Summer 2008

Image Pre-Processing Techniques for Hazard Detection in Poor Visibility Conditions

Girish Singh Rajput
Old Dominion University

IMAGE PRE-PROCESSING TECHNIQUES FOR HAZARD DETECTION IN POOR VISIBILITY CONDITIONS

by

Girish Singh Rajput
B.Tech (ECE) June 2006, Jawaharlal Nehru Technological University, India.

A Thesis Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

ELECTRICAL ENGINEERING

OLD DOMINION UNIVERSITY
August 2008

Approved by:

Dr. Zia-ur Rahman (Director)

Dr. K. Vijayan Asari (Member)

Dr. Jiang Li (Member)

ABSTRACT

IMAGE PRE-PROCESSING TECHNIQUES FOR HAZARD DETECTION IN POOR VISIBILITY CONDITIONS

Girish Singh Rajput
Old Dominion University, 2008
Director: Dr. Zia-ur Rahman

Runway incursion is a persistent problem that has resulted in some of the most devastating accidents in aviation history. With ever increasing air traffic and more passengers, runway safety is of utmost priority to the Federal Aviation Administration (FAA) and other agencies concerned with aviation. As the issue of aviation safety becomes increasingly important, developing a consistent application that detects runway incursions in various visibility conditions is crucial for the aviation industry. This thesis presents a novel method for detecting runway hazards in poor visibility conditions using image processing techniques. The first step is to obtain images of a runway on a clear day and compute the smoothness coefficients. Then edge detection is performed using the SUSAN edge detection algorithm. Finally, database of the smoothness coefficients and edge detected images is developed. For foggy images we compute the smoothness coefficients. Typically, foggy images have low contrast. Hence, before we perform edge detection, we enhance the image using Multi-Scale Retinex (MSR). The MSR provides the improvement in contrast and color constancy that is needed to enhance foggy images, by performing non-linear spatial/spectral transforms. After enhancement, the next step is to run the same edge detection algorithm with appropriate thresholds. Finally, we determine a hazard by comparing the edge detected images of images taken under clear and foggy conditions.

ACKNOWLEDGMENTS

I would like to express my sincere thanks to Dr. Zia-ur Rahman for his support, advice, and motivation over the past two years. Without his constant guidance, it would not have been possible to complete this thesis.

I would like to thank Dr. K. Vijayan Asari and Dr. Jiang Li for agreeing to be on my committee and for their valuable time.

I would like to thank my family and all my friends for being there for me all the time.

Finally, I would like to thank the NASA Aviation Safety Program for funding this project. This work was supported under the NASA cooperative agreement NNL07AA02A.

TABLE OF CONTENTS

Chapter Page

Chapter Page

LIST OF FIGURES

Figure Page

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Surveys reveal that around twenty eight percent of the U.S. economy is aviation related [1]. For this reason, the aviation ministry and professionals involved in this field are under immense pressure to improve the business of aviation. Improving safety mechanisms that lead to more on-time arrivals and departures thus streamlining aviation would go a long way towards achieving this end. This is especially important in light of the increasing number of runway incursions that are giving rise to concerns over aviation safety.

1.1 Motivation

With aircraft traffic predicted to increase over the coming years, increasing congestion at airports, prevention of runway incursions is of utmost priority for the United States Federal Aviation Administration (FAA) and other national civil aviation regulatory agencies. The FAA defines a runway incursion as "any occurrence involving an aircraft, vehicle, person, or object on the ground that creates a collision hazard or results in a loss of required separation when an aircraft is taking off, intending to take off, landing, or intending to land". The International Civil Aviation Organization (ICAO) defines an incursion as "the incorrect presence of aircraft, vehicle, or person on the protected area of a surface designated for landing or take-off of aircraft"† [2]. FAA reports suggest that the United States National Airspace System has around 500 airports with air traffic control towers handling around 171,200 aircraft operations daily. From

---

† The reference model for this work is *IEEE Transactions on Image Processing*

2003-2006, out of these 250 million operations there were about 1306 reported incidents of runway incursions. The United States National Transportation Safety Board (NTSB) placed "reduction of runway incursions" on the list of "Most Wanted" transportation safety improvements in 1990, and it has been on that list ever since [3-7]. Runway incursion ratio can be calculated as the number of incursions to the number of operations. These incursion rates are expressed as the number of incursions for every 100,000 operations. Table 1.1 shows the number of airport operations, the total number of runway incursions, and the incursion rate for each year over the 1988-2001 period. We note that while the number of airport operations increased by only 10 percent over this 13 year period, the incursion rate more than doubled.

Table 1.1: Report by the subcommittee on Aviation Hearing on Runway Incursions, Focusing on the Technology to Prevent Collisions 06/26/01 [8].

CALENDAR YEAR	TOTAL RUNWAY INCURSIONS	AIRPORT OPERATIONS	INCURSION RATE
1988	187	62,501,059	0.30
1989	223	62,625,548	0.36
1990	281	65,506,291	0.43
1991	242	62,421,635	0.39
1992	219	63,018,680	0.35
1993	186	61,946,482	0.30
1994	200	62,452,572	0.32
1995	240	62,074,306	0.39
1996	275	61,817,425	0.44
1997	292	64,440,947	0.45
1998	325	66,211,734	0.49
1999	321	68,672,240	0.47
2000	431	67,480,097	0.64
2001*	166	N/A	N/A

The FAA classifies runway incursions as *operational errors/deviations, pilot deviations,* and *vehicle/pedestrian deviations* based on the last link in the chain of events that resulted in the incursion. Incursions usually occur due to any of the following three situations:

1. Arrival of an aircraft on an occupied runway,

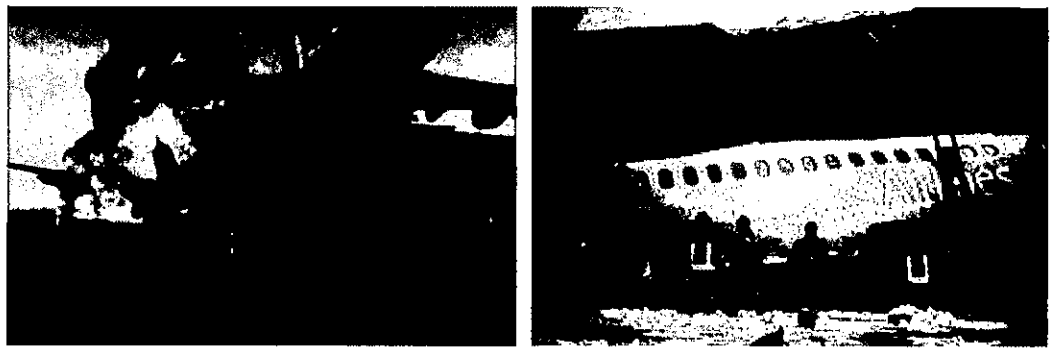
2. Two aircrafts converging on intersecting runways, or

3. Multiple departures of aircrafts that are too closely spaced on the same runway.

These incursions could be caused by communication breakdowns, errors in surface navigation, or some other factors. The FAA, the ICAO and several other agencies are developing operational guidelines and equipment to address these situations.

Most of these problems can be avoided if the safety procedures are followed properly. Detection of hazards under clear visibility conditions typically does not pose too difficult a task for the pilot. However, the problem is significant under poor visibility conditions when the pilot may find it difficult to even spot the runway, let alone detect any hazards on it. There have been several incidents and accidents in the recent past because of poor visibility conditions.

Some of the examples of the runway collisions because of poor visibility conditions are the collision at the Tenerife Airport in the Canary Islands on March 27, 1977 [9] and the collision at the Milan Linate Airport in Italy on October 08, 2001 [10]. The collision at the Tenerife Airport in the Canary Islands involved two 747s and killed 583 people. The collision at the Milan Linate Airport involved a jetliner and a business jet killing 118 people. The common reason given for both accidents was poor visibility.



(a) Crash at the Tenerife Airport (b) Crash at the Milan Linate Airport

Figure 1.1: Examples of runway incursions.

1.2 Existing Systems

In collaboration with several other agencies the FAA is trying to improve the safety standards by undertaking several safety programs to mitigate runway incursions. Some of the programs involve the use of radar to detect objects on the runway.

1.2.1 Airport Surface Detection System

The Airport Surface Detection System (ASDE) [11] is a radar system that aids air traffic controllers in detecting surface radar targets and sequencing aircraft movement on active runways during low visibility conditions. It can be used in low visibility conditions to enhance the controller's situational awareness, detect and display targets and aid movement area clearance.

1.2.2 Airport Movement Area Safety System

The Airport Movement Area Safety System (AMASS) [12], an extension of ASDE-3, is a runway collision alert system that provides tower air controllers with automated conflict warning and alerts to reduce the risk of runway collisions.

1.2.3 Airport Target Identification System

The Airport Target IDentification System (ATIDS) [13] is used as a surface beacon surveillance system to provide Flight Number Identification to ASDE and AMASS and is capable of locating and identifying aircraft in flight or on the ground.

1.2.4 Final Approach Runway Occupancy Signal

The Final Approach Runway Occupancy Signal (FAROS) [14] is an automated safety system designed to notify the pilot on the approach to landing about hazards on the runway. FAROS uses ASDE to detect hazards on the runway.

1.3 Drawbacks of Existing Systems

An extensive review of the FAA's safety programs suggests that the FAA is trying very hard to reduce the number of runway incursions by deploying several new systems. In spite of these efforts, the number of incursions has not decreased over the past years. This is because of several limitations of the existing systems.

1.3.1 Affordability

The major problem with the existing systems is their affordability. Some of the most advanced systems that have the capability to reduce the number of runway

incursions are limited to busy airports because of their massive costs. As a result, incursions occur at smaller airports. This hampers the FAA's goals to reduce the number of incursions. The systems that are designed should therefore not only serve their purpose; they must also be cost effective.

1.3.2 Adaptability

The designed system should provide a solution in varying situations. The problem with the existing systems is their reliance on the human-in-the-loop to avoid runway incursions. Human error is the major factor for many runway incursions. While most of the existing systems are very good at detecting a possible incursion, they depend on ground controllers to inform the pilot about hazards on the runway. Because human errors cannot be easily mitigated, the system's design should have very little dependence on humans. Additionally, some runway incursions occur because of the inability of the pilot to see the runway in poor visibility conditions, so incursion avoidance systems must also be effective in all weather conditions.

1.3.3 Alerting

Regarding the real time scenario being considered, where seconds of delay can result in a devastating accident, the system that produces alerts must have a fast response time. Most of the existing systems need the ground controller to notify the pilot about a hazard, so a system that directly alerts the pilot about the hazard is needed so that the pilot gets the maximum time to abort the landing sequence.

1.4 Proposed Research

This thesis addresses the aviation safety issue and aims to build an automated system to assist pilots in avoiding collisions on the runway in poor visibility conditions. There are a variety of problems that a pilot could face in landing an aircraft in poor visibility conditions, from locating the runway to determining if the runway is clear. Poor visibility conditions can be defined as fog, smoke, haze or dim or dark lighting conditions. Hazard detection of runways in poor visibility conditions is one of the current research areas of NASA's aviation safety program and this thesis is a part of the NASA program. It uses image processing techniques to detect hazards on the runway in poor visibility conditions.

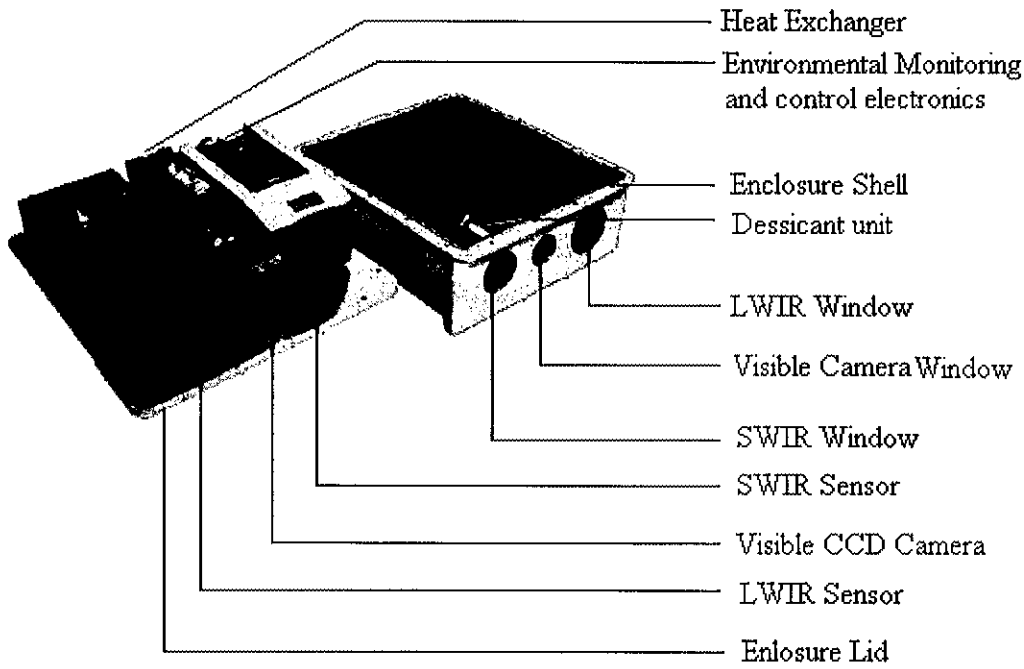


Figure 1.2: The EVS pod used on the NASA Aries Boeing 757.

The initial approach uses the visible camera from NASA's Enhanced Vision System (EVS) developed under the Aviation Safety Program. The EVS provides

enhanced images of the flight environment to assist pilots flying in poor visibility conditions. The system has a long-wave infrared (LWIR), a short-wave infrared (SWIR), and a visible band camera, all mounted in a pod that is flown beneath a NASA 757 Aries aircraft [15]. Figure 1.2 shows the cameras and other components mounted in the pod, and figure 1.3 shows the pod installed on the 757 [16-17] aircraft. The LWIR and the SWIR imagers are the primary cameras for current aviation safety research efforts. The LWIR is a Lockheed Sanders LTC500 thermal imager and senses radiation in the 7.5–14 μm band. It can image background scenery, terrain features and obstacles at night and in other low visibility conditions. The SWIR is a Merlin Near-Infrared (NIR) camera that senses in the 0.9–1.68 μm region and is optimal for detecting peak radiance from runway and taxiway lights even in poor visibility conditions. The visible-band camera is a Bowtech BP-L3C-II CCD that detects the 0.4–0.78 μm band and picks up runway markings, skyline and city lights in good visibility conditions. Additionally, we make use of ancillary information about the aircraft attitude, altitude, speed, and heading to correct for the platform motion. This information is available from the aircraft navigation system.

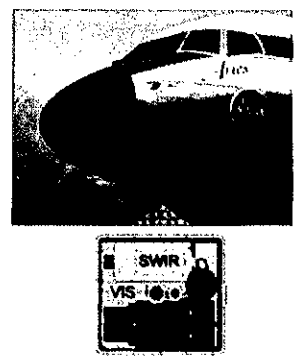


Figure 1.3: The EVS pod mounted on the NASA Aries Boeing 757. The sensors point down at a 4° angle with respect to the belly of the aircraft.

The proposed algorithm uses Multi-Scale Retinex (MSR) for image enhancement, followed by the Smallest Univalue Segment Assimilating Nucleus (SUSAN) edge detection algorithm to detect the hazards on the runway. MSR provides enhancement of low contrast and brightness imagery, as well as color constancy. These attributes are required to enhance poor visibility images. The MSR performs as non-linear spatial/spectral transform on the original image data resulting in context-dependent, locally adaptive output that is independent of illumination condition. Edge detection is a low-level image processing technique that derives the structural information about the surfaces of the objects in the image. There are a variety of operators available for edge detection. However, there are some drawbacks that are common to most of these operators. Some of the most common drawbacks are problems with connectivity and localization accuracy resulting in incomplete edge definition. The SUSAN edge detection operator is used in this thesis because of its ability to overcome most of these drawbacks. Additionally, its performance speed is better than most of the well known edge detection operators.

1.5 Specific Objectives

The prime objective of this thesis is to develop an algorithm that would be used to assist the pilot in poor visibility conditions by detecting hazards on the runway. The specific objectives include

1. Computing the Smoothness Coefficient
2. Enhancing the imagery using Multi-Scale Retinex
3. Designing the SUSAN edge detection operator

4. Detecting hazards on the runway
5. Testing the proposed algorithm for images in different lighting conditions
6. Comparing the results from the proposed algorithm with other edge detection operators.

