DESIGN OF TOOL SET FOR PROCESSING RASTER IMAGES

A DISSERTATION

Submitted in partial fulfilment of the requirements for the award of the degree

of

MASTER OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

By

SANDHYA N.

ER & DCI NOIDA

IIT Roorkee-ER&DCI, Noida
C-56/1, "Anusandhan Bhawan"
Sector 62, Noida-201 307

FEBRUARY, 2003

CANDIDATE'S DECLARATION

I hereby declare that the work presented in this dissertation titled **"DESIGN OF TOOL SET FOR PROCESSING RASTER IMAGES"**, in partial fulfillment of the requirements for the award of the degree of **Master of Technology** in **Information Technology**, submitted in **IIT, Roorkee – ER&DCI Campus, Noida,** is an authentic record of my own work carried out during the period from August, 2002 to February, 2003 under the guidance of **Mr.P.N.GOSWAMI**, Director, research and Development, Electronics Research and Development Center of India, Noida.

The matter embodied in this dissertation has not been submitted by me for award of any other degree or diploma.

Date: 24/02/03

Place: Noida

(Sandhya.N.)

CERTIFICATE

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

Guide:

(Mr. P.N. Goswami)

Director R&D,

ER&DCI, Noida.

Date: 24/02/2003

Place: Noida

ACKNOWLEDGEMENT

The successful completion of this project " Design And Implementation Of Toolset for Processing raster Images",is attributed to the great and indespensable help I have received from different people of my institute.

I take this oppurtunity to thank **Prof.Prem Vratt** ,Director,IIT-Roorkee and **Shree.R.K.Verma** ,Executive Director,ER&DCI,Noida,who were the back bones of this M.Tech (IT) course.

Special thanks to **Prof.A.K.Awasthi, Dean,PGS&R**,IIT-Roorkee, for his kind coordiantion and cooperation in running this course successfully.

I express my profound gratitude to **Prof.R.P.Agrawal** ,Course coordinator, M.Tech(IT) and **Mr.V.N.Shukla** Director, Special.Applications, ER&DCI,who contributed a lot to make this M.Tech cvourse a grand success.

My sincere thanks to my project guide **Mr.P.N.Goswami**, Director, Research&Development, ER&DCI,for his proper guidance,excellent spirit of coordination,his constructive criticisms and his help in formulation of this project.

I extend my thanks to **Mr.Munish Kumar** and **Dr.P.R.Gupta,** staff members, M.Tech(IT),ER&DCI for their proper guidance,help and support.

I am indebted to **Mr.T.Venkatesan**, L-cube innovative solutions,chennai, who extended a helping hand whenever I needed.without him,this project would have been an uphill task.

Above all, my heartfelt thanks to my family and friends,who offered moral support while suffering from my neglect for the past two years.

(N.Sandhya)

CONTENTS

ABSTRACT

Image processing refers to all the various operations, which can be applied to the image data. One can imagine various real-life applications, which needs images to be manipulated. Images form an important helper in information exchange. It may be possible to explain a concept that needs a very long language based description or a very detailed figure with a set of photographs or a film in a very short time.

The image processing tool set is the perfect environment for the budding artists and those who are new to computer graphics. This tool set allows the users to dabble, get the feel of the very different world of computer art and experience the thrill of creation without breaking their pocket book. The tool set is a nice environment to analyse and process raster images of the type the windows applications doesn't support normally. The interface and new features are such easy and familiar and user friendly. The image processing tool set is a special application package designed for nonprogrammers, who don't have to worry about how the graphics is actually working in a computer. The user will be able to communicate in his own terms with the package due to the advantage of a user friendly graphical interface.

Enhancing the details in an X-ray image for a better visualization, OCR (Optical Character Recognition) and constructing a 3D world from 2D images are examples of possible applications.

The application package should be able to perform the following functions:

- Edge detection for raster type images.
- Low pass, high pass, band pass and user defined filtering for raster type images.
- Perform arithmetic operations of Add, Sub, Multiply, Divide, And and Or functions on images.
- Hide an image into another image by encryption and should be able to retrieve the original when decrypted.
- Threshold an image in a given range of pixel values.
- Slice an image in given level of slicing.
- Produce a gray scaled, binary represented, or contrasted output for given image.
- Perform sharpening and embossing functions with the selected image.
- Flip, Mirror, Rotate and Zoom a selected image.

1.3 Application

The image processing tool set can be used to perform some special application function on all types of raster images mentioned above. It can be applied on images for analyzing them, processing, storing and displaying the processed images. It can be used to re-touch images, design web page images etc. The image processing tool set is a multi- purpose program ,intended to let users paint , draw, and process images.It is a raster based tool set. It can acquire, display, edit, enhance, analyze and animate images.

It supports many standard image-processing functions, including contrast enhancement, smoothing, sharpening, edge detection, median filtering, and steganography functions with user defined kernels.

Enhancing the details in an X-ray image for a better visualization, OCR (Optical Character Recognition) and constructing a 3D world from 2D images are examples of possible applications.

1.4 Organization Of Dissertation

The thesis is organized in such a manner to follow the waterfall model of software development life cycle.

The First chapter, the one, which you are reading is the Introduction.It gives a brief introduction about the objective of this project, its scope and the areas of application.

The Second chapter deals with the literature survey and background study needed for the development of the project.

The Third chapter gives an idea about the steps in general image analysis.

The Fourth chapter deals with the design issues, design and implementation of algorithms of the project.

The Fifth chapter is about how the project modules have been planned, pictorially.

The Sixth chapter shows the various interfaces designed for end users The Seventh chapter shows some of the snapshots of the outputs.

The seventh chapter concludes the dissertation followed by the references and appendix.

Chapter 2

LITERATURE SURVEY

2.1 Background

2.1.1 Image Processing [3]

An image is digitized to convert it to a form which can be stored in a computer's memory or on some form of storage media such as a hard disk or CD-ROM. This digitization procedure can be done by a scanner, or by a video camera connected to a frame grabber board in a computer. Once the image has been digitized, it can be operated upon by various image-processing operations.

Image processing operations can be roughly divided into three major categories, Image Compression, Image Enhancement, Restoration, and Measurement Extraction.
Image compression is familiar to most people. It involves reducing the amount of memory needed to store a digital image. Image defects, which could be caused by the digitization process or by faults in the imaging set-up (for example, bad lighting), can be corrected using Image Enhancement techniques. Once the image is in good condition, the Measurement Extraction operations can be used to obtain useful information from the image.

2.1.2 Still Images As A Processing Element [3]

Images can be powerful elements for communication, Particularly for media communications.one can easily express his ideas more conveniently through picturing his thoughts rather than verbal or text methods. The simplest form of images is the still image.

Still images are among the elements which can be changed in application with minimal efforts and expenses, if the contents are available. Still images are captured by digitizing and compressing multimedia device systems

Technically, these can be classified as follows.

2.2 Types Of Images

There are mainly two types of image file formats.

- Vector images
- Raster images

Apart from these two types, there is also a meta image format.

2.2.1 Vector Format: [5]

An image stored in vector format is defined mathematically, by lines, curves, circles etc, which are stored as mathematical formulas. This makes the file size very small. These images don't lose focus when they are zoomed, since the lines are re-rendered. With Illustrator and other vector-based applications you are working in a world of shapes. Vector images, also called object-oriented or draw images, are defined mathematically as a series of points joined by lines. Graphical elements in a vector file are called objects. Each object is a self-contained entity with properties such as color, shape, outline, size, and position on the screen, included in its definition. Since each object is a self-contained entity, you can move and change its properties over and over again while maintaining its original clarity and crispness, and without affecting other objects in the illustration. These characteristics make vector-based programs ideal for illustration and 3D modeling, where the design process often requires individual objects to be created and manipulated.

A vector image gives a good quality. However, it is not possible to store the photos, scanned images in this format. Examples of vector images are drawings, diagrams and illustrations.

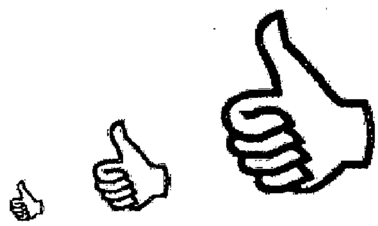


Figure.2.1 A Vector Image

2.2.2 Raster Format: [5]

A raster image is an 2-D or 3-D array of bytes. It is essentially a large, rectangular matrix or array of numbers, a certain number of units wide by a certain number of units high, that corresponds to the image's size. In other words, it as if a very fine grid was placed on top of the image dividing it into fantastically small pieces and allowing each piece to be individually defined. In so doing, the binary computer can simply define each element of the matrix or subspace of the image. Color information should be provided to view such images. With PhotoShop and other bitmap-based applications you are working in a world of color and photographic quality images.

Bitmap images, also called raster or paint images, are made of individual dots called pixels (picture elements) that are arranged and colored differently to form a pattern. When you zoom in, you can see the individual squares that make up the total image. However, the color and shape of a bitmap image appear continuous when viewed from a greater distance. Because each pixel is colored individually, you can easily work with photographs with 16,000 colors and can create photorealistic effects such as shadowing and intensifying color by manipulating select areas, one pixel at a time.
Increasing the size of a bitmap has the effect of increasing individual pixels, making lines and shapes appear jagged. Reducing the size of a bitmap also distorts the original image because pixels are removed to reduce the overall image size. Also, because a bitmap image is created as a collection of arranged pixels, its parts cannot be manipulated individually. Because you cannot easily change the size of bitmap images, the quality of your output is dependent on the decisions you make about resolution early in the process. Bitmap images look best when printed at their original size and proportions.

Drawings, photos, images, etc needed to be converted into digital format if they are to be used in a document, which is done by scanning the image and then storing as a raster file.

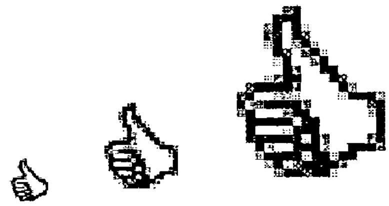


Figure.2.2 A Raster Image.

2.2.3 Meta Format :

An image in meta format is a combination of the two basic image formats-raster and vector.Such a format can include both raster and vector information.Photos are stored in raster format. In some cases, we may want to put describing numbers,text and arrows in these images.Both text and arrows should be saved as vector information but photos should be stored as raster format.

Examples of meta format are scanned images with text included in it.

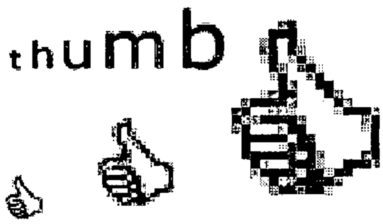


Figure.2.3 A Meta Image.

2.3 Types Of Raster File Formats

Bitmap files (BMP) :

Bitmap files have some confusion associated with them. Some refer to any pixel-based image as a bitmap file. However, a true bitmap image file refers to the standard Windows image format. This type of file is mostly used on DOS- and Windows-based machines. A bitmap file is a raster- (or pixel-) based format that only supports the RGB color space and bit depths of 1, 4, 8, or 24 bits per channel. These attributes make bitmap images unsuitable for use in a high-end print production workflow.

Graphics Interchange Format (GIF) :

The Graphics Interchange Format (GIF) is a compressed format developed to minimise file size and bandwidth. GIF files only require additional size when the colour of horizontally adjacent pixels changes. In other words, a horizontal line of a single colour takes the same amount of space no matter how long it is. GIF uses a lossless compression scheme, which is optimised for images with regions of solid colour. This makes GIF format ideal for solid colour images like logos, cartoons, and text stored as graphics.

Joint Photographic Experts Group (JPEG) :

The Joint Photographic Experts Group (JPEG) developed this format specifically for photographic images. JPEG files use a lossy compression technique (slightly adjusting the colours) to give a visually similar image that takes up less space. The scale of compression (and colour change) can be set during production. The JPEG format is preferred for photographs, textures, colour gradients, or images that require more than 256 colors.

Tagged Image File Format (TIFF) :

A Tagged Image File Format (TIFF) file is the most widely used file format in desktop publishing today. TIFF files can be compressed by using an LZW lossless compression approach or JPEG lossy compression. For high-end print production, it is the best practice to use either LZW compression or a very small amount of JPEG compression. The JPEG compression approach is a lossy compression that will degrade image quality when used in large amounts.

2.4 Image Representation and Resolution

An image is ideally represented by a function f(x,y) where x and y are real numbers that denote spatial coordinates and the value of f at any point (x,y) is the color or intensity at that point. In case of intensity the value is a single real number but in case of color the value is usually a 3-tuple e.g. [r,g,b]. It is possible to represent the ideal image by one or more surfaces given by z = f(x,y) , where z is interpreted as the surface corresponding to intensity or a component of color.

For digital processing, the ideal image experiences two important operations: Sampling and Quantization. The result is the digital image I(i,j) which is defined on a discrete coordinate system (i,j) where i and j are integers and the values are integers representing the quantization levels of intensity or components of color. A digital image of size MxN is represented by a matrix

$$\mathbf{I} = [I(i,j)] = \begin{bmatrix} I(0,0) & I(1,0) & \ldots\ldots\ldots & I(M,0) \\ I(0,1) & I(1,1) & \ldots\ldots\ldots & I(M,1) \\ . & & & \\ . & & & \\ . & & & \\ . & & & \\ I(0,N) & I(1,N) & \ldots\ldots\ldots & I(M,N) \end{bmatrix}$$

The elements of this digital array are called picture elements or simply pixels. The value of the pixels are non-negative scalars. For an intensity image the values are usually referred to as gray-levels or gray-values. Pixel (0,0) is located at top left corner of the image. Such an array is the input to the image processing systems.

Screen resolution can be termed in many ways. Some of these are explained below.

Spatial Resolution:

Measure of the smallest object that can be resolved by the sensor , For instance, a spatial resolution of 1 meter is coarser than a spatial resolution of 0.5 meter sometimes refers to the number of distinguishable lines (dots) per inch that a device can create.

dpi = dots per inch

Scanners typically n - 1200 dpi

Monitors typically 70-80 dpi

Printers typically 300 - 600 dpi

Spectral Resolution:

Refers to the specific wavelength intervals in electromagnetic spectrum that a sensor can record wide intervals in the electromagnetic spectrum are referred to as coarse spectral resolution, and narrow interval are referred to as fine spectral resolution.

Radiometric Resolution:

Refers to the dynamic range, or number of possible data file values. For instance, in 8-bit data, the data file values range from 0 to 255 for each pixel

Temporal Resolution:

Refers to how often a sensor obtains imagery of a particular area. For example, many satellite sensors can view the same area of the globe once every n days.One can also talk of resolution in terms of pixels, e.g. 640x480.

2.5 Why 'Raster' images as processing elements?

There are two main reasons why I took raster image format as a processing element than vector formats. First, bitmapped images are better than vectors at providing the photo-realism of an original scene. That's why digital images that are either created from scanned analog photographs or captured by digital cameras are stored as bitmapped images. Secondly, bitmapped images are supported by web browsers, whereas vector images are not.

Moreover, computer monitors and printers are raster devices. Monitors display images as a matrix of dots. To display a vector image on a monitor, the vectors must be "rasterized" into the required dot patterns (pixels) by hardware or software. To print images on a printer, a computer rasterizes vector images into a pattern of ink dots. Optical scanners and fax machines work by transforming text or pictures on paper into rasterized bitmapped images.

Raster images give continuous tone images like photographs. They can be used on the web which currently don't support vector formats. Only format that will show smooth gradients and subtle detail necessary in photographic images. They allow for color correction much easier then vector images.

Chapter 3

FUNDAMENTAL STEPS IN IMAGE ANALYSIS

Although the architecture of image analysis systems is quite flexible they usually fit into the following block diagram shown. [5]

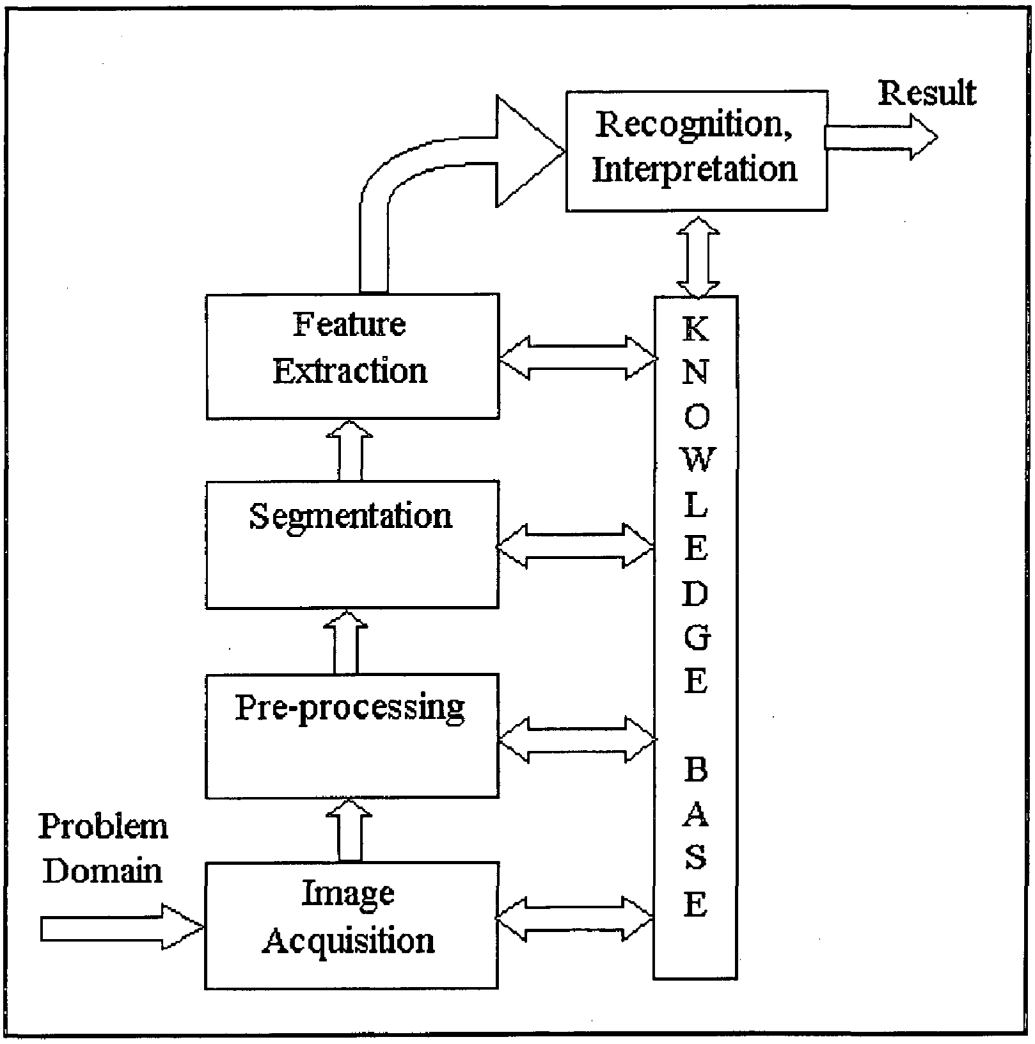


Figure.3.1 Steps In Image Analysis.

