating pairs of independent standard normally distributed random numbers. That is, random numbers with zero expectation and unit variance, given a source of uniformly distributed random numbers. The Box-Muller transform has two forms: the basic form and the polar form. In the basic form two samples from the uniform distribution on the interval (0, 1] are taken, and these two samples are mapped to two normally distributed samples.

If x and y are two uniformly distributed independent random variables in the interval (0, 1], then these variables are mapped to two independent normally distributed random variables with $\mu=0$ and $\sigma=1$ as follows:

$$Z_0=\sqrt{-2\ln(x)}\cos 2\pi y \tag{3.1}$$

$$Z_1=\sqrt{-2\ln(x)}\sin 2\pi y \tag{3.2}$$

Z_0 and Z_1 are independent random variables with normal μ=0 and σ=1. This generates Gaussian values with μ=0 and σ=1.The Gaussian values with zero mean and standard deviation of one are multiplied by $\frac{1}{m}$ of the standard deviation of the images, σs, and added to the quantized gray level values of the image.



(a) Before dithering (b) After dithering

Figure 3.26: Application of Gaussian dithering with

μ=0 and σ= $\sigma s/16$.

The value of m may be any integer value greater than zero. The algorithm is implemented using different values of m. From Figure 3.26 it is observed that there is no difference in the images before dithering and after dithering.

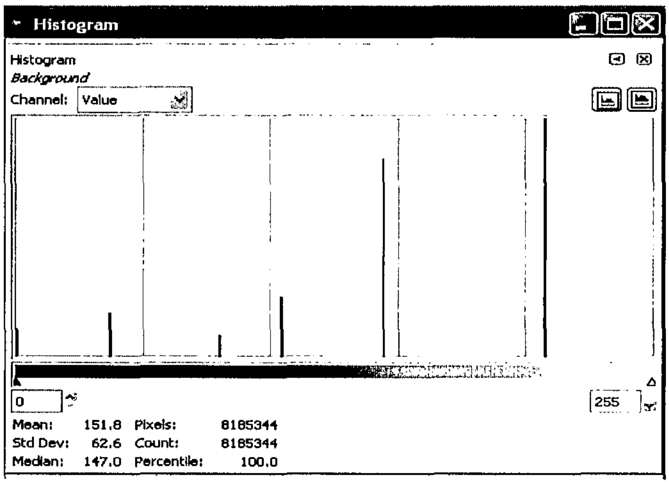


Figure 3.27: Histogram before dithering.

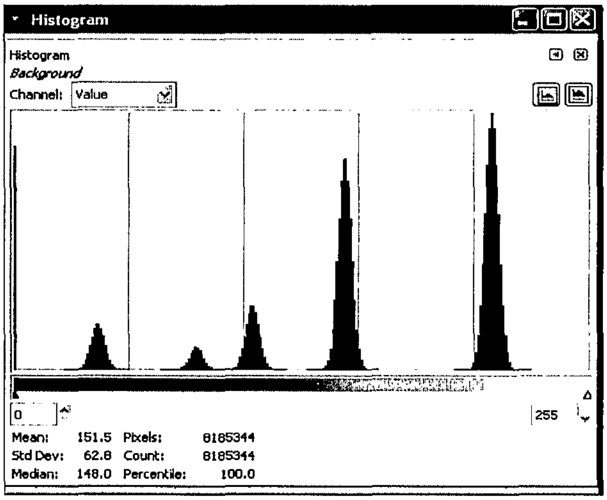


Figure 3.28: Histogram after dithering.

From Figure 3.28 it is observed that the quantization errors are spread by Gaussian dithering. We further tested the algorithm for σ =σs/8 to see if there is a better result of dithering, and it is observed that there is no difference in the images before dithering and after dithering.

(a) Before dithering (b) After dithering

Figure 3.29: Application of Gaussian dithering with μ=0 and σ= σs/8.

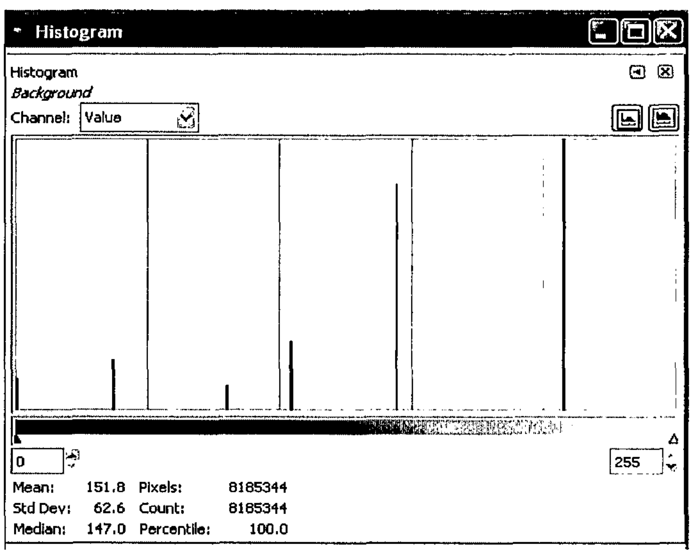


Figure 3.30: Histogram before dithering.

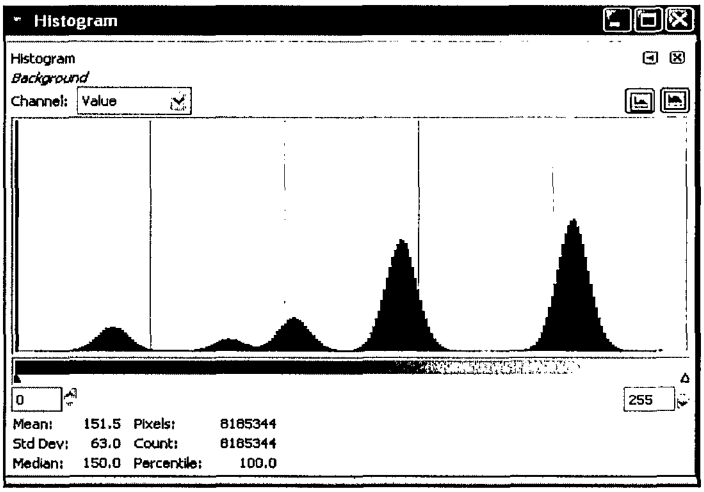


Figure 3.31: Histogram after dithering.

A final test of dithering was conducted for σ=σs/4 to see if there is a better result of dithering, but as Figure 3.32 shows, there is no difference in the images before dithering and after dithering. It is observed that the effect of false contouring is not mitigated after dithering for these cases. This may be because there are very few peaks obtained after applying the d-peak algorithm. Therefore, even if the quantization errors are spread they still remain confined to the quantization bin, and there is no mitigation of the false contouring effect.

(a) Before dithering (b) After dithering

Figure 3.32: Application of Gaussian dithering with μ=0 and σ=6s/4.

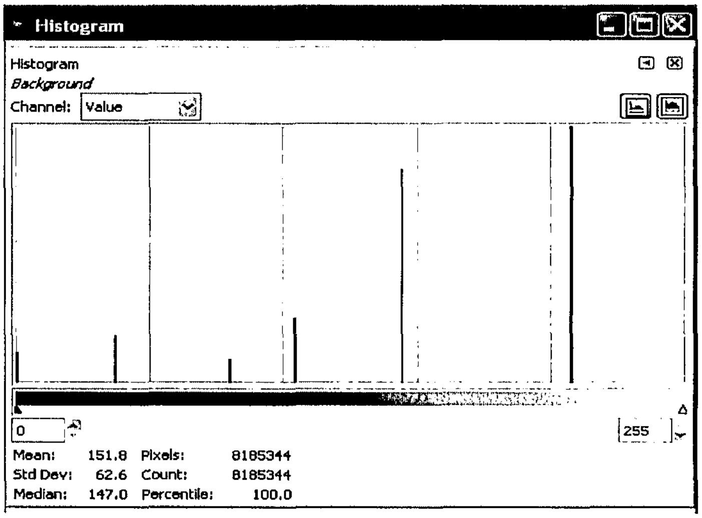


Figure 3.33: Histogram before dithering.

