ZÁPADOČESKÁ UNIVERZITA V PLZNI

Faculty of Electrical Engineering Plzen

Department of Applied Electronics and Telecommunications

BACHELOR THESIS

Image Enhancement Methods and Implementation in Matlab

Alaa Kassab 2012

Abstract

This bachelor thesis provides a description of image enhancement algorithms for multimedia technology. It contains two parts, theoretical and practical, the theoretical part of the project focuses mainly on methods of edge and color enhancement. Selected algorithms will be subsequently implemented and their efficiency will be verified using supplied image tests. In the practical part, Matlab will be used for evaluation of results.

A compact disc is attached to this work and it contains the thesis in pdf form and the Matlab code.

PROHLÁŠENÍ

Předkládám tímto k posouzení a obhajobě bakalářskou práci, zpracovanou na závěr studia na Fakultě elektrotechnické Západočeské univerzity v Plzni.

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně, s použitím odborné literatury a pramenů uvedených v seznamu, který je součástí této diplomové práce.

Dále prohlašuji, že veškerý software, použitý při řešení této bakalářské práce, je legální.

V Plzni dne 12.6.2012 Jméno příjmení

....

AKNOWLEDGEMENT

I would like to acknowledge and extend my heartfelt gratitude to my supervisor Eng. Raedk Holota Ph.D for his kind help, encouragement and support; also I would like to thank all the professors of Department of Applied Electronics and Telecommunication and all the people that supported me to complete my thesis. Most special thanks to my parents for the constant reminders and much needed motivation. And before all, to God, who made all things possible.

TABLE OF CONTENTS

TABLE OF FIGURES

ABBREVIATIONS

AI = Artificial Intelligence

BHPF = Butterworth High-pass Filter

EM = Electromagnetic

FSHS = Fall Scale Histogram stretch

GHPS = Gaussian High-pass Filter

HPF = High-pass Filter

IE = Image Enhancement

IHPF = Ideal High-pass Filter

ILPF = Ideal Low-pass Filter

JPEG = Joint Photographic Expert Group

LUT = Look up Table

MRI = Medical Resonance imaging

RGB = Red, Green, Blue

YCbCr = Luminance and chrominance

Matlab = Matrix Laboratory

INTRODUCTION

My project is about image enhancement, there are two parts, spatial domain and frequency domain. I will focus on edge and color enhancement, especially on the sharpening techniques and I will use its different types. I will use some examples of filters based on sharpening technique.

In the first chapter I will introduce image enhancement and its uses, advantages, principles, and types.

In the second chapter I will introduce enhancement in the spatial domain, its theory, techniques and types.

In the third chapter I will explain the enhancement in frequency domain. And in the fourth chapter will be the practical part; I will use Matlab for my calculations.

Furthermore, I shall thorough fully introduce the image processing in detail.

One picture is worth more than ten thousand words. The ability to see is one of the truly remarkable characteristics of living beings. It enables them to perceive and assimilate in a short span of time an incredible amount of knowledge about the world around them. The scope and variety of that which can pass through the eye and be interpreted by the brain is nothing short of astounding.

It is thus with some degree of trepidation that we introduce the concept of visual information, because in the broadest sense, the overall significance of the term is overwhelming. Instead of taking into account all of the ramifications of visual information; the first restriction we shall impose is that of finite image size, In other words, the viewer receives his or her visual information as if looking through a rectangular window of finite dimensions. This assumption is usually necessary in dealing with real world systems such as cameras, microscopes and telescopes for example; they all have finite fields of view and can handle only finite amounts of information.

The second assumption we make is that the viewer is incapable of depth perception on his own. That is, in the scene being viewed he cannot tell how far away objects are by the normal use of binocular vision or by changing the focus of his eyes.

An image may be defined as a two-dimensional function, f(x, y), where x and y are spatial (plane) coordinates, and the amplitude of f at any pair of coordinates (x, y) is called the intensity or gray level of the image at that point. The digital images are defined when x, y, and the amplitude values of f are all finite, discrete quantities. The field of digital image processing refers to processing digital images by means of a digital computer. Note that a digital image is composed of a finite number of elements, each of which has a particular location and value. These elements are referred to as picture elements, image elements, peels and pixels. Pixel is the term most widely used to denote the elements of a digital image. Vision is the most advanced of our senses, so it is not surprising that images play the single most important role in human perception.

The area of image analysis is in between image processing and computer vision. There are no clear-cut boundaries in the continuum from image processing at one end to computer vision at the other. However, one useful paradigm is to consider three types of computerized processes in this continuum: low-, mid-, and high-level processes. Low-level processes involve primitive operations such as image preprocessing to reduce noise, contrast enhancement, and image sharpening. A low-level process is characterized by the fact that both its inputs and outputs are images. Mid-level processing on images involves tasks such as segmentation, description of those objects to reduce them to a form suitable for computer processing, and classification (recognition) of individual objects. A mid-level process is characterized by the fact that its inputs generally are images, but its outputs are attributes extracted from those images. Finally, higher-level processing involves "making sense" of an ensemble of recognized objects, as in image analysis, and, at the far end of the continuum, performing the cognitive functions normally associated with vision. Based on the preceding comments, we see that a logical place of overlap between image processing and image analysis is the area of recognition of individual regions or objects in an image. As a simple illustration to clarify these concepts, consider the area of automated analysis of text. The processes of acquiring an image of the area containing the text, preprocessing that image, extracting the individual characters, describing the characters in a form suitable for computer processing, and recognizing those individual characters are in the scope of what we call digital image processing. As will become evident shortly, digital image processing, as we have defined it, is used successfully in a broad range of areas of exceptional social and economic value. **[1][3][4]**

1 Image Enhancement

The main definition of enhancing is to make something greater in value, desirability or attractiveness. The term of enhancement implies a process to improve the visual quality of the image. Image Enhancement transforms images to provide better representation of the subtle details. The principal objective of enhancement is to process an image so that the result is more suitable than the original image for a specific application. Image enhancement processes consist of a collection of techniques that seek to improve the visual appearance of an image or to convert the image to a form better suited for analysis by a human or a machine. In an image enhancement system, there is no conscious effort to improve the fidelity of a reproduced image with regard to some ideal form of the image, as is done in image restoration.

Actually, there is some evidence to indicate that often a distorted image, for example, an image with amplitude overshoot and undershoot about its object edges, is more subjectively pleasing than a perfectly reproduced original. Enhancement of an image is necessary to improve appearance or to highlight some aspect of the image is converted from one into another acquired, scanned, transmitted, copied or printed many types of noise can be present in the image. Image enhancement has come to specifically mean a process of smothering irregularities or noise that has somehow corrupted the image. The term “image enhancement” has been widely used in the past to describe any operation that improves image quality by some criteria. However, in the recent years the meaning of the term has evolved to denote image-preserving noise smoothing.

This primarily serves to distinguish it from similar-sounding terms, such as image restoration and image reconstruction, which also taking specific meaning. Image enhancement has played and will continue to play an important role into different fields such as medical, industrial, military and scientific applications. In addition to these applications, image enhancement is increasingly being used in consumer electronics. Internet Web users, for instance, not only rely on built-in image processing protocols such as JPEG (Joint Photographic Expert Group) and interpolation, but they also have become image processing users equipped with powerful yet inexpensive software such as Photoshop. Users not only retrieve digital images from the Web but they are now able to acquire their own by use of digital cameras or through digitization services. Image enhancement is an indispensable tool for researchers in a wide variety of fields:

1. In forensics, image enhancement is used for identification, evidence gathering and surveillance. Images obtained from fingerprint detection, security videos analysis and crime scene investigations are enhanced to help in identification of culprits and protection of victims.

2. In atmospheric sciences IE is used to reduce the effects of haze, fog, mist and turbulent weather for meteorological observations. It helps in detecting shape and structure of remote objects in environment sensing. Satellite images undergo image restoration and enhancement to remove noise.

3. Astrophotography faces challenges due to light and noise pollution that can be minimized by IE. For real time sharpening and contrast enhancement several cameras have in-built IE functions. Moreover, numerous softwares allow editing such images to provide better and bright results.

4. In oceanography the study of images reveals interesting features of water flow, remains concentration, geomorphology and bathymetric patterns to name a few. These features are more clearly observable in images that are digitally enhanced to overcome the problem of moving targets, deficiency of light and obscure surroundings.

5. IE techniques when applied to pictures and videos help the visually impaired in reading small print, using computers and television and face recognition. Several studies have been conducted that highlight the need and value of using IE for the visually impaired.

6. The technique of image enhancement is often employed by virtual restoration of historic paintings and artifacts in order to reduce stains and crevices. Color contrast enhancement, sharpening and brightening are just some of the techniques used to make the images bright. IE is a powerful tool for restorers who can inform decisions by viewing the results of restoring a painting beforehand. It is evenly useful in discerning text from worn-out historic documents.

7. In the field of e-learning, IE is used to clarify the contents of chalkboard as viewed on streamed video; it improves the content readability and helps students to focus on the text. Similarly, collaboration through the whiteboard is facilitated by enhancing the shared data and diminishing artifacts like shadows and blemishes.

8. Medical imaging uses IE techniques for reducing noise and sharpening details to improve the visual representation of the image. Since minute details play a critical role in diagnosis and treatment of disease, it is essential to highlight important features while displaying medical images. This makes IE a necessary aiding tool for viewing anatomic areas in MRI, ultrasound and x-rays to name a few.

9. Numerous other fields including law enforcement, microbiology, biomedicine, bacteriology, climatology, meteorology, etc., benefit from various IE techniques. These benefits are not limited to professional studies and businesses but extend to the common users who employ IE to cosmetically enhance and correct their images. **[1] [3][5]**

The following image explains the different types of image enhancement techniques.

