Edge Detection Technique Using Fuzzy Logic

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering
in
Computer Science & Engineering

Thapar University, Patiala

By:

Vandana Mittal
(80632027)

Under the supervision of:
Mrs. Shalini Batra
Lecturer, CSED

MAY 2008

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR UNIVERSITY
PATIALA – 147004

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, **"Edge Detection Technique Using Fuzzy Logic",** in partial fulfillment of the requirements for the award of degree of Master of Engineering in Computer Science & Engineering submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of *Mrs. Shalini Batra*.

The matter presented in this thesis has not been submitted for the award of any other degree of this or any other university.

Vandana Mittal

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

Mrs. Shalini Batra
Computer Science and Engineering Department
Thapar University
Patiala-147004

Countersigned by

(SEEMA BAWA)
Professor & Head
Computer Science & Engineering Department
Thapar University
Patiala

(R.K.SHARMA)
Dean(Academic Affairs)
Thapar University
Patiala

ACKNOWLEDGEMENT

First and foremost, I would like to express my sincere gratitude to my advisor **Shalini Batra**, Lecturer, Computer Science & Engineering Department for her immense help, guidance, stimulating suggestions and encouragement all the time. She always provide a motivating and enthusiastic atmosphere to work with, it was a great pleasure to do this thesis under her supervision.

I would like to thank **Dr. Seema Bawa**, Head, Computer Science and Engineering Department, Thapar University, Patiala for providing necessary facilities in the Department, to work. The most valuable thing I acquired from the two year studying Thapar University is to protect, survive and endure myself in this world with increased confidence and additional faith in my abilities to achieve. I am also thankful to the authors whose works I have consulted and quoted in this work.

I am very thankful to my respectable family for their fortitude. Nothing would have ever been possible without my parent's unconditional love and encouragement. Last but not the least I would like to thank God for not letting me down at the time of crisis and showing me the silver lining in the dark clouds.

Vandana Mittal

(80632027)

ABSTRACT

Digital image processing is a subset of the electronic domain wherein the image is converted to an array of small integers, called pixels, representing a physical quantity such as scene radiance, stored in a digital memory, and processed by computer or other digital hardware. Interest in digital image processing methods stems from two principals applications areas: improvement of pictorial information for human interpretation; and processing of image data for storage, transmission, and representation for autonomous machine perception. Edges characterize boundaries and edge detection is one of the most difficult tasks in image processing hence it is a problem of fundamental importance in image processing. Edges in images are areas with strong intensity contrasts and a jump in intensity from one pixel to the next can create major variation in the picture quality. Edge detection of an image significantly reduces the amount of data and filters out useless information, while preserving the important structural properties in an image.

Fuzzy logic represents a good mathematical framework to deal with uncertainty of information. Fuzzy image processing is the collection of all approaches that understand, represent and process the images, their segments and features as fuzzy sets. The representation and processing depend on the selected fuzzy technique and on the problem to be solved. This research problem deals with Fuzzy inference system (FIS) which represents greater robustness to contrast and lighting variations. Further tuning of the weights associated to the fuzzy inference rules is still necessary to reduce even more inclusion in the output image of pixels not belonging to edges.

TABLE OF CONTENTS

LIST OF FIGURES AND TABLES

CHAPTER 1
INTRODUCTION

A digital image is a representation of a two-dimensional image as a finite set of digital values. In image processing, the digitization process includes sampling and quantization of continuous data. The sampling process samples the intensity of the continuous-tone image, such as a monochrome, color or multi-spectrum image, at specific locations on a discrete grid. The grid defines the sampling resolution. The quantization process converts the continuous or analog values of intensity brightness into discrete data, which corresponds to the digital brightness value of each sample, ranging from black, through the grays, to white. A digitized sample is referred to as a picture element, or pixel. The digital image contains a fixed number of rows and columns of pixels. Pixels are like little tiles holding quantized values that represent the brightness at the points of the image. Pixels are parameterized by position, intensity and time. Typically, the pixels are stored in computer memory as a raster image or raster map, a two-dimensional array of small integers. Image is stored in numerical form which can be manipulated by a computer. A numerical image is divided into a matrix of pixels (picture elements).

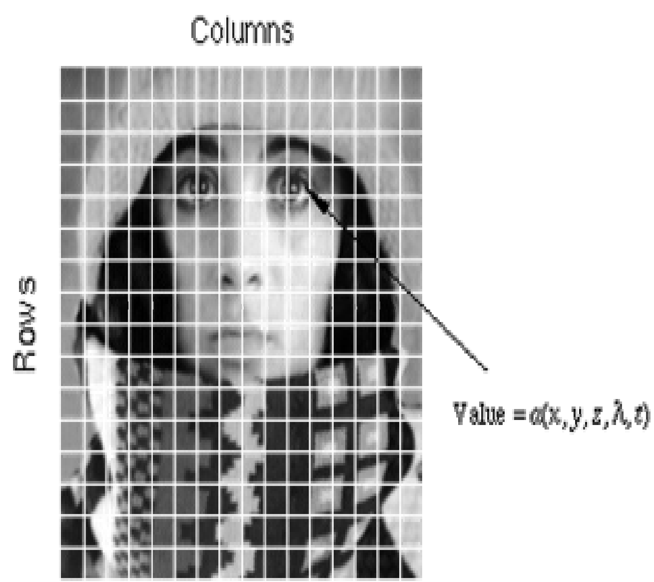

Figure 1.1: Digitization of a continuous image.

Digital image processing allows one to enhance image features of interest while attenuating detail irrelevant to a given application, and then extract useful information about the scene from the enhanced image. Images are produced by a variety of physical devices, including still and video cameras, x-ray devices, electron microscopes, radar, and ultrasound, and used for a variety of purposes, including entertainment, medical, business (e.g. documents), industrial, military, civil (e.g. traffic), security, and scientific. The goal in each case is for an observer, human or machine, to extract useful information about the scene being imaged.

1.1 Brief review of Digital Image Processing

We are in the midst of a visually enchanting world, which manifests itself with a variety of forms and shapes, colors and textures, motion and tranquility. The human perception has the capability to acquire, integrate, and interpret all this abundant visual information around us. It is challenging to impart such capabilities to a machine in order to interpret the visual information embedded in still images, graphics, and video or moving images in our sensory world. It is thus important to understand the techniques of storage, processing, transmission, recognition, and finally interpretation of such visual scenes. A two-dimensional image that is recorded by sensors is the mapping of the three-dimensional visual world. The captured two dimensional signals are sampled and quantized to yield digital images.

1.1.1 Image processing operations

Image processing operations can be roughly divided into three major categories:

a) Image Restoration
b) Image Enhancement
c) Image Compression
d) Image Segmentation

Image Restoration: Restoration takes a corrupted image and attempts to recreate a clean image. As many sensors are subject to noise, they results in corrupted images that don’t reflect the real world scene accurately and old photograph and film archives often show considerable damage.

Thus image restoration is important for two main applications:

a) Removing sensor noise,
b) Restoring old, archived film and images.

It is clearly explained in the figure 1.2 and figure 1.3.

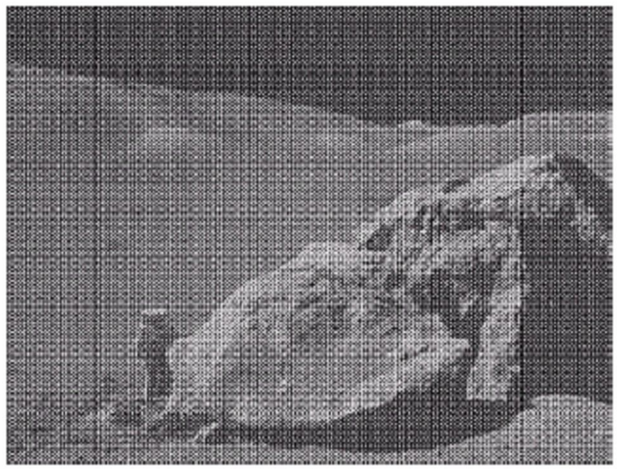

Figure 1.2: Original image

Figure 1.3: Image after restoration

Image Enhancement: Image Enhancement alters an image to makes its meaning clearer to human observers. It is often used to increase the contrast in images that are substantially dark or light. Enhancement algorithms often play attention to humans' sensitivity to contrast.

Figure 1.4: Original image

Figure 1.5: Image after enhancement

Image Compression: Image compression is the process that helps to represent image data with as few bits as possible by exploiting redundancies in the data while maintaining an appropriate level of quality for the user. The human eye has less spatial sensitivity to color than for luminance information. Large amounts of data are used to represent an image, so image has to be compressed when transferring from one place to another. Transform Coding has been a very popular technique for digital image compression. It is used in the International Standard for Still Image Compression - JPEG (Joint Photographic Experts Group).