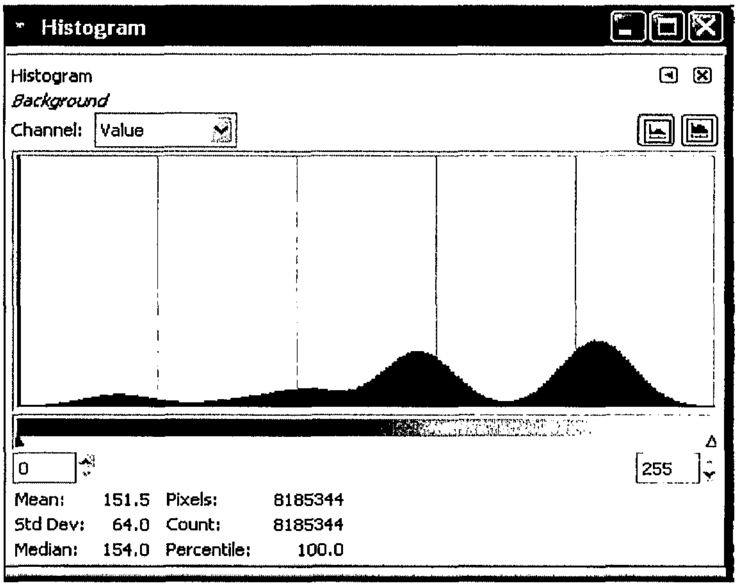


Figure 3.34: Histogram after dithering.

3.3 Edge Detection

Edges are of fundamental importance in image processing as edges characterize boundaries. Edges in images are areas with strong intensity contrasts, a change in intensity from one pixel to the next. Edge detecting an image significantly reduces the amount of data and filters out useless information, while preserving the important structural properties in an image.

3.3.1 Canny edge detection

Canny edge detection is the optimal edge detection algorithm. Currently, the canny operator is considered the best mask edge detection method. A large number of edge detection operators have been developed, but none have so far shown crucial advantages over the Canny-type operators in general situations [29].

Canny specified three issues that an edge detector should consider in detecting edges.

- Error rate: An edge detector should be able to detect all the edges and should not miss any edges.
- Localization: distance between the detected edges and the actual edges should be as small as possible.
- Response: it should not respond more than once to the same edge.

It basically consists of five steps:

- Firstly, the original image is filtered to remove noise using a Gaussian smoothing filter. The smoothed image is used for finding the edge gradients using either a Sobel operator or first derivative of Gaussian filter (G'). The edge strength is found by approximating the gradient values as shown in Equation,3.3

$$|G| = |G_X| + |G_Y| \tag{3.3}$$

- Edge direction is found by using G_x and G_y as shown in Equation 3.3. Once the edge orientation is known, the next step is to relate the edge direction to a direction that can be traced in an image. Edge orientation has to be resolved into one of the four directions (0°, 45°, 90°, 135°) depending on which direction the pixel value is closest to as shown in Figure 3.35.

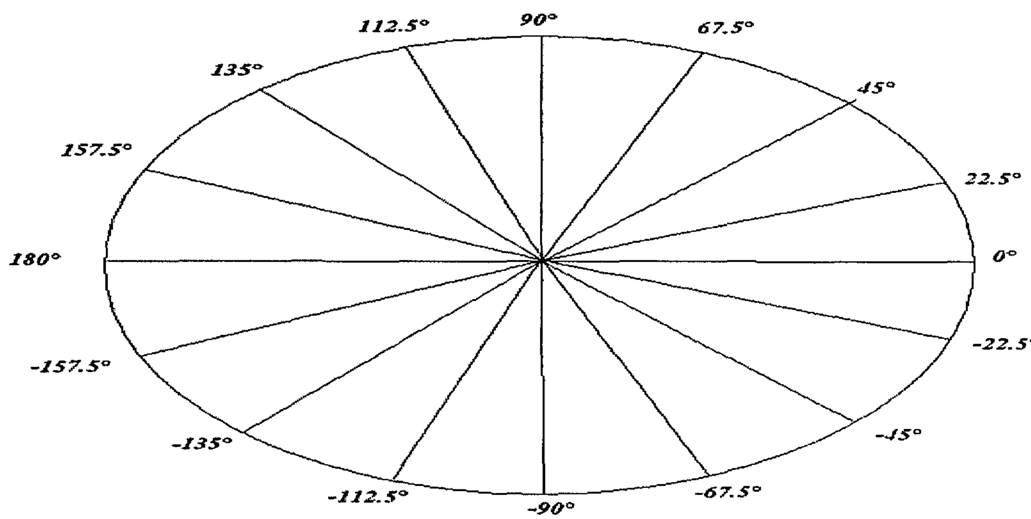


Figure 3.35: Canny operator: Tracing of the edge direction.

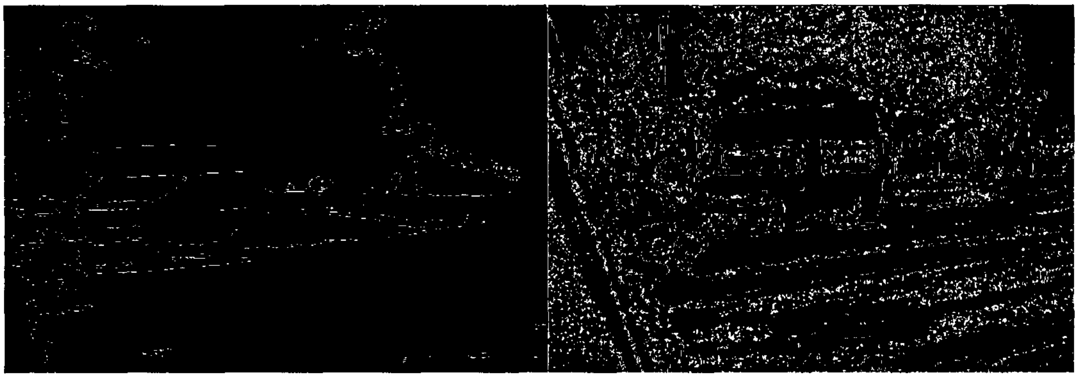
If the edge direction falls within the range

$|\theta| < \frac{\pi}{8}$ or $|\theta| < \frac{7\pi}{8}$ θ is set to 0°

$\frac{\pi}{8} < \theta < \frac{3\pi}{8}$ or $-\frac{7\pi}{8} < \theta < -\frac{5\pi}{8}$ θ is set to 45°

$\frac{3\pi}{8} < \theta < \frac{3\pi}{8}$ or $-\frac{5\pi}{8} < \theta < -\frac{3\pi}{8}$ θ is set to 90°

$\frac{5\pi}{8} < \theta < \frac{7\pi}{8}$ or $-\frac{3\pi}{8} < \theta < -\frac{\pi}{8}$ θ is set to 135°



(a) Canny on Foggy Image (b) Canny on retinexed images

Figure 3.36: Results of Canny edge detection operator on foggy and retinexed images.

From Figure 3.36 it is observed that the retinexed image is noisier but picks up more details than the foggy image.