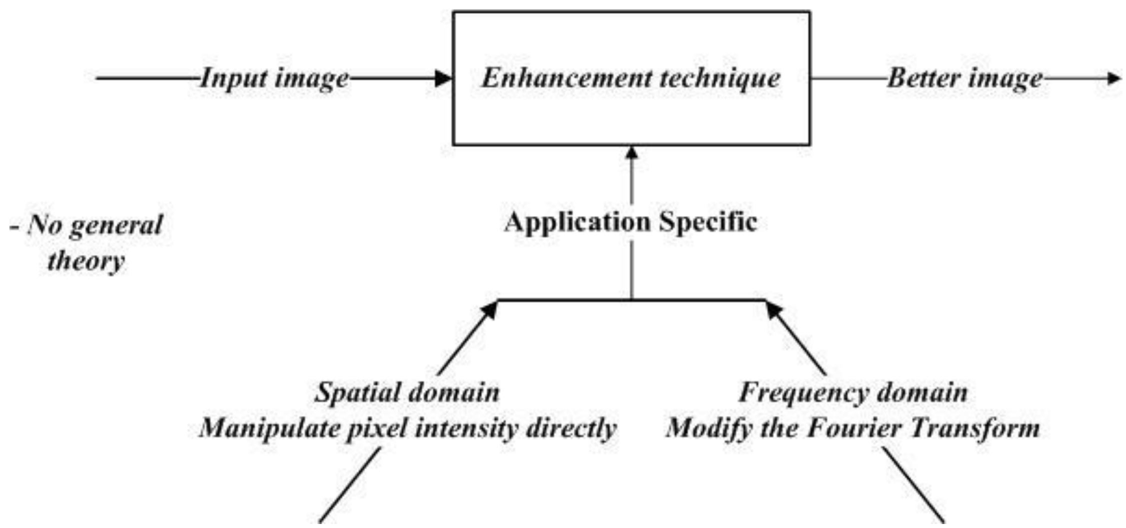

Figure 1: Image enhancement techniques

2 Spatial domain techniques

These techniques are based on gray level mappings, where the type of mapping used depends on the criterion chosen for enhancement. As an example consider the problem of enhancing the contrast of an image. Let *r* and *s* denote any gray level in the original and enhanced image respectively. Suppose that for every pixel with level *r* in original image we create a pixel in the enhanced image with level $S = T(r)$. If $T(r)$ has the form as shown in figure below:

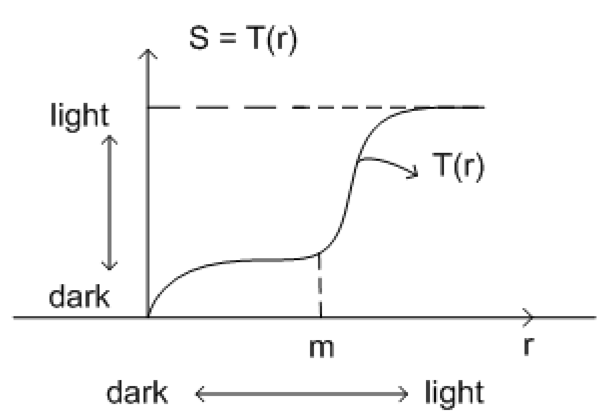

Figure 2: Example of contrast stretching in spatial domain. [3]

The effect of this transformation will be to produce an image of higher contrast than the original by darkening the levels below a value *m* and brightening the levels above *m* in the original pixel spectrum. The technique is referred to as contrast stretching. The values of *r* below *m* are compressed by the transformation function into a narrow range of *S* towards the dark end of the spectrum; the opposite effect takes place for values of *r* above *m*.

In the limiting case shown in figure, $T(r)$ produces 2-level (binary) image. This is also referred to as image thresholding. Many powerful enhancement processing techniques can be formulated in the spatial domain of an image. It does not exist a general definition of an image enhancement. When an image is processed for visual interpolation, the observer is the ultimate judge of how well a particular method works. Visual evaluation of image quality is a subjective process consequently making the definition of a "good image" an elusive standard by which to compare algorithm performance.

When the problem is one of processing images for machine perception, the evaluation task is easier. For example, if we take the problem of character recognition by a machine the best image processing method would be the one that yields the best machine recognition result… In general, if there is somewhere a clear cut criterion of performance imposed on a problem there is usually a certain amount of trial and error before one selects a particular image processing approach. **[3][5]**

2.1 Point processing

Point processing techniques are among the simplest of all image enhancement techniques, considering processing methods that are based only the intensity of single pixels.

$$g(x,y)=t[f(x,y)]$$

Simplest case: Neighborhood is (x,y)

[$g(.)$ depends only on the value of f at (x, y)]

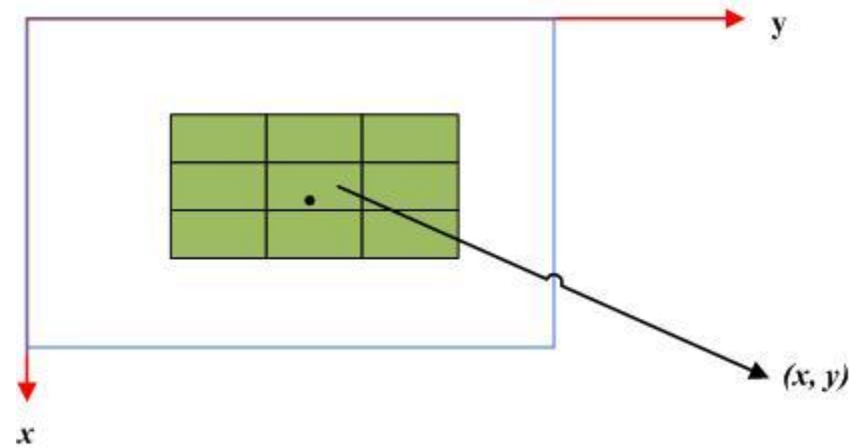

Figure 3: Simplest case of Neighborhood (x, y)

2.1.1 Gray level transformation

Image Negative is a typical grey scale transformation that does not depend on the position of the pixel in the image. The output grey value s is related to the input grey value as follows:

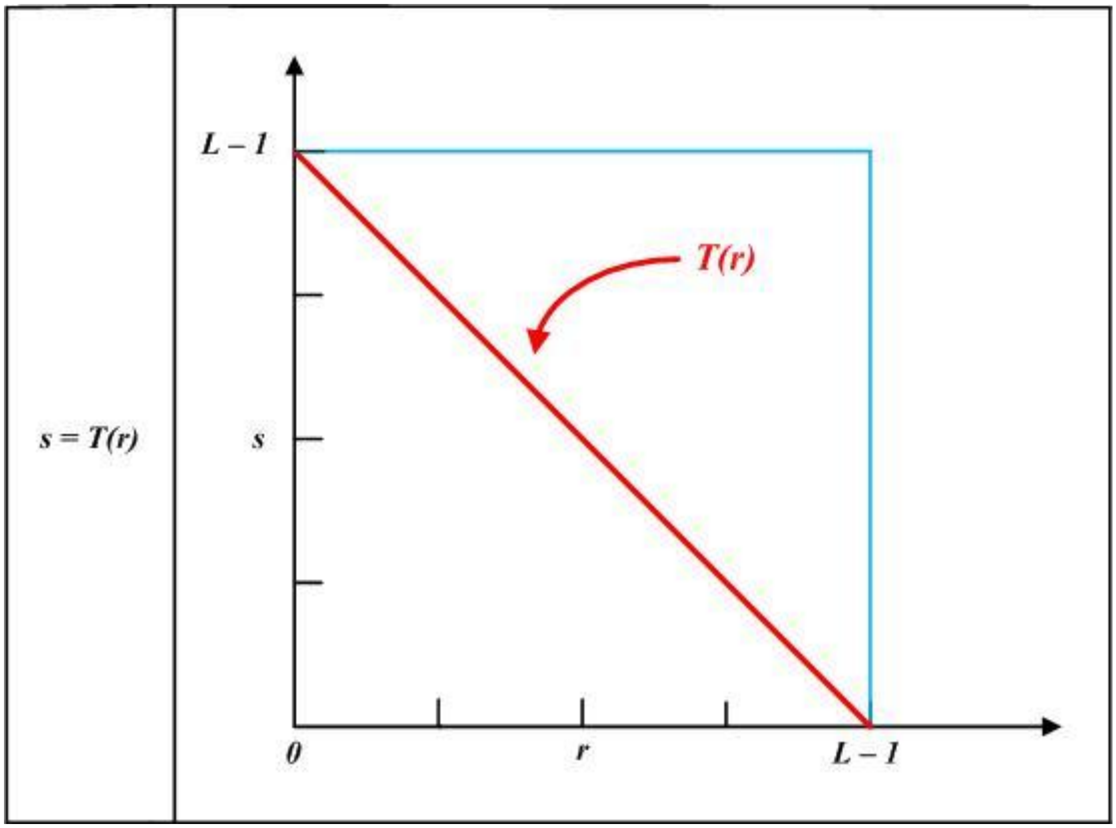

Figure 4: Gray level transformation function for obtaining the image negative of an image.[3]

Negatives of digital images are useful in numerous applications, such as displaying medical images and photographing a screen with monochrome positive film with the idea of using the resulting negatives as normal slides.

Figure 5: Example of a Negative image.

2.2 Histogram processing

The histogram in the context of image processing is the operation by which the occurrences of each intensity value in the image is shown. Normally, the histogram is a graph showing the number of pixels in an image at each different intensity value found in that image. For an 8-bit grayscale image there are 256 different possible intensities, and so the histogram will graphically display 256 numbers showing the distribution of pixels amongst those grayscale values.