1.6 Thesis Outline

Chapter 2 presents the various approaches that could be used to detect hazards on the runway. The possible use of thermal imagery is presented. Some basic segmentation algorithms like thresholding and clustering are discussed. Segmentation is followed by various image enhancement techniques and edge detection operators.

Chapter 3 presents the proposed algorithms for detection of hazards on the runway and the theory associated with it. The algorithm is discussed in three basic steps: (i) collecting the imagery and storing in database, (ii) enhancing the imagery using Multi-Scale Retinex, and (iii) performing edge detection using the SUSAN operator.

Chapter 4 presents the experimental results of Multi-scale Retinex and the SUSAN operator and compares their results with other techniques on clear and foggy images. This chapter also presents the results of the proposed algorithm.

Chapter 5 provides the conclusions and directions for future work. This chapter discusses the FAA's safety goals and the algorithm's ability to accomplish those goals.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

With the increasing demand for air travel, runway safety is of prime importance for the Federal Aviation Administration (FAA), airlines and airports. Runway safety is an aviation safety concern involving measures to prevent runway incursions and overruns. This research work primarily deals with the detection of hazards on the runway in poor visibility conditions. Image processing techniques are used as a part of this research work to detect hazards on the runway. This chapter deals with the approaches and techniques that could be used for the detection of hazards.

2.2 Thermal Imaging

Thermal imaging detects radiation in the infrared range of the electromagnetic spectrum. Thermal imaging is based on the fact that all objects emit infrared radiation based on their temperature—according to the Law of black body radiation. The amount of radiation emitted by an object increases with the temperature. Thermographic cameras such as uncooled microbolometer detectors detect these radiations and produce images using the temperature variations of those radiations. When viewed by a thermographic camera, warmer objects stand out well against cooler backgrounds.

Based on the principle of thermal imaging [18], it was thought that thermal imaging can be used to detect the hazards on the runway by mounting a thermographic camera to the camera mount at the belly of the aircraft. The thermographic camera would differentiate the runway from the hazards on it because the amount of radiated energy of

the runway would differ from the amount of radiation of an object on the runway. Also, objects on the runway could be detected not only in poor visibility conditions but also in dark conditions because the thermal imaging works with the same efficiency in dark conditions. Because of this advantage, thermal imaging finds its application in military operations for night vision.

But the problem with the application of thermal imaging for hazard detection on runways would arise in cases where the object is on the runway for a long time and emits the same amount of radiation as the runway. In such a case, the object on the runway cannot be detected. Considering such disadvantages and the cost involved, use of the thermal camera was disregarded.

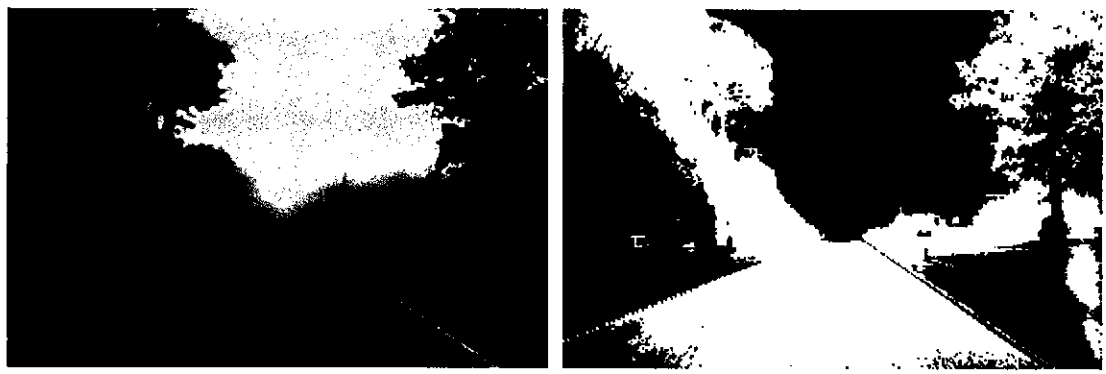
2.3 Image Segmentation

Image segmentation can be defined as the process of partitioning an image into multiple regions. It is used to distinguish objects from the background; thus it can be used to detect the runway and hazards on the runway. Several image segmentation algorithms have been developed in the past and can be used to solve a segmentation problem based on the application. Some of the image segmentation algorithms [19-20] that were tried were *thresholding* and *clustering*.

2.3.1 Thresholding

Image segmentation using thresholding can be performed by setting all the pixels with gray level values over a certain threshold value to a foreground value and all the remaining pixels with gray scale values below the threshold to a background value. This

technique can be used to identify the runway from the background, and thus can be used to detect the hazards on the runway. A simple example of thresholding is shown in figure 2.1. If *f(x,y)* is the input image, then the output image is obtained by setting all the pixel values between certain thresholds to white and all the other pixel values to black.



(a) Original image (b) Segmented image

Figure 2.1: Results of segmentation using thresholding.

From figure 2.1, it is observed that the road component in the original image is segmented from the rest of the background. In a similar manner, this technique can be useful to segment the runway from the background in the given problem. However, finding an appropriate threshold would be a problem because all the runways do not look the same (in terms of color).

2.3.2 Clustering

Clustering can be defined as the process of classifying the objects into different groups or partitioning the data set into different clusters. The data set is partitioned based on parameters like distance measure. The data set can be clustered based on *hierarchical algorithms* or *partitional algorithms* [21]. Hierarchical clustering is based on finding

successive clusters by using previously established clusters. Hierarchical clustering is agglomerative in the sense that it begins by assuming each point in the data set is a separate cluster and ends by merging the clusters to form a larger cluster. However, partitional clustering assumes the entire data set is one complete cluster and proceeds to divide the cluster into smaller clusters.

2.4 Image Enhancement

The main aim of image enhancement is to provide a better interpretation or perception of the information in the image or to provide a better input for other automated image processing techniques. The image enhancement techniques [22] can be classified as

1. Spatial Domain techniques and
2. Frequency Domain techniques.

2.4.1 Spatial Domain Image Enhancement Techniques

The spatial domain techniques refer to procedures that operate directly on the aggregate of pixels composing an image. The functions in the spatial domain can be expressed as

$$g(x,y) = T[f(x,y)] \tag{2-1}$$

where $f(x,y)$ is the input image, $g(x,y)$ is the output image, and T is an operator on f, defined over some neighborhood of (x,y). Spatial domain techniques can be further classified as point processing techniques and mask processing techniques. In point processing techniques, the pixel value at each point in the image (f) is transformed per the

transformation (T) without depending on the neighboring pixels. However, in mask processing techniques (also called as spatial filtering), the pixel values are changed according to the pixel's neighbors.

2.4.2 Point Processing Techniques

Enhancement using point processing techniques is based on changing the pixel value at a given point using the gray value at the same point. The pixel values at a given point are transformed to new values using the transfer function, as indicated in equation (2-1). Because these image enhancement techniques are based on the intensity of a single pixel, these techniques are also called intensity transformations. Some of the simple intensity transformations are image negatives, log transformation and power-law transformation. Figure 2.2 shows the plots of these simple intensity transforms.

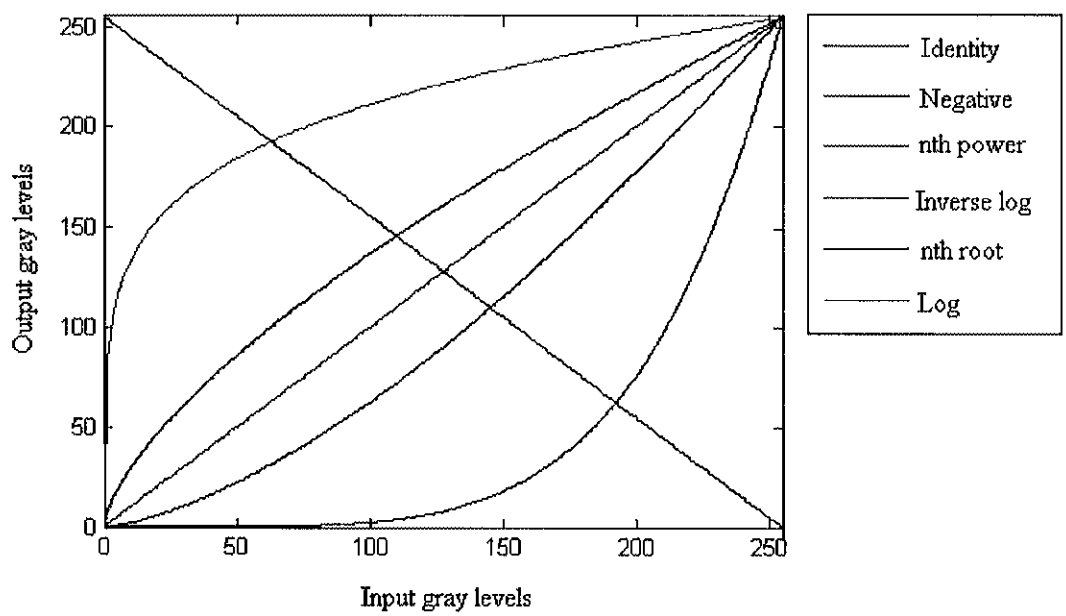


Figure 2.2: Basic gray level transformation functions.

2.4.2.1 Image Negative Transformation

The negative of an image whose gray levels are in the range of [0, L-1] is given by a transformation function as follows:

$$g(m,n) = L-1-f(m,n) \qquad where \begin{matrix} m=1,2,3,...,M \\ n=1,2,3,...,N \end{matrix} \tag{2-2}$$

By reversing the gray level values of an image, the resulting image appears as a photographic negative of the original image. Image negative can be used to enhance the white region that is embedded in the dark regions of the image.

2.4.2.2 Log Transformations

The log transformation is given by

$$g(m,n) = c\log(1+f(m,n)) \tag{2-3}$$

where c is a constant. From figure 2.2, it can be observed that the log transformation maps the narrow range of lower range gray levels in the image to a wider range of output gray levels and the wider range of higher gray level values to a lower range of output gray levels. So, the log transformation can be used in applications that require the expansion of the lower range gray scale values (dark pixels) and compression of higher range gray scale values (bright pixels). The inverse log transformation performs the opposite operation. The inverse log transformation results in compression of the lower gray scale values and expansion of the higher gray scale values. Thus, the important characteristic of log transformation is to compress the dynamic range of images which have a large variation in pixel values.

2.4.2.3 Power-Law Transformation

The power-law transformation is given by

$$g(m,n) = c[f(m,n)]^{\gamma} \tag{2-4}$$

where c and γ are constants. Changing the value of γ, results in the change in the plot of the transformation function. For $\gamma>1$, the transformation is similar to log transformation. However, the transformation is similar to inverse log transformation for $0<\gamma<1$. For $\gamma=1$, the transformation reduces to an identity transformation. Figure 2.3 shows the plot for the power-law transformation for varying values of γ.

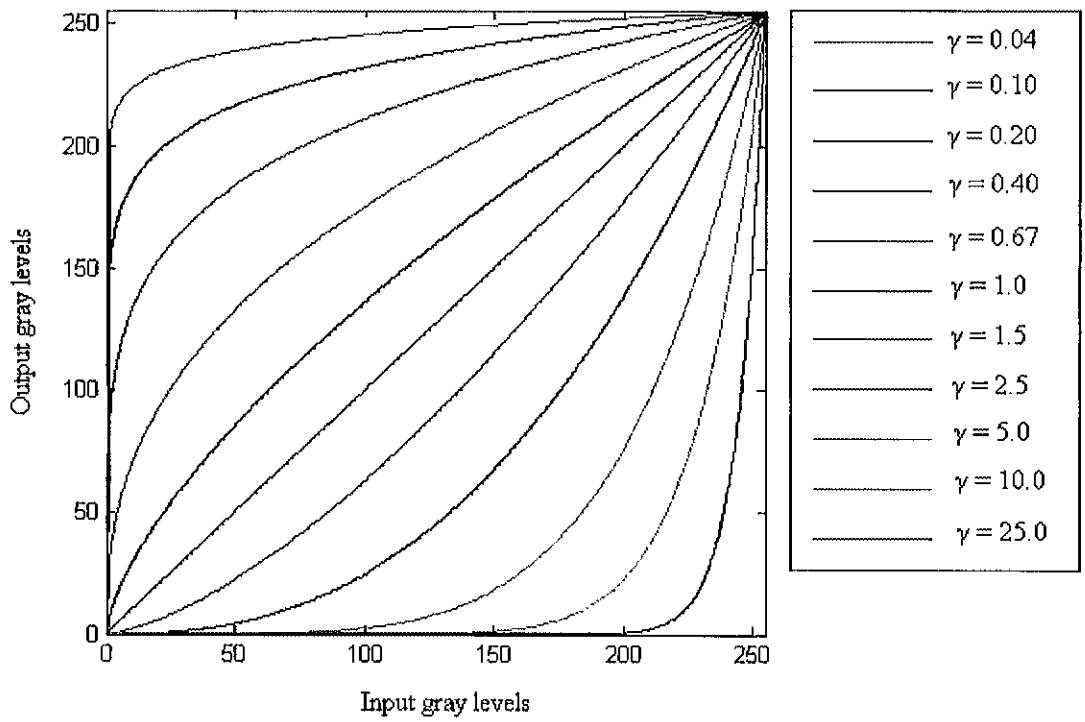


Figure 2.3: Plots of power-law transformation for various values of γ.

2.4.3 Contrast Stretching

Contrast stretching [23] is a simple piecewise linear function that is used to increase the dynamic range of the gray levels in the image to be processed. Contrast stretching attempts to improve the contrast in an image by stretching the range of

intensity values it contains to span a desired range of values. The contrast stretching can be given as follows:

$$\begin{aligned} &for(l=0;l<L;l++) \\ &T[l]=Quantize(L\frac{l-lo}{hi-lo}); \\ &\text{where } 0\le lo<hi\le L-1, \text{ and,} \end{aligned} \tag{2-5}$$

$$Quantize(x)=\begin{cases} 0 & x<0 \\ \lfloor x \rfloor & 0\le x<L \\ L-1 & x\ge L \end{cases} \tag{2-6}$$

It clips a portion of the input at either peak (i.e. it makes a certain portion of lower gray levels to black and a portion of higher gray values to white and stretches the values in between linearly). Figure 2.4 shows that the contrast stretching simply stretches out the histogram without affecting the shape of the histogram.

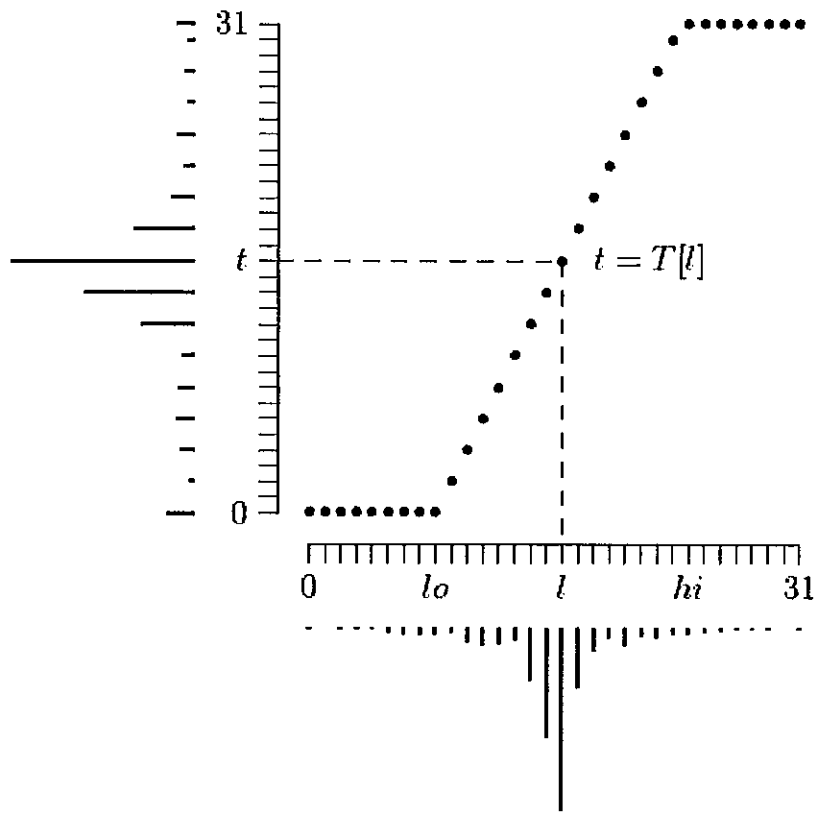


Figure 2.4: Contrast stretching function.