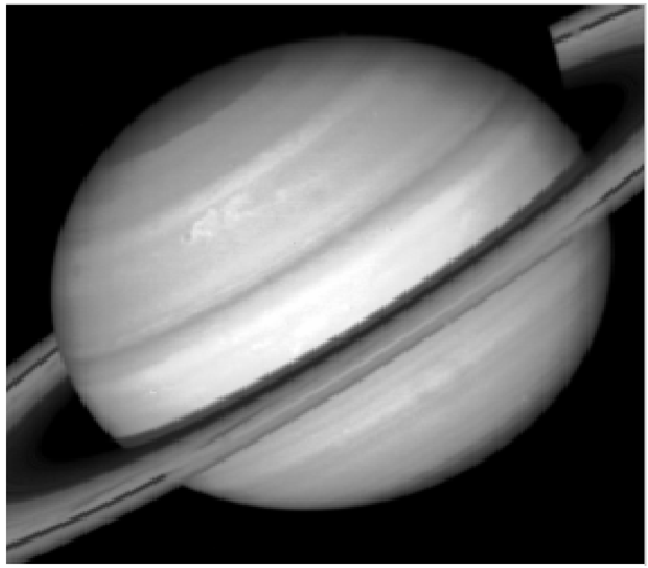

Figure 1.6: Original image

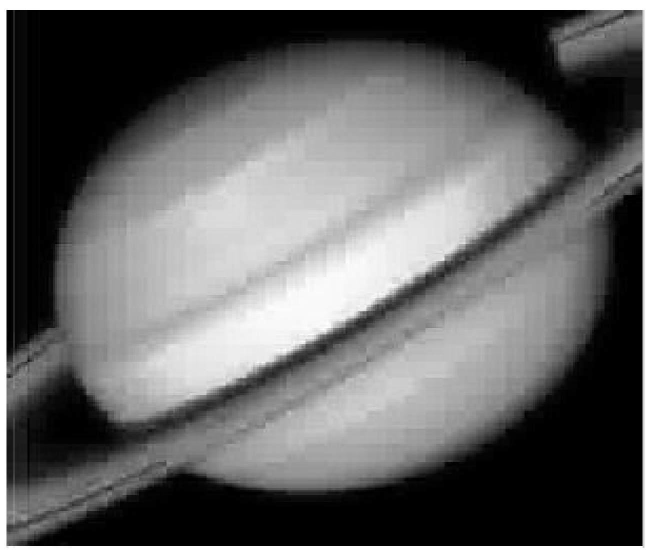

Figure 1.7: Image after compression

Image Segmentation: Segmentation is the process that subdivides an image into a number of uniformly homogeneous regions. Each homogeneous region is a constituent part or object in the entire scene. The objects on the land part of the scene need to be appropriately segmented and subsequently classified. Partitioning of an image is based on abrupt changes in gray level. If edges of the image can be extracted and linked, the region is described by the edge contour that contains it. The principal areas of interest within this category are the detection of edges of a digital image. An edge corresponds to local intensity discontinuities of an image. In the real world, the discontinuities reflect a rapid intensity change, such as the boundary between different regions, shadow boundaries, and abrupt changes in surface orientation and material properties. For example, edges represent the outline of a shape, the difference between the colors and pattern or texture. Therefore, edges can be used for boundary estimation and segmentation in scene understanding. They can also be used to find corresponding points in multiple images of the same scene. For instance, the fingerprint, human facial appearance and the body shape of an object are defined by edges in images. In a broad sense the term edge detection refers to the detection and localization of intensity discontinuities of these image properties. In a more restrictive sense, it only refers to localizations of significant change of intensity. Points of these localizations are called edges or edge elements. Edges are piecewise segmentation. They are both useful in computation of geometrical features such as shape or orientation. Edge detection is grounded on the assumption that physical 3-dimensional shapes in the scene, such as object boundaries and shadow boundaries, are clues for the characterization of the scene.

Figure 1.8: Original image

Figure 1.9: Image after segmentation

1.2 Brief review of Edge Detection

Unlike the real world, images do not have edges. An edge is sharp change in intensity of an image.

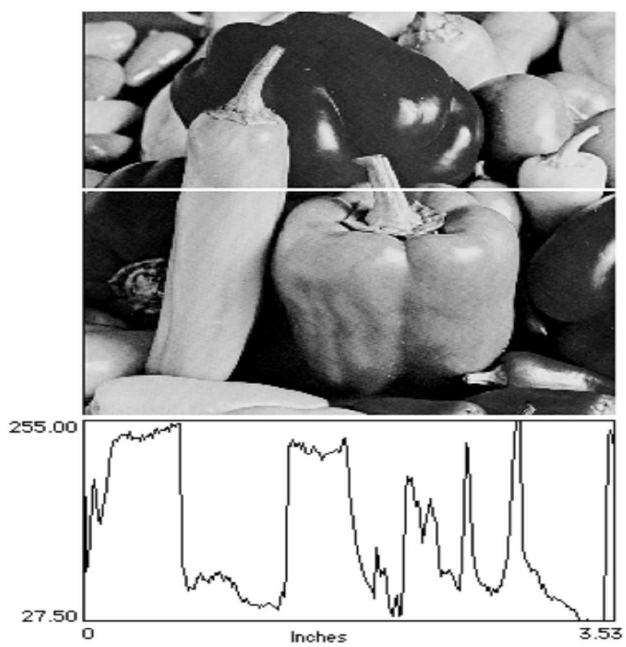

Figure 1.10 Sharp changes in intensity of image

But, since the overall goal is to locate edges in the real world via an image, the term edge detection is commonly used. An edge is not a physical entity, just like a shadow. It is where the picture ends and the wall starts, where the vertical and the horizontal surfaces of an object meet. If there were sensor with infinitely small footprints and zero-width point spread functions, an edge would be recorded between pixels within in an image. In reality, what appears to be an edge from the distance may even contain other edges when looked close-up. The edge between a forest and a road in an aerial photo may not look like an edge any more in a image taken on the ground. In the ground image, edges may be found around each individual tree. If looked a few inches away from a tree, edges may be found within the texture on the bark of the tree. Edges are scale-dependent and an edge may contain other edges, but at a certain scale, an edge still has no width. If the edges in an image are identified accurately, all the objects are located and their basic properties such as area, perimeter and shape can be measured. Therefore edges are used for boundary estimation and segmentation in the scene. Since computer vision involves the identification and classification of objects in an image, edge detection is an essential tool.

1.2.1 Types of Edges

All edges are locally directional. Therefore, the goal in edge detection is to find out what occurred perpendicular to an edge. The following is a list of commonly found edges.

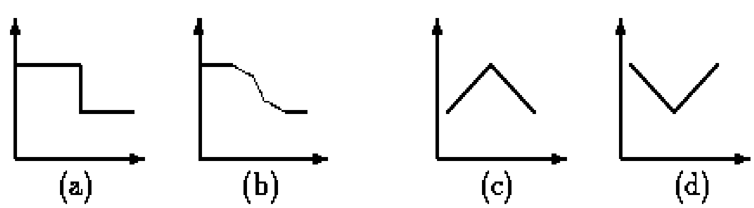

Figure 1.10: Types of Edges (a) Sharp step (b) Gradual step (c) Roof (d) Trough

A Sharp Step, as shown in Figure 1.10(a), is an idealization of an edge. Since an image is always band limited, this type of graph cannot ever occur. A Gradual Step, as shown in Figure 1.6(b), is very similar to a Sharp Step, but it has been smoothed out. The change in intensity is not as quick or sharp. A Roof, as show in Figure 1.6(c), is different than the first two edges. The derivative of this edge is discontinuous. A Roof can have a variety of sharpness, widths, and spatial extents. The Trough, also shown in Figure 1.6(d), is the inverse of a Roof.

Edge detection is very useful in a number of contexts. Edges characterize object boundaries and are, therefore, useful for segmentation, registration, and identification of objects in scenes [1]. A straightforward example of edge detection is illustrated in Figure 1.11 Original picture has a uniform grey background. The edge enhanced version of the same image has dark lines outlining the three objects Figure 1.12. Note that there is no way to tell which parts of the image are background and which are object, only the boundaries between the regions are identified.

Figure 1.11: Synthetic image with a grey background

Figure 1.12: Edge enhanced image showing only the outlines of the objects.

The goal of the edge detection process in a digital image is to determine the frontiers of all represented objects, based on automatic processing of the color or gray level information in each present pixel. Edge detection has many applications in image processing and computer vision, and is an indispensable technique in both biological and robot vision [3]. The main objective of edge detection in image processing is to reduce data storage while at same time retaining its topological properties, to reduce transmission time and to facilitate the extraction of morphological outlines from the digitized image.

1.2.2 Criteria for Edge Detection

There are large numbers of edge detection operators available, each designed to be sensitive to certain types of edges. The Quality of edge detection can be measured from several criteria objectively. Some criteria are proposed in terms of mathematical measurement, some of them are based on application and implementation requirements. In all five cases a quantitative evaluation of performance requires use of images where the true edges are known.

a) **Good detection**: There should be a minimum number of false edges. Usually, edges are detected after a threshold operation. The high threshold will lead to less false edges, but it also reduces the number of true edges detected.
b) **Noise sensitivity:** The robust algorithm can detect edges in certain acceptable noise (Gaussian, Uniform and impulsive noise) environments. Actually, an edge detector detects and also amplifies the noise simultaneously. Strategic filtering, consistency checking and post processing (such as non-maximum suppression) can be used to reduce noise sensitivity.
c) **Good localization:** The edge location must be reported as close as possible to the correct position, i.e. edge localization accuracy (ELA).
d) **Orientation sensitivity:** The operator not only detects edge magnitude, but it also detects edge orientation correctly. Orientation can be used in post processing to connect edge segments, reject noise and suppress non-maximum edge magnitude.
e) **Speed and efficiency:** The algorithm should be fast enough to be usable in an image processing system. An algorithm that allows recursive implementation or separately processing can greatly improve efficiency.

Criteria of edge detection will help to evaluate the performance of edge detectors. Correspondingly, different techniques have been developed to find edges based upon the above criteria, which can be classified into linear and non linear techniques.

1.2.3 Various Techniques for Edge Detection

Edge detection of an image reduces significantly the amount of data and filters out information that may be regarded as less relevant, preserving the important structural properties of an image. Therefore, edges detected from its original image contain major information, which only needs a small amount of memory to store. The original image can be easily restored from its edge map. Various edge detection algorithms have been developed in the process of finding the perfect edge detector. However, most of them may be grouped into two categories, namely, gradient based edge detection and Laplacian-based edge detection. In the gradient based edge detection, we calculate an estimate of the gradient magnitude using the smoothing filter and use the calculated estimate to determine the position of the edges. In other words the gradient method detects the edges by looking for the maximum and the minimum in the first derivative of the image. In the Laplacian method we calculate the second derivative of the signal and the derivative magnitude is maximum when second derivative is zero, In short, Laplacian method searches for zero crossings in the second derivative of the image to find edges. The original image can be easily restored from its edges.