2.2.1 Normalized Histogram

Normalized histogram is the histogram in which the no. of pixels for each intensity level is divided by the total no. of pixels in the image. Hence, if the whole of the image is of the same color, e.g. a white paper picture, then its normalized histogram contains only one non-zero number, and that is 1 for the 255th intensity level, which is the intensity level for the white color.

Normalized Histogram (i) = (Total Number of Pixels of Intensity i)/(Total Number of Pixels)

The image is scanned in a single pass and a running count of the number of pixels found at each intensity value is kept. This is then used to construct a suitable histogram. This operation is performed in a single pass algorithm, surfing each pixel once, and increment the number of occurrences of the specific intensity level depending upon the intensity of that particular pixel.

$$p(r_k) = n_k / n$$

For k = 0,1,…,L-1, and $p(r_k)$ gives an estimate of the probability of occurrence of gray level n_k, and n is total number of pixels

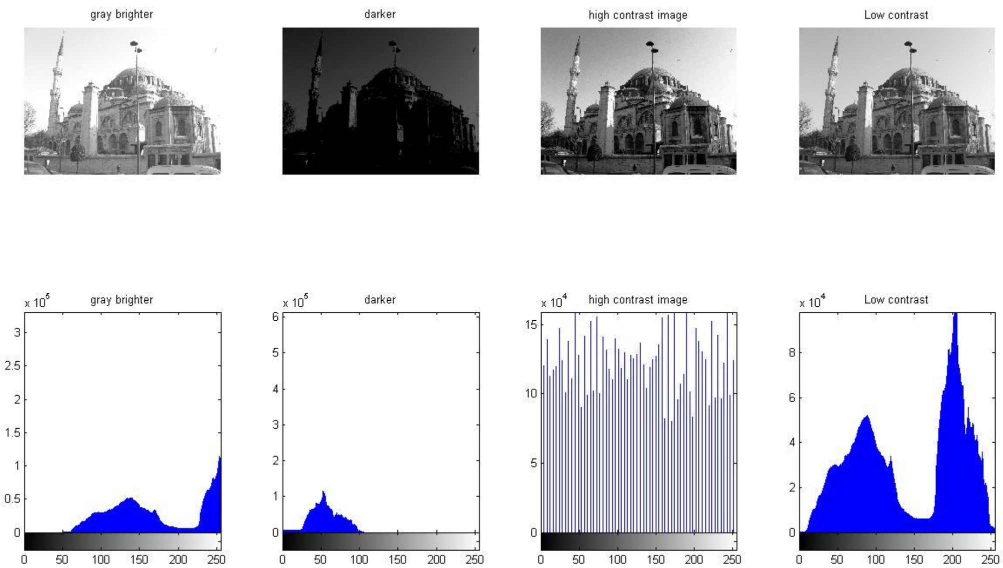

Figure 6: Four basic image types: dark, light, low contrast, high contrast, and their corresponding histograms.

2.2.1 Histogram equalization

Histogram equalization is the technique by which the dynamic range of the histogram of an image is increased. Histogram equalization assigns the intensity values of pixels in the input image such that the output image contains a uniform distribution of intensities. It improves contrast and the goal of histogram equalization is to obtain a uniform histogram. This technique can be used on a whole image or just on a part of an image.

Histogram equalization redistributes intensity distributions. If the histogram of any image has many peaks and valleys, it will still have peaks and valley after equalization, but peaks and valley will be shifted. Because of this, "spreading" is a better term than "flattening" to describe histogram equalization. In histogram equalization, each pixel is assigned a new intensity value based on its previous intensity level.

As the low-contrast image’s histogram is narrow and centered toward the middle of the gray scale, if we distribute the histogram to a wider range the quality of the image will be improved.

We can do it by adjusting the probability density function of the original histogram of the image so that the probability spread equally.

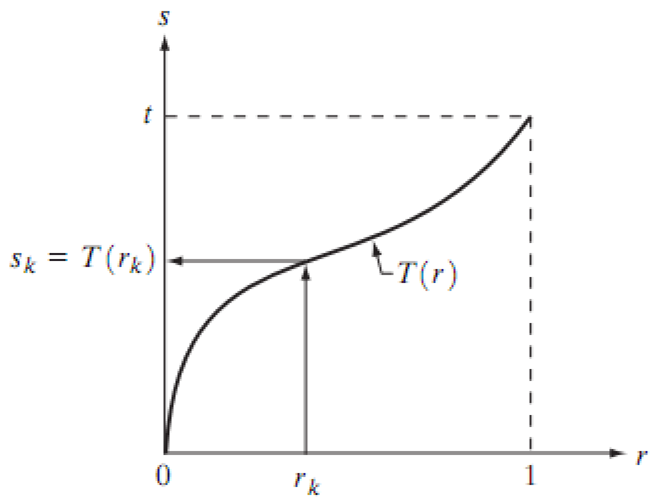

Figure 7: gray-level transformation function that is both single valued and monotonically increasing.[3]

Let the variable *r* represent the grey level of the pixels in the image to be enhanced. Assume that the pixel values are normalized to lie in the range $0 \le r \le 1$ with $r = 0$ represents black , $T(r)$ represents white in the gray scale.

For r_k, we consider transformations of the form $S_k = T(r_k)$ which produce a level S_k for every pixel value r_k in the original image. It is assumed that the transformation function satisfies the conditions:

(1) $T(r_k)$ is singled valued and monotonically increasing in the interval { $0 \le r \le 1$};

(2), $0 \le T(r_k) \le 1$ *for* $0 \prec r_k \le 1$

One of the most important nonlinear point operations is histogram equalization, also called histogram flattening. The idea behind it extends that of FSHS: not only should an image fill the available gray scale range, but it should be uniformly distributed over that range.

Histogram equalization is the technique by which the dynamic range of the histogram of an image is increased. Histogram equalization assigns the intensity values of pixels in the input image such that the output image contains a uniform distribution of intensities. It improves contrast and the goal of histogram equalization is to obtain a uniform histogram. This technique can be used on a whole image or just on a part of an image.

Histogram equalization redistributes intensity distributions. If the histogram of any image has many peaks and valleys, it will still have peaks and valley after equalization, but peaks and valley will be shifted. Because of this, "spreading" is a better term than "flattening" to describe histogram equalization. In histogram equalization, each pixel is assigned a new intensity value based on its previous intensity level. **[3][5][6]**

2.3 Basic Spatial Filtering (Mask processing)

A filter is a special kind of tool designed to take an input layer or image, apply a mathematical algorithm to it, and return the input layer or image in a modified format. Enhance filters are used to compensate for image imperfections. Such imperfections include dust particles, noise, interlaced frame and insufficient sharpness.

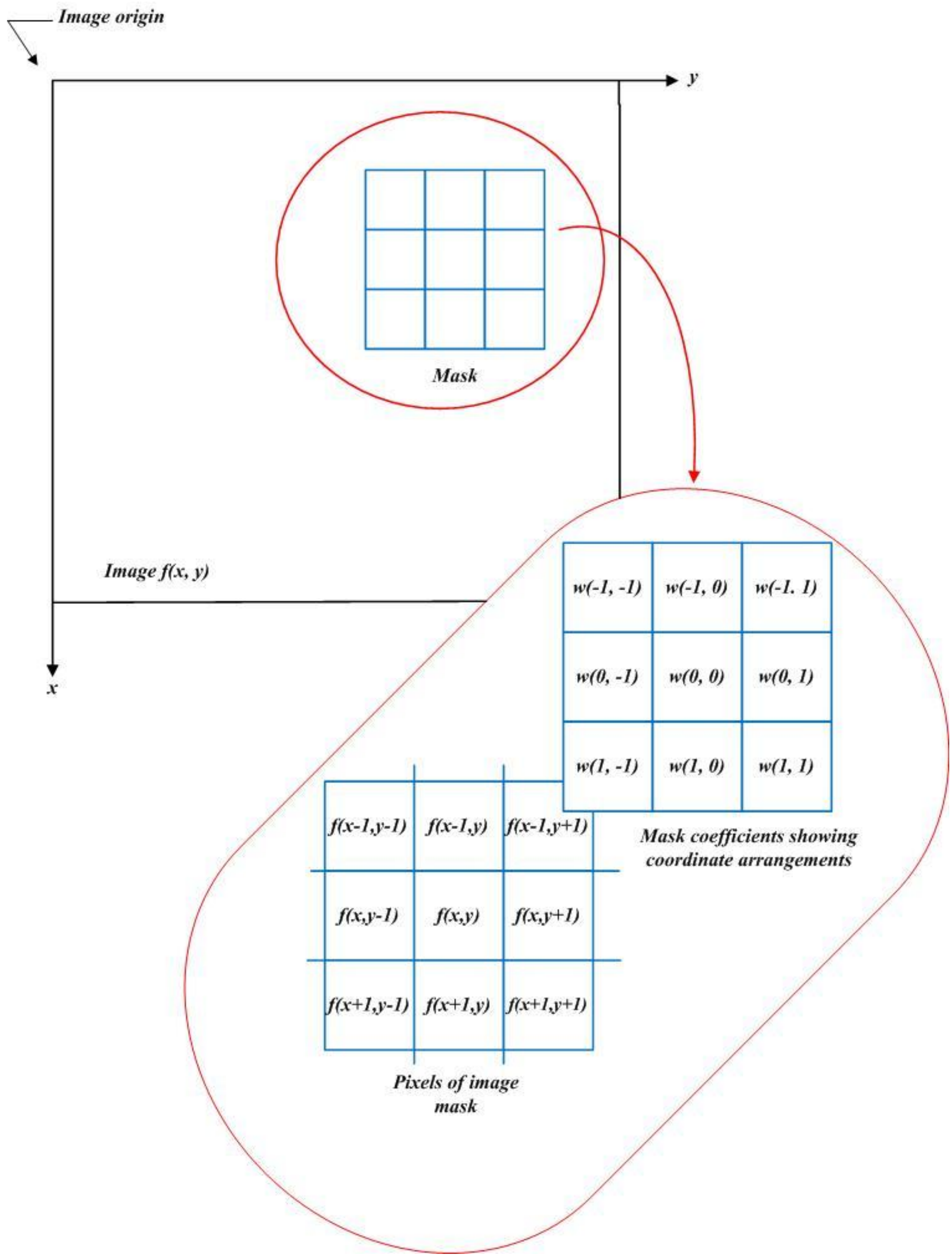
Image origin
y
Mask
Image f(x, y)
x
w(-1, -1)
w(-1, 0)
w(-1. 1)
w(0, -1)
w(0, 0)
w(0, 1)
w(1, -1)
w(1, 0)
w(1, 1)
Mask coefficients showing coordinate arrangements
f(x-1,y-1)
f(x-1,y)
f(x-1,y+1)
f(x,y-1)
f(x,y)
f(x,y+1)
f(x+1,y-1)
f(x+1,y)
f(x+1,y+1)
Pixels of image mask