The values of *lo* and *hi* determine the amount of clipping. The values of *lo* and *hi* can be chosen using two approaches namely manual contrast stretching and automatic contrast stretching.

Manual contrast stretch requires the user to explicitly supply two parameters that are used to contrast stretch the image. These parameters are typically the minimum value (*lo*) and the maximum value (*hi*). The parameters are usually based upon knowledge gained by first inspecting the original image histogram.

Automatic contrast stretch uses an implicit approach to specify *lo* and *hi* to control the amount of gray level clipping. The *clo* and *chi* values determine the amount of low-end and high-end clipping desired respectively. Typical values for *clo* and *chi* are 0.005 in which case approximately 0.5% of the image pixels will have their value clipped to black and another 0.5% will be clipped to white.

2.4.4 Histogram Equalization

Histograms of digital images are the basis for several spatial domain image processing techniques. Images can be enhanced significantly by manipulating the histogram of the image. The histogram of an image with gray levels in the range [0, L-1] is given as

$$h(r_k) = n_k \quad \text{for } k = 0,1,2,...,L-1 \tag{2-7}$$

where r_k is the k^{th} gray level and n_k is the number of pixels in the image with the gray value r_k. The normalized histogram is an image obtained by the ratio of number of pixels with the gray value r_k to the total number of pixels in the image. So, the normalized histogram is given by

$$p(r_k) = n_k / n \quad \text{for } k = 0,1,2,...,L-1 \tag{2-8}$$

where $p(r_k)$ is an estimate of the probability of occurrence of gray level r_k. The transformation function for discrete images is given by

$$\begin{aligned} s_k = T(r_k) &= \sum_{j=0}^{k} p_r(r_j) \\ &= \sum_{j=0}^{k} \frac{n_j}{n} \qquad \text{k} = 0,1,2,...,\text{L}-1 \end{aligned} \tag{2-9}$$

The transformation function implies that the resultant processed image is obtained by mapping each pixel with the gray level r_k in the input image into a corresponding pixel with level s_k in the output image.

2.5 Edge Detection

The edge detection algorithms are used to reduce the amount of data to be processed and filter out the information that can be regarded as irrelevant, while preserving the structural information in the image. The edge detection algorithm is used after enhancing the image to obtain the structural details of the runway. Edge detection techniques can be used to detect hazards on the runway because they have many advantages. Edge detection techniques provide an illumination independent image. The process of edge detection enhances the detail in the image even in poor visibility conditions and eliminates the shading effect induced by the enhancement algorithm. Also, the edge-only image is more immune to random noise and is easy to store. Based on these attributes, edge detection was used in this research.

An edge can be defined as the set of connected pixels lying on the boundary between two regions. An ideal edge is a set of connected pixels, each of which is located

at the orthogonal step transition in gray levels. However, factors like optics, sampling and image acquisition imperfections lead to a degree of blurring in the images. The resulting edges do not appear as an orthogonal step but can be closely modeled to a "ramp like" profile. Figure 2.5 shows the model for an ideal edge and the ramp-like edge.

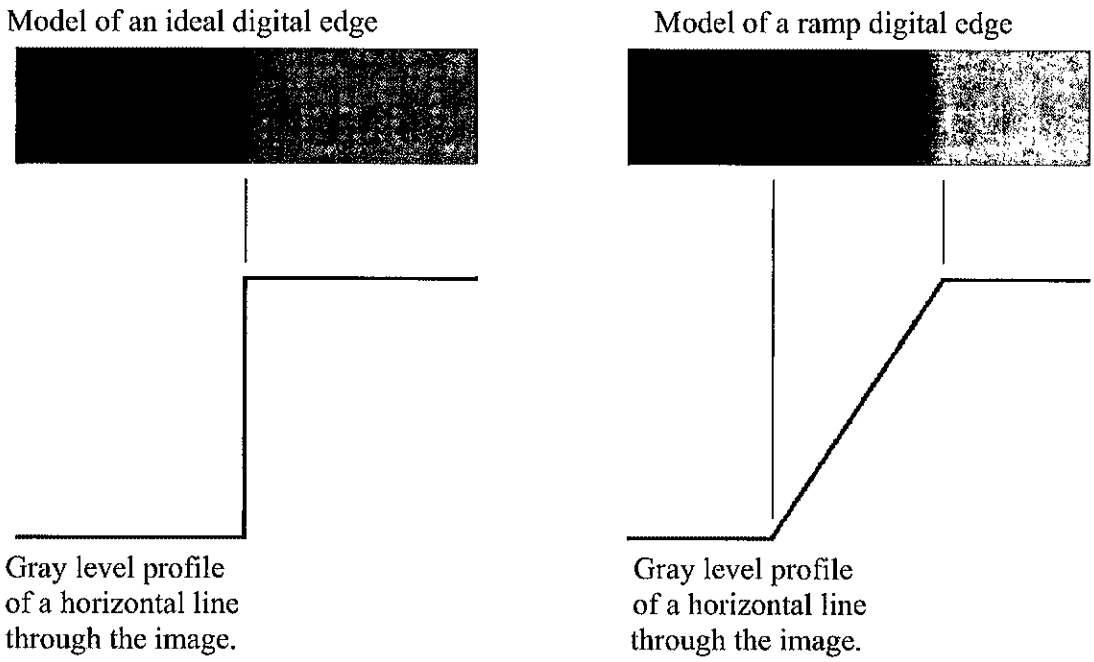


Figure 2.5: Profiles of ideal edge and ramp like edge.

The slope of the edge is inversely proportional to the amount of blurring in the image. So, edges of the blurred images tend to be thick, and the edges of the sharp images are thin. There are many edge detection techniques, but most of the techniques are classified into two groups: *search-based* and *zero-crossing based.* The search-based methods detect the edges by computing the edge strength followed by searching the edge orientation. Search-based methods use the first-order derivative to compute the measure of edge strength. The first derivative is positive at the points of transition into and out of the ramp and is zero at the areas of constant gray level. As the name suggests, the zero-

crossing based methods detect the edges by searching for the zero crossings in the second-order derivative expression computed from the image. The second derivative is positive at the transition associated with the dark pixel, negative at the transition associated with the light pixel and zero along the ramp and constant gray areas in the image. The profiles of the first and second derivative edges are as shown in figure 2.6.

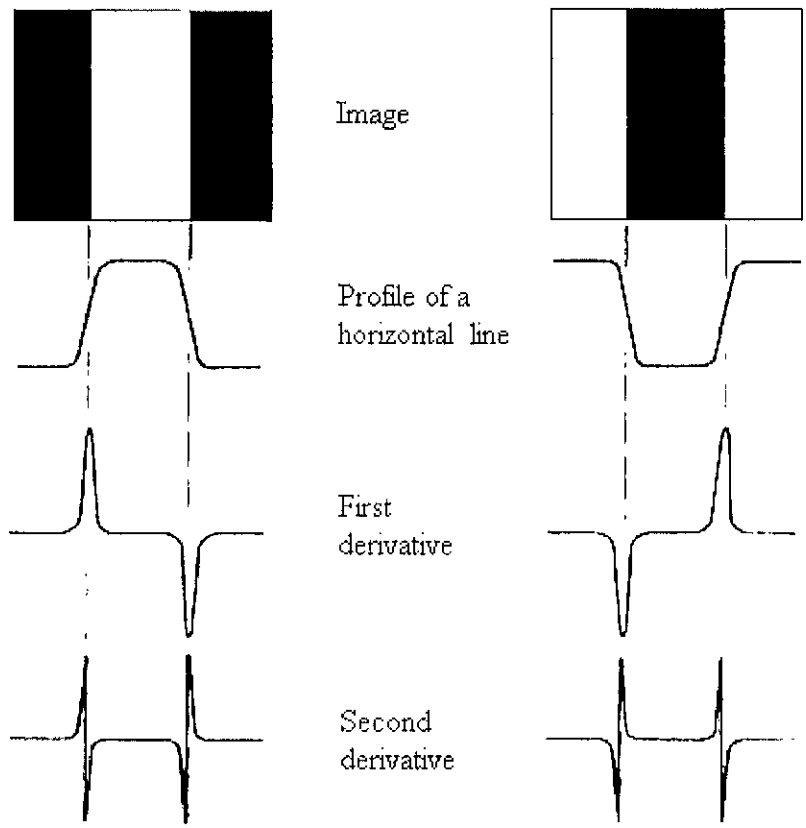


Figure 2.6: Edge profiles for first and second order derivates.

Because the second derivatives of the image are more immune to noise, this work is primarily focused on edge detection in the image using the first order derivatives. Some of the popular edge detection operators using the first-order derivatives are Prewitt, Roberts, Sobel and Canny edge detection algorithms. The following sub-sections discuss these operators in further detail.

2.5.1 Roberts edge detection operator

The Roberts cross operator [24] is an edge detection operator that performs a quick two-dimensional gradient measurement on the image by using a 2 x 2 convolution kernel. Because it uses a 2 x 2 mask, its implementation is awkward as it does not possess a clear center. Implementation using a 3 x 3 kernel is much better because the center pixel is clearly defined. The gradient vector of the image *f(x,y)* at location *(x,y)* is given by

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \tag{2-10}$$

where the magnitude of the gradient vector is given by

$$G = \sqrt{\left(G_x^2 + G_y^2\right)} \tag{2-11}$$

And the direction of the gradient vector is given by

$$\Theta = \tan^{-1}\left(G_y / G_x\right) \tag{2-12}$$

The angle of the gradient vector is measured with respect to the x-axis, and the direction of the edge is perpendicular to the gradient vector at that corresponding pixel. So if A is the original image and G_x and G_y are the results of convolving the image in the vertical and horizontal directions respectively, then they are given by

$$G_x = \begin{bmatrix} +1 & 0 \\ 0 & -1 \end{bmatrix} * A \text{ and } G_y = \begin{bmatrix} 0 & +1 \\ -1 & 0 \end{bmatrix} * A \tag{2-13}$$

where '*' is the convolution operator. The gradient's magnitude and direction can be calculated by equations 2-11 and 2-12. The main reason for using the Roberts operator is

its quick performance. However, this method of edge detection is very sensitive to noise because it uses a 2 x 2 kernel.

2.5.2 Prewitt edge detection operator

The Prewitt operator is an edge detection operator that calculates the maximum response of the set of eight convolution kernels to find the local edge direction for each pixel in the image. These eight kernels can be obtained by taking the kernel in one direction and rotating the coefficients circularly. The magnitude of each pixel in the output is the maximum response of the eight convolutions. This kind of edge detection is also called *edge template matching* because a set of templates are matched with the image, with each template representing the direction of the edge in a certain orientation. Mathematically, the operator uses a 3 x 3 kernel, which is convolved with each pixel in the image to get the gradient. Typically, the Prewitt operator can be given by equation 2-14 where G_x and G_y is the convolution kernel and A is the input image.

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix} * A \text{ and } G_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ +1 & +1 & +1 \end{bmatrix} * A \qquad (2\text{-}14)$$

where '*' is the convolution operator. In a similar manner, each pixel is convolved with the six other convolution kernels that are rotated, to find the maximum response for each pixel and, the output magnitude image.

2.5.3 Sobel edge detection operator

The Sobel operator [25-26] is an edge detection operator that is used to compute the approximation of the gradient of the image intensity function. It is a two-dimensional spatial gradient measurement of the image; thus, it emphasizes the regions of high spatial

frequency that correspond to edges. Mathematically, the gradient of the image intensity function at each image point is a two-dimensional vector with components given by the derivatives of the horizontal and vertical directions.

The Sobel operator is based on convolving the image with a 3 x 3 kernel in the horizontal and vertical directions. The kernels used by the Sobel operator are designed to give a maximum response to the edge running horizontally and vertically with respect to the pixel grid. If A is the original image, and G_x and G_y are the results of convolving the image in vertical and horizontal direction respectively, then they are given by

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * A \text{ and } G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} * A \tag{2-15}$$

where '*' is the convolution operator. The gradient's magnitude and direction can be calculated by the following equations.

$$G = \sqrt{\left(G_x^2 + G_y^2\right)} \text{ and } \Theta = \tan^{-1}\left(G_y / G_x\right) \tag{2-16}$$

The Sobel operator is slower than the Roberts operator, but the larger convolution kernel smoothens the input image to a larger extent, thus making the operator less sensitive to noise.

2.5.4 Canny edge detection operator

Gradient based, edge detection algorithms that were previously discussed are very sensitive to noise; thus, the edge detection results may vary depending on the noise characteristics of an image. Also, the coefficients and the size of the kernel are fixed and cannot be adapted for a given image. Thus, there is a need for an adaptive edge detection

algorithm that is adaptable to the varying noise levels in the image to help distinguish the image content from the noise artifacts.

The Canny operator [27] was designed to be an optimal edge detector operator that would satisfy the following criteria.

1. *Good detection*: the operator must be able to find as many edges in the image as possible.
2. *Good localization*: the edges found in the image must be close to the edges in the original image.
3. *Minimal response*: the edges in the original image must be marked only once and false edges must not be created by the noise in the image.

The Canny operator uses the calculus of variations—a method to find the function which optimizes the given functional to satisfy the criteria. The Canny operator is a multiple stage process. The first step is to use the Gaussian convolution to smooth the image. The next step is to compute the gradient vector by taking the first derivative of the image to highlight the high frequency components in the image. The gradient vector can be found by using a 3 x 3 kernel, similar to the kernel used in the Sobel and Prewitt operators. The gradient magnitude and direction can be found using equation 2-17.

$$G = \sqrt{\left(G_x^2 + G_y^2\right)} \text{ and } \Theta = \tan^{-1}\left(\frac{G_y}{G_x}\right) \tag{2-17}$$

where G_x and G_y return the values of the first derivatives of each pixel in the vertical and horizontal directions respectively. Having found the edge direction, the next step would be to relate the edge direction to a direction that can be traced in an image. After finding the edge directions, the algorithm tracks the pixels along the edge in the edge direction and sets those pixels to zero, resulting in a thin line in the output image. This process of

setting the pixels along the edges to zero is called *non-maximum suppression.* The final step is to eliminate streaking in the image by using thresholding with hysteresis. Thresholding with hysteresis requires two thresholds—T_1 and T_2 (such that $T_1 > T_2$). Tracking can only start at a point if the pixel value is greater than T_1. Tracking continues in both directions from that point until the pixel value falls below T_2.

The effect of the Canny operator is based on three parameters: the width of the Gaussian kernel and the upper and lower thresholds. If the width of the Gaussian kernel is increased, the detector's sensitivity to noise decreases. This results in the loss of fine details in the image. The localization error also increases with an increase in the width of the Gaussian kernel. Usually, the upper tracking threshold can be set too high, and the lower tracking threshold can be set too low to get good results. Lowering the upper threshold to a very low value can lead to an increase in the spurious edges in the image, and elevating the lower threshold to a very high value will result in the breaking up of the noisy edges.

2.6 Summary

In this chapter, different approaches for detection of hazards on the runway were investigated. The use of thermal images can be the simplest process for the detection of hazards on the runway, but in foggy conditions, if the object is present on the runway for a long time, there would not be much difference in temperatures of the object and the runway. In such conditions, the object on the runway cannot be detected. Image segmentation techniques like thresholding and clustering could be alternate approaches. Image segmentation based on thresholding can provide good results provided the gray

scale value of the runway is precisely known. Clustering algorithms can be used, but the number of clusters has to be provided in advance for segmentation.

An alternate approach would be to enhance the image to make it clear and perform edge detection on the enhanced image. Several image enhancement and edge detection techniques were studied. A block diagram for the detection of hazards using image enhancement and edge detection is presented in the following chapter.

CHAPTER 3

ALGORITHMS FOR HAZARD DETECTION ON RUNWAYS

This chapter presents the algorithms to detect hazards on the runway [28] that are developed in this research work. The first step is to compute the smoothness coefficient in the image to determine if the image is foggy or not. The next step would be to enhance the image using Multi-Scale Retinex. After the process of image enhancement, the next step is to detect the edges in the enhanced image. The last step is to determine if there is any hazard on the runway.