Two main methods:-

a) **Gradient-based method**: Gradient-based methods (referred in Appendix A) detect edges by looking for maxima and minima in the first derivative of the image.
b) **Laplacian (zero-crossing) based method**: The Laplacian based methods (referred in Appendix B) search for zero crossings in the second derivative of the image in order to find edges, usually the zero-crossings of the Laplacian or the zero-crossings of a non-linear differential expression.

A number of edge detection techniques are available but there is no single detection method that performs well in every possible image context. Various edge detection techniques are used for edge detection like Canny edge detection (referred in Appendix C), Krisch edge detection which are applied on various images. Choice of edge detector to be used depends upon the image properties like noise sensitivity, orientation sensitivity, speed and efficiency.

1.2.4 Motivation behind Edge Detection

The purpose of detecting sharp changes in image brightness is to capture important events and changes in properties of the world. For an image formation model, discontinuities in image brightness are likely to correspond to:-

a) Discontinuities in depth
b) Discontinuities in surface orientation
c) Changes in material properties
d) Variations in scene illumination

In the ideal case, the result of applying an edge detector to an image may lead to a set of connected curves that indicates the boundaries of objects, the boundaries of surface marking as well curves that correspond to discontinuities in surface orientation. If the edge detection step is successful, the subsequent task of interpreting the information contents in the original image may therefore be substantially simplified. Unfortunately, however, it is not always possible to obtain such ideal edges from real life images of moderate complexity. Edges extracted from non-trivial images are often hampered by fragmentation i.e. the edge curves are not connected, missing edge segments, false edges etc., which complicate the subsequent task of interpreting the image data.

1.3 Brief review of Fuzzy Logic

Fuzzy logic is a powerful problem-solving methodology with a myriad of applications in embedded control and information processing [2]. Fuzzy provides a remarkably simple way to draw definite conclusions from vague, ambiguous or imprecise information. In a sense, fuzzy logic resembles human decision making with its ability to work from approximate data and find precise solutions. As shown in Figure 1.13

fuzzy logic and probability theory are the most powerful tools to overcome the imperfection.

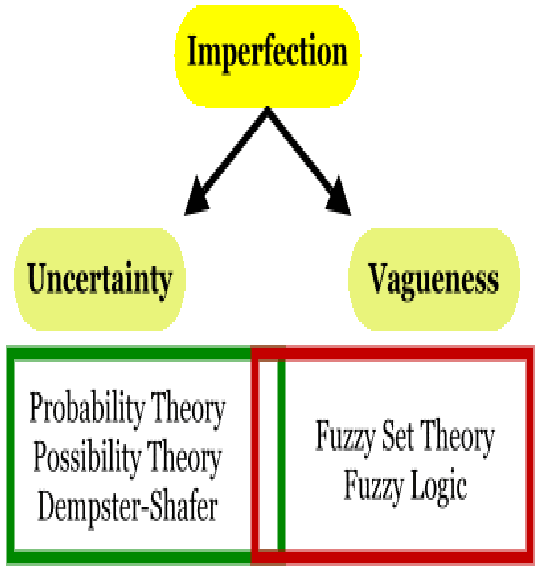

Figure 1.13: Imperfection and theories to handle it.

Unlike classical logic which requires a deep understanding of a system, exact equations, and precise numeric values, fuzzy logic incorporates an alternative way of thinking, which allows modeling complex systems using a higher level of abstraction originating from our knowledge and experience. Fuzzy logic allows expressing the knowledge with subjective concepts such as very hot, bright red and very small height, which are mapped into exact numeric ranges.

The notion central to fuzzy systems is that truth values (in fuzzy logic) or membership values (in fuzzy sets) are indicated by a value on the range [0.0, 1.0], with 0.0 representing absolute Falseness and 1.0 representing absolute Truth. For example, let us take the statement:

"Jane is old."

If Jane's age was 75, we might assign the statement the truth value of 0.80. The statement could be translated into set terminology as follows:

"Jane is a member of the set of old people."

This statement would be rendered symbolically with fuzzy sets as:

mOLD(Jane) = 0.80

Where m is the membership function, operating in this case on the fuzzy set of old people, which returns a value between 0.0 and 1.0.

Fuzzy logic and Fuzzy set theory provide a rich and meaningful addition to standard logic. The mathematics generated by these theories is consistent, and fuzzy logic may be a generalization of classic logic. Fuzzy Logic has been gaining increasing acceptance during the past few years. There are over two thousand commercially available products using Fuzzy Logic, ranging from washing machines to high speed trains. Nearly every application can potentially realize some of the benefits of Fuzzy Logic, such as performance, simplicity, lower cost, and productivity [2].

1.3.1 Structure of Fuzzy Image Processing

Fuzzy image processing is not a unique theory. Fuzzy image processing is the collection of all approaches that understand, represent and process the images, their segments and features as fuzzy sets. The representation and processing depend on the selected fuzzy technique and on the problem to be solved. Fuzzy image processing has three main stages: image fuzzification, modification of membership values, and, if necessary, image defuzzification. (Figure 1.14)

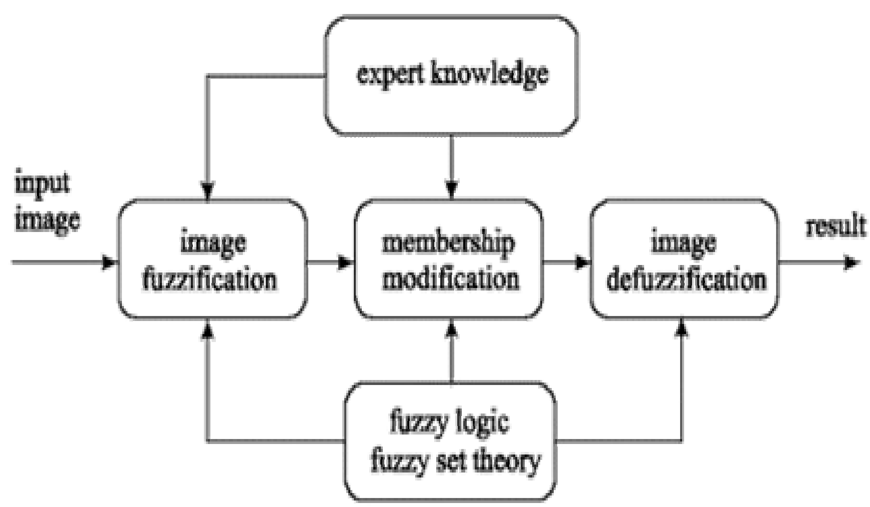

Figure 1.14: The general structure of fuzzy image processing

The fuzzification and defuzzification steps are due to non availability fuzzy hardware. Therefore, the coding of image data (fuzzification) and decoding of the results (defuzzification) are steps that make possible to process images with fuzzy techniques. The main power of fuzzy image processing is in the middle step i.e. modification of membership values, Figure1.15. After the image data are transformed from gray-level plane to the membership plane (fuzzification), appropriate fuzzy techniques modify the membership values.

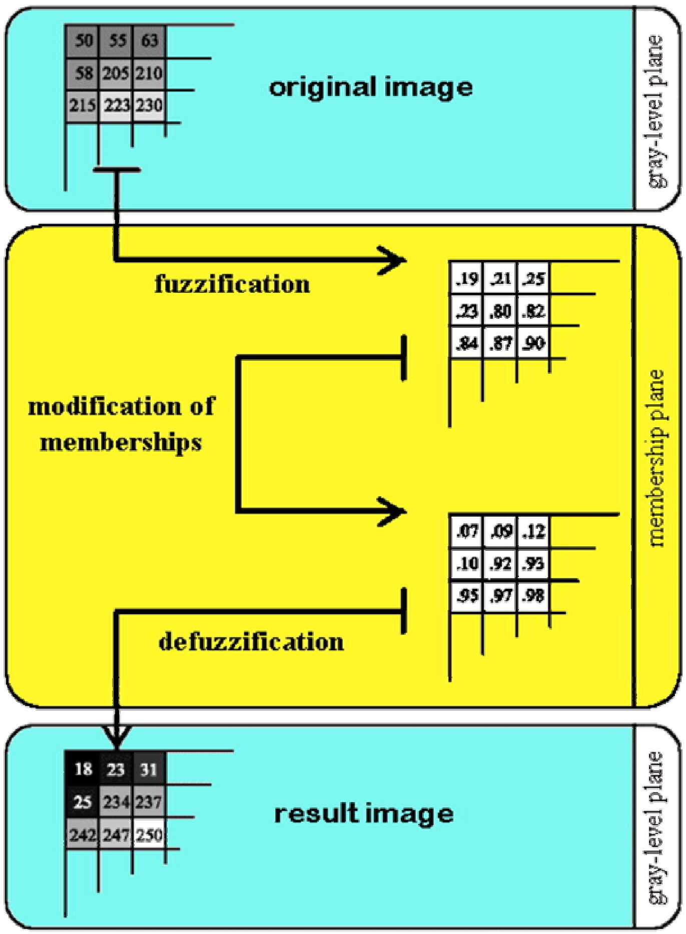

Figure 1.15: Steps of Fuzzy Image Processing

1.3.2 Motivation behind Fuzzy Logic

In many image processing applications, we have to use expert knowledge to overcome the difficulties (e.g. object recognition, scene analysis). Fuzzy set theory and fuzzy logic offer us powerful tools to represent and process human knowledge in form of fuzzy if-then rules. On the other side, many difficulties in image processing arise because the data/tasks/results are uncertain. This uncertainty, however, is not always due to the randomness but to the ambiguity and vagueness. Beside randomness which can be managed by probability theory we can distinguish between three other kinds of imperfection in the image processing (Figure 1.16):-

a) Grayness ambiguity
b) Geometrical fuzziness
c) Vague (complex/ill-defined) knowledge