Figure 8:The mechanics of spatial filtering.[3]

2.3.1 Smoothing spatial filters

Smoothing filters are used for blurring and for noise reduction. Blurring is used in preprocessing steps, such as removal of small details from an image prior to (large) object extraction, and bridging of small gaps in lines or curves. Noise reduction can be accomplished by blurring with a linear filter and also by non-linear filtering. **[3]**

2.3.2 Sharpening spatial filters

Sharpening is one of the most wonderful conversions, if you apply to an image; it brings out the image details that were not shown before. Usually if we mention the term “sharpening “ we recall edges, essentially it is to emphasize edges and make them more prominent so they will be easier for the eye to pick out, and no new details are added or created, Sharpening is a simple illusion. The main objective of sharpening is to highlight significant detail in an image or to enhance detail that has been blurred either in error or natural effect of a particular method of the image acquisition. Implementations of image sharpening vary and include applications ranging from electronic printing and medical imaging to industrial inspection and autonomous guidance in military systems.

The image blurring could be accomplished in the spatial domain by pixel averaging in a neighborhood. Since averaging is analogous to integration, it is logical to conclude that sharpening could be accomplished by spatial differentiation. This, in fact, is the case, and the discussion in this section deals with various ways of defining and implementing operators for sharpening by digital differentiation. Fundamentally, the strength of the response of a derivative operator is proportional to the degree of discontinuity of the image at the point at which the operator is applied. Thus, image differentiation enhances edges and other discontinuities and deemphasizes areas with slowly varying gray-level values. **[1][3]**

3 Frequency domain techniques

Frequency domain methods are based on modification of Fourier transform of an image. Let $g(x, y)$ be an image formed by the convolution of an image $f(x, y)$ and a position invariant operator $h(x, y)$

$$g(x, y) = h(x, y) * f(x, y)$$

From convolution theorem, we have

$$G(u, v) = H(u, v) * F(u, v)$$

Where *G, H, F* are *F T* of *g, h* and *f* respectively. The transform $H(u, v)$ is referred to as the transfer function of the process.

We are interested in the frequency domain relation because discrete convolution is often more efficiently carried out in the frequency-domain using fast Fourier transform algorithm. In discrete convolution wraparound error (aliasing) can be taken care of by assuming the functions to be periodic with periods chosen in a specified way. In an image, wraparound error gives rise to distortion around the edges. In practice this error is tolerable, even when images are not extended as required.

In a typical image enhancement problem $f(x, y)$ is given and the goal after computation of $F(u, v)$ is to select $H(u, v)$ so that the desired image given by:

$$g(x, y) = F^{-1}\{H(u, v)F(u, v)\}$$

exhibits some highlighted features of $f(x, y)$ For example edges in $f(x, y)$ can be accentuated by using a function $H(u,v)$ which emphasizes the high frequency components of $F(u,v)$.**[3]**

3.1 Smooting Frequency-Domain Filters

Edges and other sharp transitions (such as noise) in the gray levels of an image contribute significantly to the high-frequency content of its Fourier transform. Hence smoothing (blurring) is achieved in the frequency domain by attenuating a specified range of high-frequency components in the transform of a given image.

We consider three types of low-pass filters: ideal, Butterworth Gaussian filters. These three filters cover the range from very sharp (ideal) to very smooth (Gaussian) filter functions. The Butterworth filter has a parameter, called the filter order. For high values of this parameter the Butterworth filter approaches the form of the ideal filter. For lower-order values, the Butterworth filter has a smooth form similar to the Gaussian filter. Thus, the Butterworth filter may be viewed as a transition between two "extremes". **[1]**

3.1.1 Ideal low-pass filter

The simplest low-pass filter we can envision is a filter that "cuts off" all high-frequency components of the Fourier transform that are at a distance greater than a specified distance, from the origin of the (centered) transform. Such a filter is called a two-dimensional ideal low pass filter and has the transfer function:

$$H(u,v) = \begin{cases} 0 & \text{if } D(u,v) \le D_0 \\ 1 & \text{if } D(u,v) > D_0 \end{cases}$$

Where D_0 is a specified nonnegative quantity, and $D(u,v)$ is the distance from the point to the center of the frequency rectangle. **[3]**

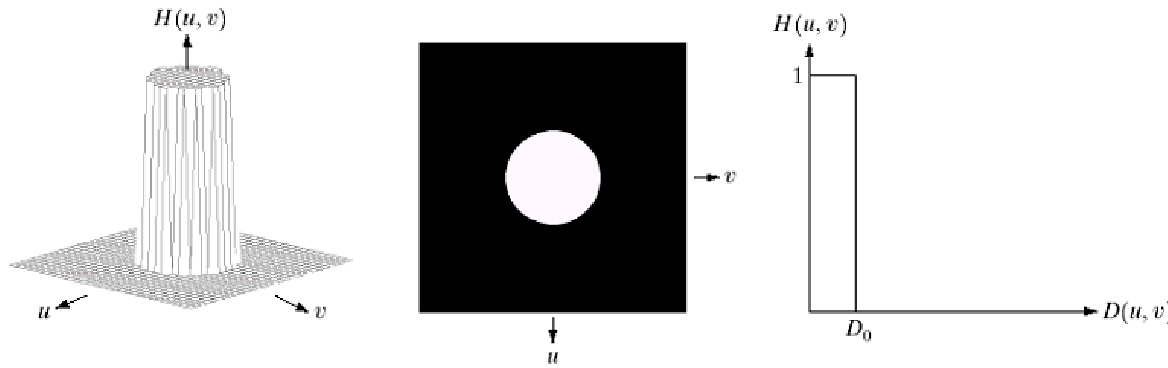

Figure 9: (a) Perspective plot of an ideal low-pass filter function. (b) Filter displayed as an image. (c) Filter radial cross section.[3]

3.2 Sharpening Frequency Domain Filters

To sharpen the image in the frequency domain, it can be accomplished by high pass filtering process which attenuates low frequency components without disturbing high frequency information in the Fourier transform. We intend in this filter is to perform precisely the reverse operation of the ideal low pass filter, the transfer function of the HPF can be expressed by this relation:

$$H_{hp}(u,v) = 1 - H_{lp}(u,v)$$

Where $H_{lp}(u,v)$ is the transfer function of the corresponding low pass filter, that's when: the low pass filter attenuates frequencies, the high pass filter pass them, and vice versa. So high-pass filter is a filter that passes high frequencies well, but attenuates (reduces the amplitude of) frequencies lower than the cutoff frequency. The actual amount of attenuation for each frequency varies from filter to filter; sometimes it's called a low-cut filter.

It is useful as a filter to block any unwanted low frequency components of a complex signal while passing the higher frequencies; high pass filter technique works so well at sharpening images is because any areas in the image which are not an edge are left untouched. The main areas that have sharpening applied to them are the edges, which is exactly what we need. In this section we concentrate on Ideal, Gaussian and Butterworth high pass filters.Butterworth filter represents a transition between the sharpness of the ideal filter and the total smoothness of the Gaussian filter, to illustrate of these filters. **[3]**