3.1 Proposed Algorithm: In Brief

The process of hazard detection, encapsulated in Figure 3.1, can be described by the following steps:

1. Collecting the imagery and the GPS information: the navigation data and the imagery are correlated with each other to correct for the impact of platform—airplane—motion on the acquired imagery.
2. Enhancing the imagery with the multi-scale Retinex (MSR) algorithm. The MSR provides (almost) illumination independence. It can be applied to all imagery since it does not degrade "good" imagery.
3. Performing edge detection to obtain salient features like runway edges in the image. The determination of runway boundaries is important in distinguishing between "hazards"—objects of a particular size on the runway—and other objects. The SUSAN edge detection algorithm is used to detect edges after performing image enhancement. The SUSAN operator is non-linear and employs intensity information of an image for edge and corner detection. This method is

simple, but it can acquire the edges and corners in the image with precise localization even for noisy images because it is insensitive to local noise.

4. Determining if the objects on the runway are hazardous by determining their objects from the temporal sequence of images.

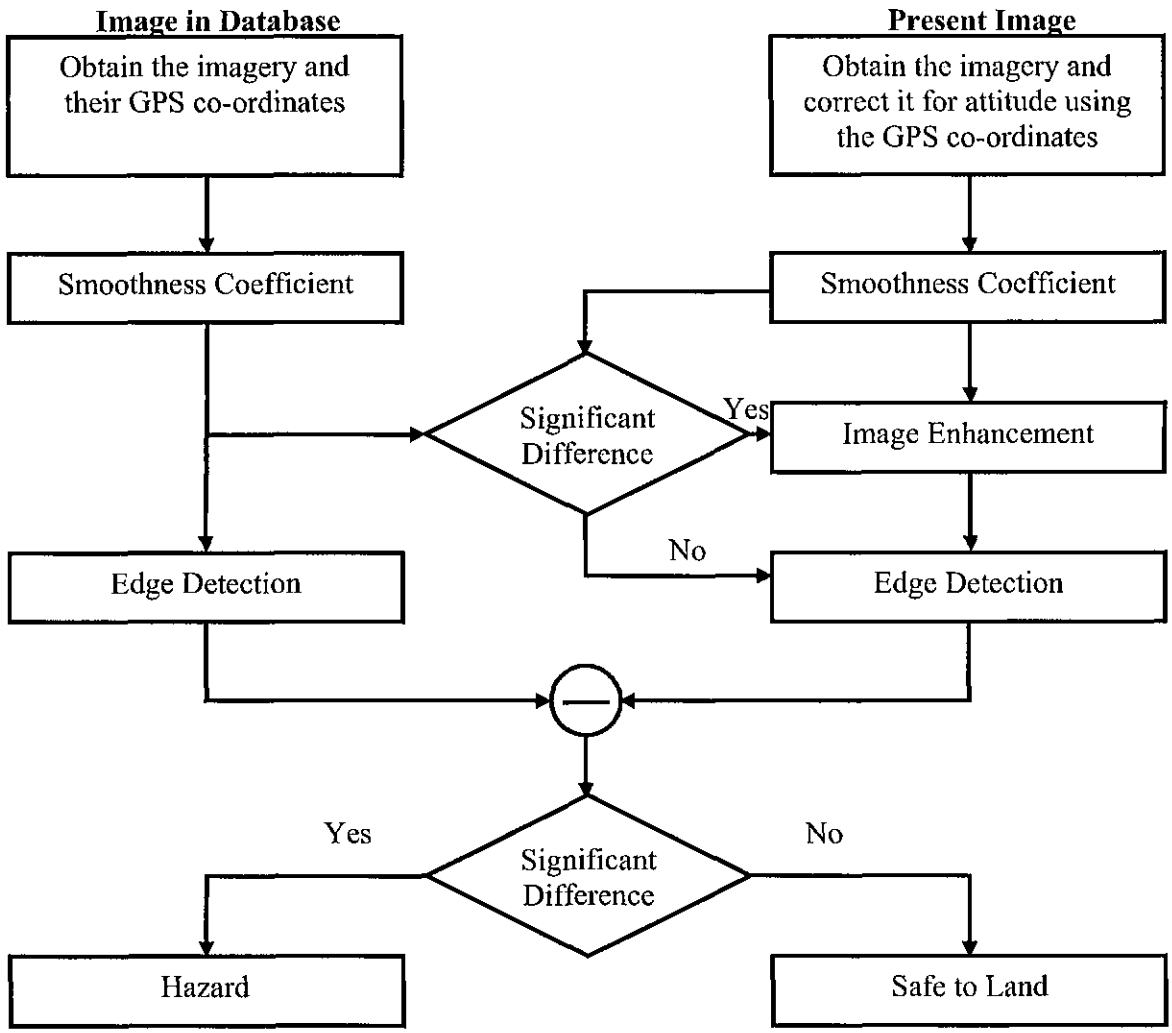
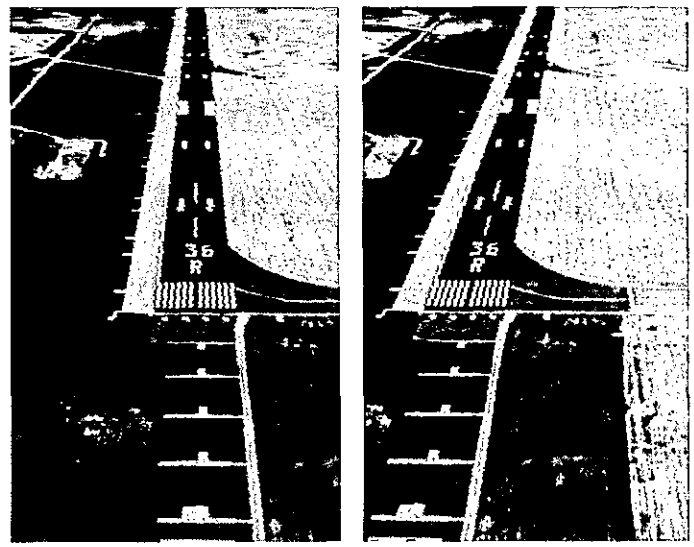


Figure 3.1: Illustration of flow for detection of hazards on runway.

The process is based on comparing the imagery on a clear day with that acquired under poor visibility conditions. The first step is to obtain the imagery on a clear day with its corresponding GPS information and store it in a database. This provides a canonical

representation of the runway without hazards. The GPS information is required for comparison of corresponding frames from foggy imagery. Additionally, the aircraft is not always in a steady state and is tilting on either side before landing; for the aircraft does not have the same field-of-view all the time.



(a) Original image (b) Image shifted by 45°

Figure 3.2: Effect of aircraft tilt on runway imagery.

Figure 3.2 shows an example of such changes. Even though these images are acquired from the same GPS locations, they do not look identical. This is due to the motion of the aircraft. We use the GPS and the aircraft's attitude and altitude information to address this issue. Comparing the GPS information for a given situation with the GPS information in the database, the image can be rotated for correct registration. The next step is to compute the smoothing coefficient for the imagery from the clear image. The smoothing coefficient is the measure of the smoothness in the image. It can be calculated by measuring the amount of energy at high frequencies. The value is larger for smooth images and smaller for high contrast images. The smoothing coefficient value for the

clear image is stored in the database. After calculating the smoothing coefficient, the next step is to perform edge detection using the SUSAN edge detection algorithm. SUSAN edge detection is preferred over other state-of-the-art technologies because of its speed and accuracy. The edge detected image is also stored in the database for future comparisons. Thus, the database for a given runway consists of clear imagery with GPS information, smoothing coefficient and edge detected image.

In a given situation, when an aircraft is getting ready to land in foggy conditions, the imagery is obtained from the camera pod and is corrected for the attitude and altitude by using the corresponding GPS information. The SC is calculated for the imagery and is compared with the values in the database for the corresponding GPS information. Typically, foggy images are low contrast images and have a higher SC than that of clear images, so if the SC is higher for the imagery when compared to the value in the database, then it indicates that the imagery is foggy. If the imagery is foggy, it has to be enhanced using an image enhancement algorithm. Image enhancement is not required if the SC is lower than the value in the database. Multi- Scale Retinex is used to enhance the imagery because of its ability to enhance the imagery having wide ranging lighting and exposure variations.

The next step is to perform edge detection of the obtained image. The threshold for the enhanced image is lower in order to obtain all the edges in the image. The threshold is higher for the clearer image which does not require enhancement because the transition at the edges is clear. After the edge image is obtained, it is subtracted from the edge image in the database. If a significant difference is noticed, then it can be concluded

that the runway is not clear and it is hazardous to land the aircraft on the runway; otherwise, it can be said that the runway is clear for landing.

3.2 Moving Object Detection on the Runway

Detection of moving objects on the runway is simpler when compared to the detection of static objects on the runway. Detection of moving objects on the runway does not require the storage of the imagery in the database which is the case in the detection of static objects on the runway. Moving objects can be detected by the inter-frame analysis of the runway. Figure 3.3 shows the algorithm for detection of moving objects using the inter-frame analysis. To detect moving objects on the runway, the first step is to obtain the imagery and correct it for its attitude. The next step is to obtain the imagery and enhance it using Multi-Scale Retinex. After enhancing the image, the edge image is obtained using the SUSAN edge detection algorithm. The process of enhancing and edge detection is performed on the succeeding frame too. These two frames, containing the edge information, are registered on a common grid to correct for the motion of the airplane.

This can be done using the GPS information from the two frames. Then, the two frames are subtracted to obtain a difference image. If the two images do not have a significant difference, then the difference image does not contain a lot of information. However, if the two images have a significant difference, then the difference image contains a lot of information. If the difference image does not have significant difference, then the runway is said to be clear for landing, but if the difference image has a significant difference, then it is said that the runway has a moving object on it.

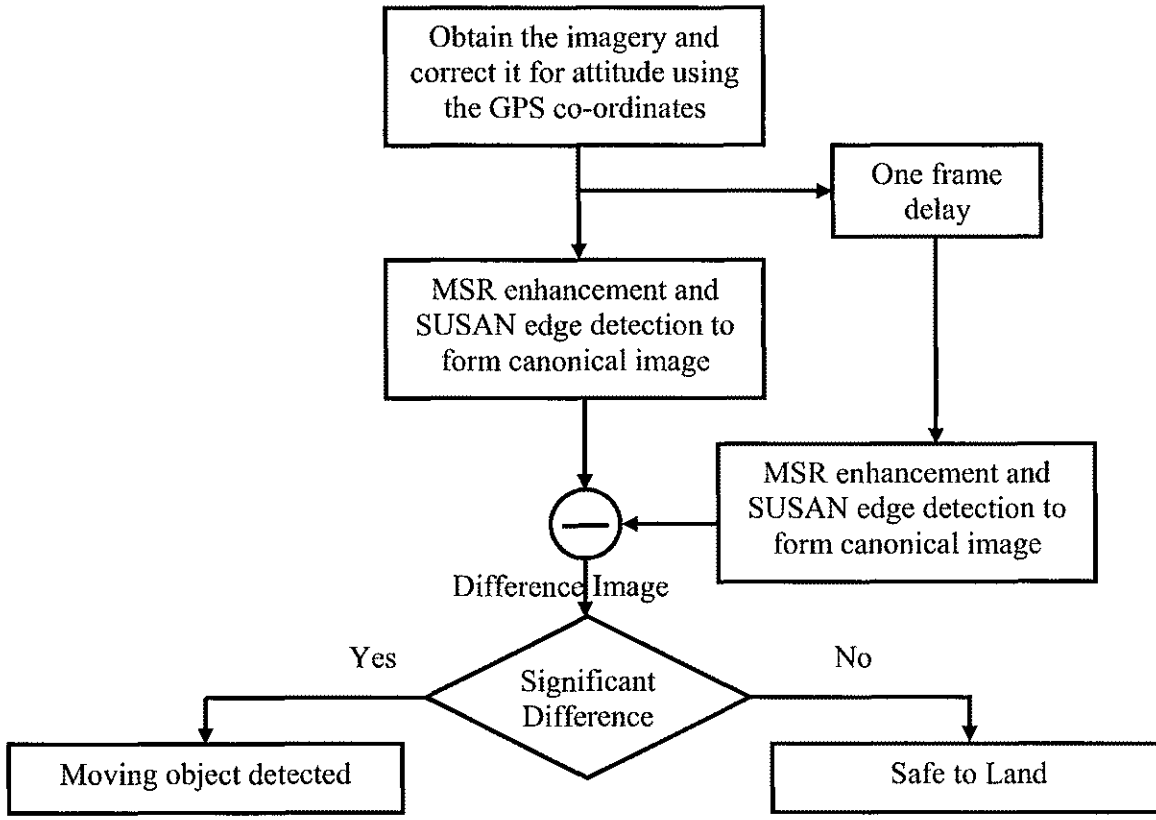


Figure 3.3: Block diagram for detection of moving objects on the runway.

3.3 Smoothness Coefficient

The degree of smoothness in the image depends upon the amount of energy in the high frequencies. Hence, the smoothness in the image can be quantified by measuring the energy in the image at high frequencies. The smoothing coefficient [29] is given by

$$S^2 = \frac{M_1 M_2}{\sum_{\omega_1,\omega_2} \left| \hat{I}(\omega_1,\omega_2)\hat{G}(\omega_1,\omega_2;\zeta = 1.0;\rho_c = 0.3) \right|^2} \qquad (3\text{-}1)$$

where $\hat{I}$ is the discrete Fourier transform of $M_1 \times M_2$ input image, and $\hat{G}$ is a high-pass filter given by

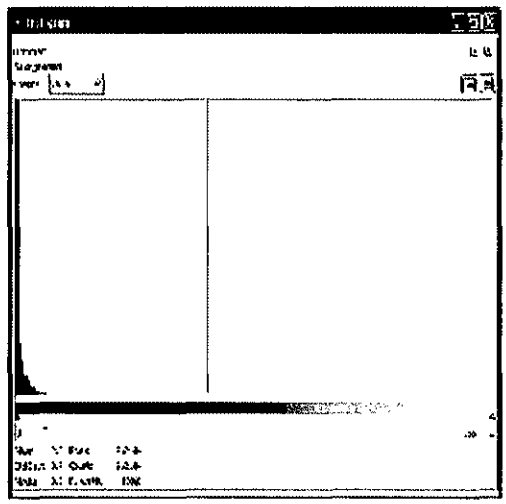
$$\hat{G}(\omega_1,\omega_2;\zeta;\rho_c) = \exp[-(\omega_1^2+\omega_2^2)/\rho_c^2] - \zeta \exp[-2.56(\omega_1^2+\omega_2^2)/\rho_c^2] \qquad (3.2)$$

The smoothness coefficient, S, represents the reciprocal of the amount of energy in the high-pass filtered version of the input image, I; the higher the amount of energy, the more high frequency information there is. Since the high-frequency information is directly correlated to the fine details in an image, the smoother an image the larger the S. Because $\hat{G}$ is in the frequency domain, a two dimensional Discrete Fourier Transform (DFT) and Inverse Discrete Fourier Transform (IDFT) were performed and is given by $\hat{I}(\omega_1,\omega_2)$.

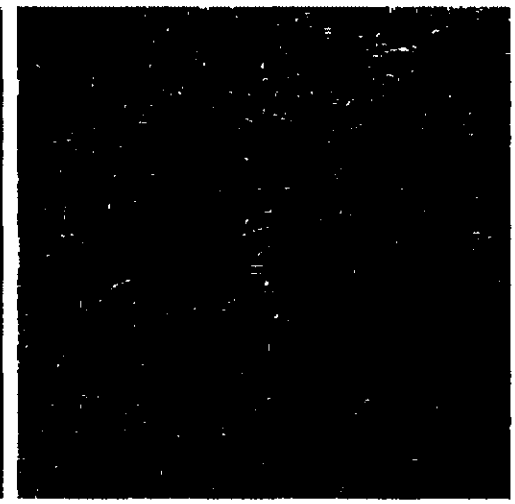
The Gaussian high pass filter is a lateral inhibition filter that is used to bring out the features in the image. ω_1, ω_2 and ρ_c are used to determine the amount of attenuation in the pass band and ζ determines the amount of inhibition. The high pass filter behaves as a pure Gaussian blurring filter for $\zeta = 0$ and can be used for interpolation and noise reduction. For a ζ value of 1.0, the Gaussian high pass filter acts as a difference-of - Gaussian filter. The Difference-of-Gaussian [30] (DoG) filter acts as an edge detection operator by performing two Gaussian blurs on the image, with a different blurring radius for each, and subtracting them to produce the resulting edge detected image. The Gaussian blurring radii are the important parameters in the Difference-of-Gaussian filter. If the size of the smaller radii is increased, then the resulting edges in the image are thicker. If the size of the larger radius is decreased, then the threshold for edge detection increases, and some of the edges are not detected. However, better results are obtained if the radius of the second Gaussian blur is larger than the radius of the first Gaussian blur. The following results were obtained for $\rho_c = 0.3$ and $\zeta = 1$.