Image Acquisition:

In the design of an image analysis system we have to choose an acquisition method. The rest of the system depends on this crucial choice. There are three important parameters that should be decided:

1. Type of the image: One important parameter is whether to use a color image or a gray-level image. A color image is usually more difficult to process because a single number cannot represent a color and the perception of color by humans is somewhat complicated. In gray-level image the values usually correspond to intensity of light incident on a location on the image plane.
2. Number of colors or intensity levels: This corresponds to the number of quantization levels to be used. As the number of quantization levels increases the representation becomes more accurate and visually much more appealing but the storage space needed increases. An optimum number is chosen for the problem domain.
3. The resolution of the image: In sampling the 2D plane the density of points used determines the level of detail that can be discriminated. Increasing the resolution is desirable for the processing algorithms but the price paid is the excessive amount of space needed to store the images and longer processing time.

Pre-Processing:

The key function of pre-processing is to improve the image in ways that increases chances for the success of other processes. This part is completely application dependent and there are various alternatives. Enhancing contrast, removing noise, isolating regions are some examples.

Segmentation:

Segmentation partitions an input image into its constituent parts or objects. This is one of the most difficult problems in image processing. For example in an OCR application text lines, words and individual characters should be isolated.

Feature Extraction:

After segmentation the objects of interest, their locations are found and some quantitative features of the objects are calculated. These features are selected according to the purpose of the application. If a blood cell is detected in a microscopic imagery and if the purpose is to classify the detected cells, some features like the average color or intensity of the interior region of the cell or the shape of the nucleus may be used as a distinguishing feature.

Recognition Interpretation:

The purpose is to assign a label or meaning to the objects in the image. This stage is related to Pattern Recognition and Computer Vision areas.

Knowledge Base:

The processing the algorithms make use of our prior-knowledge about the problem domain. It enables us to narrow the search space in processing. If we know that the objects to be found are nearly circular we simply search for circular contours on the image. A priori knowledge can control all the other stages in processing by giving hints about the related features of the objects to be found in the image.

Chapter 4

DESIGN STUDY AND IMPLEMENTATION

4.1 Primitive Issues In Image Processing

Whatever the application is, a real-life image processing system addresses at least one of the primitive problems, namely acquisition, restoration, compression, enhancement and analysis.

These are the main steps in image processing.

Image Acquisition:

The processing starts with acquisition of a digital image. Possibly a color TV camera, a flat bed scanner or a fingerprint sensor. This stage is fully implemented in hardware but the other stages can be implemented in software. (Hardware implementation is used when speed-up especially real-time operation is needed) .The digital image is obtained by two important operations namely: Sampling and Quantization. In these operations the values of the ideal image function at a finite number of points is calculated and then they are truncated to the closest predetermined quantization levels. The digitized image is represented by one or more 2D Array (or Matrix) of integers in the software. The elements of the matrix are referred to as pixels.

Image Restoration:

The purpose of image restoration is removal of known degradations in an image. The source of image degradations is the image sensor itself or the environment. When the degradation can be modeled or expressed by a formula an inverse degradation can also be formulated. The restoration problem deals with estimating the parameters of the degradation process and approximating the inverse of the degradation process. Image restoration algorithms are mathematical and complex. The imaging system is usually modeled by a space varying linear system and additive noise. The purpose is to estimate the input to the linear system given the output i.e. the image we manipulate. A fundamental result in filtering theory used commonly for image restoration is called Wiener Filter.

Image Enhancement:

Image enhancement is the processing of images to improve their appearance to human viewers or to enhance other image processing modules' performance. The objective of image enhancement is dependent on the application context and criteria for enhancement is often subjective or too complex to be easily converted to useful objective measures.

Image enhancement is closely related to image restoration but they are in principle quite different things. In image enhancement there is no assumption on a known degradation, the objective is just to make the processed image better in some sense than the unprocessed image.

Image Analysis:

Image analysis is concerned with making quantitative measurements from an image to produce a description of it. The purpose of image analysis is automatic extraction of information from an image. It is closely related with a different branch called Computer Vision. In computer vision the purpose is extracting high-level information like the location of an object in an image or constructing a 3D world from a 2D-image sequence. All of these processes usually start with the extraction of low-level information like edges, contours or uniform regions from an image, which are the topics of image processing.

Image Compression:

The images are stored either for visual inspection or automated analysis. Even with moderate resolutions the storage requirements of images is usually very high. Moreover transmission of image data requires large capacity transmission channels, which are very expensive. Because of their wide applications, data compression is of great importance in image processing. We are already familiar with compressed image formats like JPEG and GIF. But it is also possible to develop compression algorithms specific to the application, as the standard schemes are still not satisfactory for all types of images. The research is going on and the solution to the problem is still open.

4.2 Design Analysis

It is possible to formulize image processing simply as 2D signal processing but especially perceptual aspects of images needs very different modeling schemes to be applied to the images. If the purpose is to come up with a visually appealing image from a low quality image, models of visual perception of contrast, spatial frequencies and color should be considered. In image analysis (e.g. extracting the boundary of an object in the scene) some local structural properties of the 2D signal should be evaluated. In image compression it is possible to describe an image as a member of an ensemble and apply some statistical models. There is no unique theory that satisfies the needs of all possible applications, for that reason processing techniques from very different disciplines are applied to the image-processing problem.

The design of the project is divided into five main modules:

- Image view
- Enhance
- Filtering
- Edge detection
- Application

The design charts will be discussed in chapter 6.

4.2.1 Image View

It has mainly the functions mentioned below:

- Rotation
- Flip
- Mirror
- Zooming

An image operator, or transform, is a function which takes an image for its input and produces an image for its output. The domain of a transform operator is often called the spatial or space domain. The range of the operator is often called the transform domain. The transforms implemented here are rotation, scaling, flip, mirror and zooming of images.

Translation:

Translation is a simple transformation that is calculated directly from the conversion matrix for two frames, one a translate of the other. The translation matrix is most frequently applied to all points of an object in a local coordinate system resulting in an action that moves the object within this system.

Rotation:

Rotations are complex transformations. The primary complexity is that in three-dimensions, rotations are performed about an axis - usually specified by a point and a vector direction. The general idea is to develop this transformation about the three coordinate axes of a frame, and then generalize this to be able to rotate about a general axis in the frame. The general idea for the matrix comes from rotation about a point in two-dimensions.

Mirroring and **Flip** are types of rotating algorithms.

Zooming:

Zooming , like translation is a simple transformation which just scales the coordinates of an object. It is specified either by working directly with the local coordinates, or by expressing the coordinates in terms of frames.

4.2.2 Filters

This module has four types of filter functions:

- High pass filter
- Low pass filter
- Median pass filter
- User defined filter

High-pass filter:

A high-pass filter is a linear spatial filter which attenuates the low spatial frequencies of an image and accentuates the high spatial frequencies of an image. It is typically used to enhance details, edges and lines. High-pass filtering is a linear filtering operation and is the opposite of low-pass filtering. As it can be easily guessed it is used to emphasize the high- frequency content of the image. Enhancing the details in image is often visually desirable. Sharpening is an example of high-pass filtering.

Low-pass filter:

A low-pass filter is a linear spatial filter which attenuates the high spatial frequencies of an image and accentuates the low spatial frequencies of an image. It is typically used to suppress small undesired details, eliminate noise, enhance coarse image features or smooth the image. Low-pass filtering is a linear filtering operation and is related to the frequency content of the signal. Intuitively, frequency content of a signal can be guessed by observing the rate of change of pixel values in the image. Rapid change of gray-level values imply high frequency content and almost uniform gray-level values imply low frequency content. Low-pass filtering reduces the high frequency components of the images. Smoothing and Blurring operations are low-pass filtering operations.

Median filter:

This filter does the same operation as the high pass and low pass but the range of operation falls between the LPF and HPF.

User defined filter performs the filtering as per the user's requirement.

4.2.3 Applications

This module contains three functions:

- Steganography
- Noise Addition and Removal
- Comparison

Steganography:

Steganography, the art of hiding messages inside other messages, is gaining new popularity with the current industry demands for digital watermarking and fingerprinting of audio and video. It can be used to secretly transmit messages without the fact of the transmission being discovered. Often, using encryption might identify the sender or receiver as somebody with something to hide.

Digital images are typically stored in either 24-bit or 8-bit per pixel files. 24-bit images are sometimes known as true color images. Obviously, a 24-bit image provides more space for hiding information; however, 24-bit images are generally large and not that common. A 24-bit image 1024 pixels wide by 768 pixels high would have a size in excess of 2 megabytes. As such large files would attract attention were they to be transmitted across a network or the Internet, image compression is desirable.

Noise Addition and Removal:

A particular amount of noise can be added to the selected image, as defined by the user, in order to serve analysis and study of the effect of noise in a particular image in a given background. This will help in improving the image quality in future when it is used in other applications.

Comparison:

The given image will be compared with the second image selected by the user and the result will indicate whether the two images are matching or not. This is done by comparing the image pixel by pixel and by evaluating the value of each pixel in the two images.

4.2.4 Enhance Functions

The following are the functions for image enhancing:

- Brightness and RGB adjustments
- Arithmetic Operations
- Binary Operation
- Grayscale
- Negation
- Slicing
- Thresholding
- Histogram
- Bit removal
- Bit extraction
- Diffusion
- Embossing
- Sharpening

Contrast:

Contrast function is used to produce an image which will be displayed in colors which are contrast of the original colors. for example, the contrast of white is black , and so on.

Brightness and RGB Adjustments:

This implies that the brightness of an image can be increased or decreased as per the user's choice. There are two types of adjustments in brightness. One is the regular adjustment in which all the color pixels are brightened or darkened accordingly. The other is adjusting only the brightness of a particular color like red, green or blue.
By mixing red, blue and green lights in different proportions it is possible to obtain a wide range of colors.

Thresholding:

Thresholding is an image point operation, which produces a binary image from a gray-scale image. Level slicing or density slicing is a point operation, which employs two thresholds and produces a binary image. Thresholding can also be implemented by giving a range of color threshold level, within which the image will be displayed.

Slicing:

Slicing an image breaks a large picture into many small pictures. This decreases the file size and helps the page load faster. A graphic design program, such as image ready, will allow slicing the large picture into many small pictures. In addition, the small pieces of the large image can then be linked without going through the image mapping procedure.

An Image Slicer allows you to cut an image into smaller pieces to assist with the fast loading of large images, for applications such as on the Internet. The biggest benefit is probably the fact that you can be looking at each slice as it loads, rather than waiting for one large image to display.

Diffusion:

This is a technique in which the image pixels are spreaded or diffused in larger area so as to give a blurred appearance.An image which is diffused will not be as clear to the viewer because of the jagged appearance produced during the process.This can be used to analyse an image during blurring.

Thresholding:

The key and most difficult step in most image analysis procedures is the thresholding or segmentation step used to separate the image into features and background. This process is usually performed manually, which leaves plenty of room for operator differences and inconsistency, but allows the subtle power of human vision to decide what 'should' be included as features to be measured. Having the image histogram displayed to show the threshold settings in relationship to the brightness values, and the showing the selected pixels on the image immediately to show the effect of changes in settings, are both vital tools for the human while adjusting the settings. In image processing, we have to convert edge and texture information to gray scale so that thresholding on characteristics other than simple brightness or density can be used.

Negation:

It means that when this function is applied, the negative of the given image will be displayed.The negation is performed by finding out the values of the pixels and then calculating the exactly opposite valued pixel(negated pixel value) of each element. Thus, negation of the image will produce an image in black and white pixels.

Gray-scaling:

Gray-scaling is a method in which the given image is converted into a gray scale of given specification. This can be done using a look up table. Gray-scaling is the process of converting a continuous-tone image to an image that a computer can manipulate.
While gray scaling is an improvement over monochrome, it requires larger amounts of memory because each dot is represented by from 4 to 8 bits. At a resolution of 300 dpi, you would need more than 8 megabytes to represent a single 8½ by 11-inch page using 256 shades of gray. This can be reduced considerably through data compression techniques, but gray scaling still requires a great deal of memory.
Using a lookup table that runs from 255 to 0 would invert the image (swapping light and dark regions). Gray-scale images are preferred because the shades change very gradually between palette entries. This increases the image's ability to hide information.

Embossing:

An image appears to be embossed when highlights and shadows replace light/dark boundaries and low contrast areas are set to a gray background. For producing three dimensional effects this filter can be used. This filter simulates the effect of light hitting a bumpy surface and has a variety of uses for embossing, carving and producing rough textures. This filter is more complicated than most and has a large number of settings. It also takes up to three images as inputs.

These are:

- The image to be processed
- A bump map
- An environment map

The bump map is a grayscale image, which gives the height of the simulated surface , white being high and black low. The environment map is used for reflections in shiny surfaces. For simple cases, we can ignore the extra images and simply use the input image as the bump map.

Binary Operation:

Operations based on binary (Boolean) arithmetic form the basis for a powerful set of tools that will be described. The operations are point operations and thus admit a variety of efficient implementations including simple look-up tables. These binary operations can be performed on the images chosen by the user.

Arithmetic Based Operations:

Various operations like Addition, Subtraction, Multiplication, Division, AND and OR can be performed on the images just like how the operations are performed with numbers. This will be done by taking the binary values of the images.

4.2.5 Edge Detection

An edge detector is a neighborhood operation which determines the extent to which each pixel's neighborhood can be partitioned by a simple arc passing through the pixel where pixels in the neighborhood on one side of the arc have one predominant value and pixels in the neighborhood on the other side of the arc have a different predominant value. There are four classes of edge operators: gradient operators, Laplacian operators, zero-crossing operators and morphologic edge operators.

The following edge detection algorithms are implemented in this project.

- Gaussian Edge Detection
- Sobel Edge Detection
- Prewitt Edge Detection
- Laplace Edge Detection
- Enhance Edge Detection
- Kirch Edge Detection
- Quick Edge Detection
- Robet Edge Detection
- Homogenity Edge Detection
- Faler Edge Detection

4.3 Implementation Details

4.3.1 Image View

It has mainly the functions mentioned below.

- Rotation
- Flip
- Mirror
- Zooming

These are basically fall under one function known as transformation. A transformation can be a rotation, scaling or a simple translation also. Zooming is a simple scaling method.

Rotation:

1. Create an image.
2. Open the image file you want to rotate.
3. Determine the size of the image in X and Y dimensions.
4. Compare the dimensions to determine which one is larger.
5. Use the larger of the X and Y dimensions to set new image dimensions as a larger

square.

6. Create a new image and set its dimensions to the rotated dimensions of the original image. In a 90 degrees rotation example, if the original image is 640 x 480, then the new image will be 480 x 640.
7. Rotate the previous image in the enlarged square.
8. Copy the enlarged image into the new image that you set to rotated dimensions.

The result is a rotated image and frame. The new rotated frame belongs to a new image that contains a copy of the rotated data.

Mirroring and **Flip** functions are types of rotation.

Zooming:

The matrix looks like

$$\begin{vmatrix} x' \\ Y' \\ Z' \\ 1 \end{vmatrix} = \begin{vmatrix} sx & 0 & 0 & 0 \\ 0 & sy & 0 & 0 \\ 0 & 0 & sz & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} x \\ y \\ z \\ 1 \end{vmatrix}$$

where P= (x y z) and x',y', z' are scaling parameters.

Now the original image can be preserved by uniform scaling: sx=sy=sz.

Steps are as follows:

1.Translate the fixed point to origin.

2.Scale the object relative to the coordinate origin using the matrix.

3.Translate the fixed point back to its original position.