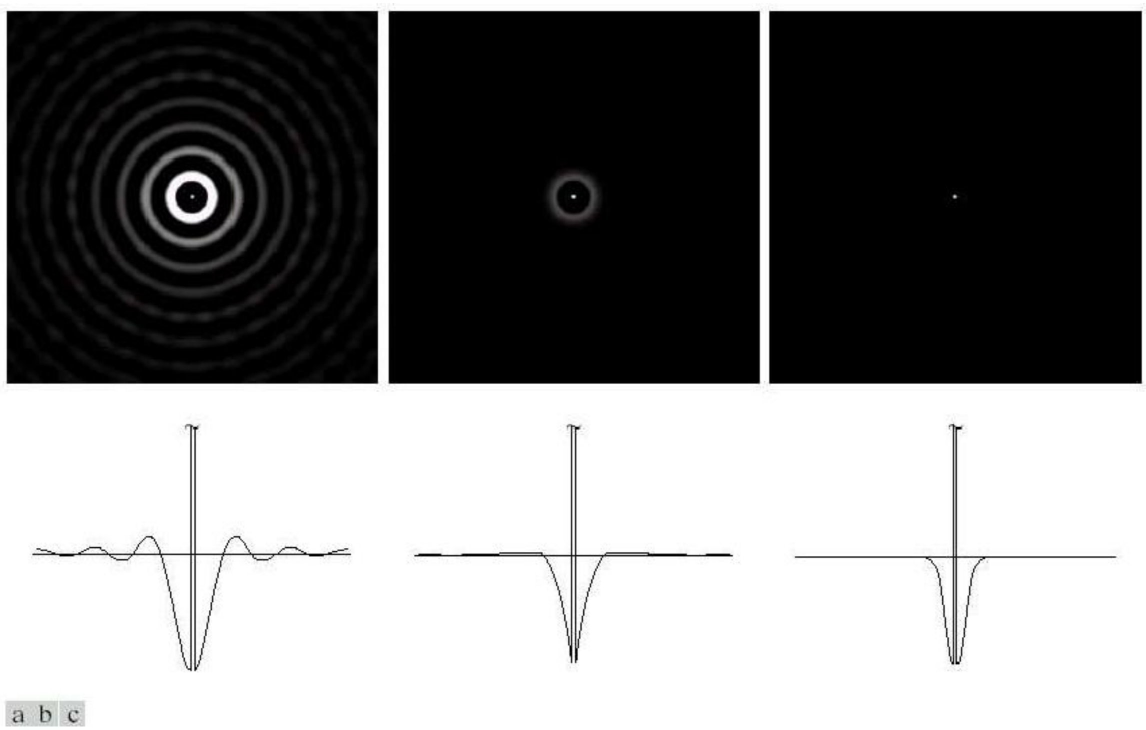

Figure 10: Spatial representation of typical (a) ideal. (b) Butterworth and (c) Gaussian frequency domain high-pass filters, and corresponding the grey-level profiles.[3]

3.2.1 Ideal high-pass filter

High pass filtering means that we filter away the low frequencies of something, and let the high frequency bands pass. In image terms, this means that the detail of an image is kept, while the larger scale gradients are removed. Luckily, it's not as complicated as it sounds.

Ideal high-pass filter IHPF is defined as:

$$H(u,v) = \begin{cases} 0 & \text{if } D(u,v) \le D_0 \\ 1 & \text{if } D(u,v) > D_0 \end{cases}$$

Where D_0 is the cut off distance measured from the distance from the origin of the frequency rectangle, and $D(u,v)$ is the distance from the origin of the Fourier transform, u & v are frequency variables of the Fourier transform. This filter is the opposite of the Ideal low pass filter that is zeroing all frequencies inside the circle of radius D_0 while passing without attenuation any frequency outside the circle. **[3]**

3.2.2 Butterworth high-pass fitler

The transfer function of the Butterworth high-pass filter (BHPF) of order n and cutoff frequency locus at distance D_0 from the origin is given by

$$H(\mu,v) = \frac{1}{1+\left[D_u / \ D_0(\mu,v)\right]^{2n}}$$

As in the case of a low-pass filters, we can expect Butterworth high-pass filters to behave smoother than IHPFs. **[3]**

3.2.3 Gaussian high-pass filter

The transfer function of the Gaussian high-pass filter (GHPF) with cutoff frequency locus at a distance D_0 from the origin given by

$$H(u,v) = 1 - e^{-D^{2(u,v)}/2D^2}$$

The results obtained are smoother than with the previous to filters Even the filtering of the smaller objects and thin bars is cleaner with Gaussian filter. **[3]**

3.2.4 Unsharp Masking

The Unsharp Mask filter sharpens edges of the elements without increasing noise or blemish. It is the king of the sharpen filters. A method by which all edges in the image are exaggerated, which produces more detail in the reproduction. In digital unsharp masking, adjoining pixel values are evaluated to locate the edges. When an edge is detected, the software exaggerates the edges by altering the value in two adjoining pixels in opposite directions, thereby increasing the edge contrast. This technique commonly used in the printing industry for crispening of edges. A signal proportional to the unsharp or low pass filtered version of the image is subtracted from the image. This is equivalent to adding a high pass signal to the image:

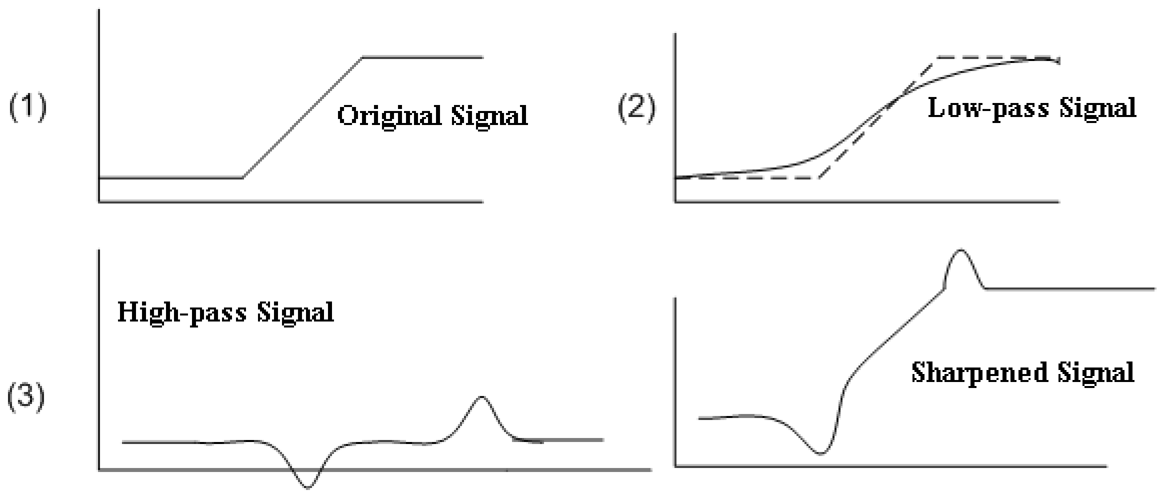

Figure 11: Unsharp Masking Operation. [1]

3.2.5 The Laplacian filtering

It is common for a single image to contain edges having widely different sharpness's and scales, from blurry and gradual to crisp and abrupt. Edge scale information is often useful as an aid toward image understanding. For instances, edges at low resolution tend to indicate gross shapes, whereas texture tends to become important at higher resolutions. An edge detected over a wide range of scale is more likely to be physically significant in the scene than an edge found only within a narrow range of scale. Furthermore, the effects of noise are usually most deleterious at the finer scales.

The Gaussian smoothing operation serves to band-limit the image to a small range of frequencies, reducing the noise sensitivity problem when detecting zero crossings. The image is filtered over a variety of scales and the Laplacian zero crossings are computed at each. This produces a set of edge maps as a function of edge scale. Each edge point can be considered to reside in a region of scale space, for which edge point location is a function of x, y and s. Scale space has been successfully used to refine and analyze edge maps.

The Gaussian has some very desirable properties that facilitate this edge detection procedure. First, the Gaussian function is smooth and localized in both the spatial and frequency domains, providing a good compromise between the need for avoiding false edges and for minimizing errors in edge position. . In fact, the Gaussian is the only real valued function that minimizes the product of spatial and frequency domain spreads. The Laplacian of Gaussian essentially acts as a bandpass filter because of its differential and smoothing behavior. Second, the Gaussian is separable, which helps make computation very efficient.

Omitting the scaling factor, the Gaussian filter can be written

$$g_c(x,y) = \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

Its frequency response $G(\Omega_x\Omega_y)$ is also Gaussian:

$$G(\Omega_x\Omega_y) = 2\pi\sigma^2 \exp\left(-\frac{\sigma^2}{2}\left(\Omega_x^2+\Omega_y^2\right)\right)$$

The σ parameter is inversely related to the cutoff frequency.

Because the convolution and Laplacian operations are both linear and shift invariant, their computation order can be interchanged.

$$\nabla^2\left[f_c[x,y] * g_c(x,y)\right] = \left[\nabla^2 g_c(c,y)\right] * f_c(x,y)$$

Here we take advantage of the fact that the derivative is linear operator. Therefore, Gaussian filtering following by differentiation is the same as filtering with the derivative of a Gaussian. The right-hand side of the above equation usually provides for more efficient computation since $\left[\nabla^2 g_c(c,y)\right]$ can be prepared in advance as a result of its image independence. The Laplacian of Gaussian (LoG) filter, $h_c(x,y)$ therefore has the following impulse response. **[1][3][6]**

$$h_c(x,y) = \nabla^2 g_c(x,y) = \frac{x^2+y^2-2\sigma^2}{\sigma 4}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

3.2.6 High-boost filtering

Often it is desirable to emphasize high frequency components representing the image details without eliminating low frequency components. In this case, the high-boost filter can be used to enhance high frequency component while still keeping the low frequency components

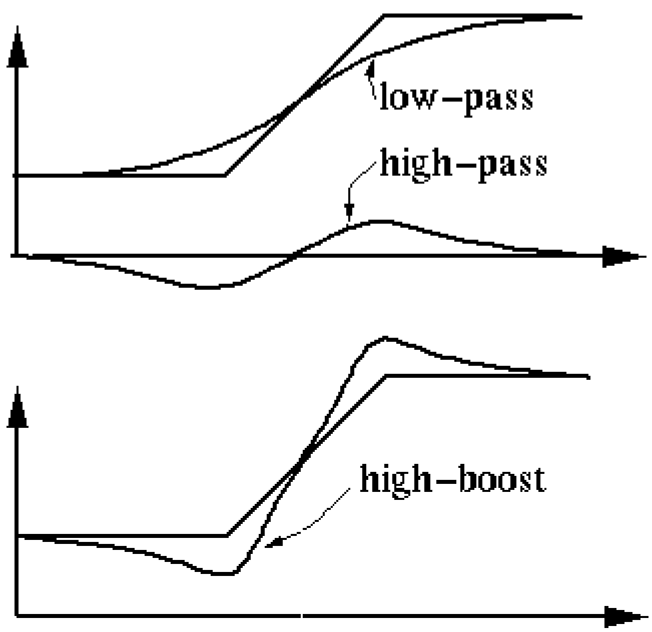

Figure 12: showing how High-Boost Filtering technique works.