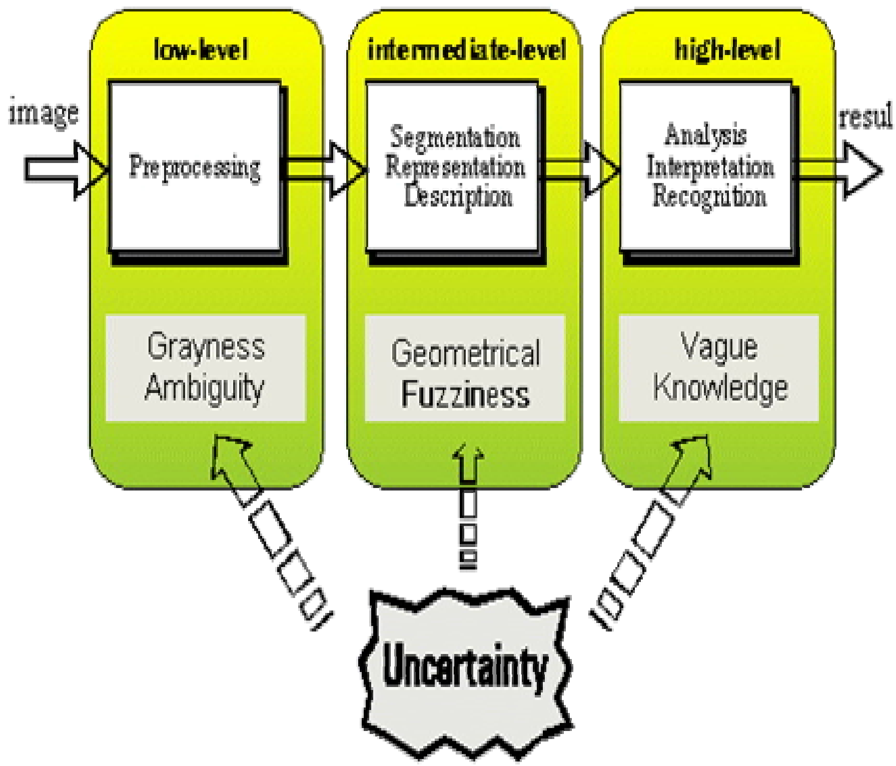

Figure 1.16: Uncertainty/imperfect knowledge in image processing

The question whether a pixel should become darker or brighter than it already is, the question where is the boundary between two image segments, and the question what is a tree in a scene analysis problem, all of these and other similar questions are examples for situations that a fuzzy approach can be the more suitable way to manage the imperfection.

General observations about fuzzy logic are:

a) Fuzzy logic is conceptually easy to understand. The mathematical concepts behind fuzzy reasoning are very simple.

b) Fuzzy logic is flexible. With any given system, it's easy to manage it or layer more functionality on top of it without starting again from scratch.

c) Fuzzy logic is tolerant of imprecise data. Everything is imprecise if we look closely enough, but more than that, most things are imprecise even on careful inspection. Fuzzy reasoning builds this understanding into the process rather than tacking it onto the end.

d) Fuzzy logic can model nonlinear functions of arbitrary complexity. We can create a fuzzy system to match any set of input-output data. This process is made particularly easy by adaptive techniques like ANFIS (Adaptive Neuro-Fuzzy Inference Systems), which are available in the Fuzzy Logic Toolbox.

e) Fuzzy logic can be blended with conventional control techniques. Fuzzy systems don't necessarily replace conventional control methods. In many cases fuzzy systems augment them and simplify their implementation.

f) Fuzzy logic is based on natural language. The basis for fuzzy logic is the basis for human communication. This observation underpins many of the other statements about fuzzy logic.

The last statement is perhaps the most important one and deserves more discussion. Natural language, which is used as a media of communication by human beings, has been shaped by thousands of years of human history to be convenient and efficient. Sentences written in ordinary language represent a triumph of efficient communication.

CHAPTER 2

LITERATURE REVIEW

In this section, work done in the area of edge detection is reviewed and focus has been made on detecting the edges of the digital images using fuzzy logic approach. Edge detection is a problem of fundamental importance in image analysis. In typical images, edges characterize object boundaries and are therefore useful for segmentation, registration, and identification of objects in a scene. Edge detection of an image reduces significantly the amount of data and filters out information that may be regarded as less relevant, preserving the important structural properties of an image [4].

In 1997 Ng Geok See, Chan Khue Hiang, proposed a technique for edge detection based on neural network. Neural network has many processing elements joined together and usually organized into groups called layers. Training is provided to the neural network in supervised or unsupervised learning mode, to force the network to yield particular result to a specific input[1].

In 1998 Zhengquan He, M.Y.Siyal, proposed a new technique based on neural network. Most of the existing techniques like Sobel[refrencee} are effective in certain senses and require more computation time. [6]. In the proposed edge detection technique a three layer BP neural network is employed to classify the edge elements in binary images into one of the predefined categories. To detect edges first binarize the image by choosing threshold by some optimal criteria and classify the edge patterns of binary images in different categories. Train the neural network on these patterns and on their noisy patterns. After the network is trained, it can recognize the input pattern as a most like pattern in our edge pattern bank. This technique is more flexible to the edge structures in the image. It can not only extract straight lines but also can extract corners and arcs edges.

In 2005 Zhang, Zhao and Li Su, proposed a technique based on the integer logarithm ratio of gray levels. In order to remove the ability of noise rejection they proposed a ratio of gray levels between the two successive image points rather than the difference of gray levels to denote the variation in the gray levels [5]. In this, division operation becomes the subtraction operation of the logarithmic ratio of gray levels. This is more convenient for calculations

In 2005 Stamatia Giannarou, Tania Stathaki, proposed a technique that allows combining the methods of different edge detection operators in order to yield improved results for edge detection in an image. This is called Receiver Operating Characteristics (ROC) analysis [7]. This technique uses the statistical approach to automatically form a optimum edge map, by combining edge images from different detectors. The characteristics of this method are to produce accurate and noise free results. One possible concern regarding these techniques is the selection of the edge detectors to be combined.

In 2006 M.Hanmandlu, Rohan Raj Kalra, Vamsi Krishna Madasu, proposed a technique based on Univalue Segment Assimilating Nucleus (USAN) area i.e. fuzzy technique. The USAN characterizes the structure of the edge present in the neighborhood of a pixel and can thus be considered as a unique feature of the pixel and is fuzzified [8]. This technique is best in yielding the large number of longest edge segments. This is used for the applications like face recognition and fingerprint identification, as it does not distort the shape of the image and is able to retain all the important edges. Appropriate fuzzification function and threshold election are important for the success of the proposed edge detection algorithm.

Later on Fast fuzzy edge detection technique was proposed. Heuristic membership functions, simple fuzzy rules, and fuzzy complements were used to develop new edge detectors [9]. Then Fuzzy edge detector using entropy optimization was proposed. The proposed fuzzy edge detector involves two phases:– global contrast intensification and local fuzzy edge detection. In the first phase, a modified Gaussian membership function is chosen to represent each pixel in the fuzzy plane [10].

To realize the fast and accurate detection of the edges from the blurry images, the Fast Multilevel Fuzzy Edge Detection (FMFED) algorithm was proposed [11]. The FMFED algorithm first enhances the image contrast by means using a simple transformation function based on two image thresholds. Second, the edges are extracted from the enhanced image by the two-stage edge detection operator that identifies the edge candidates based on the local characteristics of the image and then determines the true edge pixels using the edge detection operator based on the extreme of the gradient values.

After going through all the techniques in area of image it was realized that fuzzy is the most accurate and best technique. So the major stress will be on development of algorithms for improving the quality of detecting edges by Edge detection technique using fuzzy logic approach.

CHAPTER 3
PROBLEM STATEMENT

3.1 Problem Definition

In image processing and computer vision, edge detection treats the localization of significant variations of a gray level image and the identification of the physical and geometrical properties of objects of the scene. Edge detection is a difficult issue. Many difficulties come from the complex contents like noise, varying contrast in an image, orientation sensitivity.

Traditional edge detection techniques, such as Robert operator, Sobel operator, Laplacian of Gaussian operator are widely used. Most of the existing techniques are either very sensitive to noise and do not give satisfactory results in low contrast areas. A fuzzy theory based Edge Detector avoids these problems and is a better method for edge information detection and noise filtering than the traditional methods. Edge detection using fuzzy logic provides an alternative approach to detect edges.

First-order linear filters are mostly applied for edge detection in digital images. Nevertheless they don't allow good results to be obtained from images where the contrast varies a lot, due to non-uniform lighting [12]. But in this research fuzzy inference system is applied for edge detection to improve the performance. A non-linear image filtering technique is presented which is based on fuzzy inference systems (FIS). First, an input image is processed in different non-successive linear filtering stages, which means that the input to each filter is always the original image. The gray level in each pixel of the resulting image is then obtained by applying the FIS system to the corresponding values in the output images of the linear operators, in the same pixel.

And evaluate the efficiency of a FIS system applied to the edge detection problem. During input image pre-processing, three kinds of linear filters are applied to it: Sobel operators, used to estimate its derivatives in horizontal and vertical directions (h_{DH} and h_{DV} filters), a low-pass (mean) filter and a high-pass filter. The developed fuzzy system's purpose is to determine if pixel evaluated is or is not present in one of image's edges.