4.3.2 Filters

- High pass filter
- Low pass filter
- Band pass filter
- User defined filter

Procedure for any order filter (Nth order filter)

The Nth order filter is an extension of the median filter. It assigns to each pixel the Nth value of its neighborhood (when sorted in increasing order). The value N specifies the order of the filter, which you can use to moderate the effect of the filter on the overall light intensity of the image. A lower order corresponds to a darker transformed image; a higher order corresponds to a brighter transformed image.
Each pixel is assigned the Nth value of its neighborhood, N being specified by the user.

$P_{(i, j)}$ = Nth value in the series [$P_{(n, m)}$]
where the $P_{(n, m)}$ are sorted in increasing order.

The following example uses a 3 × 3 neighborhood:

$P_{(I-1, j-1)}$	$P_{(i, j-1)}$	$P_{(i+1, j-1)}$
$P_{(i-1, j)}$	$P_{(i, j)}$	$P_{(i-1, j)}$
$P_{(I-1, j+1)}$	$P_{(i, j+1)}$	$P_{(i+1, j+1)}$

This becomes

13	10	9
12	4	8
5	5	6

The following table shows the new output value of the central pixel for each Nth order value:

Nth Order	0	1	2	3	4	5	6	7	8
New Pixe Value	4	5	5	6	8	9	10	12	13

Figure 4.1 Pixel for Nth Order Value

Notice that for a given filter size f, the Nth order can rank from 0 to $f^2 - 1$. For example, in the case of a filter size 3, the Nth order ranges from 0 to 8 ($3^2 - 1$).
A 3x3 median filter would just be the 4th order filter, with N given by $(f^2-1)/2$.

Low order filters(LPF) tend to erode bright regions and dilate dark regions; high order filters(HPF) have the opposite effect. Median filters remove isolated pixels, whether they are bright or dark.

4.3.3 Applications

- Steganography
- Noise addition and removal
- Comparison

Steganography:

An 8-bit color images can be used to hide information. In 8-bit color images, (such as GIF files), each pixel is represented as a single byte. Each pixel merely points to a color index table, or palette, with 256 possible colors. The pixel's value, then, is between 0 and 255. The image software merely needs to paint the indicated color on the screen at the selected pixel position.
When dealing with 8-bit images, the steganographer will need to consider the image as well as the palette. Obviously, an image with large areas of solid color is a poor choice, as variances created by embedded data might be noticeable. Once a suitable cover image has been selected, an image encoding technique needs to be chosen.

The use of digital images for steganography makes use of the weaknesses in the human visual system, which has a low sensitivity in random pattern changes and luminance. The human eye is incapable of discerning small changes in color or patterns and because of this weakness text or graphic files can be inserted into the carrier image without being detected. In a 24 bit graphic file each pixel would be represented by 3 bytes, each being 8 bits long.

A white pixel would look like this:

red byte green byte blue byte

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

An 8-bit color image could hold around 300 kilobits of hidden data and a 24-bit color image around 2 megabytes. Other factors will also influence the type of image that would be used as the carrier file. Items such as compression type and color variance will need to be considered. The more gray scale that the hidden image has the easier it is to hide. This is because of the low sensitivity to random pattern changes of less than one part in 30.This technique may be performed using a simple exor function or even a complex function.

In steganography, simple ex-or function is used to hide an image into another. The steps are:

1. Choose the first image.
2. Choose the second image to hide.
3. Ex-or each pixel of first image with that of the second.
4. Give the output image.

Noise Addition:

Steps in noise addition:

1. Choose the image to which noise has to be added.
2. To each pixel, add certain value.
3. Change the pixel value and give the output image with noise.

Noise removal is reverse function of the above.

4.3.4 Enhance Functions

- Brightness and RGB adjustments
- Arithmetic
- Binary
- Grayscale
- Negation
- Slicing
- Thresholding
- Histogram
- Bit removal
- Bit extraction
- Diffusion
- Embossing
- Sharpening

The Brightness, RGB, binary, contrast , negation and gray-scaling functions are based on common implementation procedure except for some changes in the algorithms.

Gray scaling:

The conversion of an image to grayscale is achieved by setting each of the 'R', 'G' and 'B' components of every pixel value to the arithmetic mean of the three.

Steps in gray-scaling are:

1. Store the grayscale image.
2. Buffers are defined, which are one-dimensional arrays of bytes, and are populated with the pixel values of the original and new Bitmap objects .
3. Then for each pixel in the image, the 'R', 'G' and 'B' components are retrieved.
4. The first byte of the four bytes for per pixel gives the 'B' component, the second byte, the 'G' component, the third, the 'R' component and the fourth one, the 'A' or the alpha value.
5. The grayscale value for the 'R', 'G' and 'B' components thus obtained.

6. The corresponding bytes of the buffer for the new Bitmap object are set to this value
7. After the operation has been performed for all the pixels in the original image.

Binary Image:

1. Store the bitmap image.
2. Copy the bitmap Image.
3. A one-dimensional array of bytes is defined and is populated with the pixel values of the A two-dimensional array is defined to hold the binary pixel values.
4. Another Bitmap object is created.
5. For each pixel in the image, the 'R', 'G' and 'B' components for the pixel in i^{th} row and j^{th} column, both rows and columns indexed starting at '0' are retrieved.
6. If the 'r', 'g' and 'b' values are all equal to '0', then the corresponding element in the two-dimensional array is set to a value '1', otherwise to a value '0'.
7. The pixel values in the buffer populated by the pixel values of the new Bitmap object are initialized.

After the process has been repeated for all the pixels in the original Bitmap, we get the binary image in the form of the new Bitmap object and the array with '0's representing the background and '1's representing text.

Arithmetic Operations:

Operations based on binary (Boolean) arithmetic form the basis for a powerful set of tools like contrasting, gray-scaling, binary, negation etc. The operations described below are point operations and thus admit a variety of efficient implementations including simple look-up tables.

The implication is that each operation is applied on a pixel-by-pixel basis.

NOT a	
0	1
1	0

AND a	b 0 1
0	0 0
1	0 1

XOR a	b 0 1
0	0 1
1	1 0

SUB a	b 0 1
0	0 0
1	1 0

Table 4.2 Arithmetic Operations Truth Table.

The gray-value point operations that form the basis for image processing are based on ordinary mathematics and include:

Operation	Definition	Preferred data typ
ADD	c = a + b	integer
SUB	c = a - b	integer
MUL	c = a * b	integer or floating poin
DIV	c = a / b	floating poin

Table 4.3 Look Up Table for Arithmetic Operation

Bit Removal:

1. Store the bitmap image.
2. Copy the bitmap Image.
3. A one-dimensional array of bytes is defined and is populated with the pixel values of the A two-dimensional array is defined to hold the binary pixel values.
4. Another Bitmap object is created.

5. For each pixel in the image, the 'R', 'G' and 'B' components for the pixel in i^{th} row and j^{th} column.
6. The bits are removed according to the bit value stored in a buffer.
7. The pixel values in the buffer are then manipulated and the new image is created as per the stored values.

Bit Extraction is same as the above except for the extraction of bits instead of removing the correponding bits.

Diffusion function is done by shifting the bits by a certain predefined value.

4.3.5 Edge Detection Functions

- Gaussian Edge Detection
- Sobel Edge Detection
- Prewitt Edge Detection
- Laplace Edge Detection
- Enhance Edge Detection
- Kirch Edge Detection
- Quick Edge Detection
- Robet Edge Detection
- Homogenity Edge Detection
- Faler Edge Detection

The implementation procedure for the most commonly used detection method, the Sobel's and Prewitt's edge detection is as follows.

Prewit Filter (3x3)

The Prewit filter is very similar to Sobel filter. It performs a smoothing using a convolution mask defined by :

1		1

and a derivative convolution mask :

-1	0	1

The total convolution mask used to detect gradient in the X direction is :

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

While the convolution mask used to detect gradient in the Y direction is :

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

Prewit filter is a fast method for edge detection. But it was vulnerable to noise. It difference with respect to Sobel filter is the spectral response. It is only suitable for well-contrasted noiseless images.

Sobel Filter (3x3)

Sobel filter is a simple approximation to the concept of gradient with smoothing. The smoothing in direction X is done by a convolution mask :

1	2	1

The derivative filter is performed by the convolution mask :

-1	0	1

So the combined convolution mask to detect gradients in X direction is :

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

And in the Y direction is :

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

This method has the advantage of short computation time. Nevertheless it is strongly vulnerable to noise and use to produce disconnected contours. It is implicitly tuned to detect edges of three pixels wide (the transition zone). When original images are well contrasted and noiseless, this is the best adapted method to edge detection.

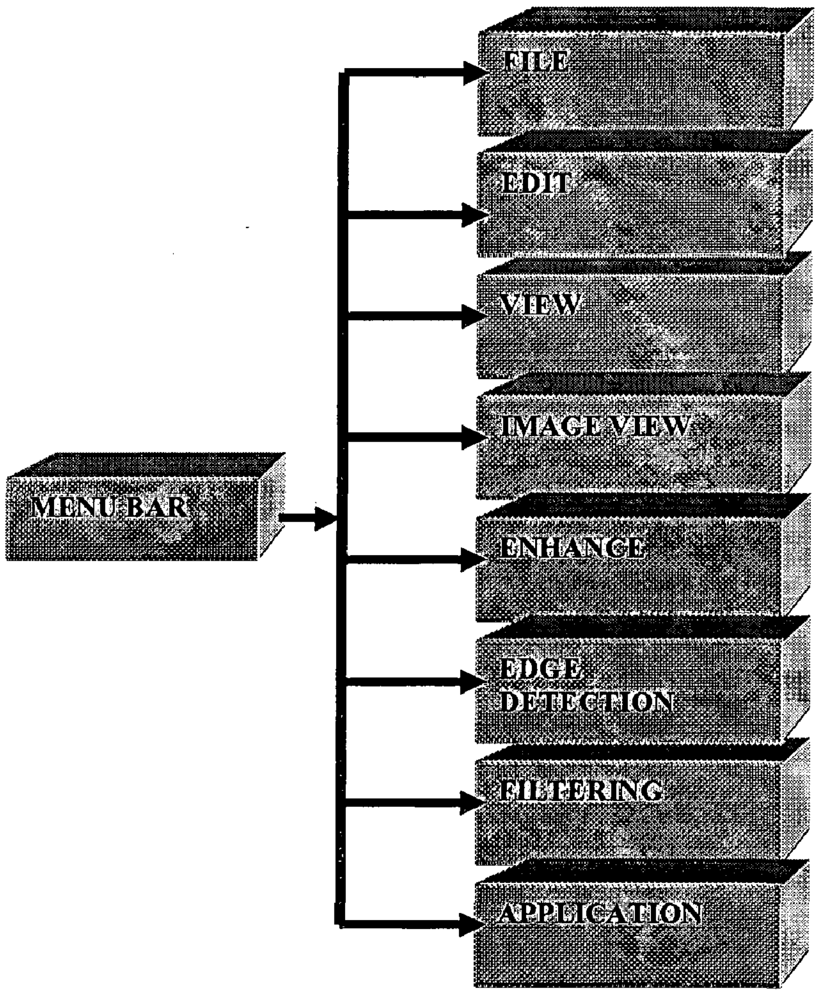
The gradient module is computed from the above results.

Smoothed image is obtained by convolving noised image with the mask:

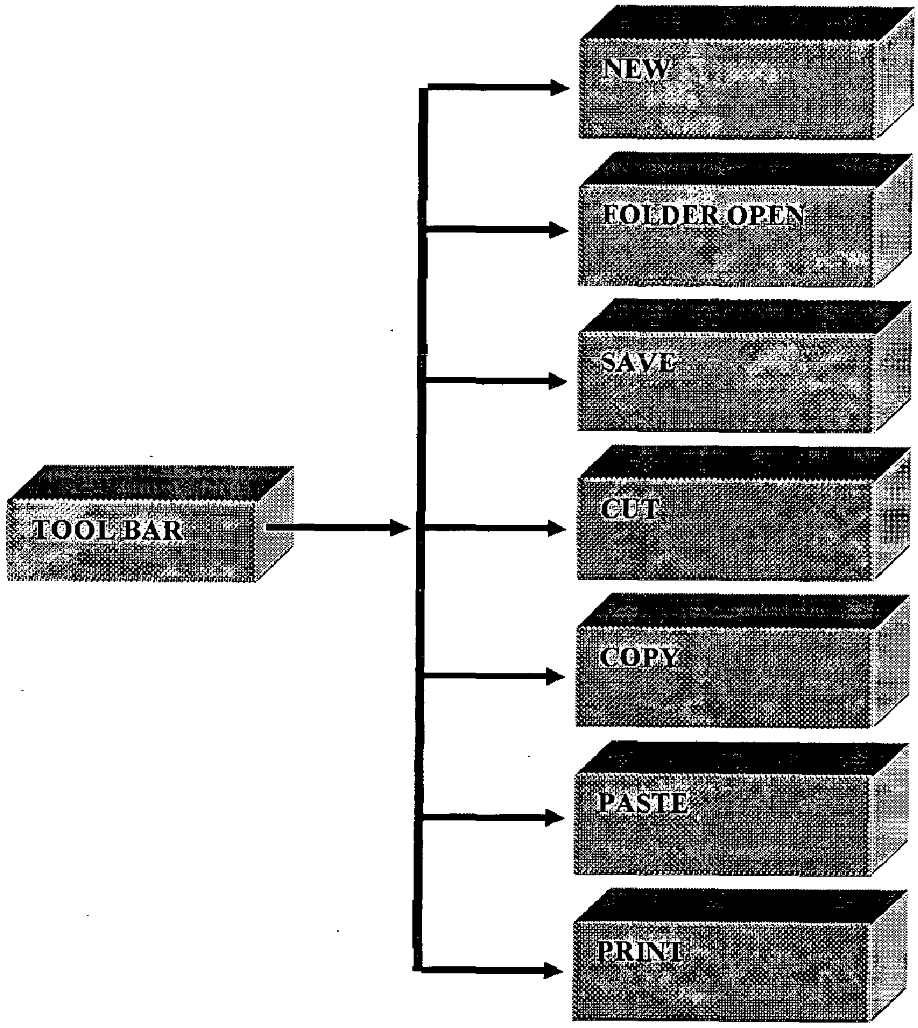
1	2	1
2	4	2
1	2	1

DESIGN CHARTS

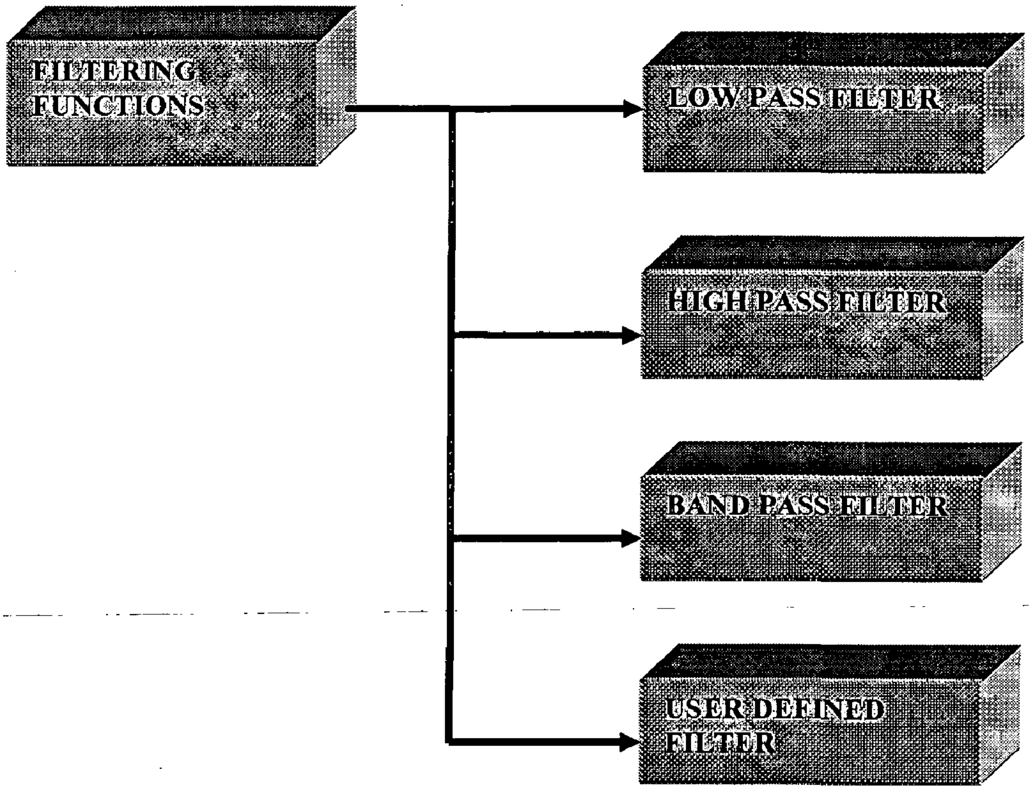
5.1 Diagram Showing Menu Functions.