Unsharp masking consists of generating a sharp image by subtracting from an image a blurred version of itself. Using frequency domain terminology, this means obtaining a high-pass filtered image by subtracting from the image a low-pass filtered version of itself, and that is:

$$f_{hp}(x,y) = f(x,y) - f_{lp}(x,y)$$

High-boost filtering generalizes this by multiplying $f(x,y)$ by a constant $A \geq 1$:

$$f_{hb}(x,y) = Af(x,y) - f_{hp}(x,y)$$

Thus, high boost filtering gives us the flexibility to increase the contribution made by the image to the overall enhanced result:

$$f_{hb}(x,y) = (A-1)f(x,y) + f(x,y) - f_{lp}(x,y)$$

or

$$f_{hb}(x,y) = (A-1)f(x,y) + f_{hp}(x,y)$$

This result is based on a high-pass rather than a low pass image. When A =1, high boost filtering reduces to regular high-pass filtering. As A increases past 1, the contribution made by the image itself becomes more dominant. **[1][3]**

4 Color image enhacement

Images and videos colored are usually displayed in the RGB color space. For example JPEG requires each color component to be handled separately, as we can see in the figures below.

R = red, G = green, B = blue.

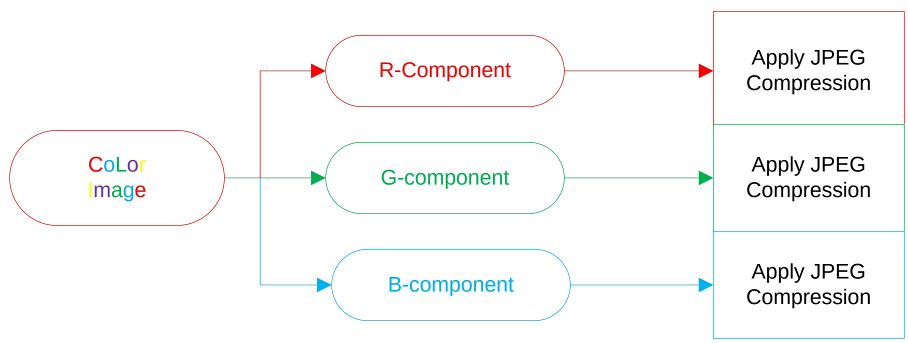

Figure 13: RGB color image [3]

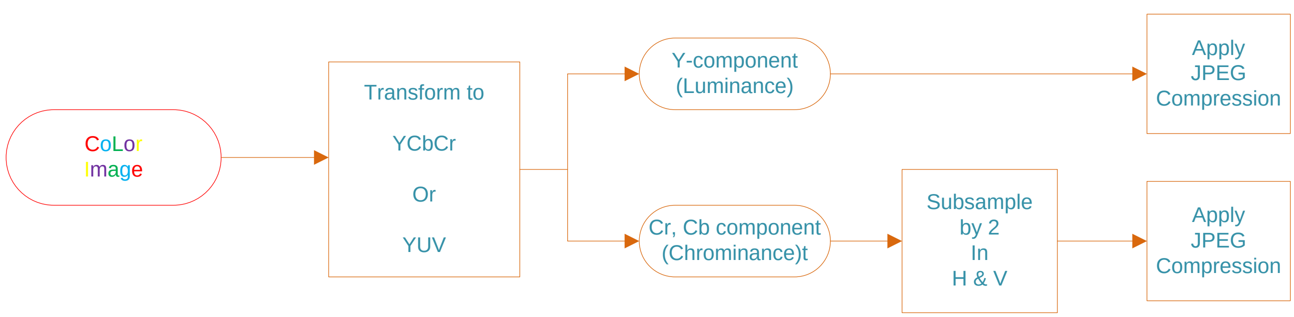

Figure 14: Transform color image to YCbCr [3]

Where

$$Y = 0.299(R - G) + G + 0.114\,(B - G)$$

$$Cb = 0.564\,(B - Y)$$

$$Cr = 0.713 (R – Y)$$

The second figure represents a block diagram of color image transforming into another representation which it is chrominance and luminance form (YCbCr). YCbCr is the most commonly used color coordinate system for the compression of image and video signals. It is divided into two components: Y components and Cb, Cr components, where Y is the luminance and Cb, Cr are the chrominance. Before applying the JPEG compression to chrominance component, we need to sub-sample it by two then we can apply a compression method. **[3]**

4.1 Natural Color Image Enhancement

The monochrome image enhancement methods described previously can be applied to natural color images by processing each color component individually. This comprises the class of intra component processing algorithms. There is also a class of inter component processing algorithms in which color pixels are combined on a pixel-by-pixel basis. Finally, there is a class of vector processing algorithms. **[5]**

4.1.1 Intra component Processing

Typically, color images are processed in the *RGB* color space. This approach works quite well for noise cleaning algorithms in which the noise is independent between the *R*, *G* and *B* components. Edge crispening can also be performed on an intra component basis, but better, and more efficient, results, are often obtained by processing in other color spaces. Contrast manipulation and histogram modification intra component algorithms often result in severe shifts of the hue and saturation of color images. Hue preservation can be achieved by using a single point transformation for each of the three *RGB* components. For example, form a sum image, and then compute a histogram equalization function, which is used for each *RGB* component.

For some image enhancement algorithms, there are computational advantages to processing in a luma-chroma space, such as, or a lightness-chrominance space, such as**. [5]**

4.1.2 Inter component Processing

The intra component processing algorithms previously discussed provide no means of modifying the hue and saturation of a processed image in a controlled manner. One means of doing so is to transform a source *RGB* image into a three component image, in which the three components form separate measures of the brightness, hue and saturation (*BHS*) of a color image. Ideally, the three components should be perceptually independent of one another. Once the *BHS* components are determined, they can be modified by amplitude scaling methods**. [5]**

4.1.3 Color Vector Processing

A color vector can be formed in three-dimensional color space based upon the *R*, *G* and *B* color components at each pixel (*j*, *k*). Now consider a moving window about the (*j*, *k*) pixel, which contains a sequence of color vectors **v**1, **v**2,...,**v***N*. For example, for a window, the neighborhood array is: For natural, noise-free images with a relatively small window, the vectors **v***n* will be similar in magnitude and direction. For images subject to noise, some of the vectors may differ significantly from one another. **[5]**

5 Matlab implementation

As its mentioned at the beginning, the taske is divided into two parts: theoritical and prcatical. The theoretical part is already done, therefore this part contains the practical part and drescirption of Image Processing Toolbox in Matlab, and my work on Matlab and Graphical user face (GUI). Matlab software has many functions to apply in image processing. How to manipulate the program depending on the user but must be practically knows what item in Matlab program will be used. Creativity in Matlab can make interesting result. Even, the complex data can be solved in Matlab. Especially when the data involved is very complex. Here, we can create some image from converting data by using some programs in it, which just applied all procedure in the Matlab program. Toolbox is a tool for making mathematical calculations. Image processing toolbox is user friendly programming language with feature more advanced. In the program also used the GUI (Graphical User Interface) to create develop the program.

5.1 Image Processing Toolbox in Matlab

Image Processing Toolbox is a wide set of algorithms that deal with images. It supports almost any type of image file. It gives the user unlimited options for processing of pictures. There are functions responsible for image enhancement, deblurring, filtering, noise reduction, spatial transformations, creating histograms, changing the threshold, hue and saturation, also for adjustment of color balance, contrast, and detection of objects and analysis of shapes.

'I' will mean exemplary input image. All descriptions are based on the website www.mathworks.com, which provides wide compendium of knowledge about all Matlab functions, including those from Image Processing Toolbox. First group of operations is responsible for changes and information concerning color transformation of images. Couples of functions can't change anything in the image but they are crucial when it comes to gain information about it, without need of opening the actual object of interests. Isbw(I) returns value 1 if the image is black&white, and value 0 otherwise. Some operations have sense only when executed on binary graphic files. For example adjusting contrast, brightness or other changes, usually made on colorful pictures, would not work with black&white images. Function isgray(I), similarly to previous one, checks colormap of the image. As the name suggests, this time function returns value 1 if the picture is grayscale and value 0 otherwise. It may also become useful while deciding if some operations can be performed on the file. Isrgb(I) informs if examined file is the RGB image. These three functions are essential when it comes to deciding about changing the colormap or color system. Knowing if the image is black&white, grayscale or RGB determines what transformations can be done to the file. There would be no point to try having some changes to image, if they are inoperative for some color models or maps. Command colormap 'map' is connected with the previously mentioned however it is not Image Processing Toolbox function. It exists in Matlab main library. Current image colormaps are set to one that stands in the brackets as a parameter. There is about twenty ready-built colormaps in Matlab.

5.1.1 Open, Save, and Display functions

Opening, closing, displaying and saving the image file are basic operations in Matlab. In addition the library of Matlab contains couple of useful commands. 'Imread' for example, deals with reading image from graphics file. As a parameter in the brackets it takes the name of the file and its extension. Among supported formats are bmp, gif, jpeg, png and tiff. Imread returns a two-dimensional array if the image is grayscale and a three-dimensional one, if the picture is color. The function mentioned above, allows also reading an indexed image and an associating colormap with it. In order to do that, instead of giving one variable as a result, user needs to put second variable that will stand for the map, just after the comma in square brackets. Complementary, function imwrite writes the image to the graphics file. It can support the file formats as we mentioned before, with the 'imread' command. Each file extension has its own syntax but there is one simple that works for most of them.