(a) Original image

(b) Result of lateral inhibition filter



(c) Histogram of (b)



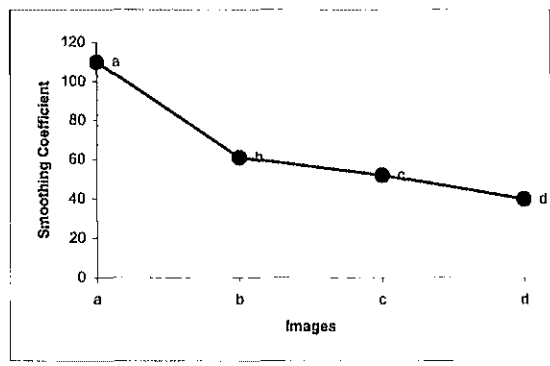
(d) Contrast stretched result of (b)

Figure 3.4: Results of lateral inhibition filter.

Figure 3.4(a) is the original image and figure 3.4(b) is the resulting output. The second image appears to be a blank image, but the histogram of the image shows that it has all the features at lower gray levels. The histogram of the image is given by figure 3.4(c).To visualize the lower gray levels in the image, the gray levels are contrast stretched resulting in figure 3.4(d).

Figure 3.4(d) shows that figure 3.4(b) shows the edges in the image. The smoothness coefficient is computed by adding all the gray values in the image scaled by the product of the rows and columns. Figure 3.5(a) shows the image of a bridge with different fog levels.

(a) Images of a scene with decreasing amount of fog



(b) The corresponding smoothness coefficient values

Figure 3.5: Smoothness Coefficient as a function of fog density; the heavier the fog, the larger the smoothness co-efficient.

From the graph, it is observed that the smoothness coefficient values decrease with the decrease in the amount of fog in the images.

3.4 Multi-Scale Retinex

A fundamental concern in the development of resilient, vision-based, automation technology is the impact of wide-ranging extraneous lighting and exposure variations on the acquired imagery. This concern can be considerably ameliorated by the application of the (MSR) image-enhancement algorithm. The MSR [31-39] is a non-linear, context-dependent enhancement algorithm that provides color-constancy, dynamic range compression and sharpening:

$$R_i(x_1,x_2) = \sum_{k=0}^{\kappa} w_k(\log(I_i(x_1,x_2) - \log(I_i(x_1,x_2) * F_k(x_1,x_2))), \quad i = 1,....,N \qquad (3\text{-}3)$$

where I_i is the i^{th} spectral band of the N-band input image, R_i is the corresponding Retinex output, '*' represents the (circular) convolution operator, F is a (Gaussian) surround function, and κ is the number of scales. The Gaussian surround function is given by:

$$F_k(x_1,x_2) = a_k^{-1} G_k(x_1,x_2) \qquad (3\text{-}4)$$

$$G_k(x_1,x_2) = \exp(-(x_1^2 + x_2^2)/\sigma_\kappa^2) \qquad (3\text{-}5)$$

$$a_k = \sum_{x_1,x_2} G_k(x_1,x_2) \qquad (3\text{-}6)$$

The σ_κ are scale parameters that control the performance of the SSR: small σ_κ lead to SSR outputs that contain the fine features in the image at the cost of color, and large σ_κ lead to outputs that contain color information but not fine detail. In order to extract consistent scene structure from any image under widely varying scene and sensor conditions, one has to think in terms of transforming the image into a "canonical" representation that effectively eliminates such undesirable variability. The MSR has

proven to be a powerful tool for doing just this. Because of its dynamic range compression and illumination independence properties, the MSR provides consistent rendering for imagery from highly diverse scene and sensor conditions. To expand the performance envelope of the MSR to handle narrow dynamic range images encountered in turbid imaging conditions such as fog, smoke, and haze, dim lighting conditions, or significant under- or over-exposures, we have developed a "smart" framework of visual quality measurements and enhancement controls that we call the Visual Servo (VS). The VS assesses the quality of the image in terms of brightness, contrast and sharpness, and controls the strength of the MSR enhancement. This combination of the VS with the MSR is called the VMSR. Figure 3.6 shows a sequence of images and their enhancement under visibility conditions that range from acceptable to unacceptable. Figure 3.6 shows that the image enhancement operator successfully compensates for changing illumination conditions and exposure errors. The camera aperture, shutter speed, and ISO setting were constant over this sequence. The enhanced image provides useful information in every case regardless of the caliber of the original data. Additionally, the enhancement can provide better-than-observer performance in many cases, especially when the obscuration is due to fog, rain, or light clouds in otherwise good illumination. Figure 3.7 shows the performance of the image enhancement operator on imagery acquired under hazy and cloudy imaging conditions. The enhancements were compared with the recollections of the observer about the extent to which he could discern features with the naked eye, or through the camera, at the time the image was acquired. In each case, according to the observer, the enhanced imagery provided more information than could be discerned either through the view-finder of the camera or with the naked eye. Although this is not a

rigorous scientific test, it does justify laying the groundwork for further testing and analysis.

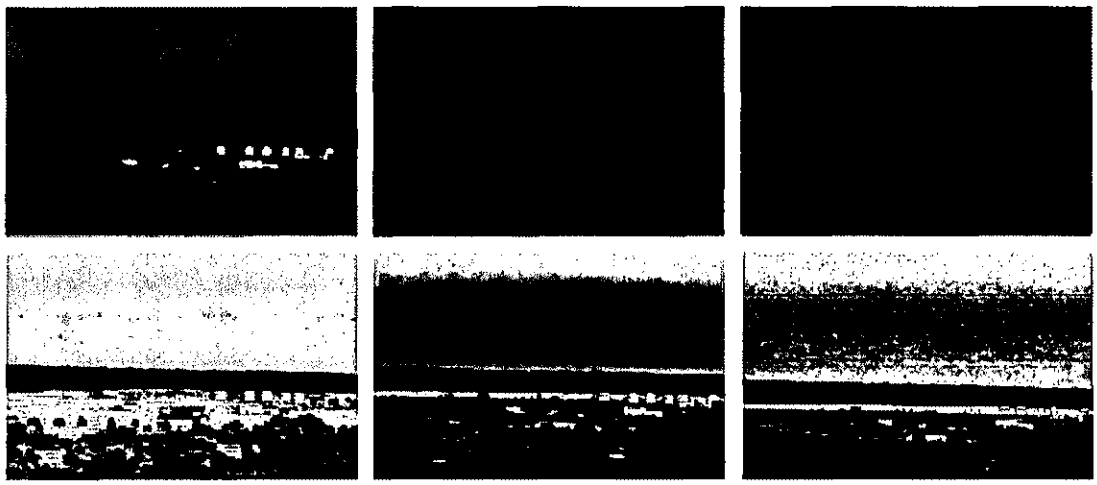


Figure 3.6: Results of Multi-Scale Retinex; (Top) Original sequence of images with changing illumination conditions; (Bottom) Corresponding Retinex outputs.

The image enhancement process also provides illumination independence, i.e., the output of the algorithm is (almost) independent of the type, or level, of illumination under which the image was acquired. This is especially critical for automatic classification and detection algorithms that rely on comparing imagery of the same scene at different times. The ability of the algorithm to produce images that are independent of the change in illumination conditions due to changing sun angle and atmospheric conditions considerably simplifies the automation process for detection and classification. Figure 3.8 shows an example illustrating the illumination independent output produced by the algorithm. The fundamental problems relating to enhancement of still imagery have been addressed in Jobson, et al. and Rahman et al. Additionally, issues relating to enhancement of imagery under poor visibility conditions have been addressed in Jobson et al. and Woodell et al.

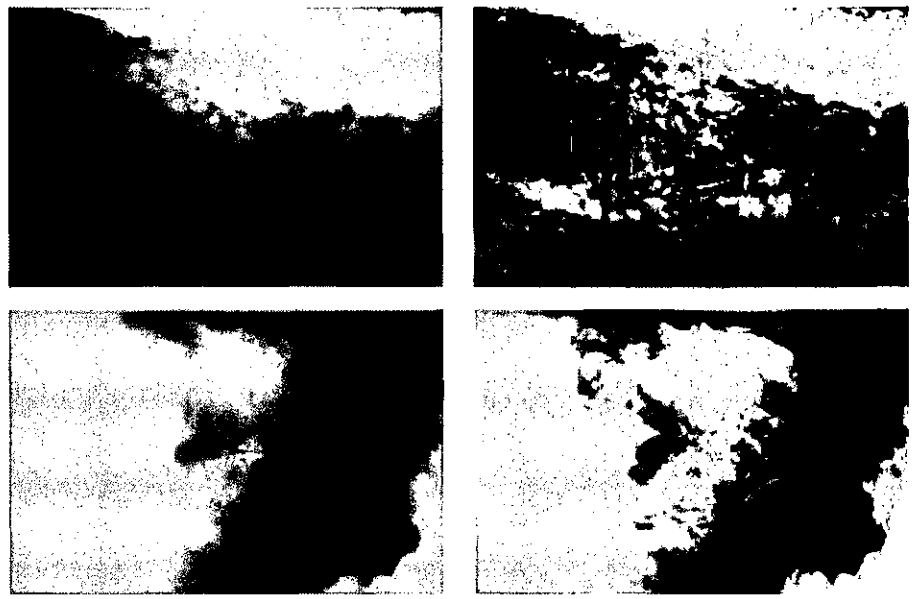


Figure 3.7: Performance of MSR on foggy images; (Left) Original Images; (Right) MSR outputs.

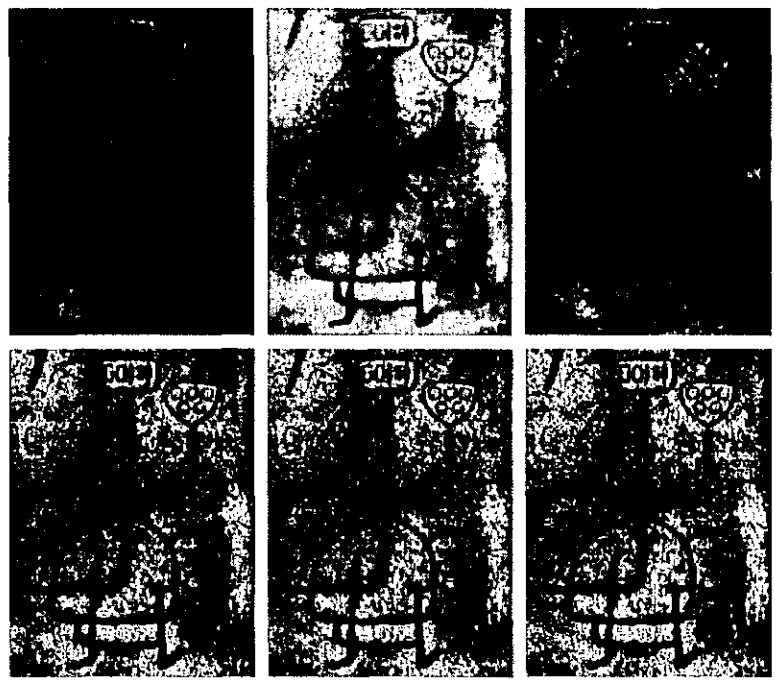


Figure 3.8: Illumination independent output of MSR. The impact of illuminant change was simulated by red, blue, and green shifting an image (top row). The MSR outputs are almost perfectly color constant (bottom row).

3.5 SUSAN Edge Detection

Edge detection is a basic low level primitive for image processing which conveys the structural information about the structures in an image. Edge detection performs the filtering operation to reduce the amount of data in the image by removing the irrelevant information and preserving the structural information in the image.

There are many edge detection operators like Canny, Sobel, Marr-Hildreth, that are widely used which have their own drawbacks. Some of the drawbacks are poor connectivity at the edges and the corners being rounded. Also, with the increase in the Gaussian filter, there is a decrease in the noise levels at the expense of accuracy in localization of edges. SUSAN [40-43] edge detection algorithm, a non-linear filtering operator, provides a solution for the drawbacks of most of the edge detection operators.

The SUSAN principle can be described by considering figure 3.9 which shows a dark rectangle on a white background. The figure consists of five circular masks at different locations on the dark rectangular block. The center pixel of each mask is called the "nucleus".

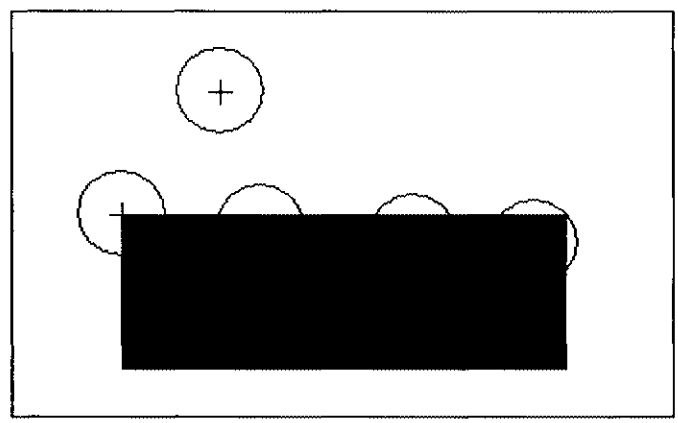


Figure 3.9: Five circular masks at different locations in an image.

If the brightness of each pixel in the mask is the same as the brightness of the mask's nucleus, then the area formed by the pixels with similar brightness values is called the USAN (Univalue Segment Assimilating Nucleus). The USAN area for the circular masks is shown in white in figure 3.10.

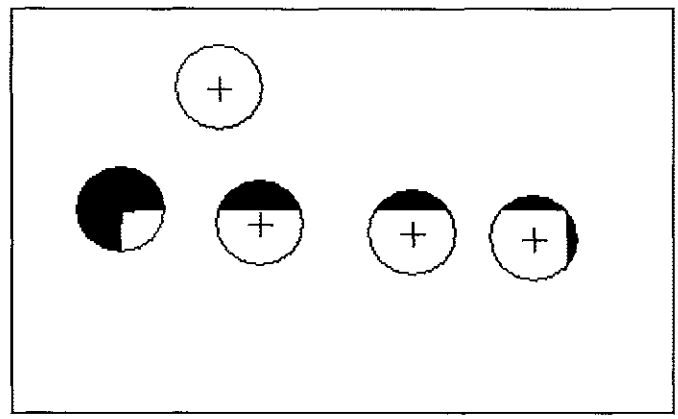


Figure 3.10: Five circular masks with similar coloring; USAN's are shown in white.

From figure 3.9 and figure 3.10, it is noticed that the USAN area is at its maximum when the nucleus of the mask lies on the flat region in the image. The USAN area is almost half as big when the nucleus lies near the edge of the rectangular block, and it further reduces with the nucleus at the corner of the rectangular block. Thus, the USAN area contains lot of structural information and can be considered as effective region finding on a small scale. Also, this method of feature detection is very different from other feature detection techniques in that it does not use any derivatives in the image, and this method does not require any noise reduction as compared to many other techniques.

This theory of associating each pixel in the mask with the nucleus of the mask is the basis for the SUSAN principle of edge detection. The acronym SUSAN (Smallest Univalue Segment Assimilating Nucleus) comes from the principle which states that an

image processed to give as output inverted USAN area has edges and two dimensional features strongly enhanced, with the two dimensional features more strongly enhanced than edges. The SUSAN edge detection operator provides better performance in the presence of noise because it does not use any image derivatives. The integrating effect in association with its non-linear response provides a strong noise rejection.