3.2 Methodology

The step-by-step methodology to be followed for detecting edges using fuzzy theory.

a) A thorough analysis of various edge detection techniques will be done.
b) Advantages and disadvantages of using Fuzzy logic will be analyzed.
c) Based upon above analysis a program will be developed for detecting edges in C, C++ or VC++.
d) Results achieved after the execution of program will be compared with the earlier outputs

CHAPTER 4
IMPLEMENTATION

4.1. Image Pre-Processing

During input image pre-processing stage, 4 linear filters are used. Sobel operators h_{DH} and h_{DV} are kernels with 3x3 elements given by.

$$h_{DH} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$h_{DV} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

$$h_{DV} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

As a high-pass filter 3x3 kernel, given by:

$$h_{HP} = \begin{bmatrix} -1/16 & -1/8 & -1/16 \\ -1/8 & 3/4 & -1/8 \\ -1/16 & -1/8 & -1/16 \end{bmatrix}$$

Filter *MF*, in turn, was chosen in such a way as to guarantee that the gray level in each pixel of the output image is the arithmetic mean of the gray levels in a 9X9 neighborhood of the same pixel in the input image.

In that manner:

$$h_{MF} = 1/25 \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Given the kernels associated with each filter, the filtered images may be computed through a bidimensional convolution operation:

a) $DH = h_{DH} * I$

b) $DV = h_{DV} * I$

c) $HP = h_{HP} * I$

d) $M = h_{MF} * I$

4.2 Inference Rules Definitions

The fuzzy inference rules are defined in such a way that the FIS system output ("Edges") is high only for those pixels that belong to edges in the input image. Robustness to contrast and lighting variations are also in mind when these rules are established. The first 3 rules are defined to represent the general notion that pixels belong to that edge where the high variation of gray level.

1. (*DH* low) and (*DV* low) → ("Edges" low).

2. (*DH* medium) and (*DV* medium) → ("Edges" high).

3. (*DH* high) or (*DV* high) →("Edges" high).

To guarantee that edges in regions of relatively low contrast can be detected, the following rules are to favour medium variations of the gray level in a specific direction in regions of low frequency of the input image (*HP* “low”):

4. (*DH* medium) and (*HP* low) → (“Edges” high).

5. (*DV* medium) and (*HP* low) → (“Edges” high).

Rules 6 and 7 are to avoid output image pixels that belong to regions of the input where the mean gray level is lower. These regions are proportionally more affected by noise, if it is uniformly distributed over the whole image. The goal here is to design a system which makes it easier to include edges in low contrast regions, but which does not favour false edges by effect of noise.

6. (*DV* medium) and (*M* low) → (“Edges” low).

7. (*DH* medium) and (*M* low) → (“Edges” low).

Rules 8 to 11 are defined to avoid forming double edges in the output image (they tend to appear due to shadows in the natural images).

8. (*DV* high) and (*DV* $(i + 1; j)$ high) → (“Edges” medium).

9. (*DH* high) and (*DH* $(i; j + 1)$ high) → (“Edges” medium).

10. (*DV* medium) and (*DV* $(i+1; j)$ high)→(“Edges” low).

11. (*DH* medium) and (*DH* $(i; j+1)$ high)→(“Edges” low).

Finally, rule 12 is defined to avoid isolated pixels in the output image, favouring only continuous lines. It also avoids including points by effect of noise, since this tends to generate isolated pixels in the image which represents the input’s edges.

12. (*DV* $(i; j + 1)$ low) and (*DH* $(i + 1; j)$ low) and (*DV* $(i; j - 1)$ low) and (*DH* $(i - 1; j)$ low) → (“Edges” low).

4.3 Implementation Details

We start by opening the program in Microsoft Visual C++.

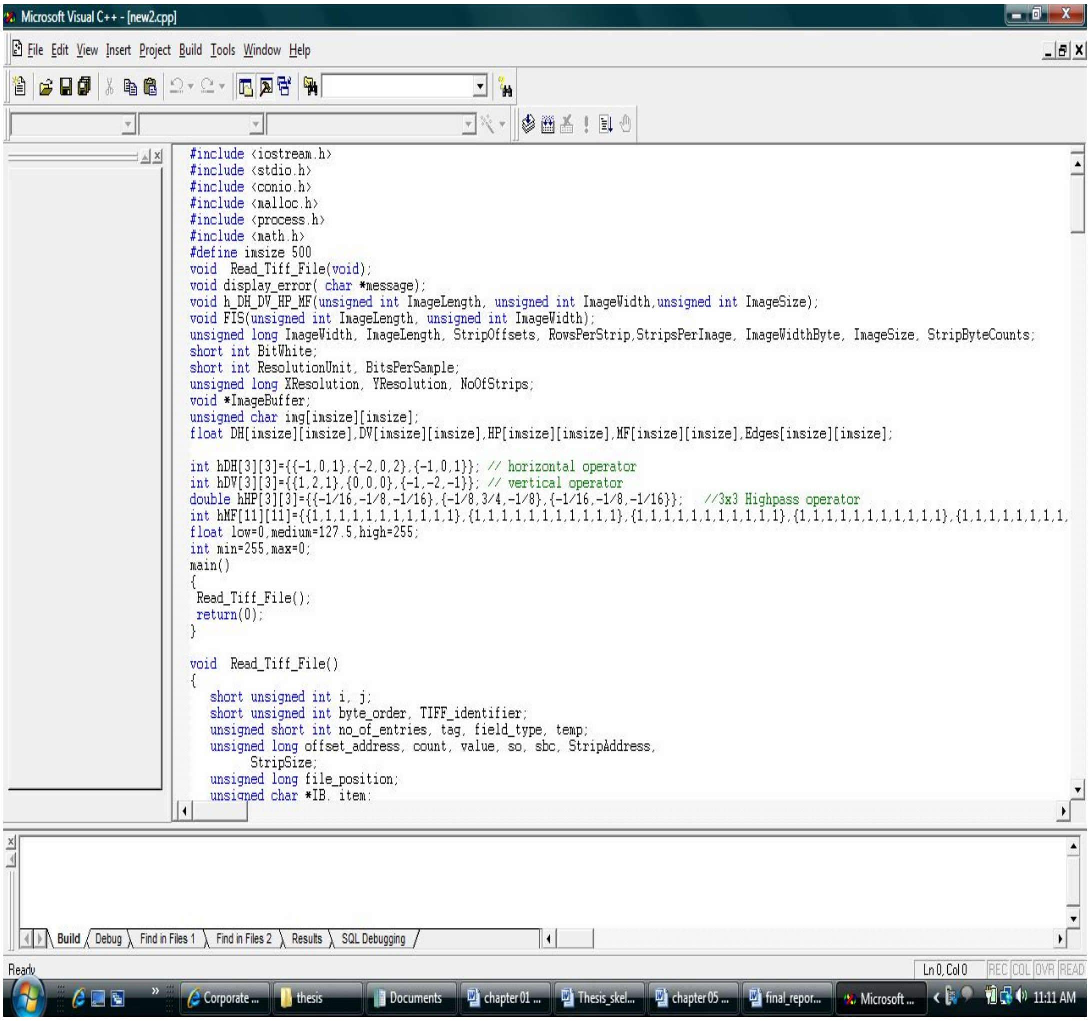

Figure 4.1: Program code

Execution of the program begins by clicking on the execute.exe option.

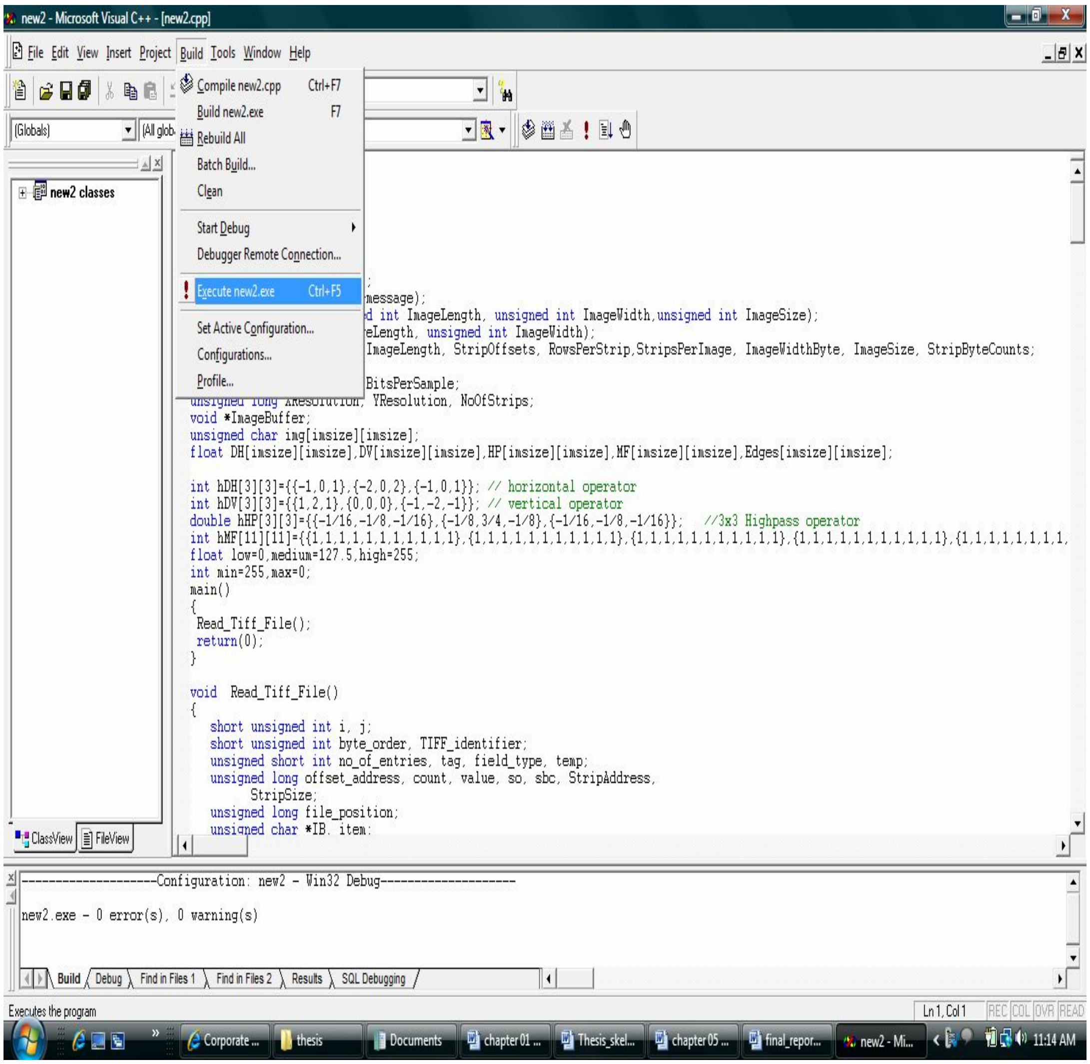

Figure 4.2: Executing the program

After executing the program, console (black screen) appears to get an input and to write an output file.