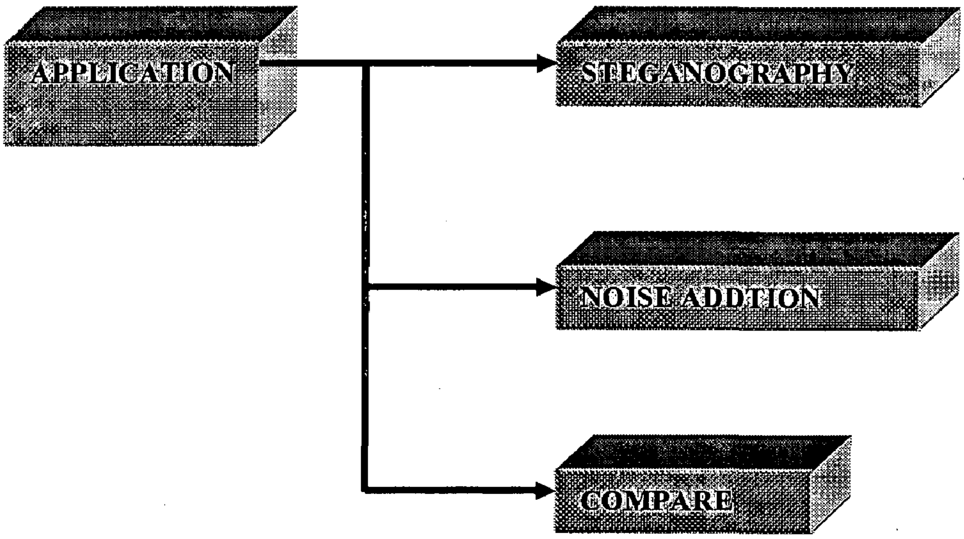
5.2 Diagram Showing Tool Bar Functions.



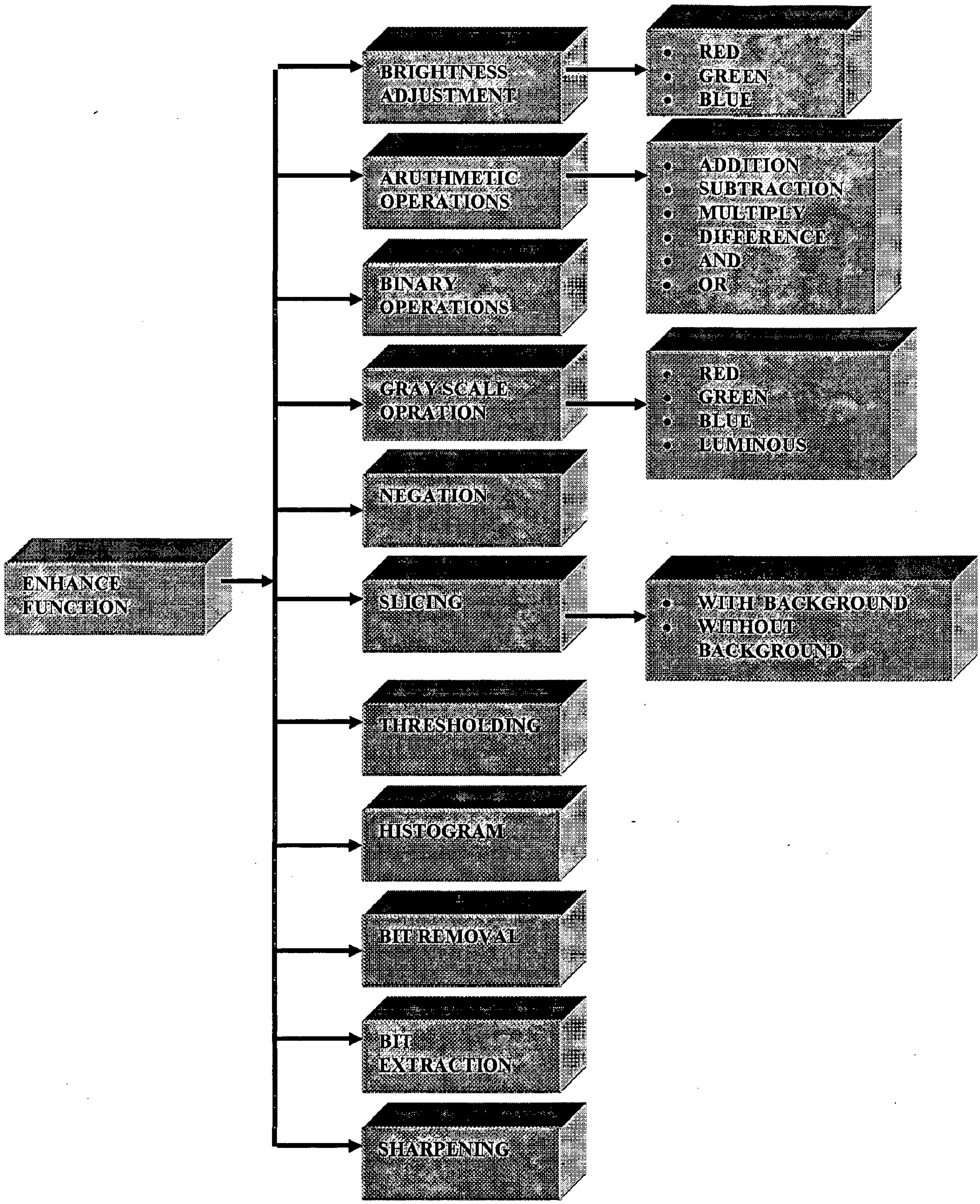
5.3 Diagram Showing Filtering Functions.



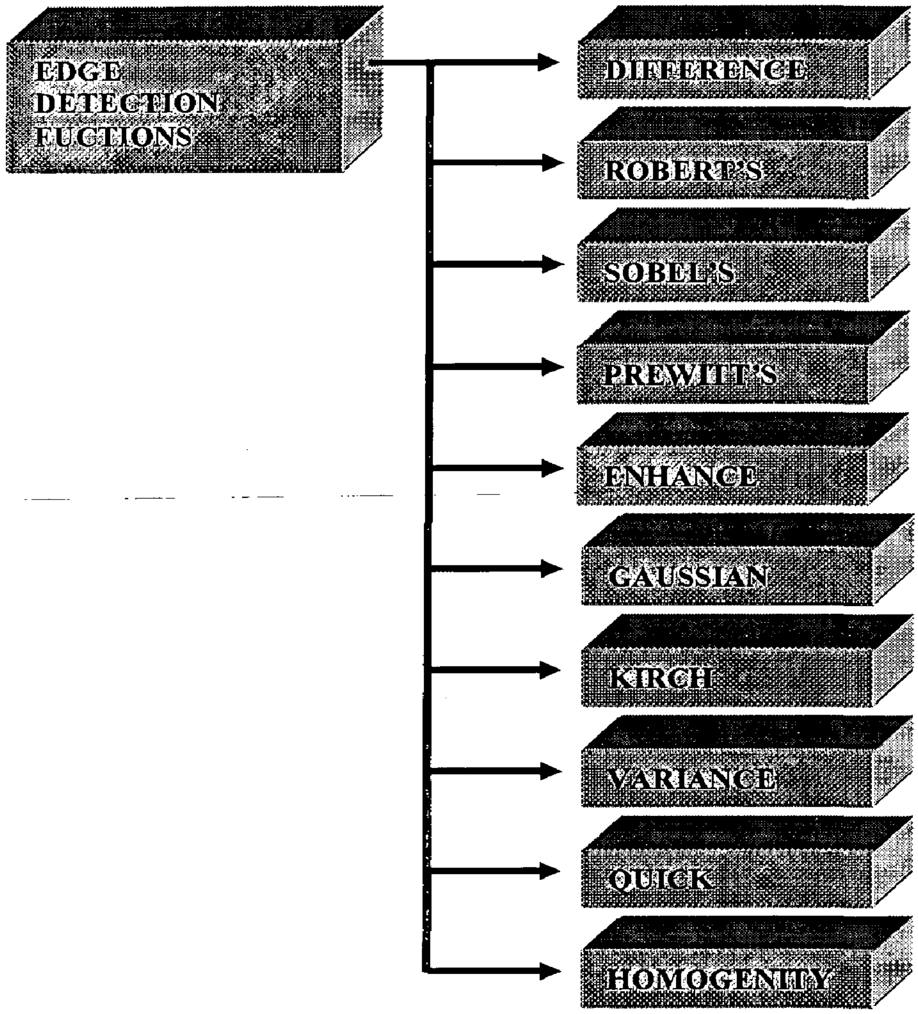
5.4 Diagram Showing Application Functions.



5.5 Diagram Showing Enhance Functions.



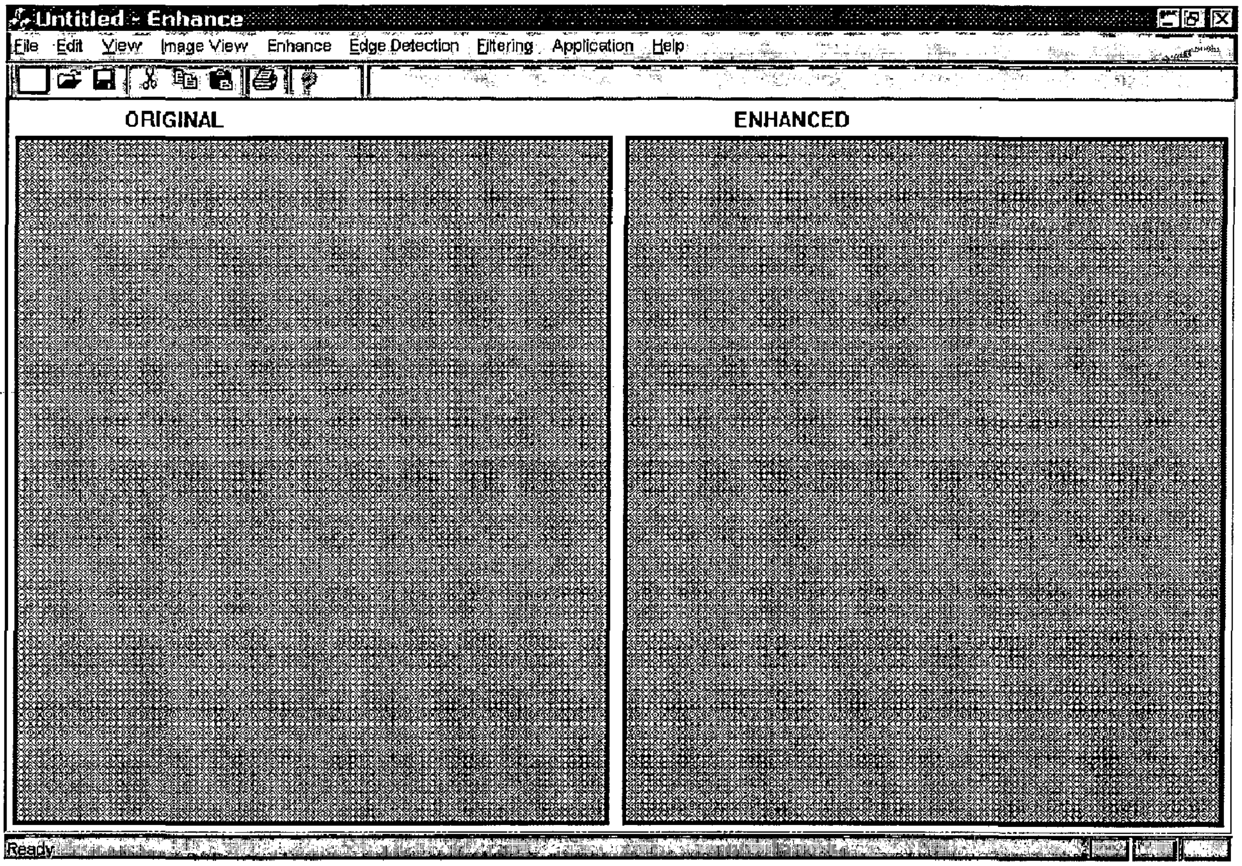
5.6 Diagram Showing Edge Detection Functions.



Chapter 6

INTERFACES

6.1 Opening Window

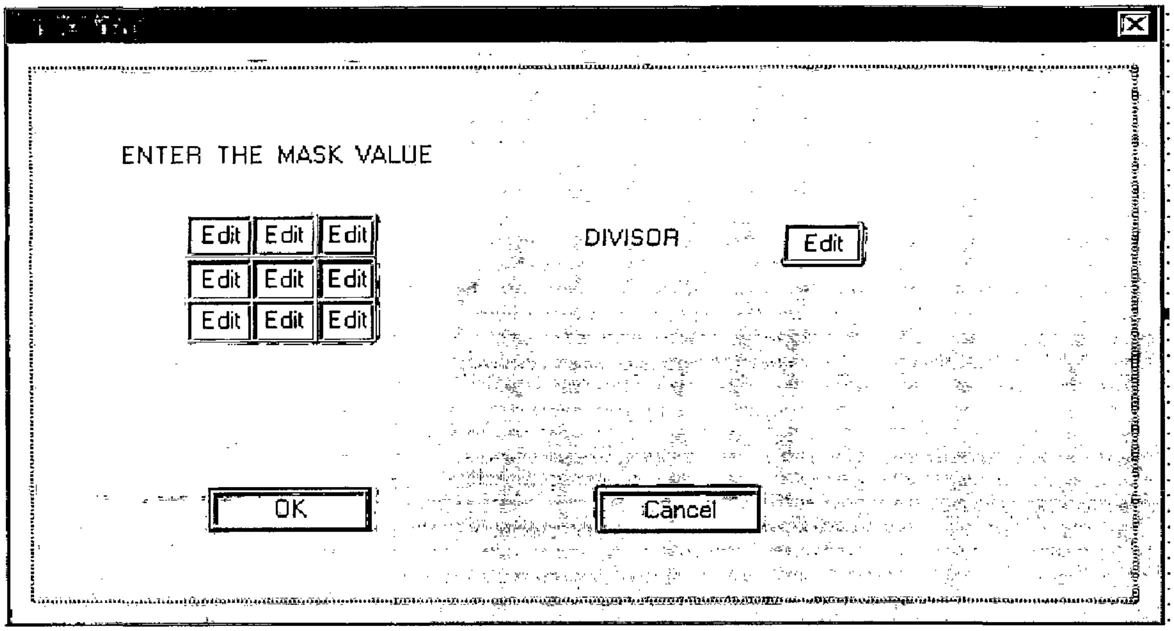


Description: This is the opening window of the image processing tool set and it contains a menu bar which has all functions implemented in it.

Orientation: As soon as the program is executed, this window appears automatically.

Action: This window shows the chosen image in the left hand side and the processed image at the right hand side.

6.2 Filtering

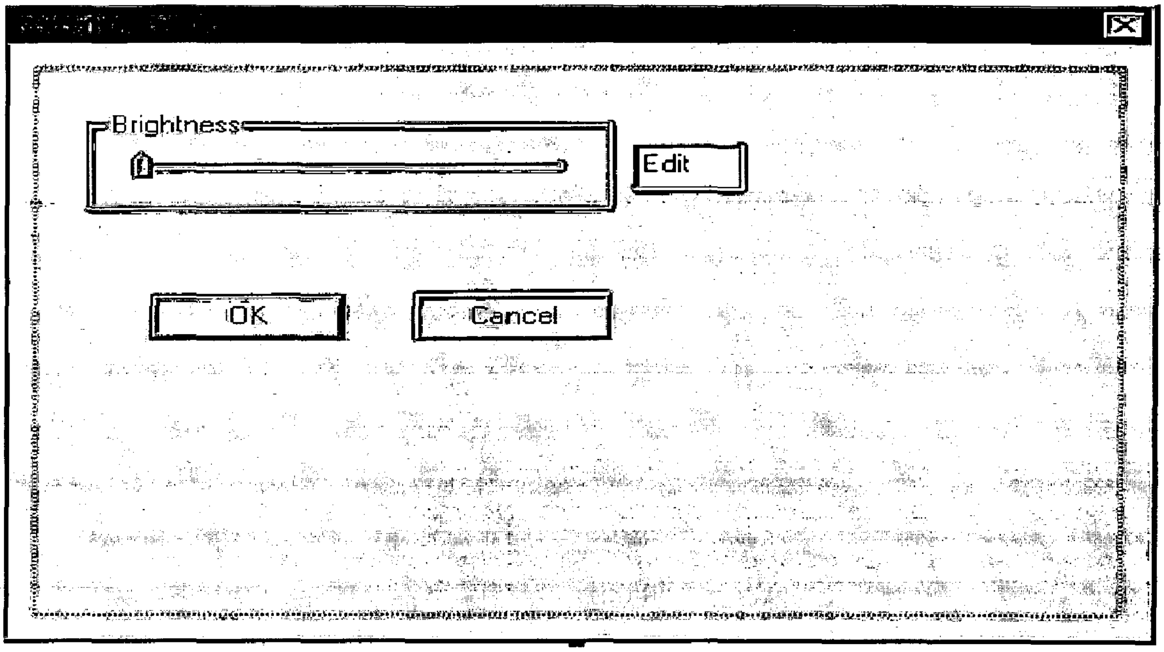


Description: This editor allows you to apply user defined digital image filtering to image data sets, such as smoothing, unsharp masking, or morphological operations. Filters directly operate in 3D and also 2-D images of raster type format.

Orientation: This screen appears only if the user defined filter is selected from the filter menu.

Action: Removes noise as defined by the user.