3.4 Quadtree Segmentation

When humans look at some scene, the eye-brain visual system subdivides a complex scene into simple objects and then detects objects of interest [13]. This image segmentation is applied to detect objects on the runway, detect cancer cells from medical image or detect a way from satellite pictures. Quadtree segmentation is being implemented in this thesis to divide the image into similar regions based on brightness and contrast of the image. Mean value is the image statistic by which the brightness of the image is measured. Mean (average gray level) for an M x N image is defined by

$$\bar{g} = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} g[m][n] \tag{3.4}$$

It should be clear that $\overline{g}$ is a real value statistic bounded between 0 and 255.Image contrast is the global amount of image gray level dispersion that is the variation about the mean gray level. Contrast of the image is numerically characterized by the statistic called variance. Variance is given by

$$\sigma_g^2 = \frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} g^2[m][n] - \overline{g}^2 \tag{3.5}$$

Variance is the global mean squared deviation about the mean gray level. This means the variance value is zero if and only if the image is uniformly gray; intensity value is constant throughout the image, but the image variance cannot be used as a measure of contrast that is dimensionally incorrect because the "units" of $\|g - \overline{g}\|$ are squared. Hence, we use the square root of the variance, $\|g - \overline{g}\|$,which is the standard deviation. Hence, the contrast of the image is numerically characterized by standard deviation.

Quadtree decomposition is a recursive operation. A condition is predefined such that the mean value and the variance value should lie in a particular range. First it starts with the original image as the input of size M x N. Then, the condition is checked, the mean value and the variance value of the image should lie in the range defined in the condition. Then the image is divided into four blocks each of size M/2 x N/2. Then each of the four blocks is checked to determine whether it satisfies the condition that is the mean and the variance value of each block is checked to see if the mean and the variance value should lie between the ranges defined in the condition. This recursive operation of checking each of the blocks is continued until the condition on the size of the block is reached. The condition on the size of the block here is that the row size and the column size of the block should be 0.08 of the image size.

A flow chart displaying the operation of the quadtree algorithm is shown below:

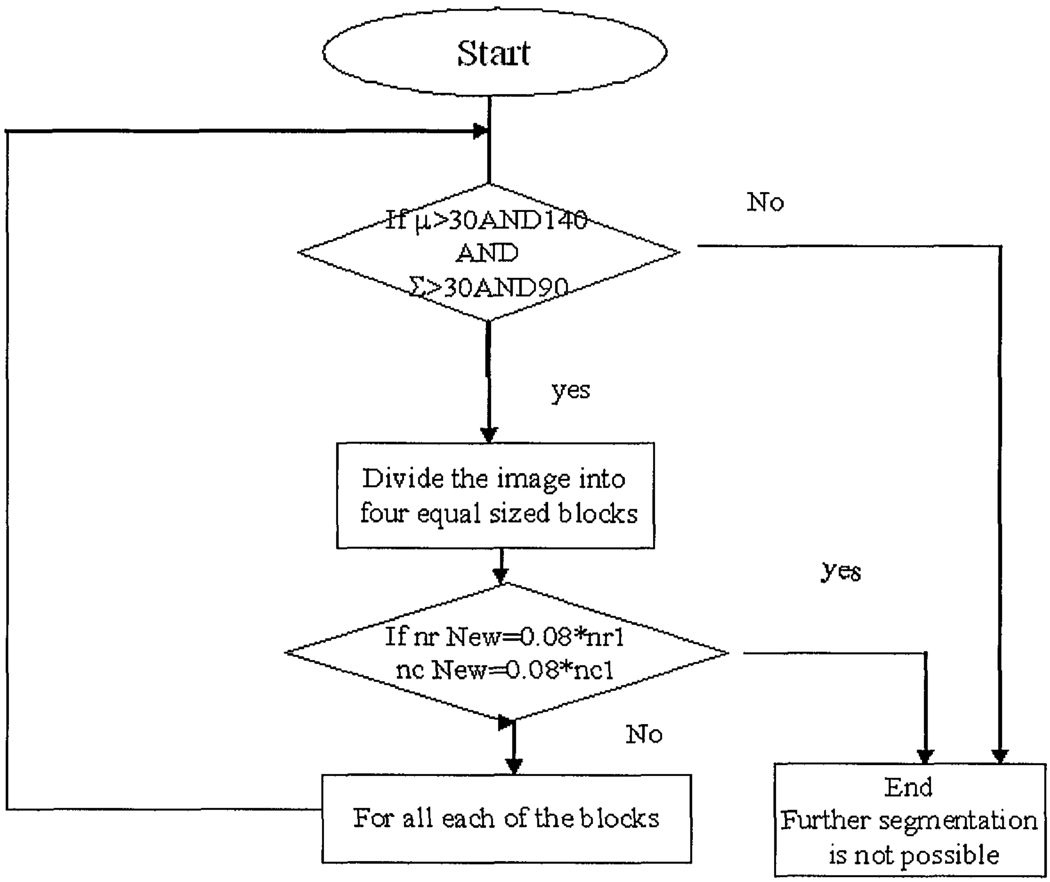
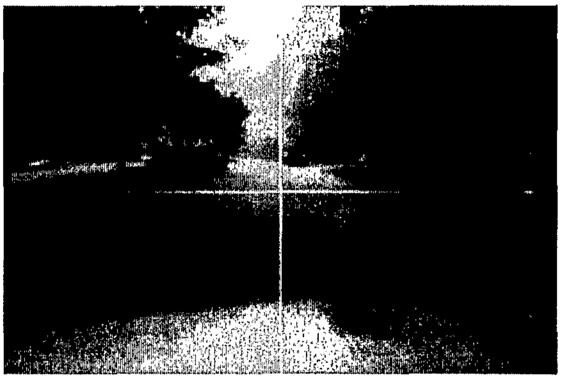
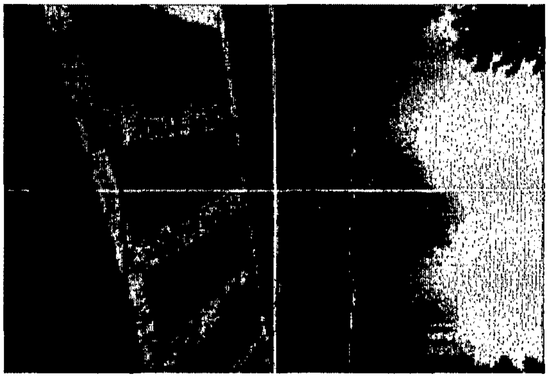


Figure 3.37: Flowchart showing the operation of quadtree algorithm.



(a) Original Foggy Image



(b) Original Foggy Image

Figure 3.38: Application of first step of quadtree algorithm.

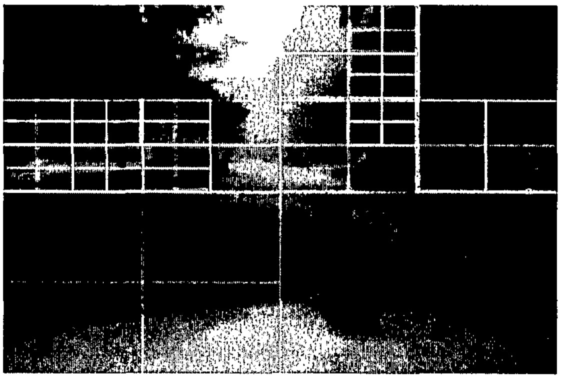


Figure 3.39: Application of quadtree algorithm on the foggy image.

The contrast and the quality of the image is degraded by the poor visibility conditions, and this degradation restricts the imaging applications to good visibility conditions. This is why most of the algorithms are applied to the retinexed images. The original foggy image is retinexed using PhotoFlair. Comparing Figures 3.25 and 3.27 we observe that the region where the car is present is picked up in the retinexed image whereas it is not picked up in the foggy image, Figure 3.25. Hence, the regions where objects are present are picked up in the enhanced images.