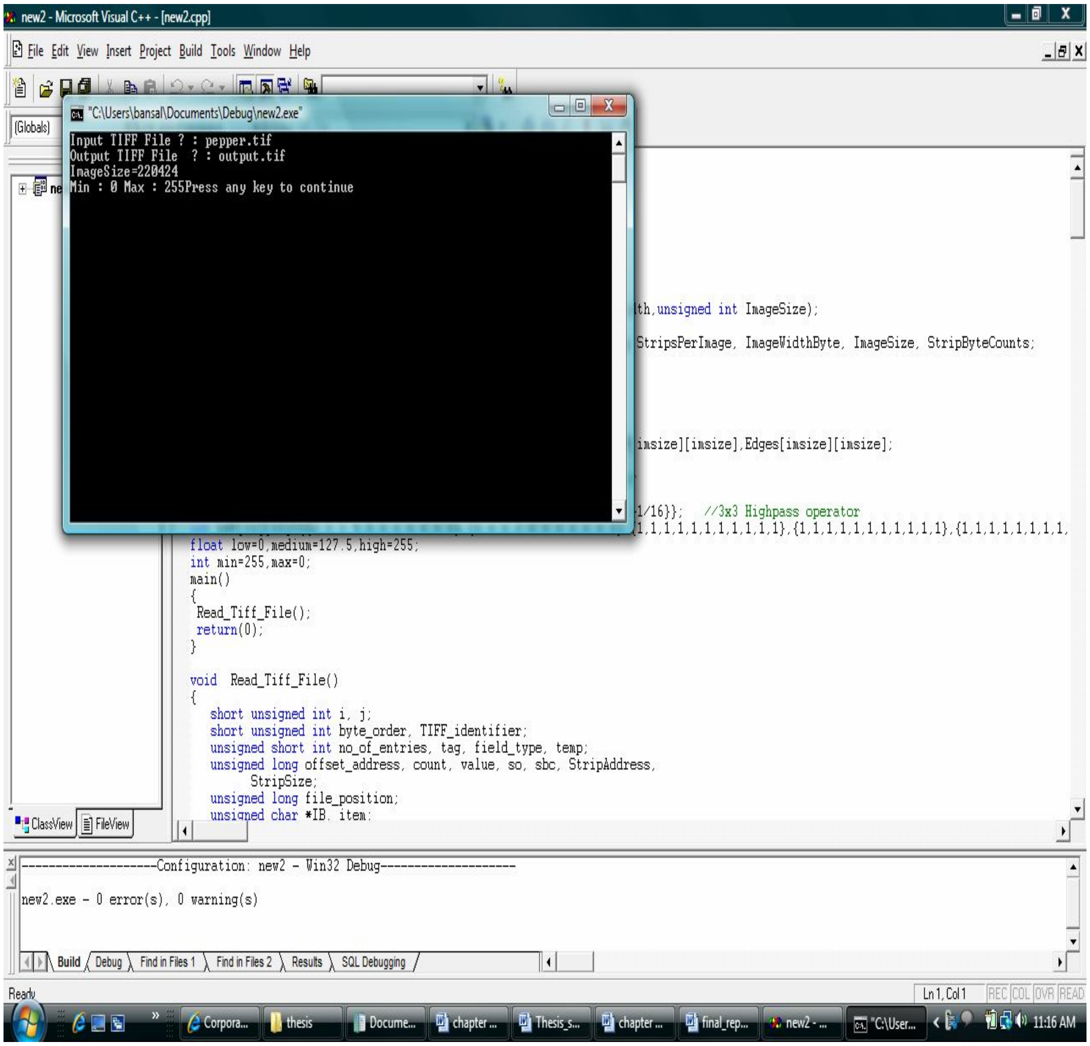

Figure 4.3: Reading an input file and Writing an output file

Input file:

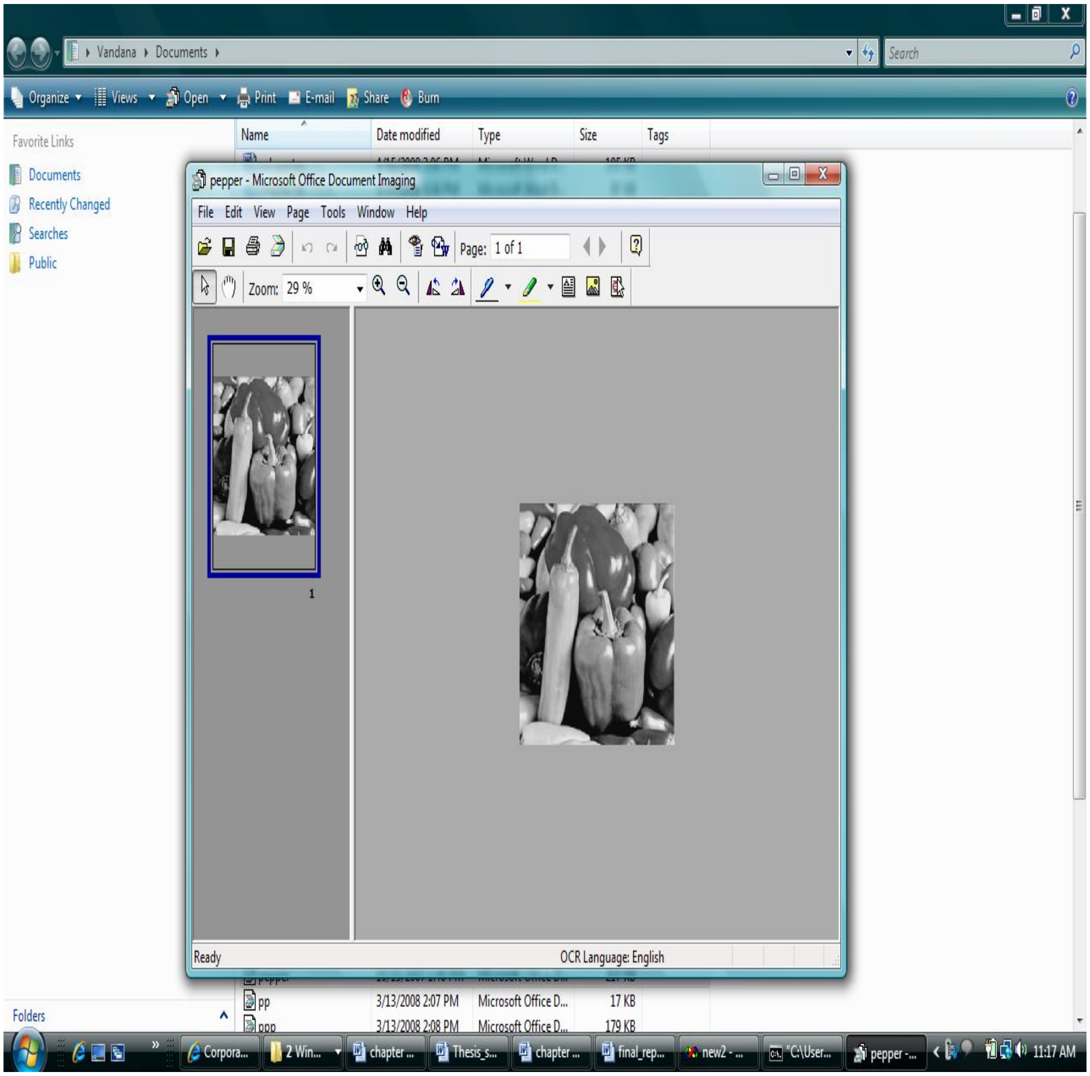

Figure 4.4: Input file

Output file:

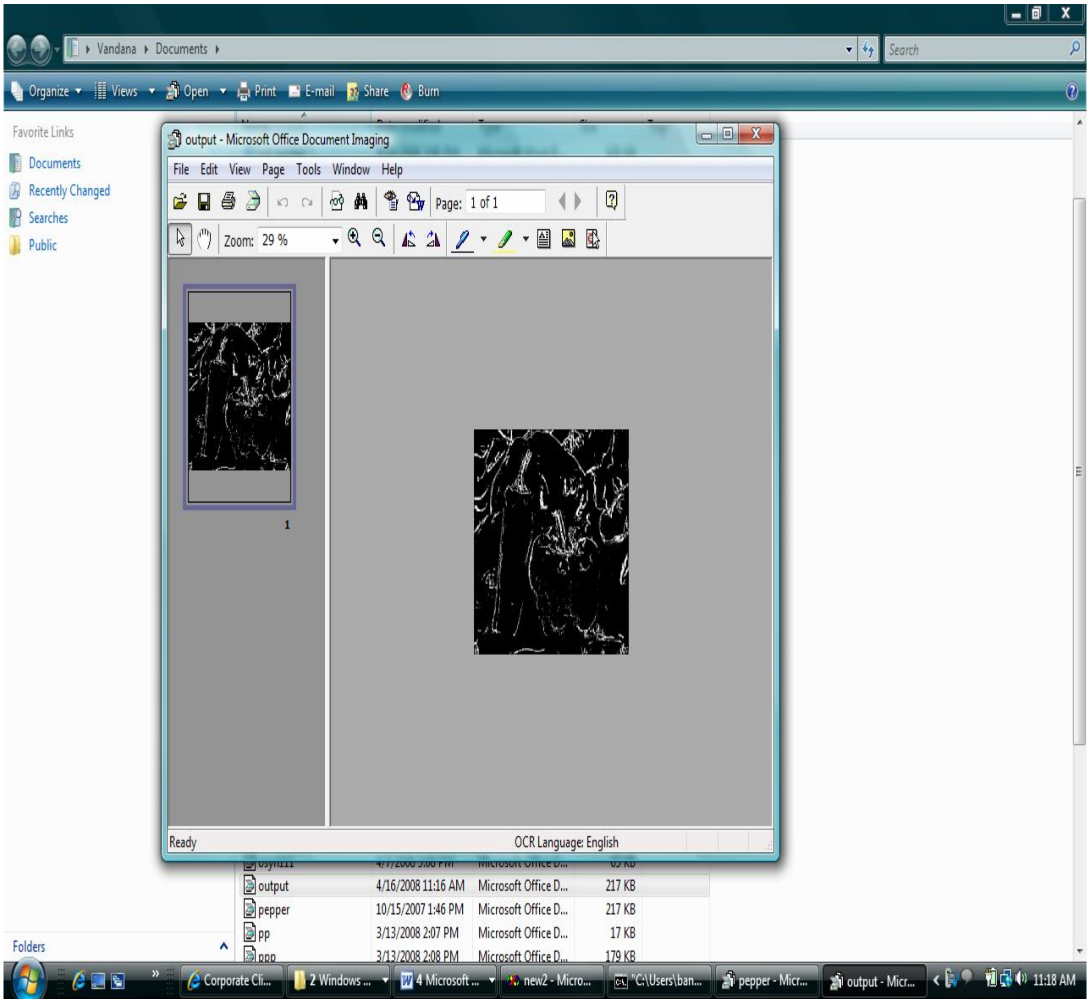

Figure 4.5: Output File

CHAPTER 5
TESTING AND RESULTS

5.1 Testing

Testing is a process of executing a program with the intention of finding error. A successful test is one that finds an undiscovered error.

Testing of an algorithm is done by giving the input of different images to check whether all the required outputs are generated and are in the desired and proper format.

5.2 Results

Better result is obtained from our algorithm as compare to other edge detectors. Digital image as input is given and edges of that image are produced. Results are shown as below:

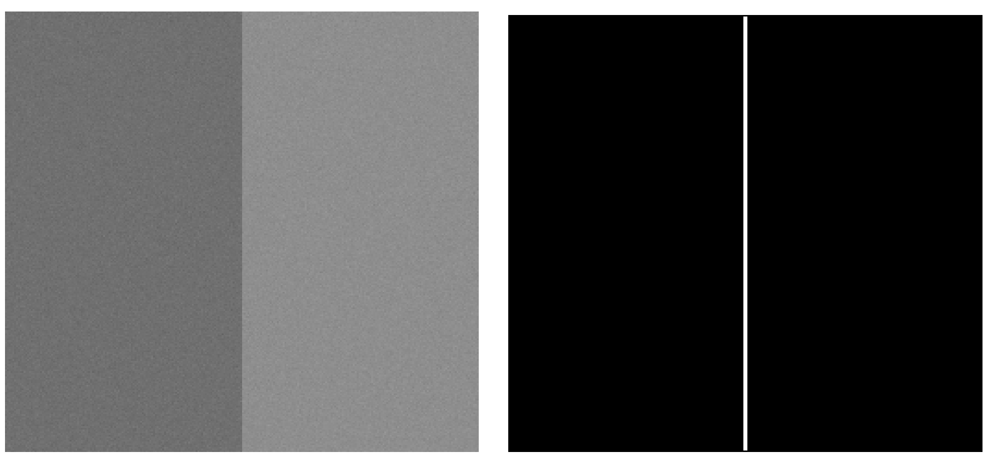

Figure 5.1: Edge Detection (a) Original image (b) Output image

Original Digital Image

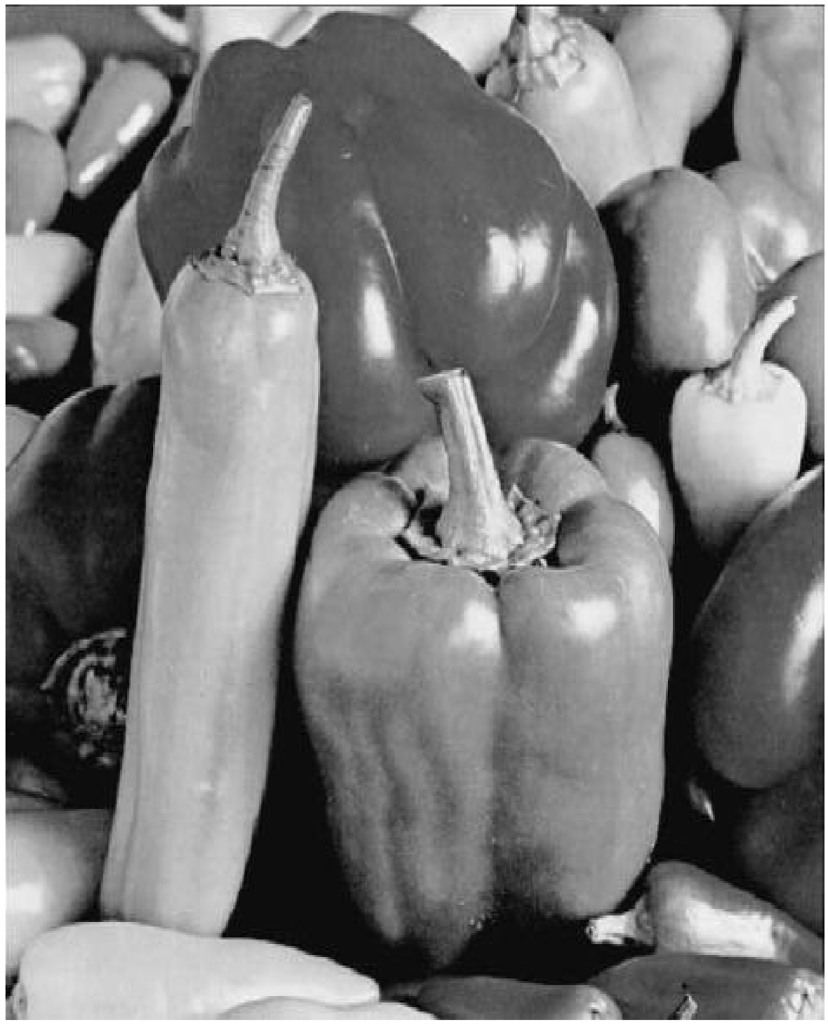

Figure 5.2: Original Image

Output Image

Figure 5.3: Output Image

CHAPTER 6
CONCLUSION & FUTURE SCOPE

6.1 Conclusion

In this thesis work, better algorithm has been proposed to improve the detection of edges by using fuzzy rules. This algorithm is adaptable to various environments. The weights associated with each fuzzy rule were tuned to allow good results to be obtained while extracting edges of the image, where contrast varies a lot from one region to another. During the performance tests, however, all parameters were kept constant.

The results allow us to conclude that, the implemented FIS system presents greater robusteness to contrast and lighting variations, besides avoiding obtaining double edges.

6.2 Future Scope

In fact, the proposed technique is to find more fine edges using fuzzy logic technique. In future, modification of fuzzy rules can produce better result. Further tuning of the weights associated to the fuzzy inference rules is still necessary to reduce even more inclusion in the output image of pixels not belonging to edges.

Our proposed technique is restricted only to gray scale images, this can be extended to color images in that case, and the detection would become significantly more complex.

REFERENCES

[1] Ng Geok See and Chan Khue hiang. **"Edge Detection using supervised Learning and voting scheme"**. Nanyang Technological University, National university of Singapore, Singapore.

[2] Hamid R. Tizhoosh, "**Fuzzy Image Processing".** www.pami.uwaterloo.ca/tizhoosh/edge.html

[3] Md. Shoiab Bhuiyan, Yuiji Iwahori, and Akira Iwata. **"Optimal edge detection under difficult imaging conditions"**. Technical report, Educational Center for Information Processing and Dept. of Electrical and Computer Engineering, Nagoya Institute of Technology, Showa, Nagoya, 466-8555, JAPAN, URL:- http://www.center.nitech.ac.jp/people/bhuiyan /pub.html.

[4] Ursula Kretschmer, Maria S. Orozco, Oscar E. Ruiz and Uwe Jasnoch, **"Edge and corner identification for tracking the line of sight".**

[5] Renyan Zhang, Guoling Zhao, Li Su. **"New edge detection method in image processing".** College of Autom., Harbin Engineering University, China.

[6] Zhengquan He, M.Y.Siyal. "**An image detection technique based on morphological edge detection and background differencing for real-time traffic analysis".** Elsevier Science Inc. New York, NY, USA

[7] Stamatia Giannarou, Tania Stathaki**. "Edge Detection Using Quantitative Combination of Multiple Operators".** Communications and Signal Processing Group, Imperial College LondonExhibition Road, SW7 2AZ London, UK.

[8] M. Hanmandlu, Rohan Raj Kalra, Vamsi Krishna Madasu, Shantaram Vasikarla. **"Area based Novel Approach for Fuzzy Edge Detection".** Department of Electrical Engineering, Indian Institute of Technology, Delhi.