Usually 'imfinfo' displays various information about the image, among all the data fields, returned by this procedure there are nine of them that are the same with every file format. Those are:

• Filename – contains name of the image;

• FileModDate – last date of modification;

• FileSize – an integer indicating the size of the file, in bytes;

• Format – graphic file extension format;

• FormatVersion – number or string describing the file format version;

• Width – width of the image in pixels;

• Height – height of the image in pixels;

• BitDepth – number of bits per pixel;

• ColorType –indicates type of the image, either 'truecolor' for RGB image, 'grayscale' for grayscale image or 'indexed' for an indexed image.

Very professional and complex function, fspecial creates predefined filters that can be used while processing the image. Each parameter takes one value, which states the type of the filter. More interesting possibilities of them might be 'disk', 'motion' and 'unsharp'. Value 'disk' returns a circular averaging filter with a radius specified by the user. The default radius is 5. The result of using this filter is the picture becoming blurred. Motion filter answers with linear motion of a camera. User will determine the amount of pixels needed to be moved and the angle of motion. Default values returns the picture as it was blurred by 9 pixels of horizontal camera movement.

5.1.2 Graphical user interface (GUI)

Graphical user interface (GUI) is a graphical display in one or more windows containing controls, it can be also defined as a set of techniques and mechanisms used to creat interactive communication between a program and user called components that enable a user to perform interactive tasks.

Matlab provides a helpful tool called GUIDE, after choosing the button guide from Matlab tool bar, a quick start window appears. From the choice of exemplary positions it is recommended to pick 'Blank GUI'. In the new window it is possible to drag and drop each object into the area of the program

• Toggle Button – once pressed stays depressed and executes an action, after the second click it returns to the raised state and performs the action again;

• Check Box – generates an action when checked and indicates its state (checked or not checked), many options might be ticked in the same time;

• Radio Button – similar to the check box, but only one option can be selected at any given time, function starts working after the radio button is clicked;

• List box – displays a list of items and enables user to select one or more from them.

• Pop-up Menu – open a list of choices when the arrow is pressed.

5.2 Working on GUI:

In the figure below I have designed a graphical user interface, It will be responsible for loading the original image to the axes and then user will be able to use the technique he needs with parameters he can get a better result.

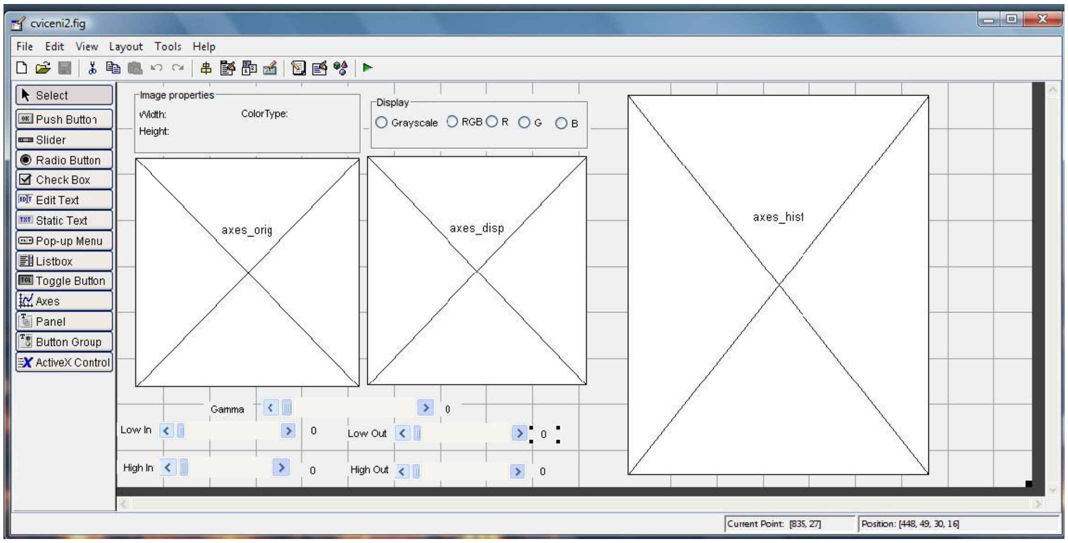

Figure 15: Building GUI on MATLAB

Menu item 'File' contains three elements, which are: Load, Save and Exit. Save with compression. It is necessary to change each element's label and tag property. The reason is enabling easier maintenance within the components. Second item Functions contains Histogram, Imadjust, and Spatial filters.

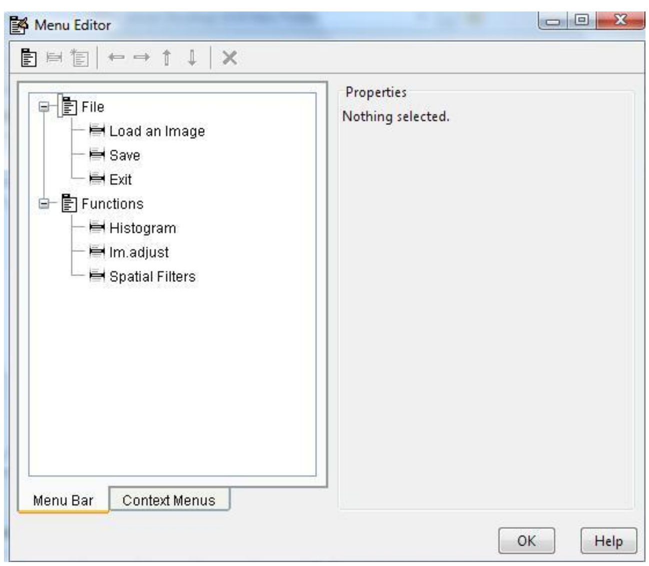

Figure 16: Menu Editor

Frist step we load the original image as it's shown in the figure 17,

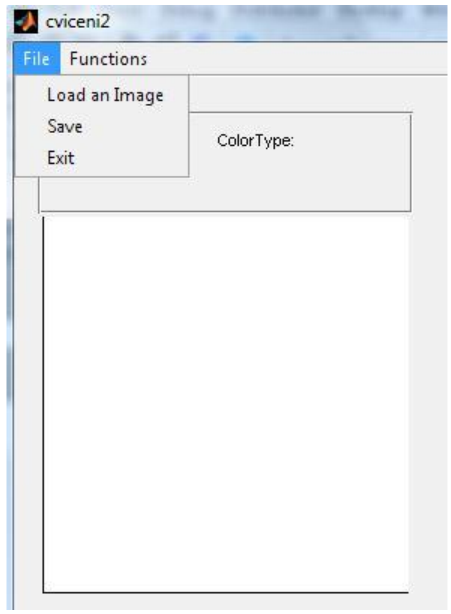

Figure 17: Menu in GUI and Loading Image.

After we load an image, GUI will read the properties of the image width, colortype and height, after we can chose the function for example the Histogram, it will show the display options grayscale, RGB , R, G and B.

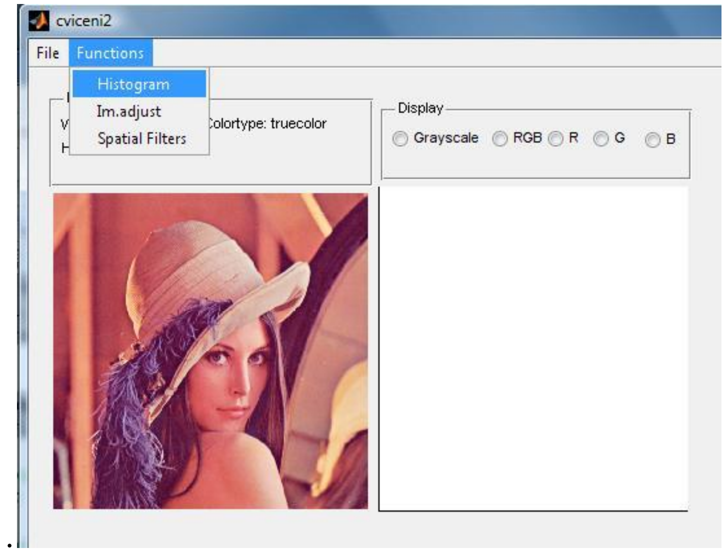

Figure 18: Histogram function in my work on GUI

For Example choosing the grayscale option we will get the the image in grayscale with it's hitsogram as well.

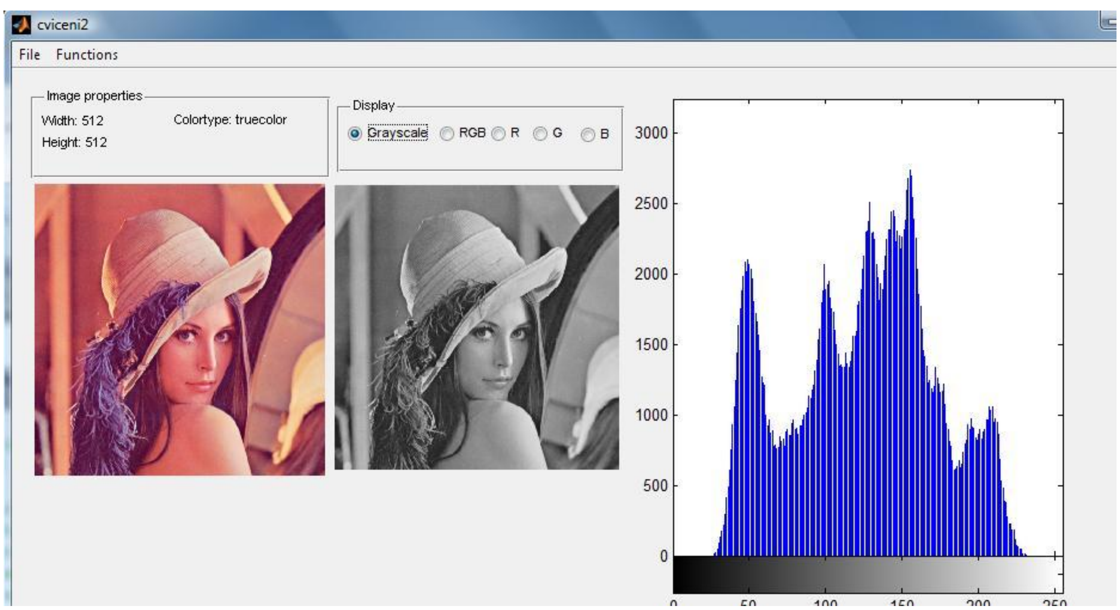

Figure 19: Example of Grayscale option in Histogram.