Figure 3.40: Retinexed Image using PhotoFlair.

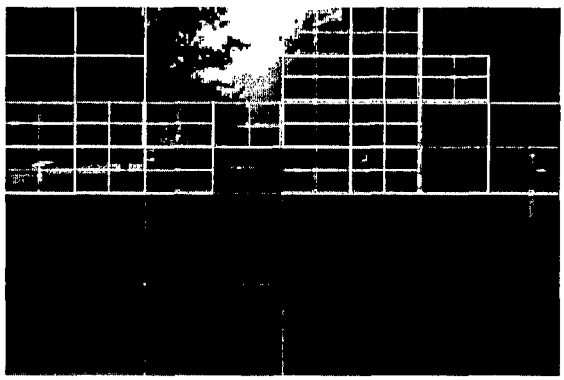


Figure 3.41 Application of quadtree segmentation on Retinexed Image.

3.5 Summary

The d-peak algorithm has been implemented on foggy images and retinexed images for the purpose of image segmentation. The d-peak algorithm is a histogram based method. This algorithm is suitable for all nature images whereas the traditional method, which uses the thresholding value for segmenting the image, is suitable only for images whose foreground and background have different brightness. Dithering is implemented on the images after applying the d-peak algorithm to mitigate the effects of false contouring, but here the implementation of the dithering technique has no effect on mitigating the false contouring effect. Quadtree segmentation is also applied to the foggy images and retinexed images to detect the regions of interest or objects.

CHAPTER 4

EXPERIMENTAL RESULTS

In this chapter, the performance of the image segmentation techniques is assessed. The algorithms are tested and evaluated with images captured in foggy conditions and on the retinexed images.

4.1 Data Set

An image data set was collected for testing and evaluating the proposed d-peak algorithm and quadtree algorithms. The effect of dithering after implementing the d-peak algorithm on foggy and retinexed images is also assessed.

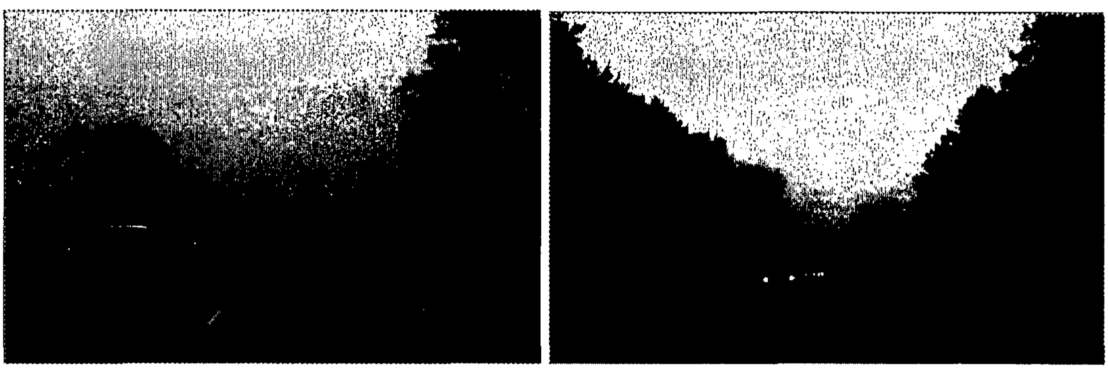
4.2 Implementation Platform

The implementation platform plays an important role in facilitating a flexible and practical development approach. The platform has to allow maximum flexibility and debugging. There are two primary development techniques: MATLAB and C. The programming in MATLAB is considerably slower; however, the debugging environment is not well suited to images with larger dimensions. The flexibility in testing and development offered in C were the critical deciding factors, so development and testing were carried out in C.

4.3 Experiments

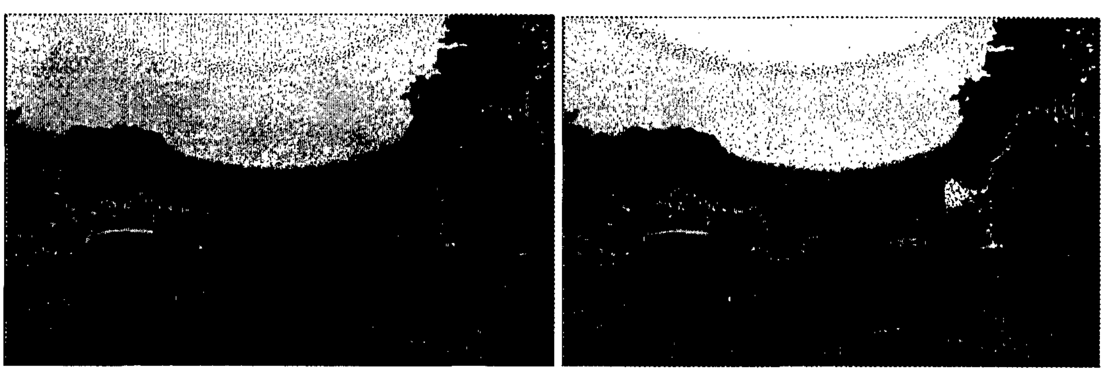
The proposed algorithms provide a novel method for detecting objects in poor visibility conditions. The major problem with the d-peak algorithm is that the pixels belonging to a particular range are to be grouped together. Thus, sampling and quantization are performed. After the quantization process the number of peaks depends on the threshold chosen. First the d-peak algorithm is applied to foggy images and then applied on the retinexed version of the images. A threshold value of P=0.10 is used for the foggy images.

4.3.1 Application of d-peak on foggy and enhanced images



(a) Foggy Image (b) Foggy Image

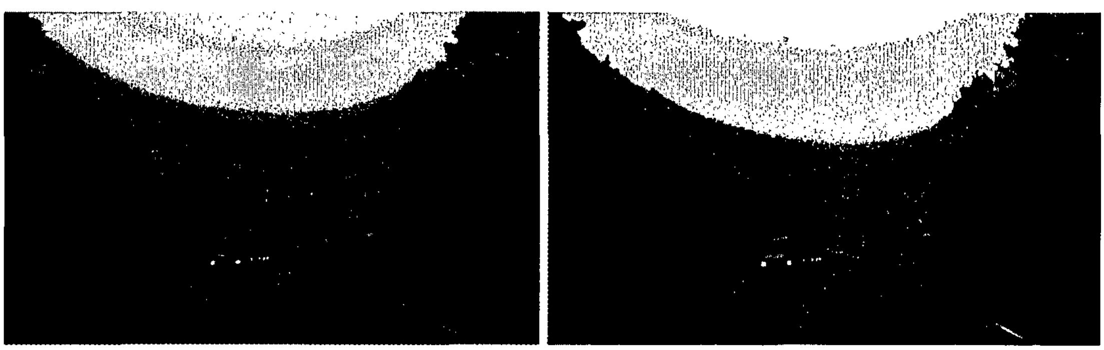
Figure 4.1: Original Foggy Images



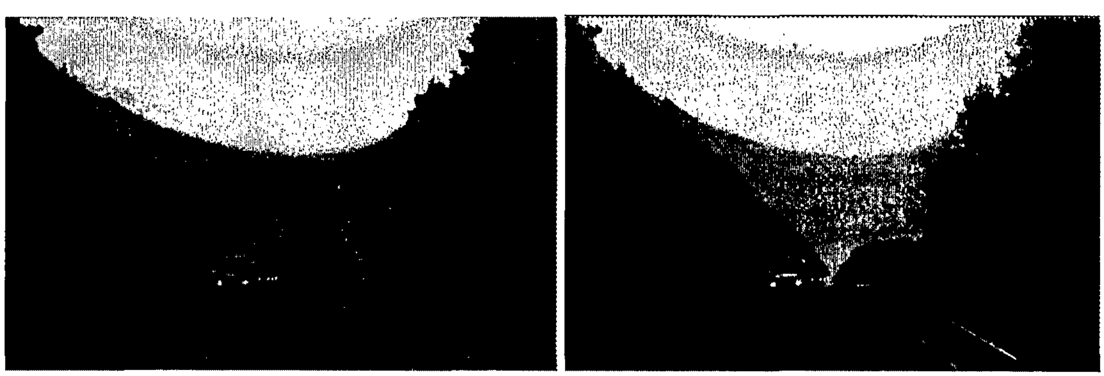
(a) Image with P=0.40 (b) Image with P=0.30

(c) Image with P=0.20 (d) Image with P=0.10

Figure 4.2: Application of d-peak algorithm with different threshold values-I.