[9] Hamid R. Ezhoosh, "**Fast Fuzzy Edge Detection".** Pattern Recognition and Machine Intelligence Lab Systems Design Engineering, University of Waterloo Waterloo, Ontario, N2L 3G1, Canada.

[10] Madasu Hanmandlu, John See, Shantaram Vasikarla. **"Fast fuzzy edge detector using entropy optimization".** Dept. of Electrical, Engineering I.I.T. Delhi, Faculty of Engineering, Multimedia University, Cyberjaya Selangor D.E. 63100, Malaysia, IT Department, Intercontinental University Los Angeles, CA 90066, USA.

[11] Jinbo Wu, Zhouping Yin, and Youlun Xiong, **"The Fast Multilevel Fuzzy Edge Detection of Blurry Images"**.

[12] Cristiano Jacques Miosso, Adolfo Bauchspiess. **"Fuzzy Inference System Applied to Edge Detection in Digital Images"**. Departamento de Engenharia Electrical Faculdade de Tecnologia Universidade de Braslia – UnB

Appendix A: Gradient Edge Detectors

The first derivative assumes a local maximum at an edge. For a gradient image f(x, y), at location (x, y), where x and y are the row and column coordinates respectively, typically consider the two directional derivatives. The two functions that can be expressed in terms of the directional derivatives are the gradient magnitude and the gradient orientation.

The gradient magnitude is defined by

$$g(x, y) \cong (\Delta x^2 + \Delta y^2)^{\frac{1}{2}}$$

$$\Delta x = f(x+n, y) - f(x-n, y) \quad and\ \Delta y = f(x, y+n) - f(x, y-n)$$

,Where n is a small integer usually unity.

This quantity gives the maximum rate of increase of f(x, y) per unit distance in the gradient orientation of g(x, y). The gradient orientation is also an important quantity. The gradient orientation is given by

$$\theta(x, y) \cong atan\ (\frac{\Delta y}{\Delta x})$$

Here the angle is measured with respect to the x-axis. The direction of the edge at (x, y) is perpendicular to the direction of the gradient vector at that point. The other method of calculating the gradient is given by estimating the finite difference.The most popular classical gradient-based edge detectors are Roberts cross gradient operator, Sobel operator and the Prewitt operator.

A.1 Robert Edge Detector

The calculation of the gradient magnitude of an image is obtained by the partial derivatives G_x and G_y at every pixel location. The simplest way to implement the first order partial derivative is by using the Roberts cross gradient operator.

Therefore

$$G_x = f(i, j) - f(i+1, j+1)$$

$$G_y = f(i+1, j) – f(i, j+1)$$

The above partial derivatives can be implemented by approximating them to two 2x2 masks. The Roberts operator masks are:

+1	0
0	-1

G_x

0	+1
-1	0

G_y

These filters have been the shortest support, thus the position of the edges is more accurate, but the problem with the short support of the filters is its vulnerability to noise. It also produces very weak response to genuine edges unless they are very sharp.

A.2 Prewitt Edge detector

The Prewitt edge detector is a much better operator than Roberts's operator. This operator having a 3 x 3 masks deals better with the effect of noise. An approach using the masks of size 3 x 3 is given below, the arrangement of pixels about the pixels [i, j].

A_0	a_1	A_2
A_7	[i,j]	A_3
A_6	a_5	a_4

The partial derivatives of the Prewitt operator are calculated as

$$G_x = (a_6 + ca_5 + a_4) - (a_0 + ca_1 + a_2)$$

$$G_y = (a_2 + ca_3 + a_4) - (a_0 + ca_7 + a_6)$$

The constant c implies the emphasis given to pixels closer to the centre of the mask. G_x and G_y are the approximation at *[i, j]*.

Setting c=1, the Prewitt operator is obtained. Therefore the Prewitt masks are as follows

-1	-1	-1
0	0	0
1	1	1

G_x

-1	0	1
-1	0	1
1	0	1

G_y

These masks have longer support. They differentiate in one direction and average in the other direction, so the edge detector is less vulnerable to noise.

A.3 Sobel Edge Detector

The Sobel edge detector is very much similar to the Prewitt edge detector. The difference between the both is that the weight of the centre coefficient is 2 in the Sobel operator. The partial derivatives of the Sobel operator are calculated as

$$G_x = (a_6 + 2a_5 + a_4) - (a_0 + 2a_1 + a_2)$$

$$G_y = (a_2 + 2a_3 + a_4) - (a_0 + 2a_7 + a_6)$$

Therefore the Sobel masks are:

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

G_x

-1	0	1
-2	0	2
1	0	1

G_y

Although the Prewitt masks are easier to implement than Sobel masks, the later has better noise suppression characteristics.

Appendix B: Laplacian of Gaussian (LOG)

The principle used in the Laplacian of Gaussian method is, the second derivative of a signal is zero when the magnitude of the derivative is maximum. The Laplacian of a 2-D function f*(x, y)* is defined as

$$\nabla^2 f(x,y) = \nabla(\nabla f(x,y)) = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2}.$$

B.1 The LOG Operator

The two partial derivative approximations for the Laplacian for a

3 x 3 region are given as

$$\nabla^2 f = 4(a_8) - (a_1 + a_3 + a_5 + a_7)$$

$$\nabla^2 f = 8(a_8) - (a_0 + a_1 + a_2 + a_3 + a_4 + a_5 + a_6 + a_7)$$

The masks for implementing these two equations are as follows

0	-1	0
-1	4	-1
0	-1	0

-1	-1	-1
-1	8	-1
-1	-1	-1

The above partial derivative equations are isotropic for rotation increments of 90^0 and 45^0, respectively. Edge detection is done by convolving an image with the Laplacian at a given scale and marks the points where the result have zero value, which is called the zero-crossings. These points should be checked to ensure that the gradient magnitude is large.

B.2 Marr and Hildreth method

The edge pixels in an image are determined by a single convolution operation. The basic principle of this method is to find the position in an image where the second derivative becomes zero. These positions correspond to edge positions. The Gaussian function firstly smoothens or blurs any edge, blurring is advantageous here because Laplacian would be infinity at unsmoothed edge and therefore edge position is still preserved. LOG operator is still susceptible to noise, but by ignoring zero-crossings produced by small changes in image intensity can reduce the effects of noise.

LOG operator gives edge direction information as well as edge points, determined in the LOG formulation is to smooth the image and the purpose of the Laplacian operator is to provide an image with zero crossings used to establish the location of edges. It always generates closed contours, which is not realistic. The Marr-Hildreth operator will mark edges at some locations that are not edges.

Appendix C: The Canny Edge Detector

The Canny's operator is one of the most widely used edge finding algorithms. The Canny approach to edge detection is optimal for step edges corrupted by white Gaussian noise. This edge detector is assumed to be output of a filter that both reduces the noise and locates the edges. Its 'optimality' is related to the following performance criteria:

a) Good detection: Both the probability of missing real edge points and incorrectly marking non-existent edge points must be minimal.
b) Good localization: The distance between the actual and detected location of the edge should be minimal.
c) Minimal response: This criteria state that multiple responses to a single edge and 'false' edges due to noise must be eliminated.

After optimizing the above criteria in a certain fashion, an efficient approximation to the required operator is the first derivative of the two-dimensional Gaussian function G(x, y) applied to the original image. For example, the partial derivative with respect to x is defined as follows:

The first step of the edge detection algorithm is to convolve the image I(x, y) with a two dimensional Gaussian filter and differentiate in the direction of n.

Candidate's edge pixels are identified as the pixels that survive a thinning process known as no maximal suppression. Any gradient value that is not a local peak should be set to zero. Each pixel in turn, forms the centre of a 3x3 neighbourhood. The gradient magnitude is estimated for two locations, one on each side of the pixel in the gradient direction, by interpolation of the surrounding values. If the value of the centre pixel is larger than these of the surrounding pixels, the pixel is considered a maximal point. Otherwise, the pixel value is set to zero.

The last step of the algorithm is to threshold the candidate edges in order to keep only the significant ones. Canny suggests hysterics threshold instead of a global threshold values. The high threshold is used to find "seeds" for strong edges. These seeds are grown to as long as an edge as possible, in both directions, until the edge strength falls below the low threshold value.

Color edge detection is called `imgCED. For colour edge detection three methods:` output fusion methods, multi-dimensional gradient methods, and vector methods. Output fusion appears to be the most popular; the goal is to perform edge detection three times, once each for red, green, and blue. Multi-dimensional gradient methods short-circuit the process somewhat by combining the three gradients into one and detecting edges only once. Advantages and disadvantages of all these detectors are discussed in Table A1.

Operators	Advantages	Disadvantages
Classical (Sober, Prewitt, Kirsch,…)	Simplicity, Detection of edges and their orientations.	Sensitivity to noise, Inaccurate
Zero Crossing (Laplacian second directional derivative)	Detection of edges and their orientations, Having fixed characteristics in all directions.	Sensitivity to noise, Reresponsing to some of existing images.
Laplacian of Gaussian (LoG) (Marr – Hildreth)	Finding correct places of edges, Testing wider area around the pixel.	Malfunctioning at corners, Covers and where the gray level intensity function varies, Not finding the orientation of edge because of usage of Laplacian filter.
Gaussian (Canny, Shencastan)	Using probability for finding error rate, Localization and response, Improving signal and noise ratio, Better detection specially in noise conditions.	Complex computation, False zero crossing, Time consuming
Colored Edge Detector	Accurate, More efficient in object recognition	Complicated, Complex compuations.

Table A1: Advantages and Disadvantages of Edge Detectors

PAPERS COMMUNICATED/PUBLISHED/ACCEPTED

[1]. Vandana Mittal and Shalini Batra "**Edge Detection Technique Using Neuro fuzzy sytem**" at National Conference on Challenges and Opportunities in Information Technology held at RIMT – Institute Of Engineering and Technology, Mandi Gobindgarh from March 28-29, 2007.