The SUSAN edge detection algorithm is implemented using a circular mask to give an isotropic response. The circular mask used has a radius of 3.4 pixels and can be obtained by using a 7 x 7 pixel mask.

$$Mask = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix} \tag{3-7}$$

The mask is applied to each pixel in the image, and the brightness of each pixel in the mask is compared to the brightness of the nucleus using the following equation:

$$c(\vec{r}, \vec{r}_0) = \begin{cases} 1 & \text{if} \left| I(\vec{r}) - I(\vec{r}_0) \right| \le t \\ 0 & \text{if} \left| I(\vec{r}) - I(\vec{r}_0) \right| > t \end{cases} \tag{3-8}$$

where $\vec{r}_0$ is the position of the center pixel of the mask, $\vec{r}$ is the position of the pixel in the mask, $I(\vec{r}_0)$ is the intensity of the nucleus, $I(\vec{r})$ is the intensity of other pixels in the mask and t is the threshold. The parameter t is used to determine the minimum contrast of features and maximum amount of noise to be ignored. For a foggy image, the value of t is smaller when compared to that of a clear image because the foggy images have lower contrast. The number of pixels in the USAN is counted and is given by

$$n(\vec{r}_0) = \sum_{\vec{r}} c(\vec{r}, \vec{r}_0) \tag{3-9}$$

The total n gives the total USAN area. After finding the value of n, the USAN area is compared with the geometric threshold and is given as

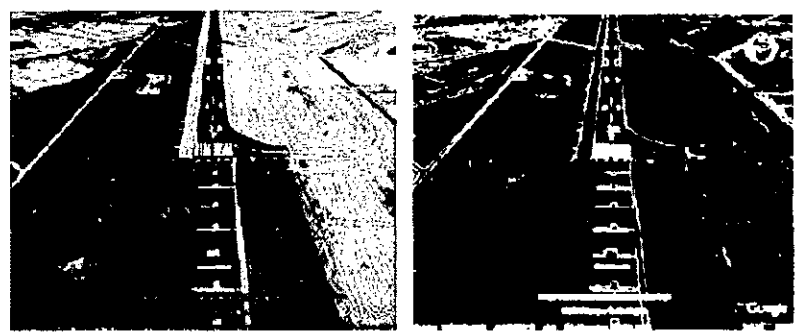
$$R(\vec{r}_0) = \begin{cases} g - n(\vec{r}_0) & \text{if } n(\vec{r}_0) \le g \\ 0 & \text{otherwise} \end{cases} \tag{3-10}$$

where the geometric threshold (g) is set to $3\,n_{\max}/4$ for optimal noise rejection with $n_{\max}$ being the number of pixels in the mask and $R(\vec{r}_0)$, the initial edge response.

The algorithm gives pretty good results, but a much more stable equation which is a smoother version of equation 3-8 is as follows:

$$c(\vec{r}, \vec{r}_0) = \exp\left(-\left(\frac{I(\vec{r}) - I(\vec{r}_0)}{t} \right)^6 \right) \tag{3-11}$$

This equation allows the pixel brightness to vary slightly without having a large effect on the value of c, even if it is near the threshold position. Figure 3.11 shows the edge image of an image in clear conditions:



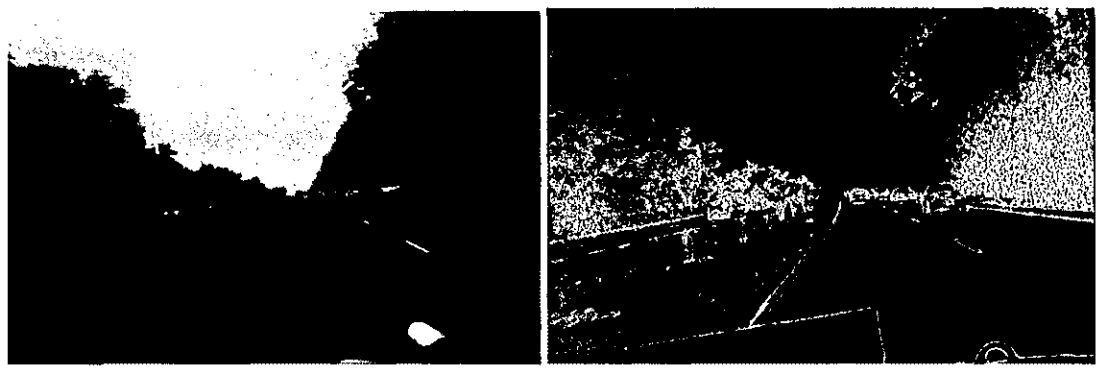
(a) Original image (b) SUSAN edge detection output

Figure 3.11: Results of SUSAN edge detection for clear day imagery.

For the images in clear visibility conditions, the threshold was set to 20, which is the default threshold value for the SUSAN edge detection operator, but when the same

default value was set to the foggy images, the resulting edge images lost a lot of information. Increasing the threshold resulted in the further loss of information, so different thresholds lower than the default threshold values were tried. The results were pretty good for a threshold of 10, but were not consistent for all foggy images. Images with less fog gave good results for the threshold of 10, but the images with a lot of fog did not produce good results. Thus, the threshold value of 5 was set for foggy images. The simulation was run on around 50 foggy images with different levels of fog, and the threshold of 5 produced good results for most of the images. Figure 3.12 shows the results obtained for some of the foggy images.

(a) Original image (b) SUSAN edge detection output



(c) Original image (d) SUSAN edge detection output

Figure 3.12: Results of SUSAN edge detection for foggy imagery.

With the decrease in the threshold value, the neighboring pixels in the mask can afford to have very little variation when compared to the nucleus of the mask. This is because the foggy images have high brightness and low contrast, and the variation between the pixels at the edges is pretty low. In order to pick up the edges in such foggy conditions, the threshold is set to a low value to extract all the possible edges. Further reduction in the threshold resulted in a very noisy image because the algorithm picked up the smallest variations in a flat region of the image indicating that they were edges. Thus, it can be said that the threshold t indicates the minimum contrast of edges to be picked and can be controlled easily.

3.6 Summary

In this chapter, the algorithms for detection of hazards on the runway in poor visibility conditions are presented. This chapter also presents the intricacies of the problem and the possible solutions. Several concepts like smoothness coefficient, Multi-Scale Retinex and SUSAN edge detection operator, which are used in the process of hazard detection, are discussed, and the results of each of these operators is provided.

CHAPTER 4

EXPERIMENTAL RESULTS

In this chapter, the performance of the algorithm to detect hazards on the runway is assessed. The performance of the algorithm is tested for different kinds of imagery in different visibility conditions. The comparative analysis of the proposed algorithm to other possible techniques for detection of hazards on the runway is also presented.

4.1 Data Set

An image dataset was collected for testing and evaluating the proposed algorithm and comparing the results. The imagery of a clear day was collected from different sources on the web and the runway images were acquired from Google Earth. The foggy images were captured using the Canon EOS-1D Mark II N camera. This camera was used to acquire the imagery because of its ability to produce high resolution images without losing any information.

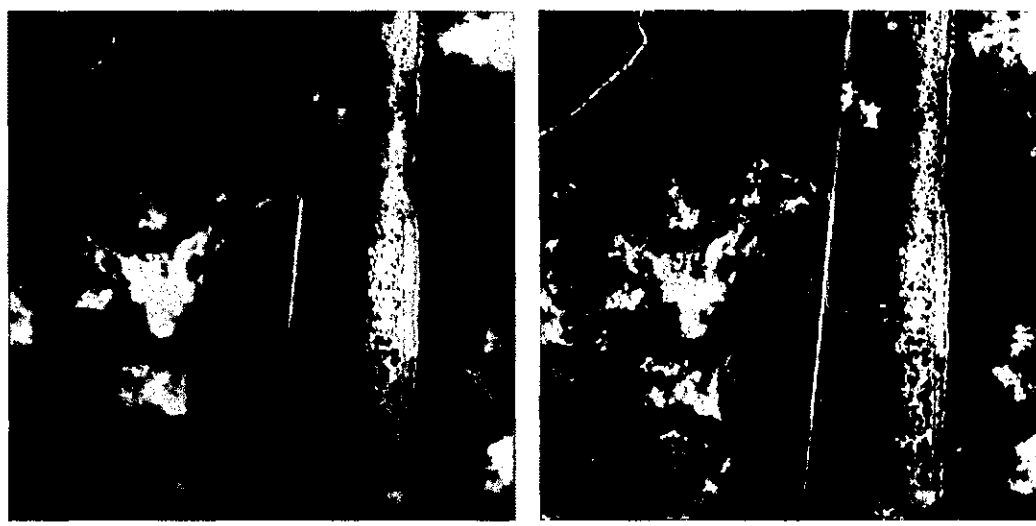
4.2 Implementation Platform

The implementation platform plays an important role in facilitating a practical and flexible development approach. The platform needs to provide adequate flexibility and a good debugging environment. Because the algorithm would be in use in a real time scenario, where time is a major constraint, the performance was a major criterion to decide on the platform to be used. Hence, C and MATLAB were used because of their maximum flexibility, low cost and fast performance. The proposed algorithm was

implemented on a personal computer with a Pentium 4 processor, 1GB RAM and a 3.2 GHz processor.

4.3 Experiment

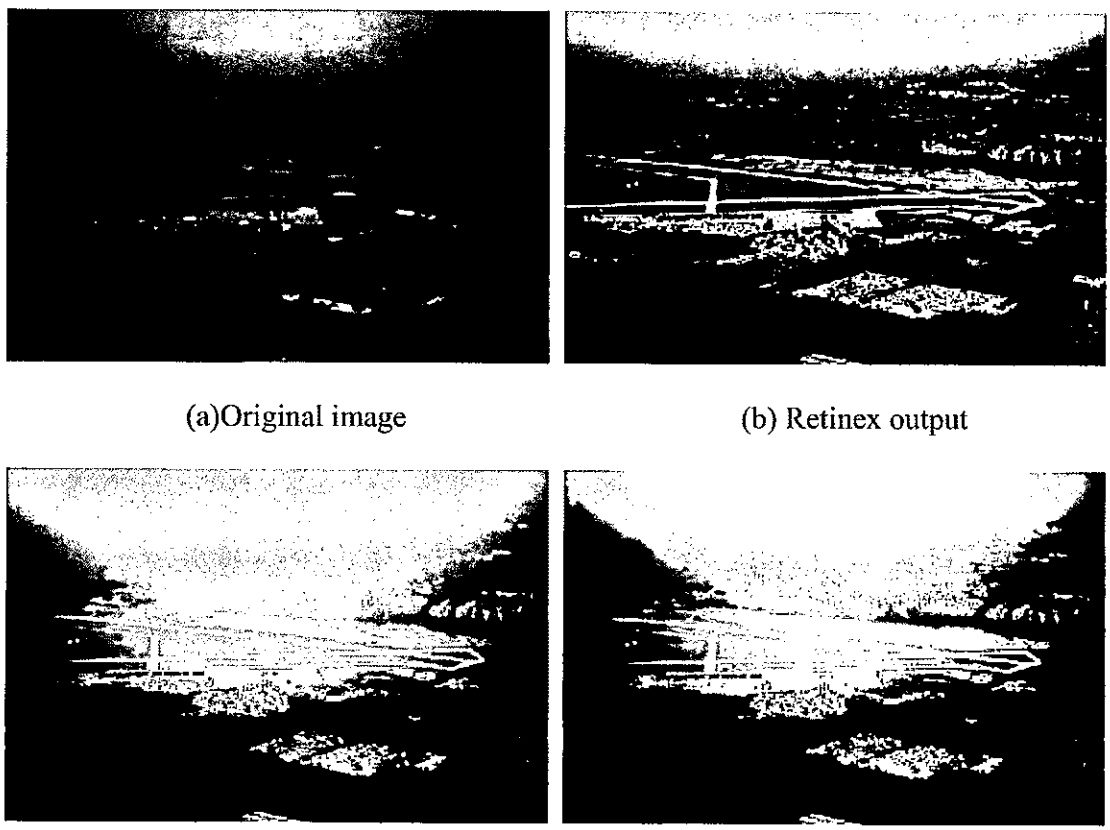
The proposed algorithm provides a novel method for detection of hazards on the runway in poor visibility conditions. The major problem in this work was to automate the process for the algorithm to work in varying illumination and weather conditions. The concept of smoothness coefficient solved most of the problems in the automation process by differentiating foggy images and clear images by considering the fact that the foggy images are smoother and have low contrast when compared to clear images. After differentiating the foggy images from the clear images, the next task was to enhance the foggy images. Multi-Scale Retinex was used to enhance the images because of its performance in poor visibility conditions. Figure 4.1 shows some of the results of the Multi-Scale Retinex.



(a)Original image (b) MSR output

Figure 4.1: Results of Multi-Scale Retinex.

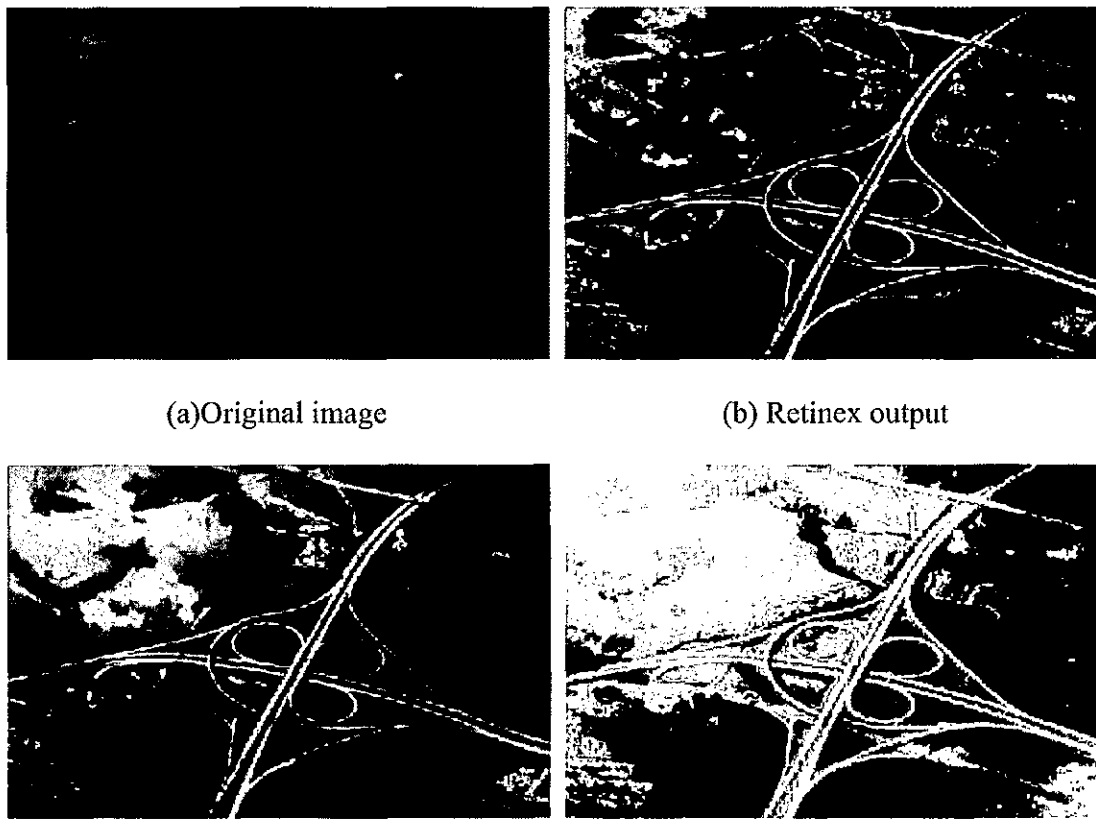
From figure 4.1, it is noticed that the Multi-Scale Retinex provides great dynamic range compression, increased sharpness and color, and accurate scene rendition, especially in foggy conditions. Several other techniques like contrast stretching, histogram equalization, etc. were tried but could not give good results. Figure 4.2 and Figure 4.3 compare the results of Retinex, contrast stretching and histogram equalization on foggy images.



(a)Original image (b) Retinex output

(c)Contrast stretching (d) Histogram equalization

Figure 4.2: Comparison of different enhancement algorithms-I.



(a)Original image (b) Retinex output

(c)Contrast stretching (d) Histogram equalization

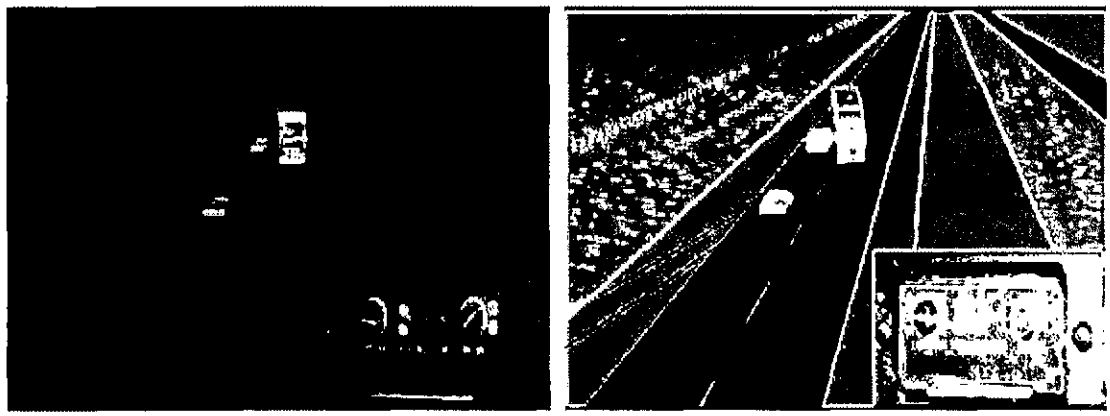
Figure 4.3: Comparison of different enhancement algorithms-II.