(a) Image with P=0.40 (b) Image with P=0.30

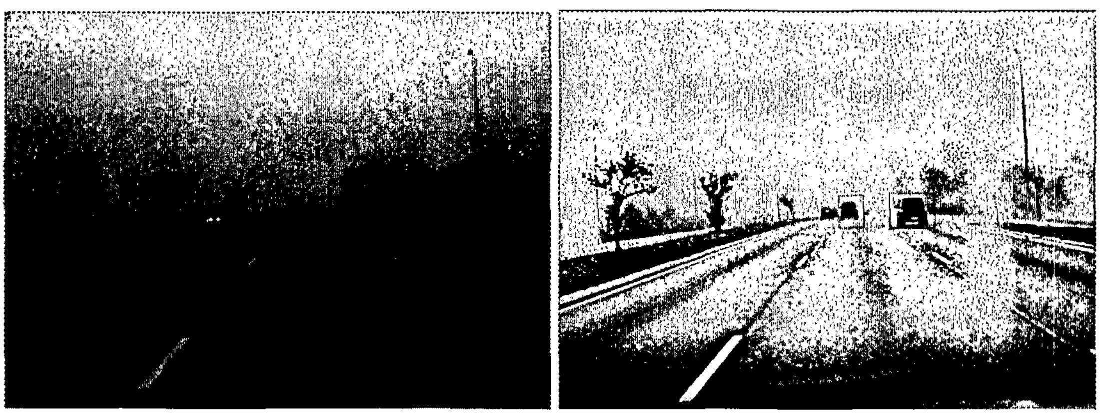


(c) Image with P=0.20 (d) Image with P=0.10

Figure 4.3: Application of d-peak algorithm with different threshold values-II.

From Figure 4.2 and Figure 4.3 it is observed that the objects of interest are clearly separated from the background if P=0.10 of the maximum value of the peak in the histogram when compared to the images if P=0. 40, 0.30, 0.20.

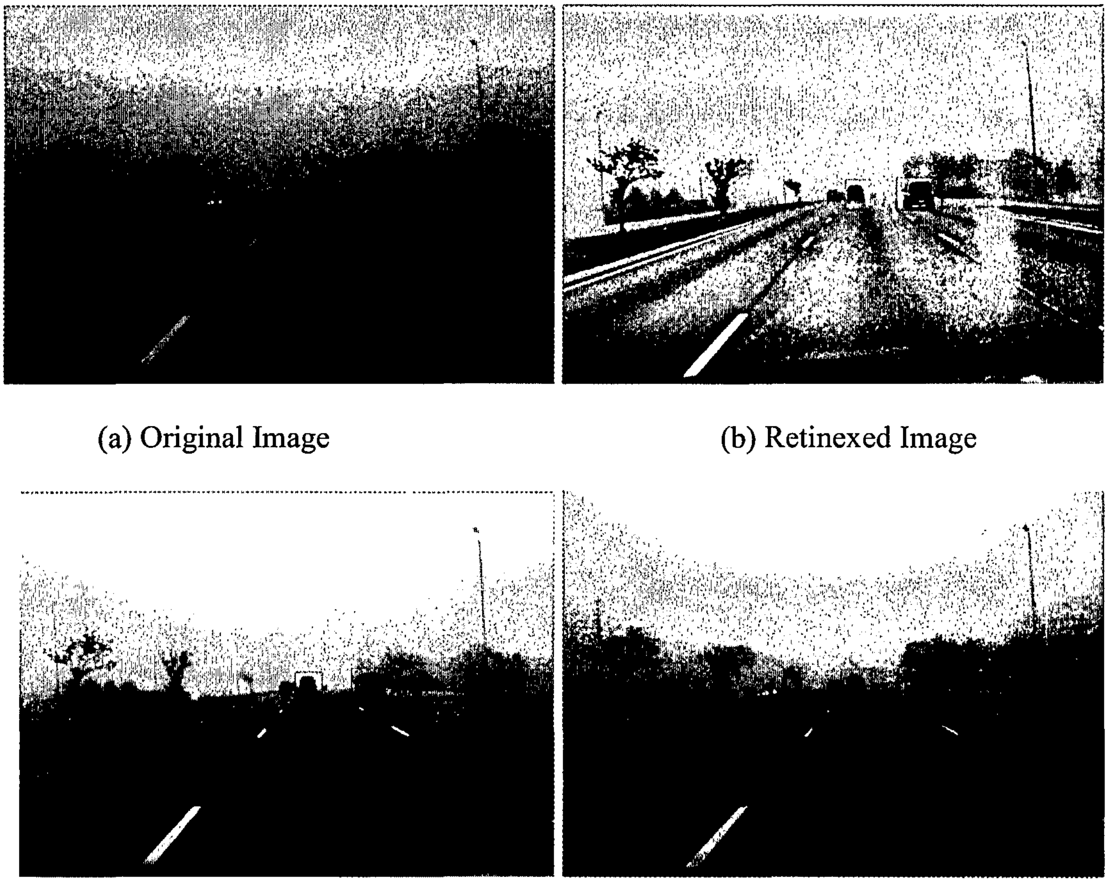
The contrast of an image is often significantly degraded by poor visibility conditions such as fog and haze. This degradation of image quality by poor visibility conditions restricts the imaging applications to good visibility conditions. Hence, the images are enhanced using Multiscale retinex because of its performance in poor visibility conditions.



(a) Original Image (b) Retinexed Image

Figure 4.4: Results of Multi –Scale Retinex (MSR).

From Figure 4.4, it is observed that the multi scale retinex provides great dynamic range compression, increased sharpness and color, and accurate scene rendition, especially in poor visibility conditions such as fog and haze. Other techniques like contrast stretching, histogram equalization, etc. were tried, but those algorithms did not give good results. Figure 4.4 and Figure 4.5 compare the results of MSR, contrast stretching and histogram equalization on images degraded by poor visibility conditions such as fog and haze.

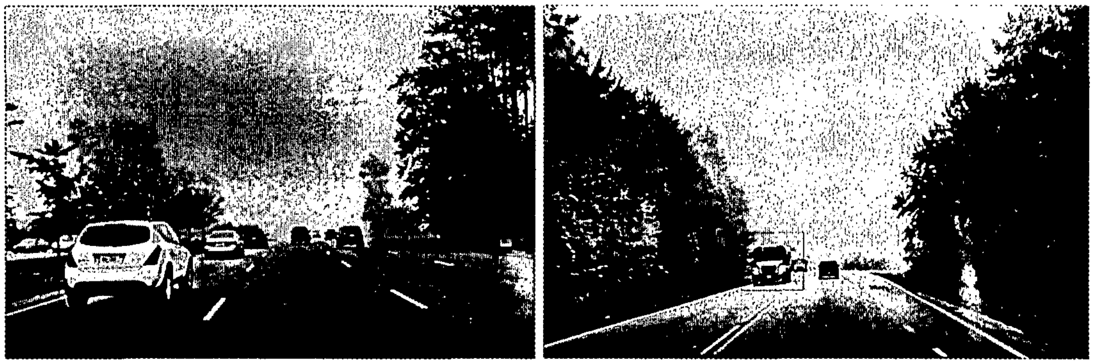


(a) Original Image (b) Retinexed Image

(c) Contrast stretching (d) Histogram Equalization

Figure 4.5: Comparison of different enhancement algorithms-I.