6.3 Brightness

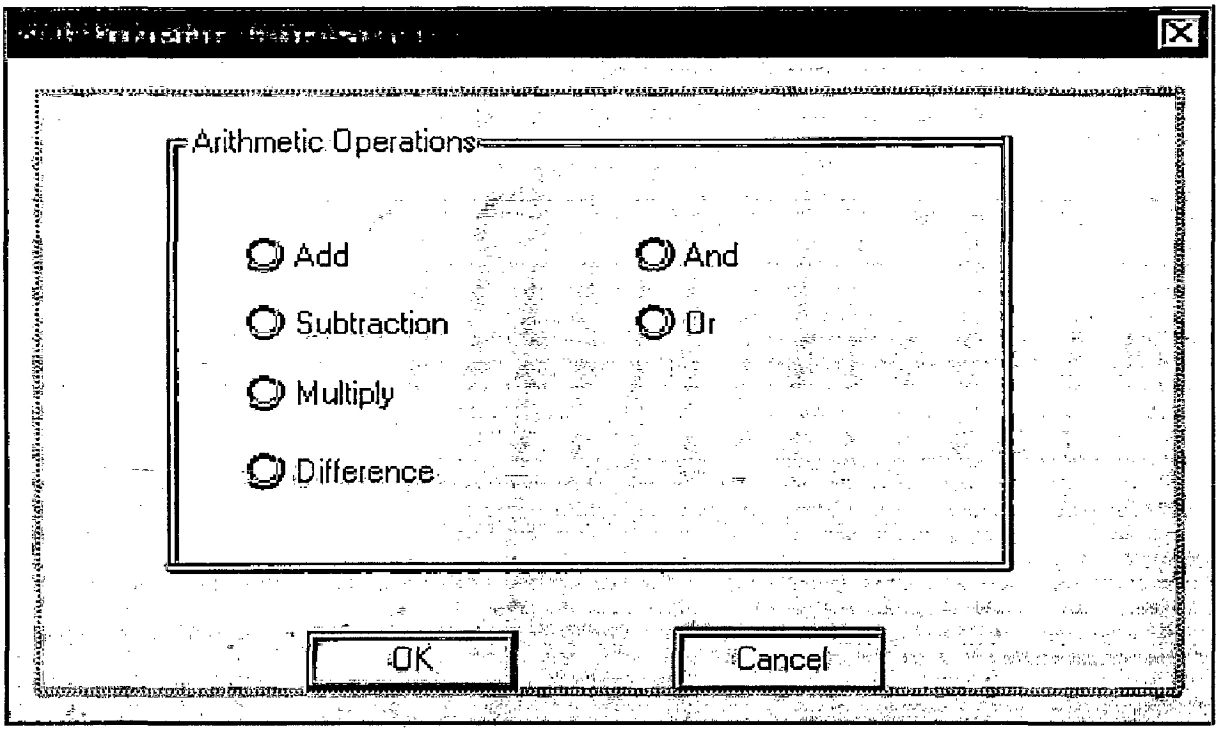


Description: This screen allows the user to adjust the brightness of an image.there is a sliding rule that allows to adjust the brightness of the image by moving the mouse through it.there is also a provision to enter the value of brightness in numbers.once 'OK' button is pressed, the brightness will be adjusted.

Orientation: This screen appears by selecting menus-enhance- brightness.

Action: Increases/decreases image brightness as defined.

6.4 Arithmetic Operations

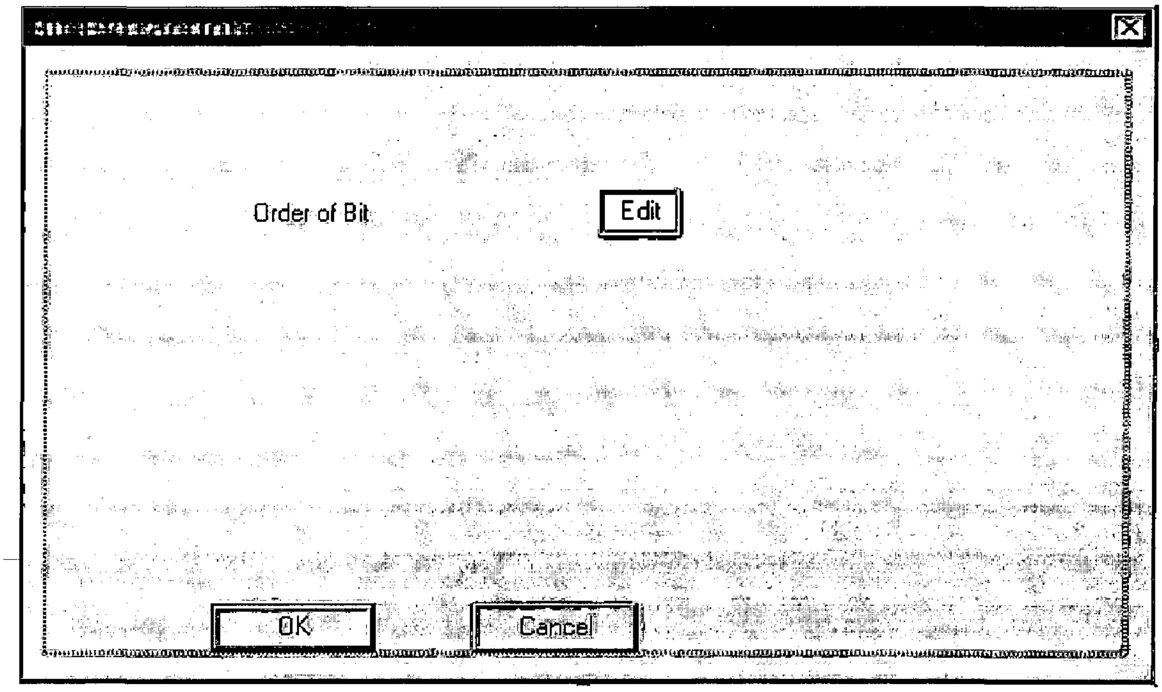


Description: This editor allows user to perform arithmetic operations just as it can be performed with numbers. The main difference is that these operations are performed on the image with itself or any other image selected by the user as second image.

Orientation: This screen appears when the menus-enhance-arithmetic operations.

Action: Performs addition, subtraction, multiplication, difference, OR, AND with the images selected.

6.5 Bit Extraction

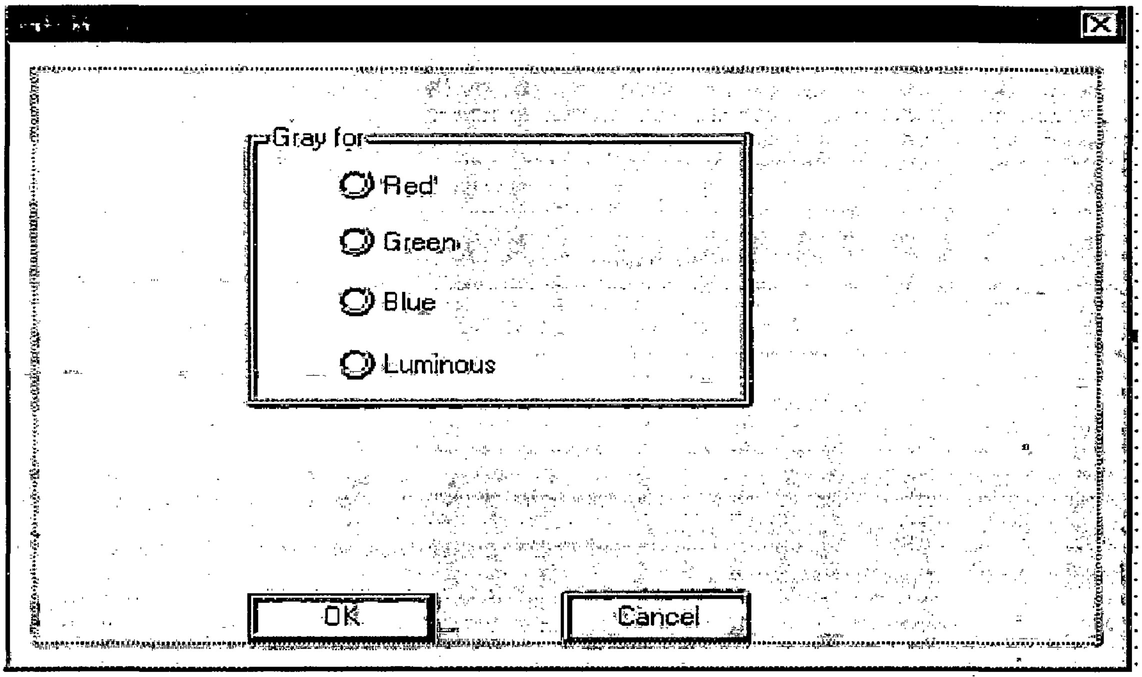


Description: This window is designed to get the number of bits to be extracted as input and gives the extracted image according to the number of bits mentioned by the user.

Orientation: This window appears when menu-enhance-bit extraction is selected.

Action: Extracts certain bits of the image as per the level selected by the user.

6.6 Gray Scaling

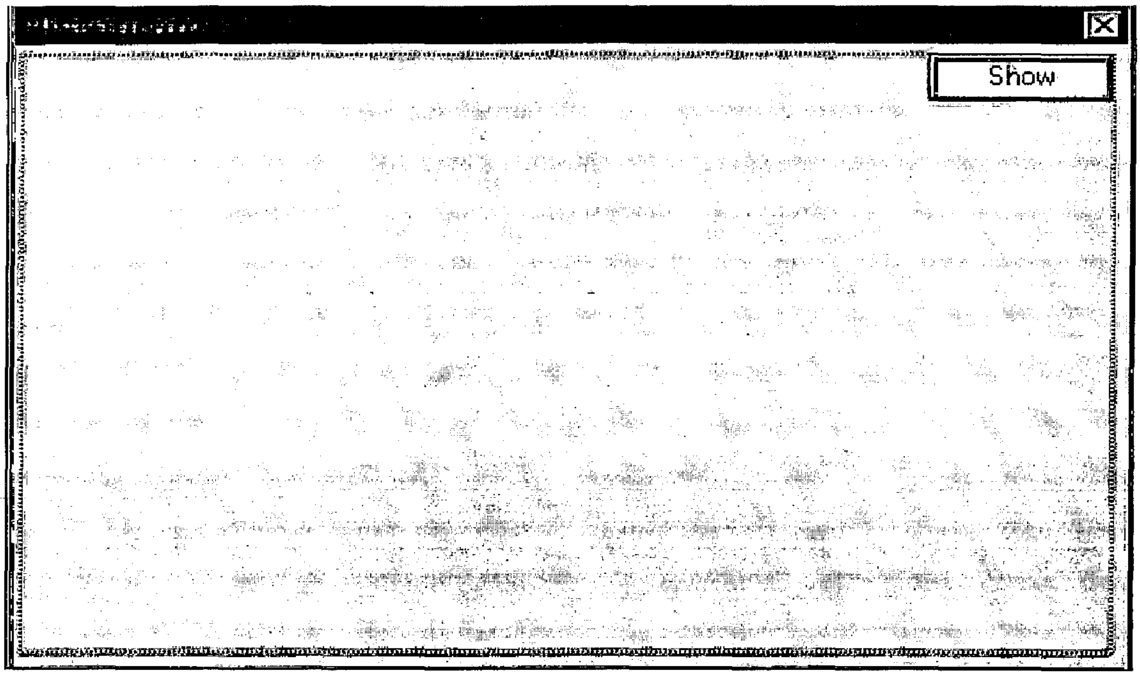


Description: The use of many shades of gray to represent an image is called gray scaling. Continuous-tone images, such as black-and-white photographs, use an almost unlimited number of shades of gray.any image given as input will be processed and the same image in various shades of gray will be displayed.

Orientation: This window appears when menus-enhance-grayscale is selected.

Action: It gives the gray scaling with respect to red, green, blue colors for the image.

6.7 Histogram

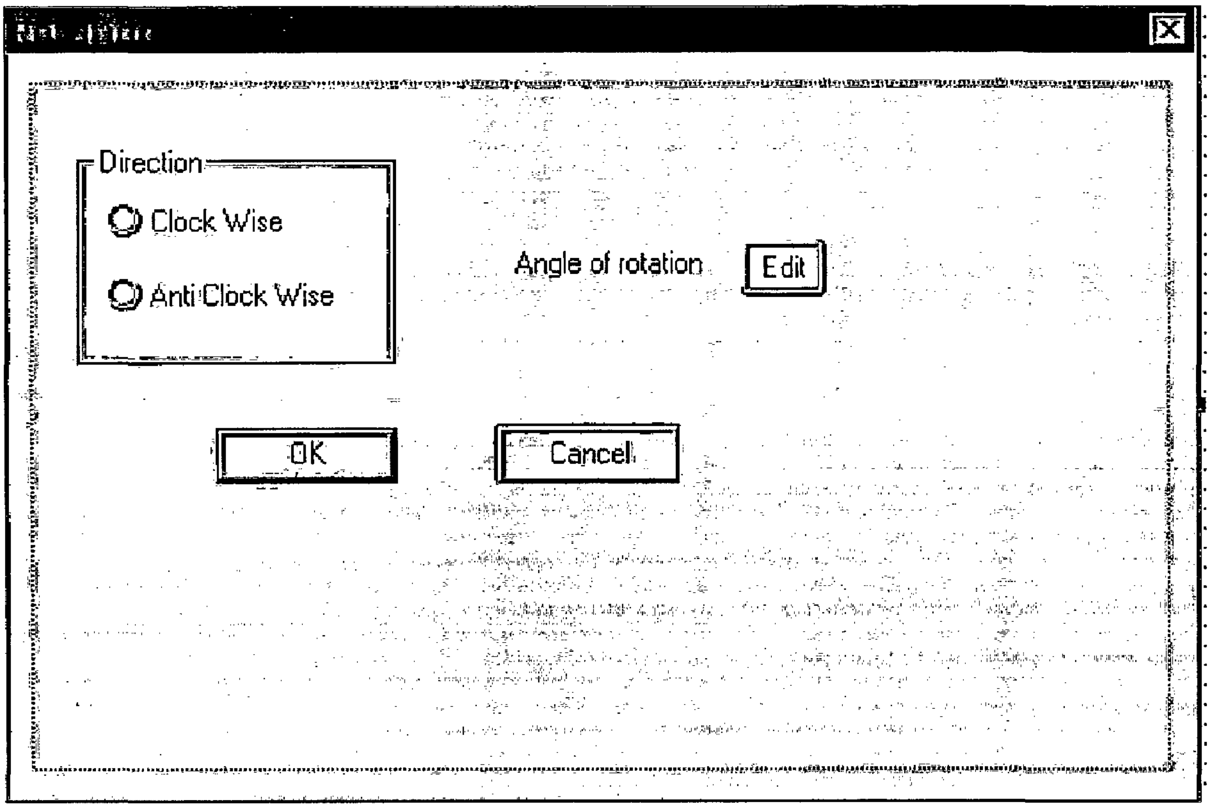


Description: Histograms analyze image color composition. The Histogram is a tool designed to help *you* decide what you need to do to improve your image. Full luminance is white while zero luminance is black. The spectrum for a color is on the horizontal axis while the portion of the image's color that matches that spectrum is displayed on the vertical axis. The Histogram Functions modify an image or a selection within that image based upon its luminance histogram.

Orientation: This window can be selected from menus-enhance-histogram.

Action: The histogram window displays a graph of the color values of the active window in RGB and luminance.

6.8 Rotation

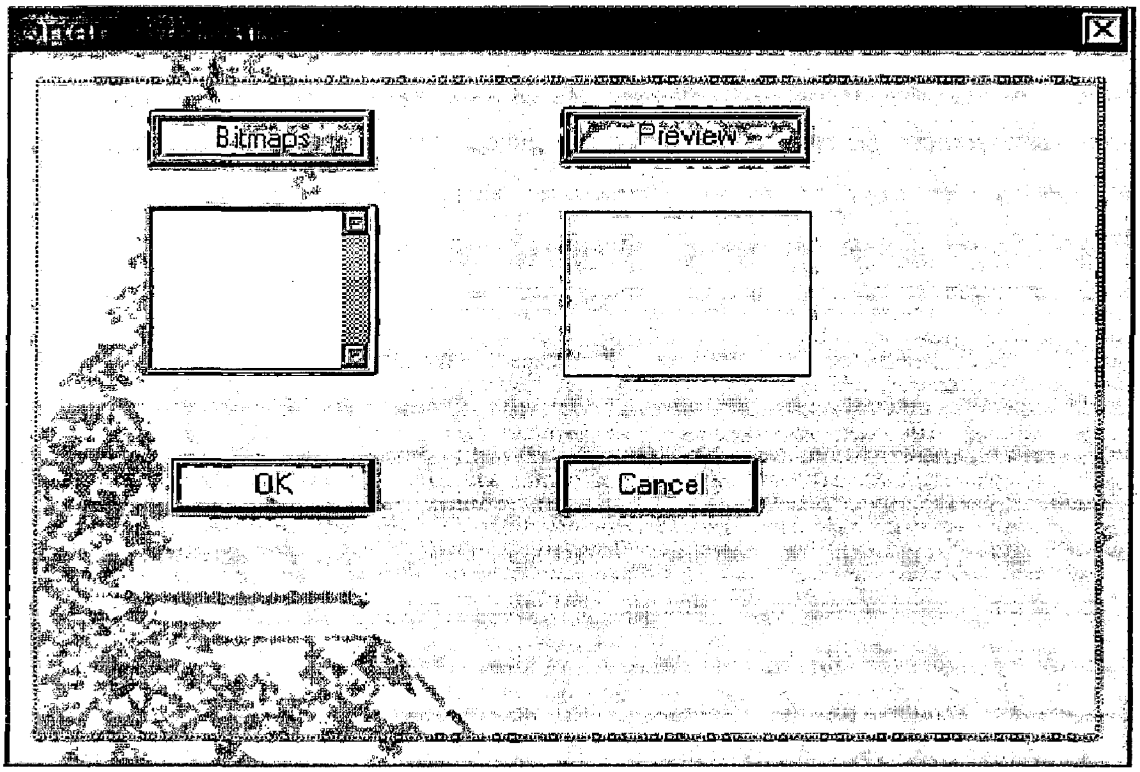


Description: This windowallows the user to rotate the images in any angle-90 degrees, 180 degrees, or even user defined angle.rotation can be done either clock wise or anti clock wise depending upon the user's choice.the angle varies between 0 and 360 degrees.