From Figure 4.2 and figure 4.3, it is noticed that the results of Retinex are much better than the results produced by contrast stretching and histogram equalization.

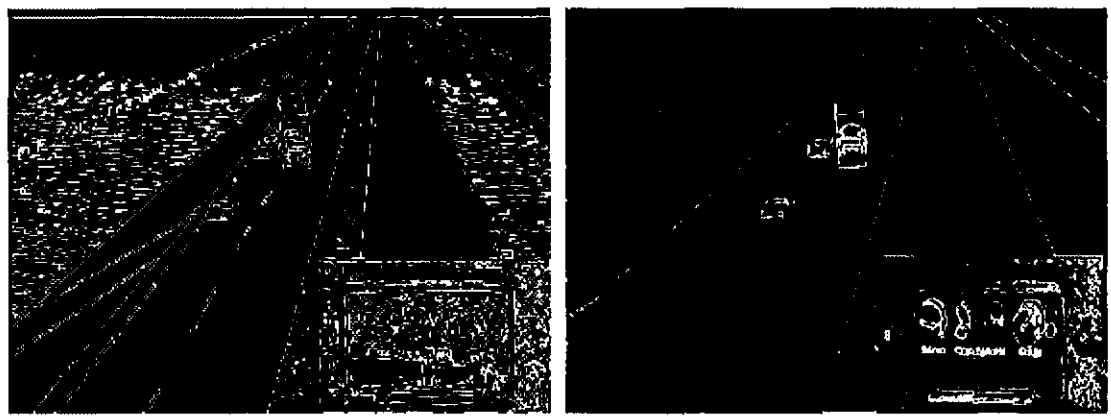
Having enhanced the image, the next step was to obtain an edge image using an edge detection operator. Several edge detection algorithms were tried, but the SUSAN edge detection algorithm was used because of its superior performance on smoother images. For a real time application, like the one considered in this research, computation time is a major constraint, so the SUSAN edge detection operator was used because of its performance speed over the state-of-the-art edge detection operators. Figure 4.4 compares the results of SUSAN, Canny, Sobel, Prewitt and Roberts's edge detection operators for

clear images. Figure 4.4(b) shows the result of the SUSAN edge detection algorithm with the threshold value set to 20. The default threshold for clear images is set to 20. The result of the canny operator, in Figure 4.4(c), has the upper threshold value as 10 and the lower threshold as 5.

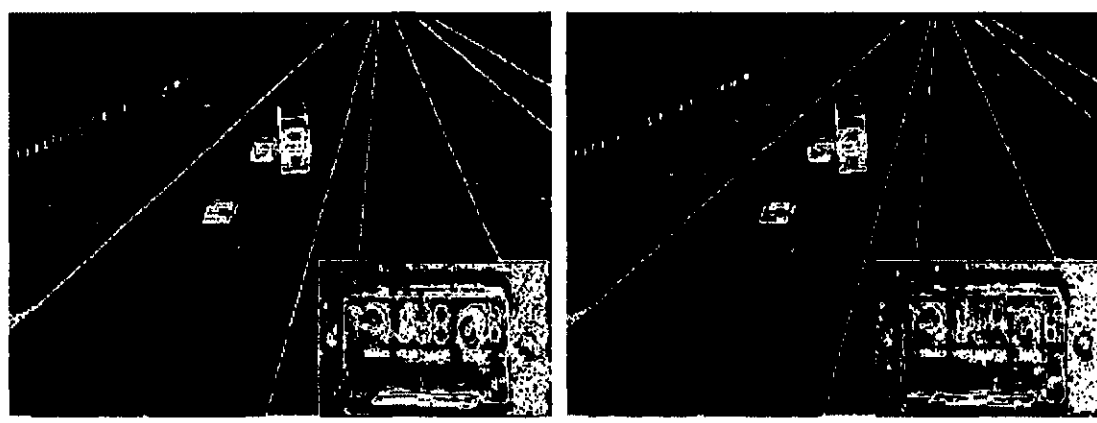
Figure 4.4 shows that the results from different edge detection operators for a clear image that appears to be the same. They do not lose any edge information, but a problem arises with foggy images. Typically, foggy images are smoother, and they have low contrast. Thus, most of the edge detection algorithms lose a lot of edge information for foggy images. Figure 4.5 shows the result of different edge detection operators for foggy images. From figure 4.5, it is observed that the results of the SUSAN edge detection operator were far better than the results produced by other techniques. The threshold value of the SUSAN edge detection algorithm was set to 5 for foggy images. Figure 4.5(c) shows the output of the canny edge detection operator. The lower threshold was set to 5, and the upper threshold was set to 10.



(a) Original image (b) SUSAN output

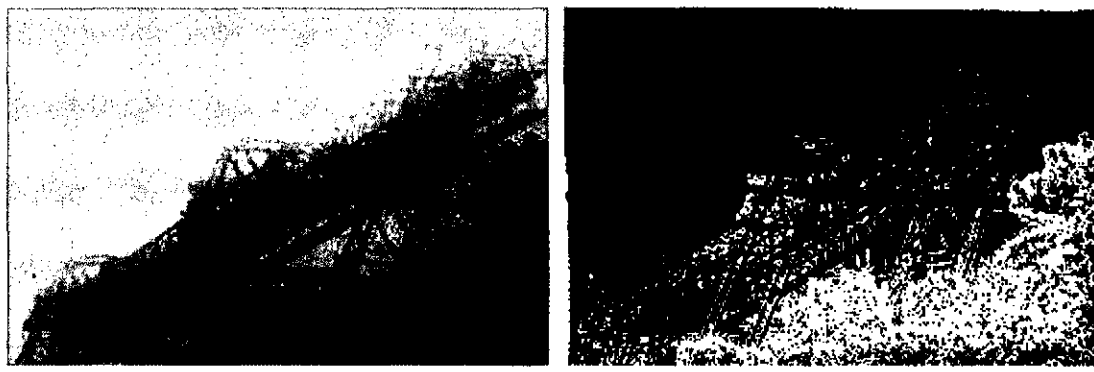


(c) Canny (d) Sobel

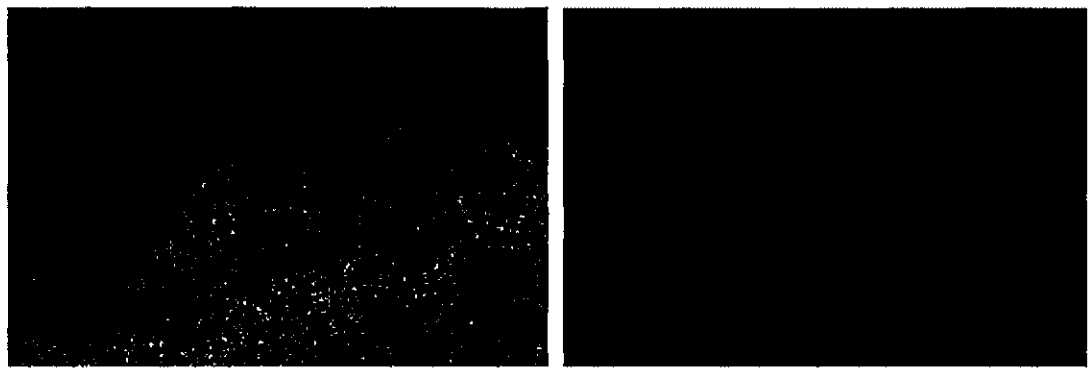


(e) Prewitt (f) Roberts

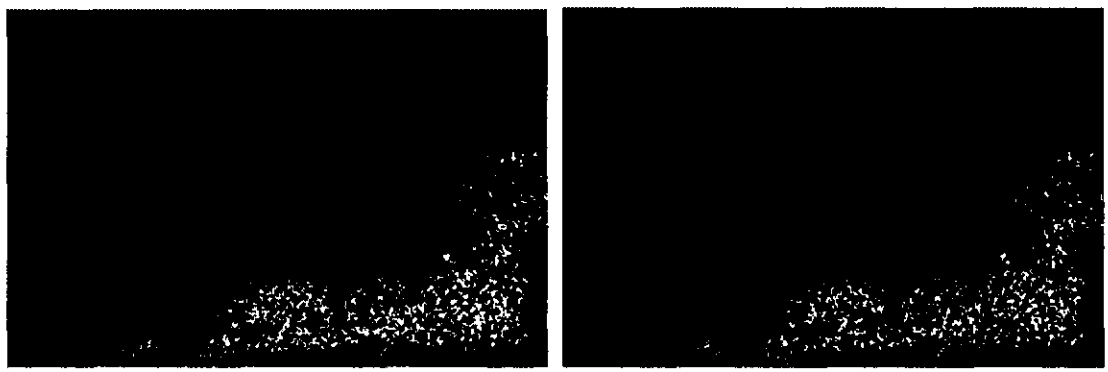
Figure 4.4: Comparison of different edge detection algorithms on clear imagery.



(a) Original image (b) SUSAN output



(c) Canny (d) Sobel



(e) Prewitt (f) Roberts

Figure 4.5: Comparison of different edge detection algorithms on foggy images.

Because the SUSAN edge detection operator produced good results in foggy and clear conditions, it was used as a part of the research to develop the proposed algorithm. Also, the SUSAN edge detection operator was better than most of the state-of-the-art

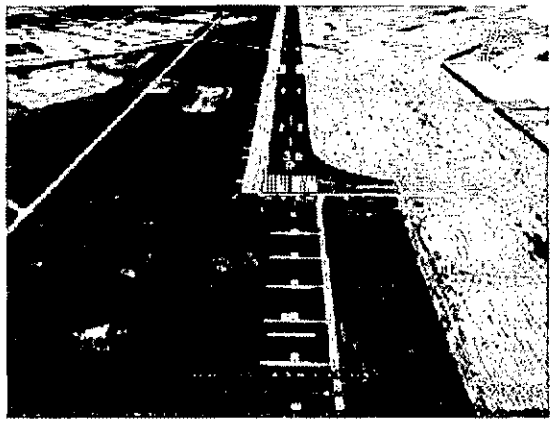
edge detection algorithms. Edge-detection techniques like Sobel and Prewitt use a small convolution kernel for estimating the first derivative of an image to extract the edges. These methods do not provide a high degree of edge localization and smoothing. Widely popular edge detection techniques like Canny find edges by minimizing the error rate, marking edges as closely as possible to the actual edges to maximize localization and marking edges only once when a single edge exists for minimal response. Canny uses the calculus of variation to satisfy the criterion and derive the optimal function which is a close approximation of the first derivative of the Gaussian function. Non-maximum suppression is performed followed by removal of edges using thresholding. Thresholding is applied with hysteresis. While performing the Gaussian convolution can be fast, the hysteresis stage can slow down the computation. Even though the results from the Canny are stable, it does not provide good edge connectivity and the corners are rounded. The scale of the Gaussian determines the amount of noise reduction. With the increase in the size of the Gaussian, the smoothing effect increases resulting in poor edge localization.

The fact that the SUSAN edge detection algorithm does not use any image derivatives explains its performance in the presence of noise. Because the SUSAN edge detection technique uses the USAN area, it provides better localization, good connectivity and no false edges. The computation speed of the SUSAN edge detection is about 10 times faster than Sobel and Canny, which is very important in applications like the one described here. Because of the integrating effect and its non-linear response, the SUSAN operator shows good tolerance to noise. The performance of the SUSAN algorithm is superior to that provided by Canny and does not depend on selecting an optimal threshold value for each of the images.

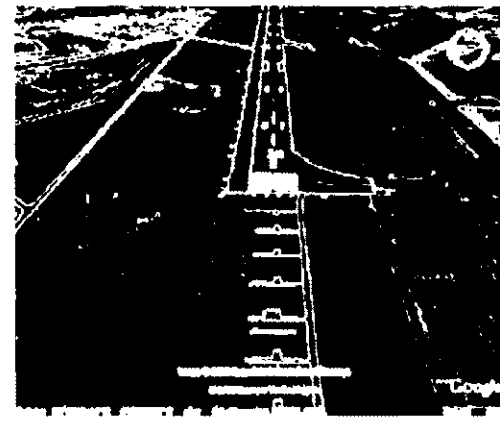
4.4 Results

In the previous section, the results of image enhancement and edge detection operators were observed. Now, the results from previous sections are used to show the results of the algorithm.

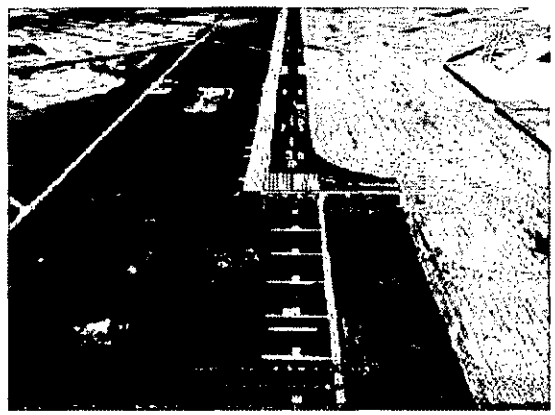
As per the algorithm, the imagery on a clear day does not need any enhancement to detect hazards on the runway. Figure 4.6 shows the sequence of images to detect hazards on the runway. Figure 4.6(a) is the image of a runway taken from Google Earth. Figure 4.6(b) is the edge image of figure 4.6(a) using SUSAN edge detection algorithm. The threshold was set to 20 when using the SUSAN edge detection operator. In a real time scenario, figure 4.6(a) and figure 4.6(b) will be stored in the database. Figure 4.6(c) shows the image of the same runway as in figure 4.6(a) but with a truck on the runway. Figure 4.6(d) is the corresponding edge image of figure 4.6(c) using the SUSAN edge detection operator. Figure 4.6(c) and figure 4.6(d) are images obtained on a given day. These images are compared with the images in the database to detect hazards. Subtracting the edge images would result in the difference image as in figure 4.6(e). If figure 4.6(e) contains a lot of information, then it would mean that the runway is hazardous.



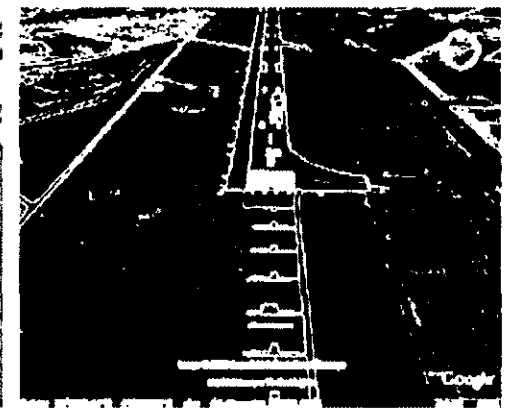
(a) Original image



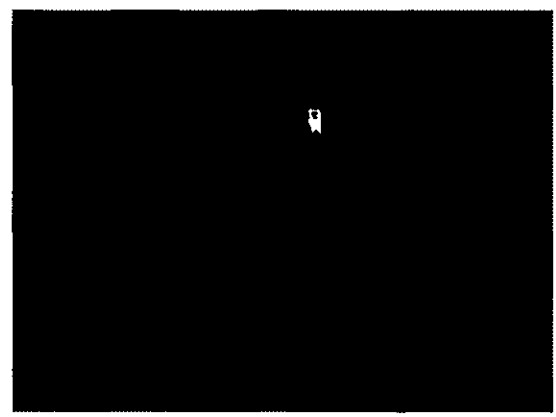
(b) SUSAN output



(c) Image with object on runway



(d) Corresponding SUSAN output

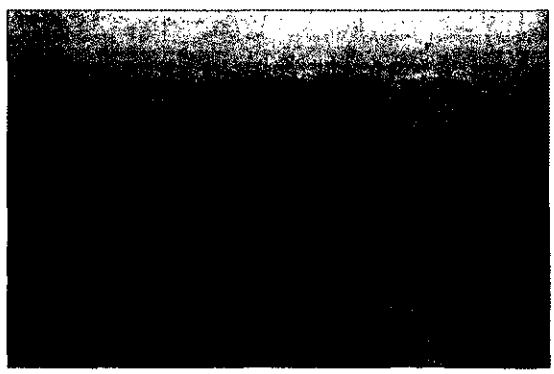


(e) Difference of edge images

Figure 4.6: Sequence of images to detect the hazards using the proposed algorithm in clear visibility conditions.