From Figure 4.5 it is observed that the results of Retinex are better than the results produced by contrast stretching and histogram equalization. After the enhancement of the foggy image, the next step is implementing the d-peak algorithm on the retinexed image.



(a) Retinexed Image (b) Retinexed Image

Figure 4.6:Retinexed Images.

(a) Image with P=0.40 (b) Image with P=0.30

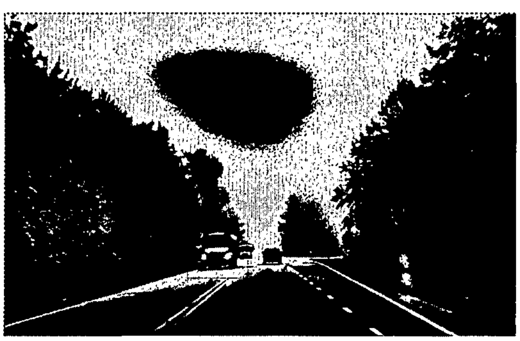
(c) Image with P=0.20 (d) Image with P=0.10

(e) Image with P=0.05

Figure 4.7: Result of applying d-peak algorithm on Retinexed Images-I.

(a)Image with P=0.40

(b) Image with P=0.30



(c)Image with P=0.20

(d) Image with P=0.10

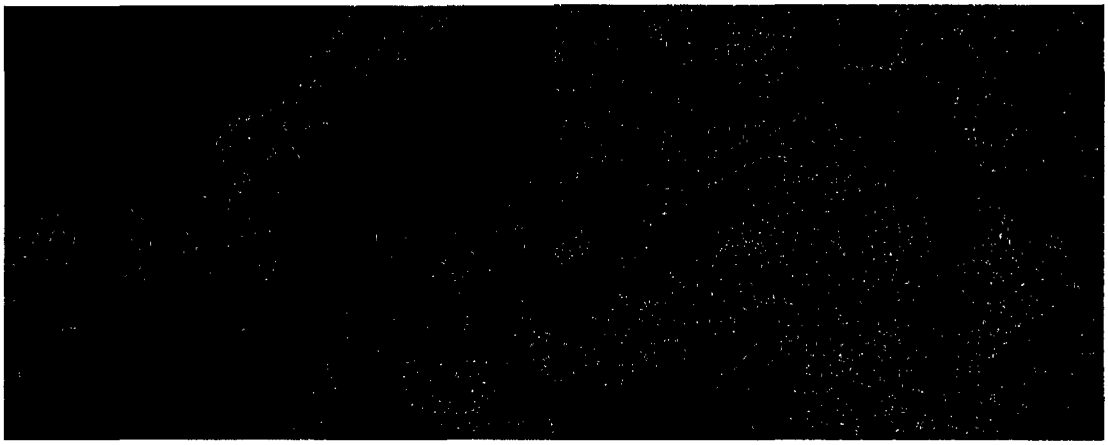
(e)Image with P=0.05

Figure 4. 8: Result of applying d-peak algorithm on Retinexed Images-II.

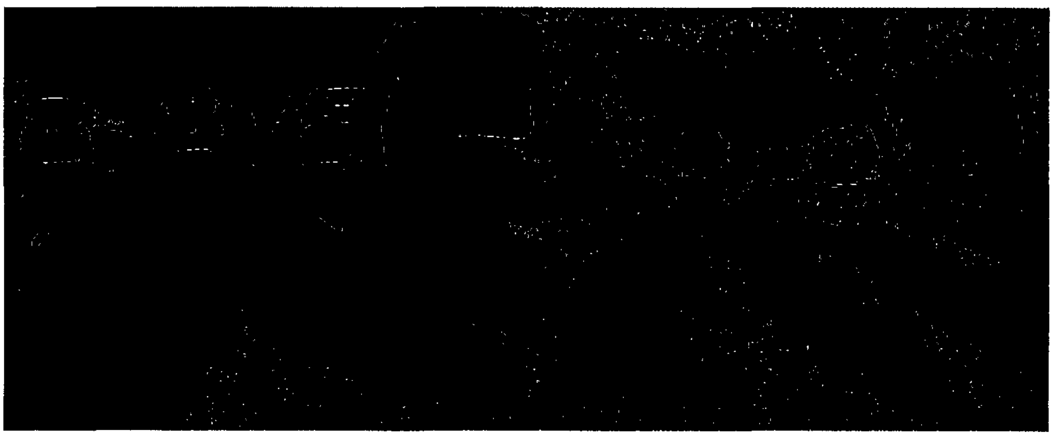
From Figure 4.7 and Figure 4.8 it is observed that the objects of interest are clearly separated from the background if P=0.05 when compared to the images if P=0.40, 0.30, 0.20, 0.10. Hence, the threshold value for retinexed foggy images is taken as P=0.05.

4.3.2 Canny Edge Detection

Use of the d-peak algorithm by itself for the purpose of image segmentation is not sufficient to interpret the images. Hence, dithering is followed by edge detection. The canny edge detection operator is applied to the segmented images, and the results are shown below in Figure 4.9:

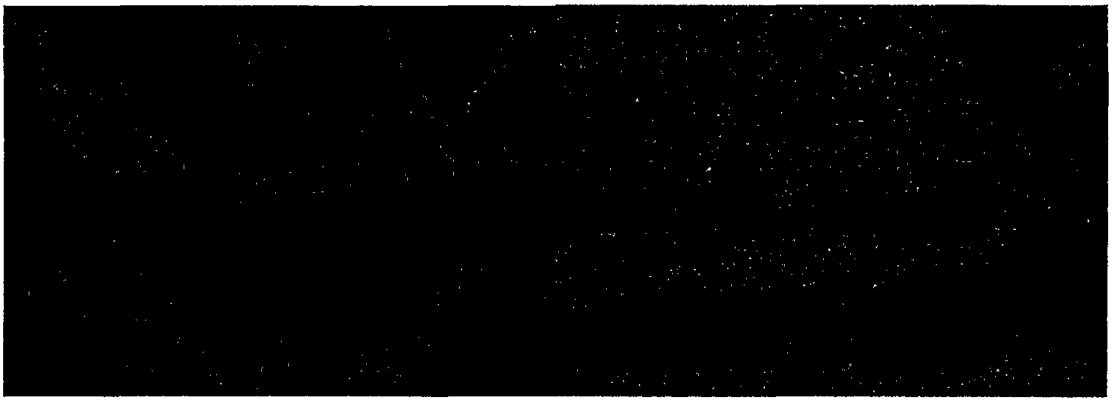


(a) Canny on foggy image (b) Canny on Retinexed image



(c) Canny on foggy image (d) Canny on Retinexed image

Figure 4.9: Results of implementing Canny edge detection operator on foggy and retinexed images-I.



(d) Canny on Foggy image (e) Canny on Retinexed Image

Figure 4.10: Results of implementing Canny edge detection operator on foggy and retinexed images-II.

It is observed from the above images that more of the details and edges are picked up in the case of enhanced images whereas in the case of foggy images edges tend to disappear, and the enhanced images are noisier when compared to the foggy images.

4.4 Quadtree Segmentation

Quadtree segmentation is one of the image segmentation techniques.