Orientation: This window appears when menus – image view- rotate is chosen.

Action: Rotates the given image in required angle clock wise or anti clock wise..

6.9 Bitmaps

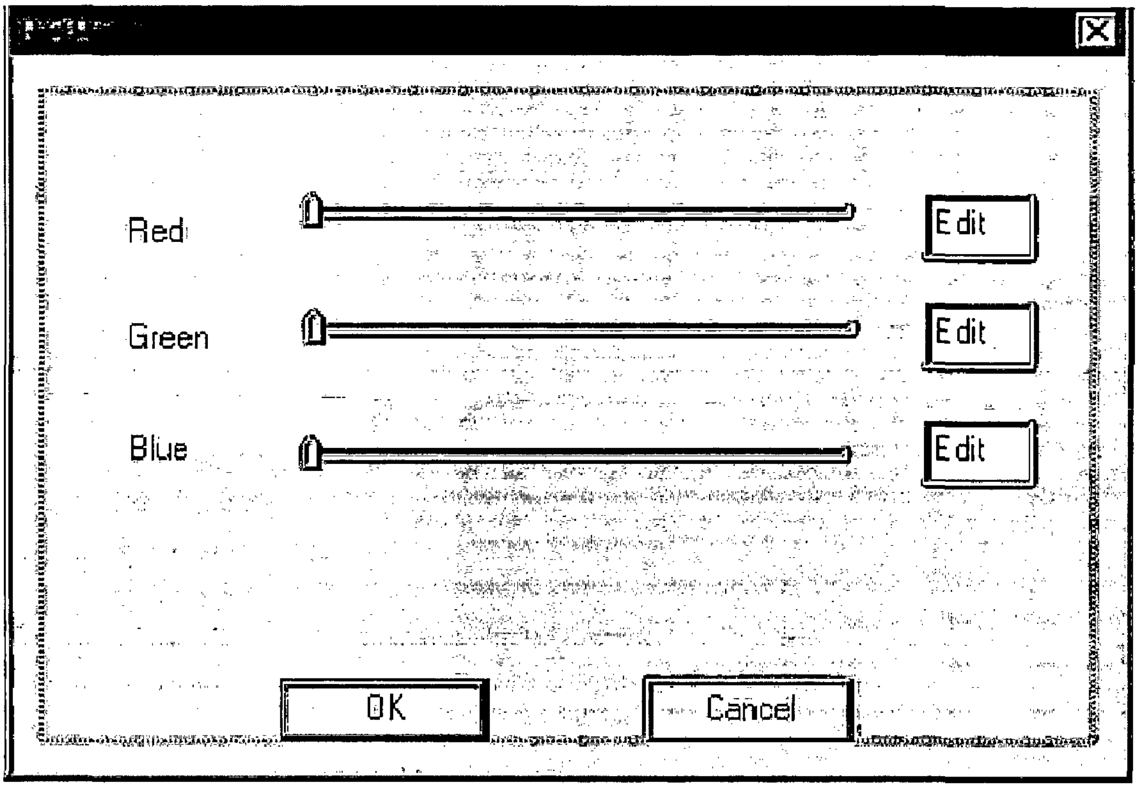


Description: The various bitmaps stored in any file in the c drive or floppy can be accessed through this window, to select for processing them.

Orientation: Menu - file - open bit map will lead to the opening of this window.

Action: Opens the selected bit map image.

6.10 RGB

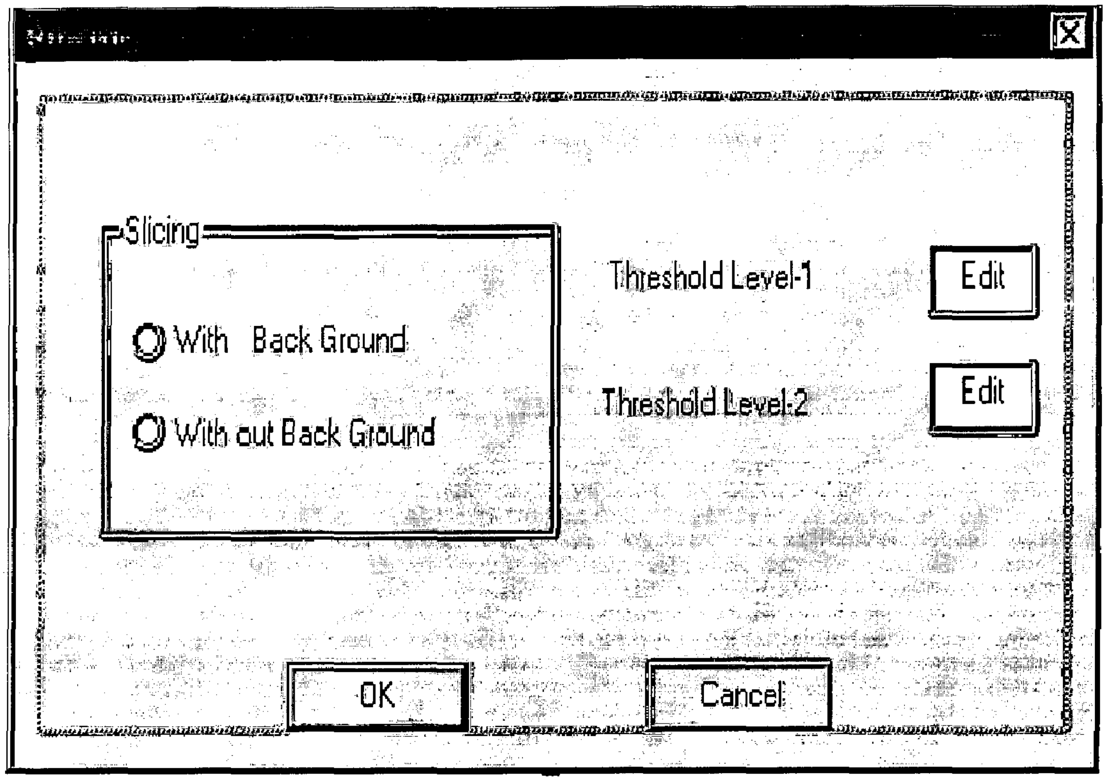


Description: RGB shows the image with respective red, blue, and green colors highlighted. that is, it shows the image with the chosen color's ratio more in the picture.

Orientation: Menus – enhance – adjustment – RGB.

Action: Adjust the image color with respect to the selected color.

6.11 Slicing

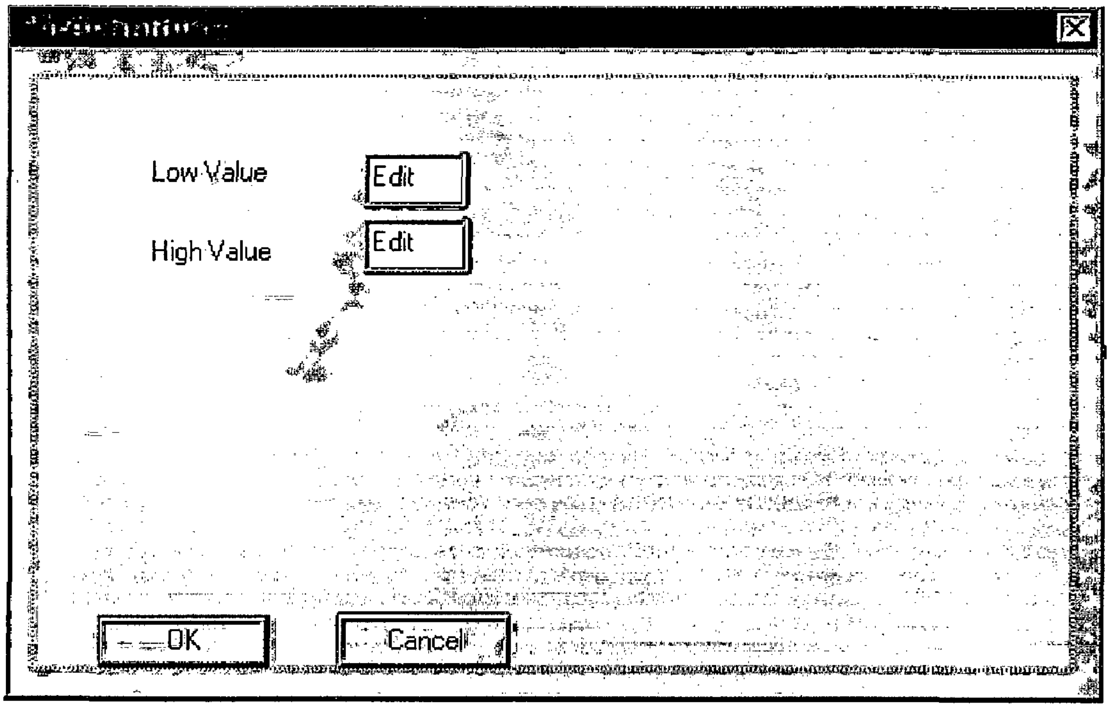


Description: This is a technique in which the image is sliced through the layer according to the range defined by the user. slicing slices the images into different layers of slices which will make dynamic loading of images easier.

Orientation: Menus – enhance – slicing will open this window.

Action: Slices an image through specified range, with or without the background image.

6.12 Thresholding



Description: This window is used to enable the image to be thresheld between a particular higher and lower values.this is actually a function which reproduces the selected image with a range of threshold as defined by the user.

Orientation: Menus – Enhance – Thresholding will open this window.

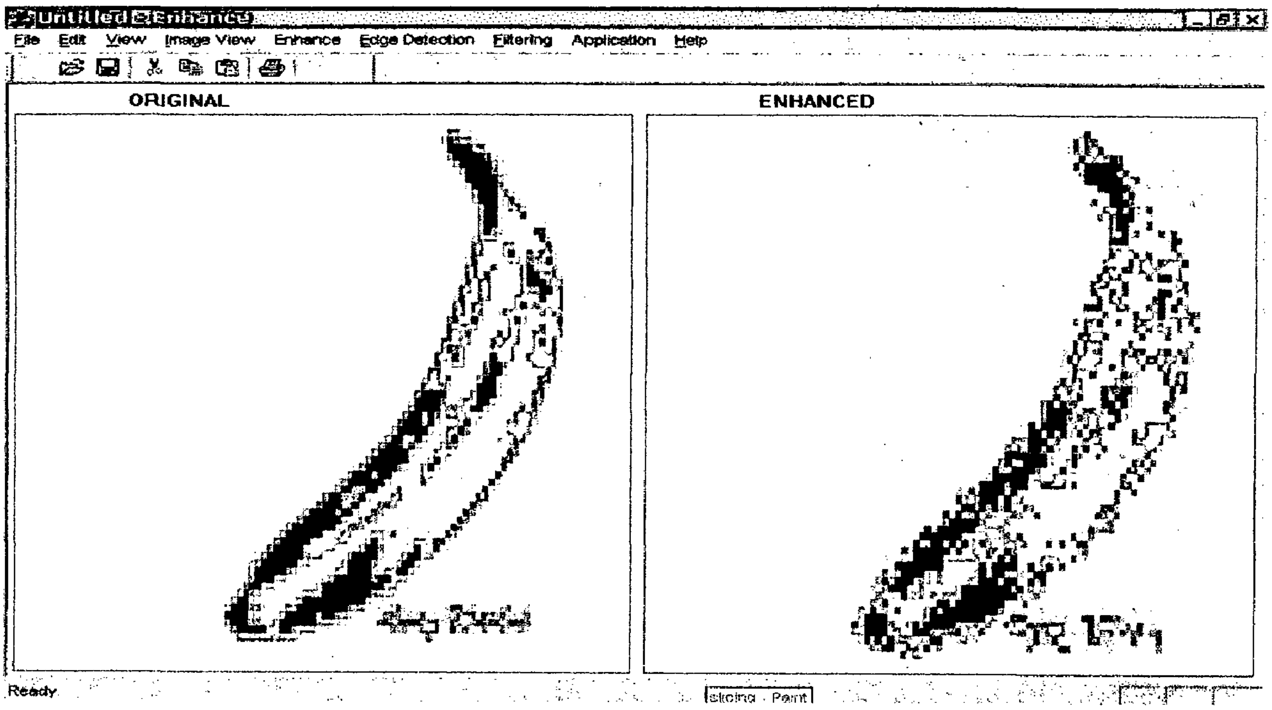
Action: Adjusts the image and thresholds it between the specified values.

SNAPSHOTS OF OUTPUTS

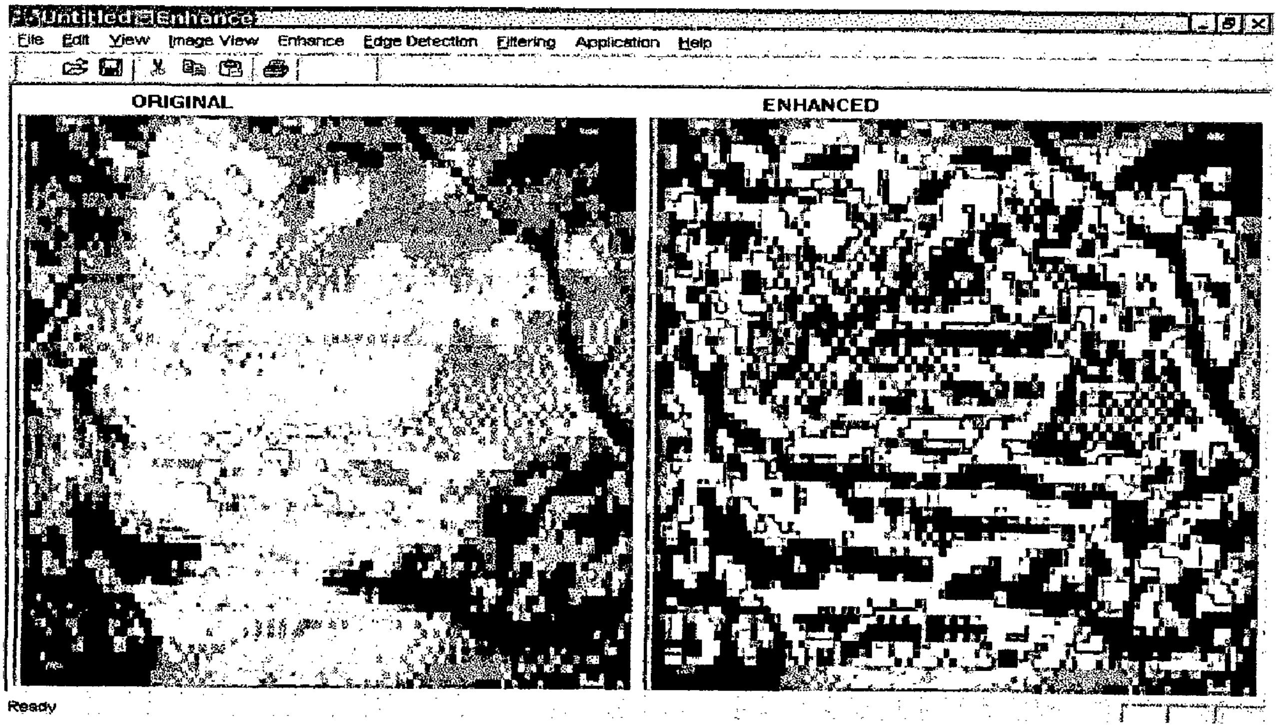
7.1 Figure showing an image and its binary output, which is in black and white:

7.2 Figure showing an image processed through mirror function:

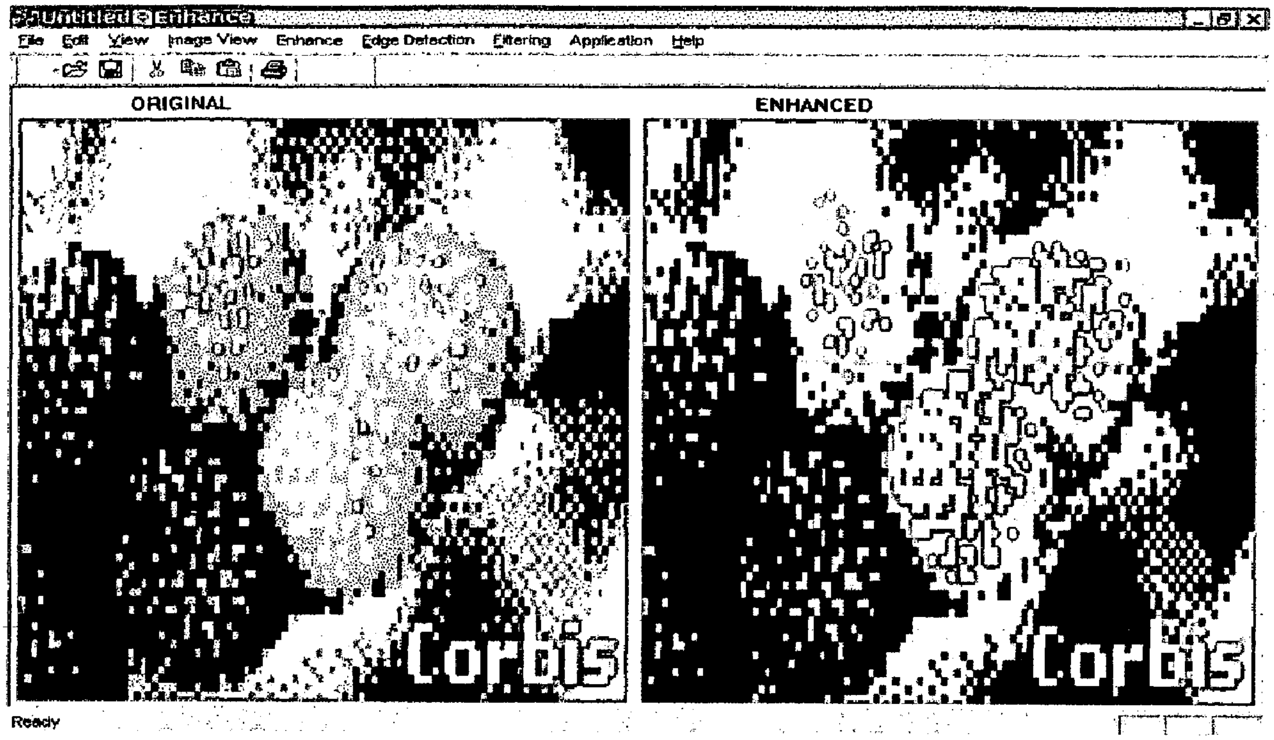
7.3 Figure showing diffused image:



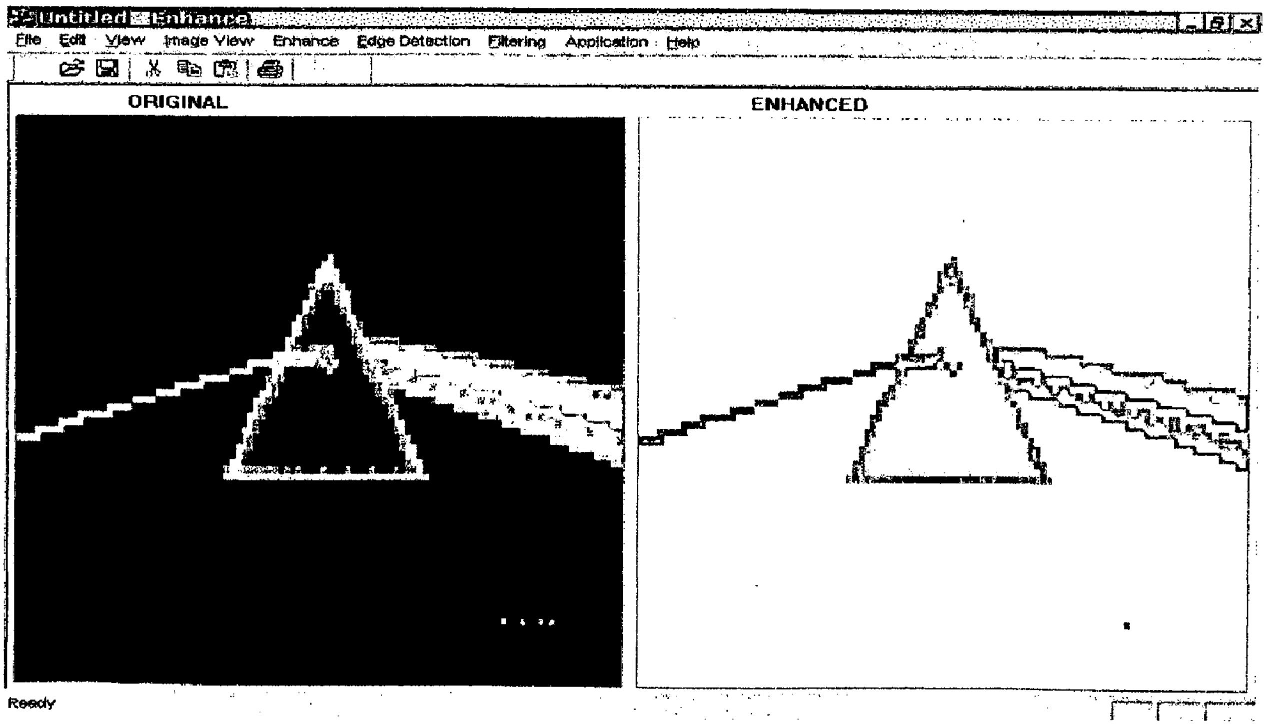
7.4 Image after passing through Gaussian edge detector:



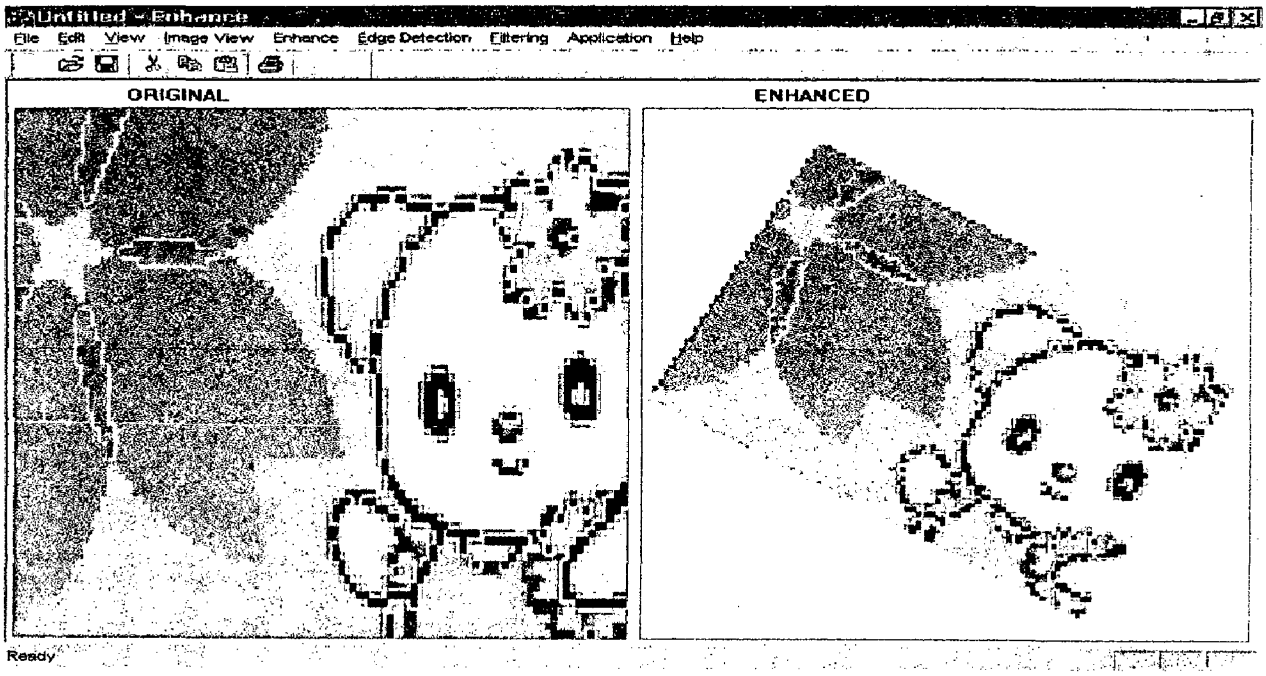
7.5 Image and its gray scaled output:



7.6 Figure showing image and its color - contrasted output:

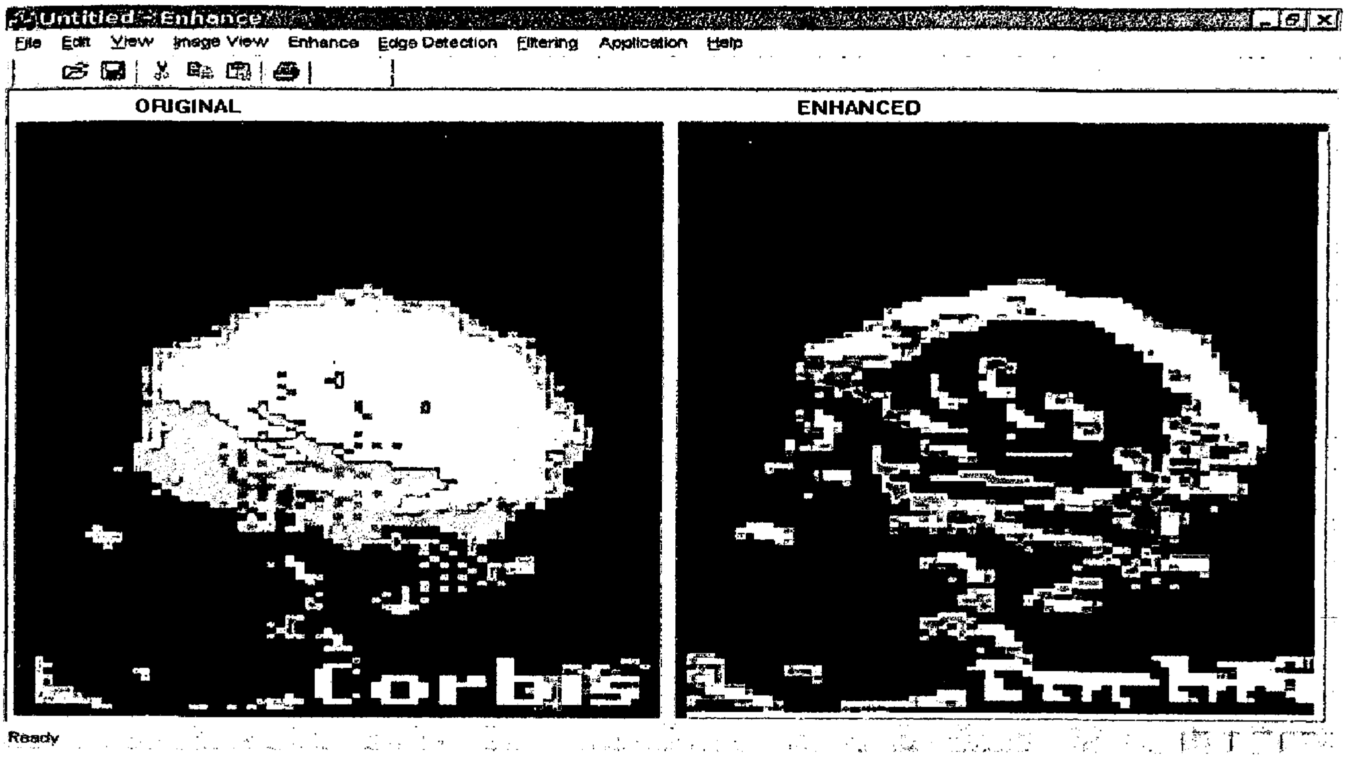


7.7 Figure showing an image rotated through 30 degrees:

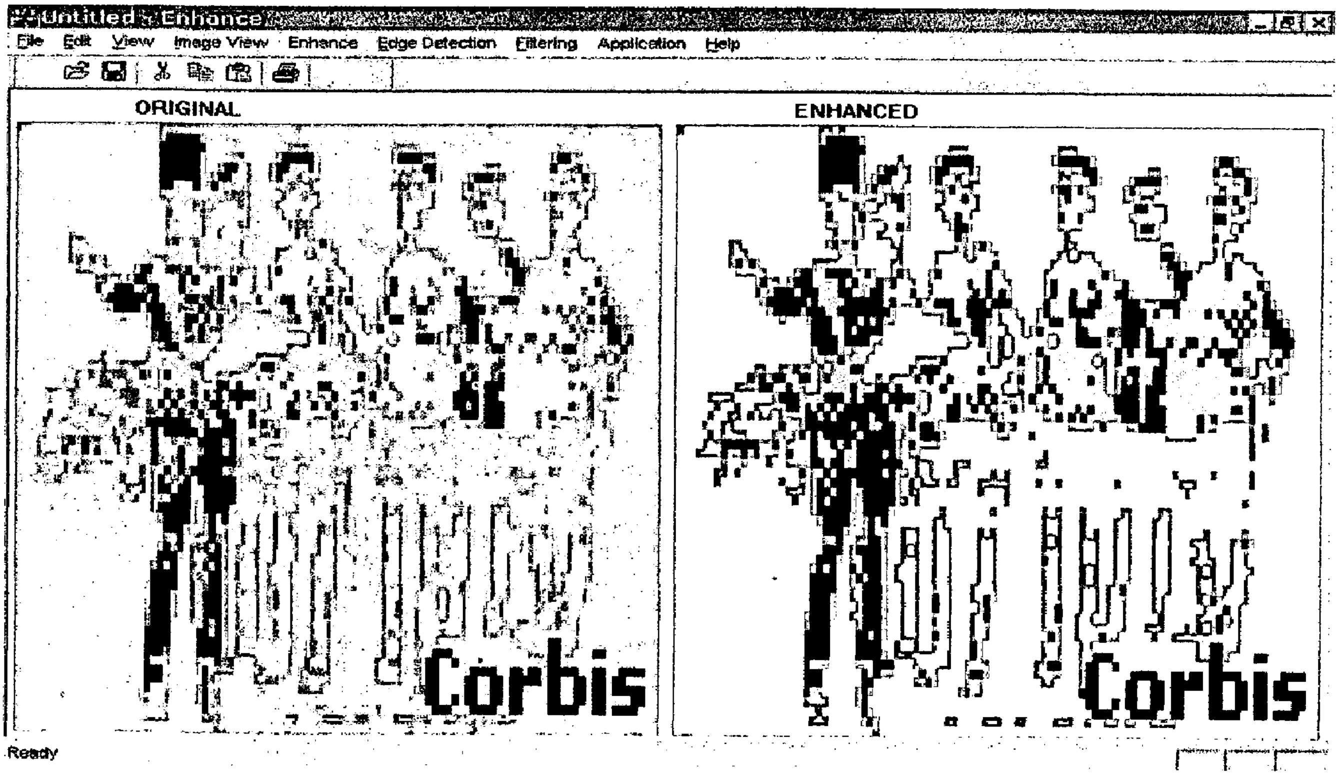


7.8 Image and its sliced output through a layer:

7.9 Image passed through sobel filter:

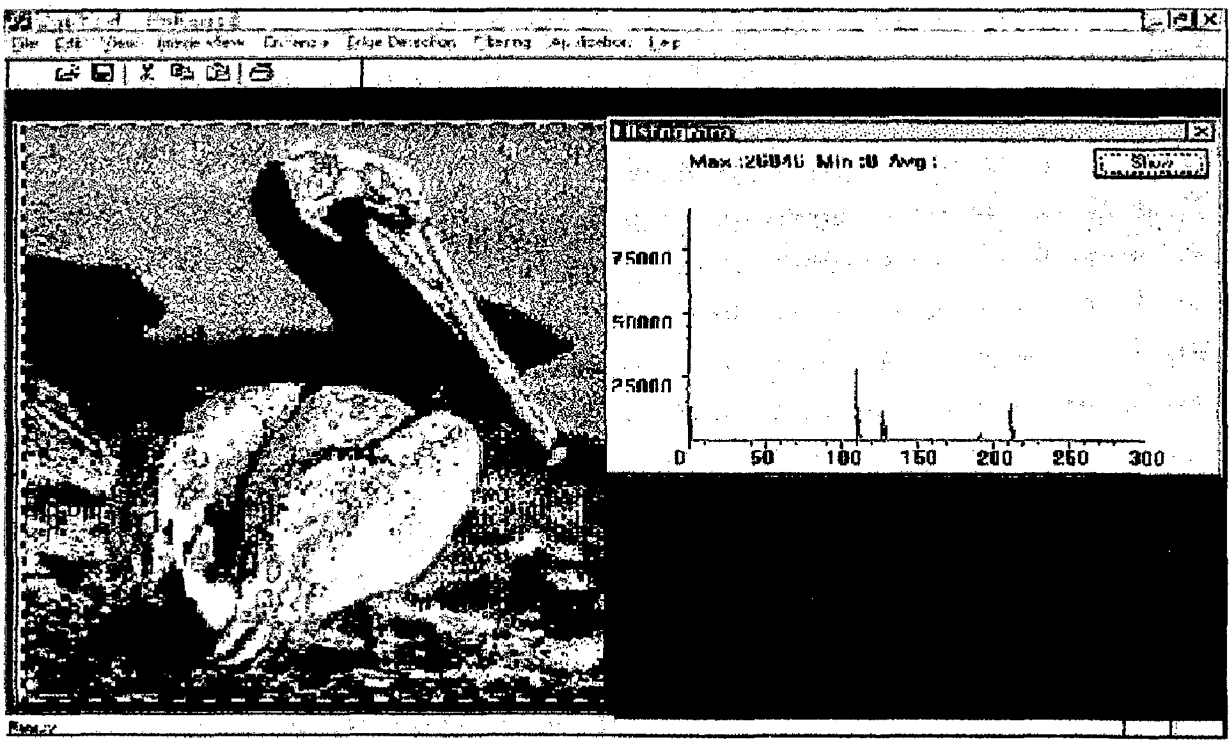


7.10 An image thresholded between the range of of 50 and 150:

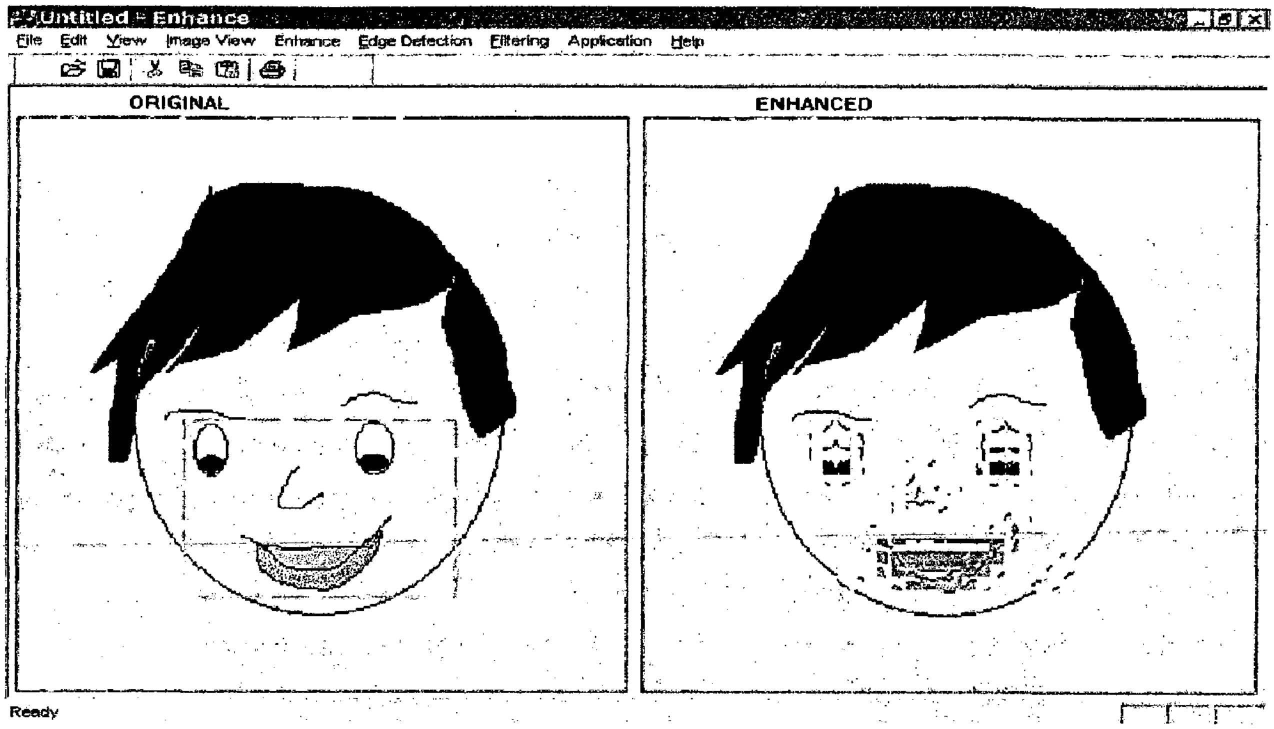


7.11 Figure showing an image and noise added to it :

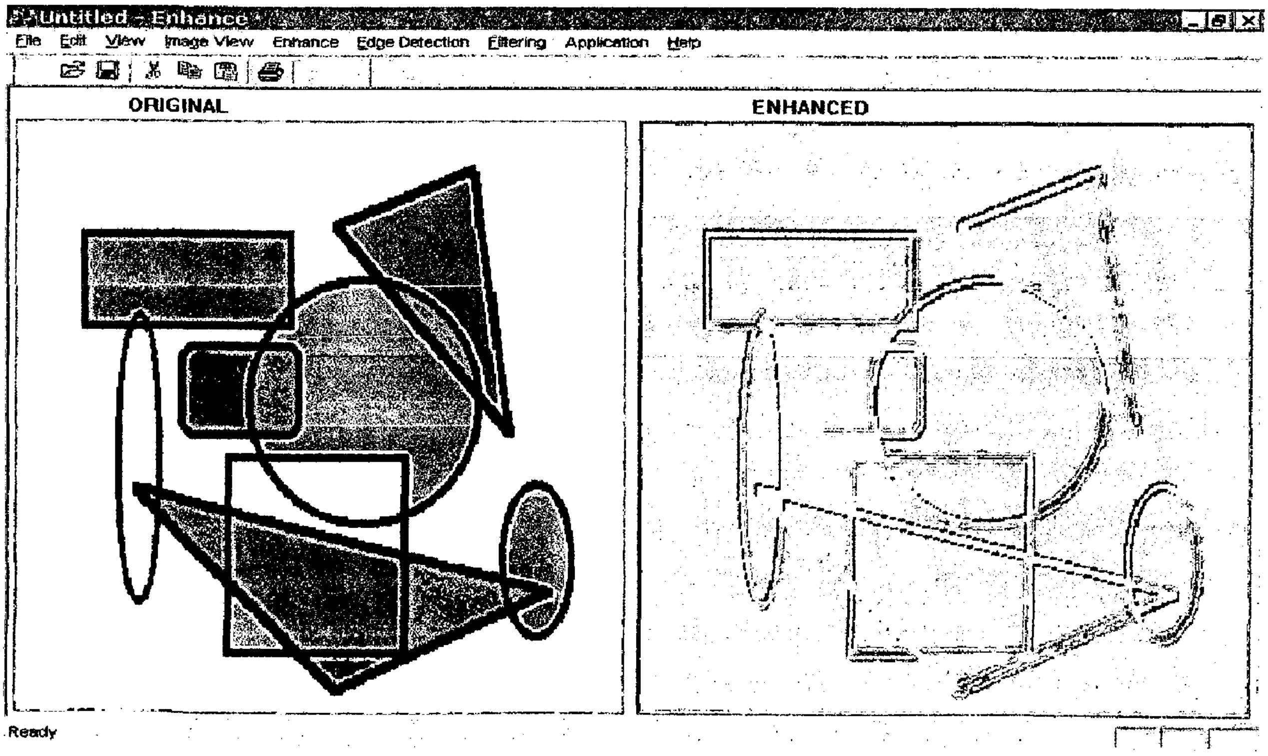
7.12 Histogram showing the level of 'RED' colors in an image:



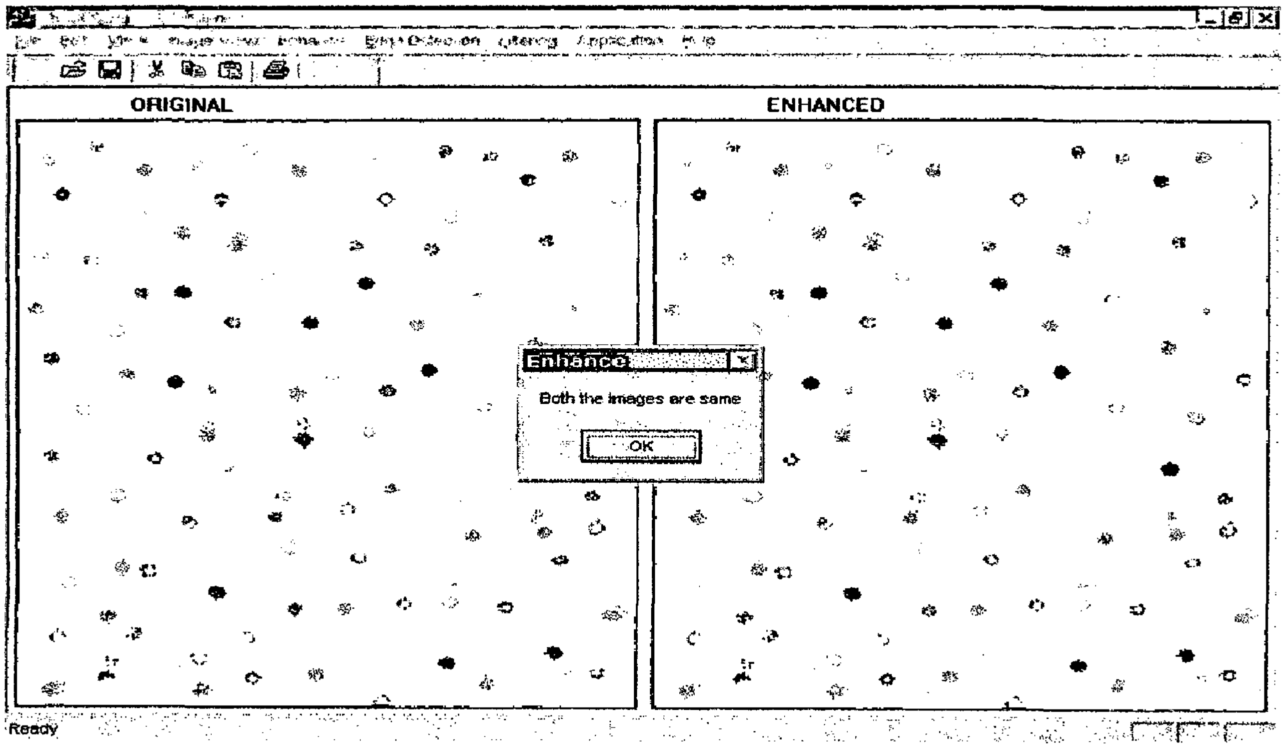
7.13 An image and it's demolished output:



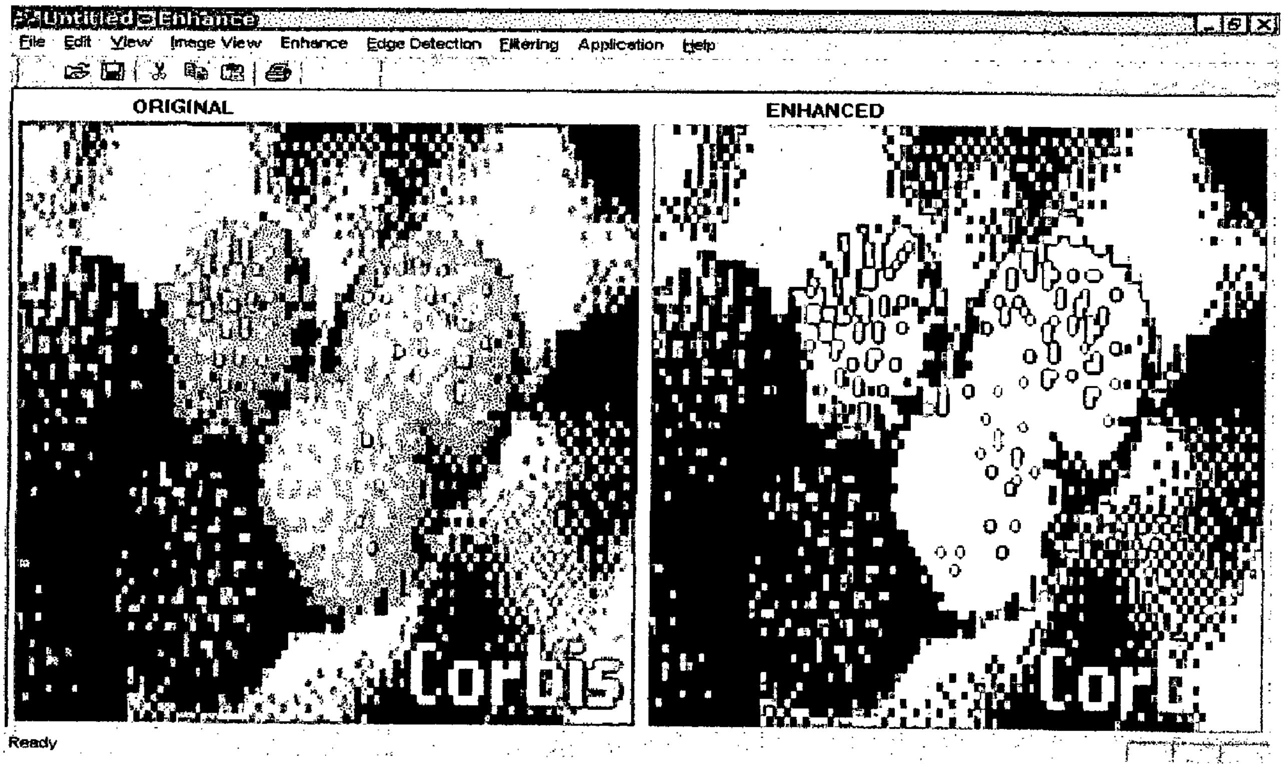
7.14 Figure showing an image passed through embossing function:



7.15 Output of an image compared with itself:



7.16 An image with arithmetic multiplication performed with itself:



Chapter 7

CONCLUSION

The project was successfully completed and the following are inferred from it:

- The basics of image processing of raster image formats, the technical descriptions of various image processing algorithms used for image manipulation, contrast enhancement, sharpening and blur removal, noise reduction, Filtering, steganography, segmentation, and image compression were studied and analysed thoroughly.
- The effect of the various parameters on the performance of image and processing algorithms was studied.
- The various methods to predict performance were studied and its limits were identified by understanding the concepts behind the algorithms.

The image processing tool set was able to perform the following functions successfully.

- Edge detection for raster type images.
- Low pass, high pass, band pass and user defined filtering for raster type images.
- Performs arithmetic operations of Add, Sub, Multiply, Divide, And and Or functions on images.
- Hides an image into another image and it is able to retrieve the original when decrypted.
- Thresholds an image in a given range of pixel values .
- Slices an image in given level of slicing
- Produces a gray scaled, binary represented, or contrasted output for given image
- Performs sharpening and embossing functions with the selected image
- It flips, mirrors, rotates and zooms a selected image.

This windows application tool set which is designed for processing 'raster'images, will help any beginner to process various kinds of raster images with comfort .There is little effort to be put by the user to enhance the image using the 'enhance'tool set, as the interface provided is fully equipped to ease the processing of raster images.

The tool set that is designed to process raster images will be a great application for designing web based images and for other applications like scanning,

Limitations of this project:

There are some problems I faced during implementation of the application module. The steganography function which is implemented has currently being done using simple Ex-Or function. That is, the image is hidden into another image by simply 'ex-or'ing the bits of first image into the second. This is not as secure as it doesn't involve complex functions. The encrypted image can be decrypted by hackers. Some noise is created in the decrypted image whenever it is retrieved from the encrypted image.

Future scope:

- At present, the tool set can be used to handle only raster images. In future it can be used to process vector images also.
- There can be a conversion provision implemented, which can convert raster images into vector and vice versa.
- The steganography function can be implemented using more complex algorithms so as to prevent easy hacking of hidden images.

REFERENCES

1. Ioannis Pitas, "Digital Image Processing Algorithms",Prentice Hall,1993
2. Rothstein, J. and Weiman, C.F.R. "Parallel and sequential specification of a context sensitive language for straight lines on grids",1980
3. Rothstein, J. and Weiman ,"Computer Graphics and Image Processing", Vol. 5, 1979, pp. 106-124
4. Rosenfeld, A. and Kak, A. "Digital Picture Processing", Academic Press, 1976.
5. Donald Hearn & M.Pauline Baker,"computer Graphics",Printice Hall of India,1999
6. Core VC++,Laiz Klande,Prentice Hall,1997
7. Teach yourself VC++, David.A.Holzgang,BP Publications,1996
8. Practical VC++,Jon Bates & TIM Tompkins,Oct,2001

APPENDIX A

A.1 VISUAL C++ 6.0

Microsoft Visual C++ is a developer studio member, which provides integrated Development for the development of project.

A.1.1 The Visual C++ Development Environment

The Visual C++ Development Environment mainly contains three areas- Workspace, Output Pane and Editor Area. Each of these areas has a specific purpose in the developer studio environment.

A.1.2 The Workspace

This is the key area to navigating the various pieces and parts of users development project. The workspace allows you to view the parts of your application in three different ways:

- Class View allow user to navigate and manipulate your source code on a C++ class level
- Resource View allow user to find and edit each of the various resources in your application, including dialog window designs, icons, and menus.
- File View allows user to view and navigate all the files that male up in the user's application.

A.1.3 The Output Pane

The Output Pane is where Developer Studio provides any information that it needs to give the user ; where user can see all the compiler progress statements, warnings, and error messages; and where the Visual C++ debugger displays all the variables with their current values as you step through the user's code.

The above screen shows the different areas in a Visual C++ development environment. The following screen shows the MFC ClassWizard using which the functions, classes, member variables, etc can be added easily.

A.1.4 The Editor Area

This is the area where user can perform all user's editing when user use Visual C++, where the code editor windows display when user edit C++ source code, and where the window painter displays when user design a dialog box. The Editor Area is even where the icon painter displays when user designs the icons for use in the user's application. The editor area is basically the entire Developer Studio area that is not otherwise occupied by panes, menus, or toolbars.

A.1.5 Visual C++ program

A Visual C++ program consists of C++ code and selected basic framework provided by Visual C++ compiler. These frameworks are to be selected by users according to the type of program he wants to develop.

A.1.6 Kinds of Visual C++ programs

- An MFC program
- An MFC DLL
- An MFC ActiveX control program
- An ActiveX Container program
- An Internet Server(ISAPI) Extension or Filter
- A Win32(non-MFC) program for Windows
- A Win32 DLL

- A Static Library
- A Console program
- A Utility project
- A program that supports a DAO or ODBC database
- An Extended Stored Procedure
- A Microsoft Cluster Resource Type

A.2 MICROSOFT FOUNDATION CLASSES

The Microsoft Foundation Class Library (MFC) is an "application framework" for programming in Microsoft Windows. Written in C++, MFC provides much of the code necessary for managing windows, menus, and dialog boxes; performing basic input/output; storing collections of data objects; and so on. All user needs to do is add user's application-specific code into this framework. And, given the nature of C++ class programming, it's easy to extend or override the functionality the MFC framework supplies.