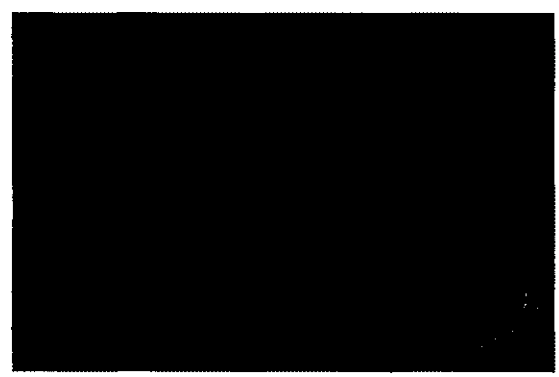
(a) Original image

(b) SUSAN output



(c) Image with object

(d) Corresponding SUSAN output



(e) Difference of edge images

Figure 4.7: Sequence of images to detect the hazards using the proposed algorithm in poor visibility conditions.

Figure 4.7(a) shows a foggy image and figure 4.7(b) is its corresponding edge image. The edge image is taken after the image is enhanced using Multi-Scale Retinex. Because of the unavailability of runway images, the algorithm was tried on other foggy images. As per the algorithm, Figures 4.7(a) and 4.7(b) are stored in the database. Figure 4.7(c) shows the similar image with a man standing (just like a hazard on the runway) and Figure 4.7(d) is its corresponding edge image. The difference image is shown by figure 4.7(e). The difference image, figure 4.7(e), shows the outline of a man along with some other detail. Thus, this edge image can be used to check for the hazard.

4.5 Summary

In this chapter, the performance analysis of the proposed algorithm has been presented. Based on the experimental results, it is observed that the proposed algorithm performs well in variable lighting and visibility conditions. The results of the Multi-Scale Retinex and SUSAN edge detection operator provide a good platform for the proposed algorithm. Thus, the proposed algorithm can be applied to detect hazards on the runway for different visibility conditions, especially foggy conditions.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

5.1 Impact of Proposed Algorithm

With increasing concern over the safety standards associated with aviation, the primary objective of the FAA is to reduce the severity and rate of runway incursions. With the ability to provide an automated alert by detecting hazards on the runway in poor and clear visibility conditions, the proposed algorithm has the ability to address the FAA goal to reduce the number of runway incursions. The proposed algorithm provides a direct warning to the pilot, and thus can reduce the number of runway incursions. Also, it provides pilots with an enhanced vision system to see the runway in poor visibility conditions. The proposed design would achieve the FAA goals as described in the FAA's Runway Safety Blue Print [44].

5.2 FAA's Goals for Runway Safety

The FAA identified eight goals for runway safety and most can be achieved using the proposed design. The ones that cannot be achieved using the proposed design are

- To develop and distribute runway safety education and training materials to controllers, pilots and all other airport users;
- To improve runway safety data collection, analysis, and dissemination;
- To increase surface safety awareness throughout aviation community.

These goals can be achieved only by the FAA's ability to educate pilots and airport authorities on the existing technologies and their usage. The rest of the goals can be achieved as follows.

5.2.1 Asses and modify procedures to enhance runway safety

The runway safety procedures can be altered using the proposed algorithm by eliminating the ground staff factor. However, this algorithm can be employed with the existing technologies to further improve runway safety.

5.2.2 Identify and implement enhancements to improve surface communications

The proposed algorithm can be used to directly alert the pilot of hazards on the runway without any intervention from the ATC operator. This reduces the burden on the ATC operator, especially at airports with heavy air-traffic. In busy airports, the ATC finds it difficult to identify the location of each aircraft on the ground; hence, it cannot identify the aircrafts or potential hazards on the runways.

5.2.3 Increase situational awareness

The proposed algorithm is intended to detect hazards on the runway in poor visibility conditions. However, the algorithm can also be used in clear conditions. The FAA safety reports suggest that most of the runway incursions occur because of miscommunication between the pilot and ground staff. This problem can be eliminated by providing an automatic alarm system to the pilot that eliminates the possibilities of runway incursions because of miscommunication. Also, the algorithm increases the situational awareness of the pilot by providing a display for the pilot to see the enhanced imagery.

5.2.4 Support and deploy new technologies that reduce the potential for collision

Because the proposed algorithm intends to mount the cameras on every aircraft, this does not affect the existing technologies being used at the airports. Some of the busy airports in the United States are equipped with systems like ASDE, AMASS, etc. The proposed design can work in tandem with the existing systems at busy airports to provide better safety at airports. However, some of the low traffic airports are not deployed with these sophisticated systems, so the proposed algorithm can be used for such airports.

5.2.5 Implement site-specific runway safety solutions

The underlying algorithm for hazard detection requires baseline information about an airport in order to detect hazards. However, the system itself resides on the aircraft and not at the airport, and the cost would be computed as a part of the aircraft. This means that this design is just as effective for smaller airports as it is for larger airports. Also, this technique can be used for terrain analysis to land the aircraft on a smoother terrain in case of emergency landings.

5.3 Conclusions

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

The significant problem when landing the aircraft, especially in poor visibility conditions, can occur due to the presence of objects on the runway. If an object is detected early enough in the landing approach, the pilot would 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 concept of smoothing coefficient provides a good platform to differentiate clear images from foggy images. Determining if the imagery is foggy is an important part of the automation process. The image enhancement process using Multi-Scale Retinex provides illumination independence (i.e. the output of the enhancement phase is almost independent of the type, or level, of illumination under which the image was acquired). This is especially critical for automatic classification and detection algorithms that rely on comparing imagery of the same scene at different times. The SUSAN edge detection operator is used to provide the edge information of the acquired imagery. Edge Detection is used because of its ability to reduce the amount of data by filtering out the unimportant information, while preserving the structural properties in an image. The SUSAN edge detection operator is used to obtain the structural information of the runway imagery to detect any objects on the runway. The SUSAN edge detection operator was used because of its performance speed over other edge detection techniques, which is important in real time scenarios like the one discussed here.

5.4 Future Work

The proposed algorithm provided good results for hazard detection on the runways in poor visibility conditions. However, the work presented is the initial stage of NASA's project for aviation safety, so more features could be added to the proposed algorithm to make it more robust with regards to varying illumination and weather conditions. Also, the tests could not be performed on the runway images because images could not be acquired in poor visibility conditions. Using this approach in a real time

scenario, using real runway imagery could result in a few changes in the algorithm. Also, this algorithm can be further improved to be used for terrain analysis for emergency landing of the aircraft.

REFERENCES

[1] Curt L. Lewis, and John Darbo, "Flight Safety Information journal: Runway awareness and Advisory System," November 30 2006, http://www.fsinfo.org/FSI-journals/ATT830843%20RAAS.pdf

[2] FAA Air Traffic Organization, "FAA Runway Safety Report," September 2007.

[3] FAA Office of Runway Safety, "FAA Runway Safety Report-Runway Incursion Trends at Towered Airports in United States," CY1998-CY2001, June 2002.

[4] FAA Office of Runway Safety, "FAA Runway Safety Report-Runway Incursion Trends at Towered Airports in United States," FY1999-FY2002, July 2003.

[5] FAA Office of Runway Safety, "FAA Runway Safety Report-Runway Incursion Trends and Initiatives at Towered Airports in United States," FY2000-FY2003, August 2004.

[6] FAA Air Traffic Organization, "FAA Runway Safety Report," August 2005.

[7] US Government Accountability Office, "AVIATION RUNWAY AND RAMP SAFETY: Sustained efforts to address leadership, technology, and other challenges needed to reduce accidents and incidents," GAO-08-29, November 2007.

[8] Robert Baron, "Runway Incursions-Where are We?," April 2002, http://www.airlinesafety.com/editorials/RunwayIncursions.htm

[9] Accident description, The Aviation Safety Network Website, http://aviation-safety.net/database/record.php?id=19770327-1

[10] Accident description, The Aviation Safety Network Website, http://aviation-safety.net/database/record.php?id=20011008-0□=fr

[11] C. Brandon, T. Blasier, F. Bouchard, V. Capezzuto, F. Coyne, and C. Franck, "Low-cost ASDE Evaluation Report: Raytheon Marine (Phase-I) Radar at MKE," DOT/FAA/ND410-96/1, July 1996.

[12] Dan Dellmyer, "Airport Movement Area Safety System (AMASS) Operational Test," DOT/FAA/CT-TN00/27, December 2000.

[13] Anthony Stevens, "Evaluation of the Airport Target identification System (ATIDS) Beacon Multilateration System,"DOT/FAA/CT-TN98/4, May 2000.

[14] Figueroa. J. and Swanson K.J., "An Operational Evaluation of the Final Approach Runway Occupancy Signal at Long Beach Airport," in Integrated Communications, Navigation and Surveillance Conference, 2007 ICNS '07, 2007.

[15] NASA Facts online, "ARIES: NASA's 'Flying Lab' Takes Wing," December 1998, http://oea.larc.nasa.gov/PAIS/FS-1998-11-41-LaRC.html

[16] C. L. Tiana, J. R. Kerr, and S. D. Harrah, "Multispectral uncooled infrared enhanced-vision system for flight test," in Enhanced and Synthetic Vision 2001, J. G. Verly ed., pp. 231–236, Proc. SPIE 4363, 2001.

[17] G. A. Woodell, D. J. Jobson, Z. Rahman, G. D. Hines, "Advanced Image Processing of Aerial Imagery," in Visual Information Processing XIV, Proc. SPIE 6246, 2006.

[18] Wikipedia Website, http://en.wikipedia.org/wiki/Thermography, Accessed April 2007.

[19] Linda G. Shapiro and George C. Stockman, "Computer Vision," pp. 279-325, New Jersey, Prentice-Hall, ISBN 0-13—3-796-3, 2001.

[20] Dzung L. Pham, Chenyang Xu, and Jerry L. Prince, "Current Methods in Medical Image Segmentation," in Annual Review of Biomedical Engineering, volume 2, pp 315-337, 2000.

[21] Cheng-Ru Lin and Ming-Syan Chen, "Combining partitional and hierarchical algorithms for robust and efficient data clustering with cohesion self-merging," in IEEE Transactions on Knowledge and Data Engineering, Volume 17, Issue 2, pp. 145 – 159, February 2005.

[22] R.C. Gonzalez and R.E. Woods, "Digital Image Processing," Addison Wesley, 2nd ed., pp.97-130, 2002.

[23] Zia-ur Rahman, "Image Enhancement by Histogram Modification," Class notes for ECE 783, Digital Image Processing, Old Dominion University, Spring 2007.

[24] Wikipedia Website, http://en.wikipedia.org/w/index.php?title=Roberts_Cross&oldid=193018212.

[25] E. Davies, "Machine Vision: Theory, Algorithms and Practicalities," Academic Press, pp 101 - 110, 1990.

[26] Wikipedia Website, http://en.wikipedia.org/w/index.php?title=Sobel_operator&oldid=193018020

[27] J. Canny, "A computational approach to edge detection," in IEEE Transactions on Pattern Analysis and Machine Intelligence 8(6), pp. 679–698, 1986.

[28] Girish Singh Rajput and Zia-Ur Rahman, "Hazard detection on runways using image processing techniques," in SPIE Symposium on Defense and Security, Enhanced and Synthetic Vision 2008, Proc. 6957, March 2008.

[29] Z. Rahman, D. J. Jobson, G. A. Woodell, and G. D. Hines, "Automated, on-board terrain analysis for precision landings," in Visual Information Processing XV, Z. Rahman, S. E. Reichenbach, and M. A. Neifeld, eds., Proc. SPIE 6246, 2006.

[30] Xiaoyin Xu and Miller E.L., "Adaptive difference of Gaussians to improve subsurface imagery," in Geoscience and Remote Sensing Symposium, 2002, IGARSS '02, 2002 IEEE International, pp: 3441- 3443 vol.6, June 2002.

[31] D. J. Jobson, Z. Rahman, and G. A. Woodell, "Properties and performance of a center/surround retinex," in IEEE Trans. on Image Processing 6, pp. 451–462, March 1997.

[32] 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," in IEEE Transactions on Image Processing: Special Issue on Color Processing 6, pp. 965–976, July 1997.

[33] Z. Rahman, D. J. Jobson, and G. A. Woodell, "Retinex processing for automatic image enhancement," Journal of Electronic Imaging 13(1), pp. 100–110, 2004.

[34] Z. Rahman, D. J. Jobson, and G. A. Woodell, "Resiliency of the multiscale retinex image enhancement algorithm," in Proceedings of the IS&T Sixth color Imaging Conference: Color Science, Systems, and Applications, pp. 129–134, IS&T, 1998.

[35] 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 Anniversary Conference, pp. 426–431, IS&T, 1997.

[36] D. J. Jobson, Z. Rahman, and G. A. Woodell, "Feature visibility limit in the nonlinear enhancement of turbid images," in Visual Information Processing XII, Z. Rahman, R. A. Schowengerdt, and S. E. Reichenbach, eds., Proc. SPIE 5108, 2003.

[37] D. J. Jobson, Z. Rahman, G. A. Woodell, and G. D. Hines, "The automatic assessment and reduction of noise using edge pattern analysis in nonlinear image enhancement," in Visual Information Processing XIII, Z. Rahman, R. A. Schowengerdt, and S. E. Reichenbach, eds., Proc. SPIE 5438, 2004.

[38] G. A. Woodell, D. J. Jobson, Z. Rahman, and G. D. Hines, "Enhancement of imagery in poor visibility conditions," in Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense IV, E. Carapezza, ed., Proc. SPIE 5778, 2005.

[39] D. J. Jobson, Z. Rahman, G. A. Woodell, and G. D. Hines, "A comparison of visual statistics for the image enhancement of foresite aerial images with those of major image classes," in Visual Information Processing XV, Z. Rahman, S. E. Reichenbach, and M. A. Neifeld, eds., Proc. SPIE 6246, 2006.

[40] S.M. Smith and J.M. Brady, "SUSAN-A New Approach to Low Level Image Processing," Technical Report TR95SMSlb, Defense Research Agency, Chobham Lane, Chertsey, Surrey, UK, 1995.

[41] R. Zhang, G. Zhao, and L. Su, "A new edge detection method in image processing," in Proceedings of the IEEE International Symposium on Communications and Information Technology, 2005 (ISCIT 2005), pp. 445–448, 2004.

[42] M. M. Perez and T. J. Dennis, "An adaptive implementation of the SUSAN method for image edge and feature detection," in Proceedings of the IEEE International Conference on Image Processing, p. 394, IEEE, 1997.

[43] W. Muyun and H. Mingyi, "Image feature detection and matching based on SUSAN method," in Proceedings of the First International Conference on Innovative Computing: Information and Control, 1, pp. 322–325, 2006.

[44] Federal Aviation Administration, "Runway Safety Blueprint 2002-04," July 2002.

GIRISH SINGH RAJPUT
Department of Electrical and Computer Engineering
Old Dominion University
757-275-5624
grajp001@odu.edu

Education:

Master of Science in **Electrical Engineering**

Old Dominion University, Norfolk, VA. (August 2008)

GPA: **4.0/4.0**

Bachelor of Technology in **Electronics and Communication Engineering**

Jawaharlal Nehru Technological University, India. (April 2006)

GPA: **3.6/4.0**

Research Publications:

Conference Proceedings

Girish Singh Rajput and Zia-ur Rahman, "Hazard detection on runways using image processing techniques," Enhanced and Synthetic Vision 2008, SPIE Symposium on SPIE Defense and Security Symposium, March 2008.

Published Abstract

Girish Singh Rajput and Zia-ur Rahman, "SUSAN Edge Detection based Hazard Detection on Runways," ODU-NSU-EVMS-VTC Research Exposition Day: Research Expo 2008 - 400 Years of Discovery, Norfolk VA, p. 108, April 2008.

Presentations:

Presented a poster titled, "SUSAN Edge Detection based Hazard Detection on Runways," at the ODU-NSU-EVMS-VTC Research Exposition Day: Research Expo 2008 - 400 Years of Discovery, Norfolk VA, p. 108, April 2008.

Awards:

Secured the First prize in the poster presentation at ODU-NSU-EVMS-VTC Research Exposition Day: Research Expo 2008, April 09, 2008.