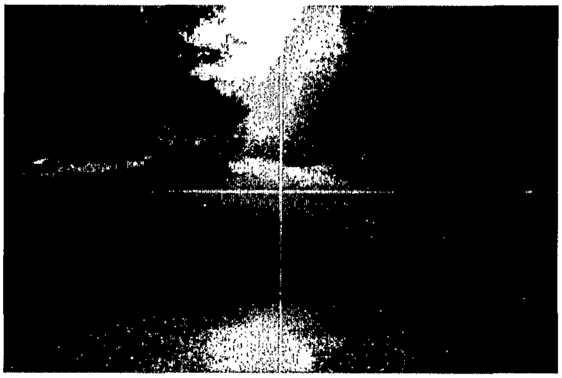


Figure 4.11: Foggy Image after applying first step of quadtree segmentation-I.

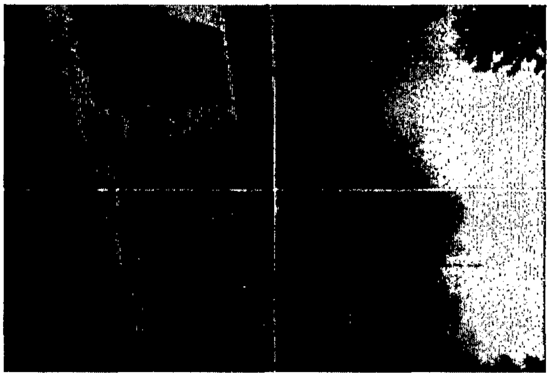


Figure 4.12: Foggy Image after applying

first step of quadtree segmentation-II

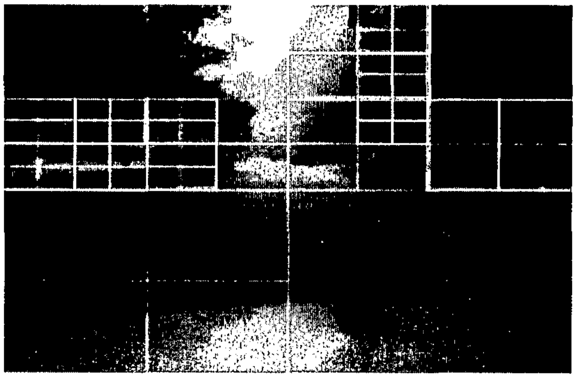


Figure: 4.13: Results of applying

quadtree segmentation on Foggy Image-I.

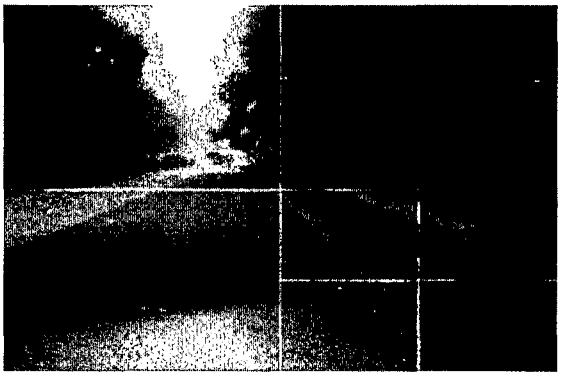


Figure: 4.14: Results of applying

quadtree segmentation on Foggy Image-II.

From Figure 4.17 and Figure 4.18 it is observed that the regions where the objects are present in the scene are not picked up by the algorithm. Since the image contrast and quality of the image is degraded by poor visibility conditions such as fog and haze, the foggy images are enhanced; then the quadtree segmentation is implemented on the enhanced images.

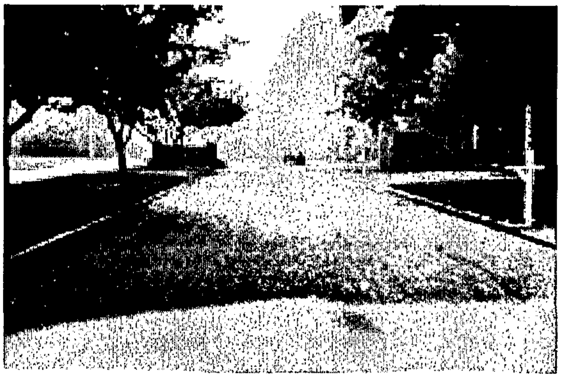


Figure 4.15: Retinexed Image using Photo flair-I.

Figure 4.16: Retinexed Image using Photo flair-II.

Even though the Retinexed images in Figure 4.15 and Figure 4.16 have significantly more brightness and contrast, the quadtree segmentation does not pick up any objects as the brightness and contrast of the image does not meet the segmentation criteria. As this segmentation algorithm was developed particularly for detecting objects and since the algorithm implementation did not yield the desired results, we modified the procedure slightly. Instead of performing the MSR on the original data, we chose to apply it to the negative of the original data. Images in Figure 4.17 and Figure 4.18 are retinexed in the negative mode using PhotoFlair software

Figure 4.17: Retinexed Image in the negative mode-I.

Figure 4.18: Retinexed Image in the negative mode-II.

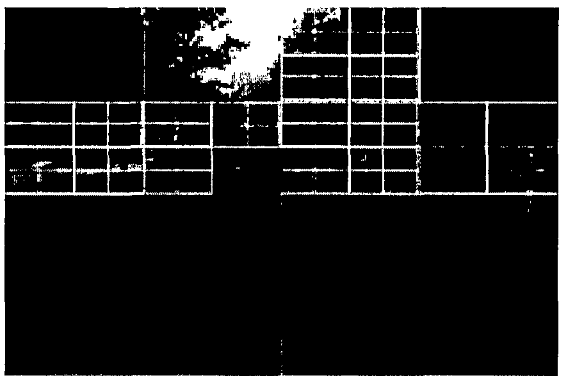


Figure 4.19 Result of applying the quadtree

Segmentation on Figure 4.17.

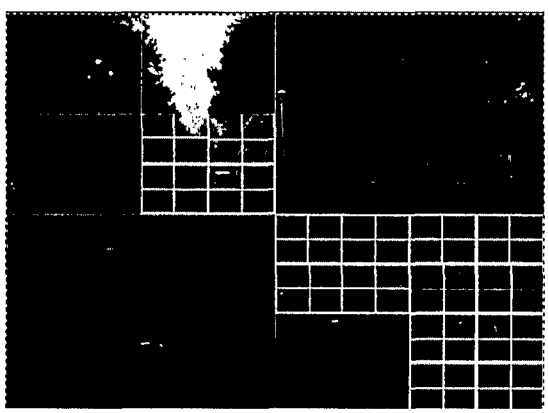


Figure 4.20 Result of applying

the quadtree segmentation on Figure 4.18.

4.5 Summary

In this chapter, the performance analyses of the proposed algorithms have been presented. Based on the experimental results, it is observed that the proposed d-peak and quadtree segmentation algorithms perform well on enhanced images. Thus, the proposed algorithms can be applied to detect objects or hazards on the runway especially in poor visibility conditions such as fog, haze and snow.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

5.1 Conclusions

In this thesis, a novel algorithm has been proposed for the detection of hazards on runways in variable lighting and poor visibility conditions.

It is dangerous for a pilot to land an aircraft safely in poor visibility conditions, due to the possible presence of objects on the runway. If an object is detected early enough before landing the aircraft, the pilot may get a chance to either change the flight path or abort the landing. The various problems involved in detection of hazards have been discussed and addressed. The contrast of an image is often significantly degraded by atmospheric aerosols such as fog and haze. This degradation of images by atmospheric aerosols restricts imaging applications to good visibility conditions. This is why foggy images are enhanced. The image enhancement techniques have been discussed. The images enhanced using Multi-Scale Retinex provide illumination independence i.e. the output of the enhancement phase is independent of the type, or level of illumination under which the image was acquired. This is especially critical for object detection algorithms which rely on comparing the imagery of the same scene at different times. The d-peak algorithm and quadtree algorithm are used to detect the regions where the objects are present.

5.2 Future Work

The proposed algorithms provided good results for object edge detection on runways in poor visibility conditions. They were tested on road images under foggy conditions and enhanced images. However, the algorithms were not tested on runway images under poor visibility conditions as we do not have access to real runway images. Using these algorithms for the detection of objects on real runway images may need to be refined.

REFERENCES

[1] T. Chen, D. Z. Asano, N. Katoh, and T. Tokuyama, "Polynomial-Time Solutions to Image Segmentation," *Proceedings of the 7th Annual SIAM-ACM Conference on Discrete Algorithms*, vol.7, pp. 104-113, 1996.

[2] P. Salmebier, and F. Marques, "Region-Based Representations of Image and Video: Segmentation Tools for Multimedia Services," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 8, pp. 1147-1169, 1999.

[3] T. Taxt, P.J. Flynn, and A. K. Jain, "Segmentation of Document Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 11, no. 12, pp. 1322-1329, 1989.

[4] Y. Amit, and D. Geman, "A Computational Model for Visual Selection," *Neural Computation*, vol.11, pp.1691-1715, 1999.

[5] P. K. Sahoo, S. Soltani, A. K. C. Wong, and Y. C. Chen, "A Survey of Thresholding Techniques Computer Vision, Graphics and Image Processing," vol. 41, pp. 233-260, 1988.

[6] C. K. Chow, and T. Kaneko, "Boundary Detection of Radiographic Images by a Thresholding Method.Frontiers of Pattern recognition," S. Watanabe, eds., Academic Press, pp. 61-82, Newyork, 1989.

[7] S. D. Yanowitz, and A. M. Bruckstein, "A New method for Image Segmentation," CVGIP, vol. 46, pp. 82-95, 1989.

[8] A. K. Jain, and M. P. Dubbuisson, "Segmentation Of X-Ray and C-Scan Images of Fiber Reinforced Composite Materials.Pattern Recognition," vol. 25, no. 3, pp. 257-270, 1992.

[9] Han Lee, and Rae-Hong Park, Comments on "An Optimal Multiple Threshold Scheme for Image Segmentation", *IEEE Transactions on Systems, Man and Cybernetic,* vol. 20, no. 3, pp. 741-742, 1990.

[10] Dithering: http://en.wikipedia.org/wiki/Dither#Different_types_of_dither, 2008.

[11] J. R. Parker, "Gray level Thresholding in badly Illuminated Images," *Pattern Analysis and Machine Intelligence*, vol. 13, no. 8, pp. 813-819, 1991.

[12] Histogram-Based-Methods: http://en.wikipedia.org/wiki/Segmentation_(image_processing)#Histogram-Based_Methods, 2008.

[13] H. Francis, Y. Chan, F. K. Lam, and Hui and Zhu "Adaptive Thresholding by Variational Method," *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp.468-473, 1998.

[14] D. J. Jobson, Z. Rahman, and G. A. Woodell, "Properties and Performance of a Center/Surround Retinex," *IEEE Transactions on Image Processing*, vol. 6, no. 3, pp. 451-462, 1997.

[15] D. J. Jobson, Z. Rahman, and G. A. Woodell, "A Multi-Scale Retinex for Bridging the Gap between Color Images and the Human Observation of Scenes," *IEEE Transactions on Image Processing; Special Issue on Color Processing,* vol. 6, no. 7, pp. 965-976, 1997.

[16] John P. Oakley, and Brenda L. Satherly "Improving Image quality in poor visibility conditions using a physical model for contrast degradation", *IEEE Transactions on Image Processing,* vol. 7, no. 2, pp. 167-179, 1998.

[17] E. Land, "An alternative technique for the computation of the designator in the retinex theory of color vision", *In proceedings of the Natl. Acad. Sci. USA,* vol.83, pp.3078-3080, 1986.

[18] Anya C. Hurlbert, "Formal connections between lightness algorithms," *Journal of the Optical Society of America A,* vol.3, no. 10, pp. 1684-1693, 1986.

[19] Anya C. Hurlbert, "The Computation of Color", *PhD thesis, Massachusetts Institute of Technology,* September 1989.

[20] G. Leedham, C. Yan, K. Takru, J. H. N. Tan, and L. Mian, "Comparison of Some Thresholding Algorithms for Text/Background Segmentation in Difficult Document Images", *in proceedings of the 7th International Conference on Document Analysis and Recognition,* vol. 2, pp. 859-864, 2003.

[21] S. Y. Chen, and W. C. Lin, "Split and merge Image Segmentation Based on Localized Feature Analysis and Statistical Tests. CVGIP: Graphical Models Image Processing," vol. 53, no .5, pp. 457-475, 1991.

[22] Eric P. Lam, "Image Quality Measure via a Quadtree Homogeneity Analysis," *In proceedings of the SPIE Visual Information Processing XVI,* vol. 6575, pp. 65750R, 2007.

[23] Z. Rahman, G. A. Woodell, and D. J. Jobson, "A Comparison of the Multiscale Retinex with Other Image Enhancement Techniques", *in proceedings of the IS&T 50th Annual Conference, 1997.*

[24] BoxMuller Transforms: http://en.wikipedia.org/wiki/Box-Muller_transform, 2008.

[25]. Z. Rahman, D. Jobson, and G.A. Woodell, "Multiscale retinex for color image enhancement," *International Conference on Image Processing (ICIP),* 1996.

[26]. Z. Rahman, D. J. Jobson, and G. A. Woodell, "Resiliency of the Multiscale Retinex Image Enhancement Algorithm," *IS&T Sixth Color Imaging Conference, 1998.*

[27] Z. Rahman, D. J. Jobson, and G. A. Woodell, "Retinex processing for automatic image enhancement," *Journal of Electronic Imaging*, vol. 13, no. 1, pp.100-110, 2004.

[28] Z. Rahman, D. J. Jobson, and G. A. Woodell, "A Multiscale Retinex for Color Rendition and Dynamic Range Compression", SPIE International Symposium on Optical Science, Engineering, and Instrumentation, *Conference on Signal and Image Processing,* 1996.

[29] J. Canny, "A computational approach to edge detection", *in proceedings of the IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 8, no. 6, pp. 679 - 698, 1986.

CURRICULUM VITAE

Triveni Vuppalapati

Mobile :(571) 217-2315

Email: tvupp001@odu.edu

Educational Background:

- M.S. in Electrical Engineering, August 2008, GPA: 3.12, Old Dominion University, Norfolk VA. *Focus:* Detection of objects in poor visibility conditions using image segmentation.
- B.Tech. in Electronics and Communication Engineering, June 2006. GPA: 3.65, Jawaharlal Nehru Technological University Hyderabad-32, AP India. *Focus*: Worked on Computer networks and security systems and medical diagnostic equipment in funded projects.

Interests:

- Object detection in poor visibility conditions using image segmentation
- Biometrics and Image Processing
- Machine learning and Modular neural networks

Areas of Expertise:

- Extensive experience in digital signal/image processing algorithms
- Well-versed on computer vision:
  - Edge detection, segmentation, dithering, and noise reduction
- Familiar with programming languages/tools (MATLAB, C, C++ etc.) and OS platforms (Window95, Window NT, etc.)

Practical Experience:

- Student Researcher, 2007-2008, Old Dominion University, Norfolk VA
  - Developed novel algorithms for the detection of objects in poor visibility conditions using image segmentation
- Academic Teaching Assistant, 2007-2008.Old Dominion University, Norfolk VA