Figure 20, shows us the Imadjust function, so after loading the original image and the we chose the technique Imadjust, we will get changeable parameters as it shown below, according to parameters we choose we will get enhanced images, after we get the best result we want we can save the enhanced image as it shown in the right side of the figure.

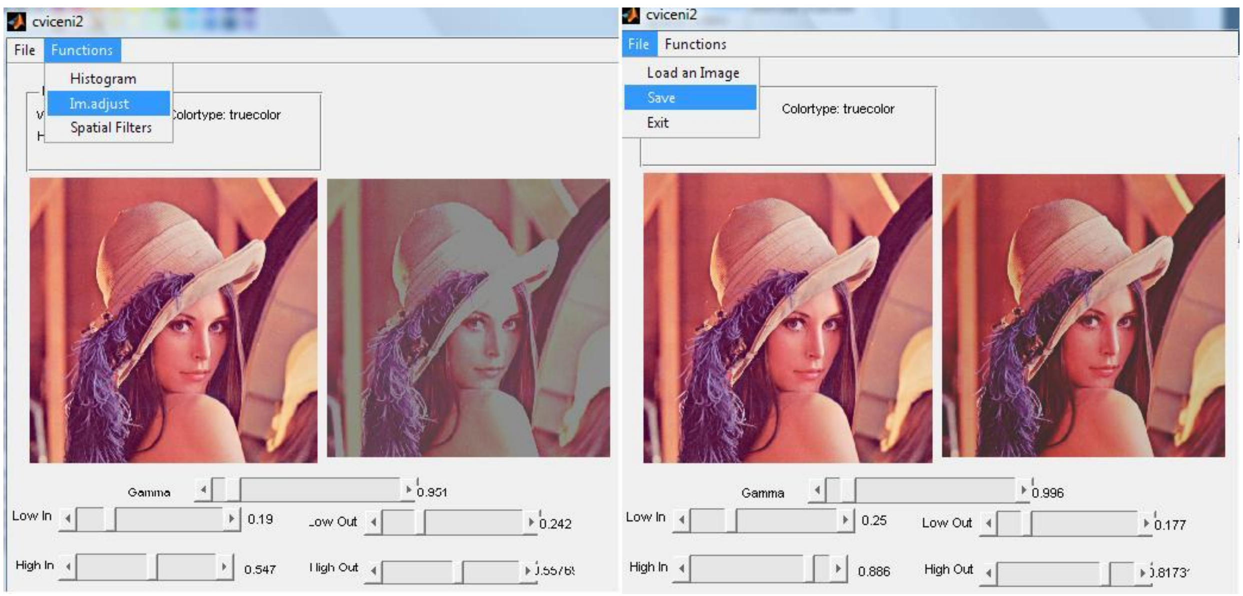

Figure 20: Imadjust function and its effect on Image by changing parameters.

5.2.1 Histogram method

The figure below shows an input of an image and easily we can compare it to the output image using the histogram method. For example for the output we can notice that between 0 and 150 it reach maximum 8000 where in the output it reach maximum 500.

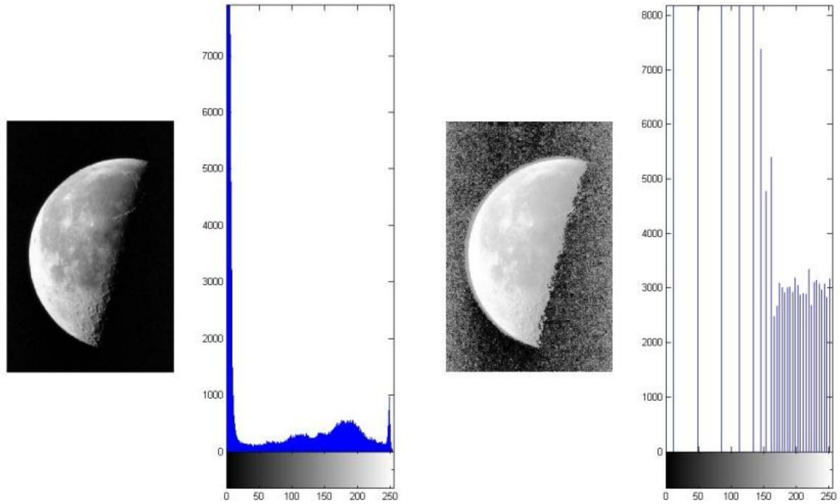

Figure 21: histogram method for a gray picture

I used another method for adjusting images but with difference gamma and we can notice the difference between these examples especially when the gamma equal to 2, 0.7 and the last one when gamma equals 1.

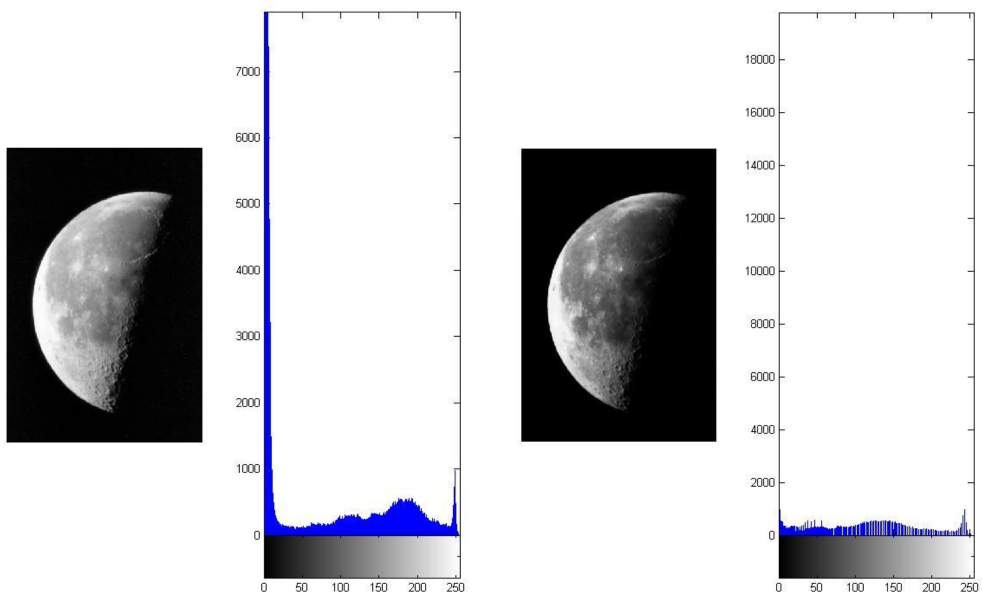

Figure 22: image adjusting for gamma =2

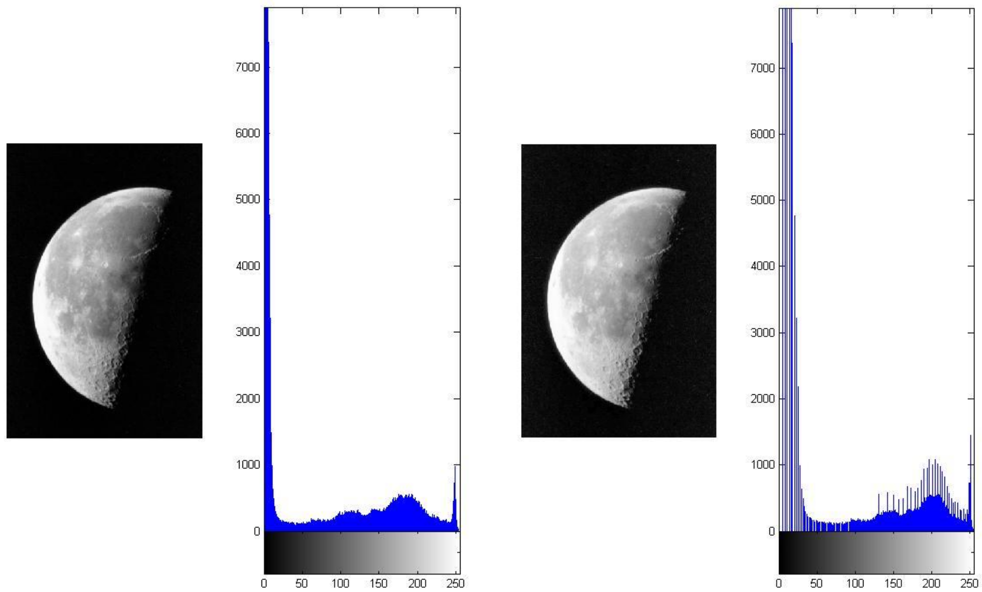

Figure 23: image adjusting for gamma= 0.7

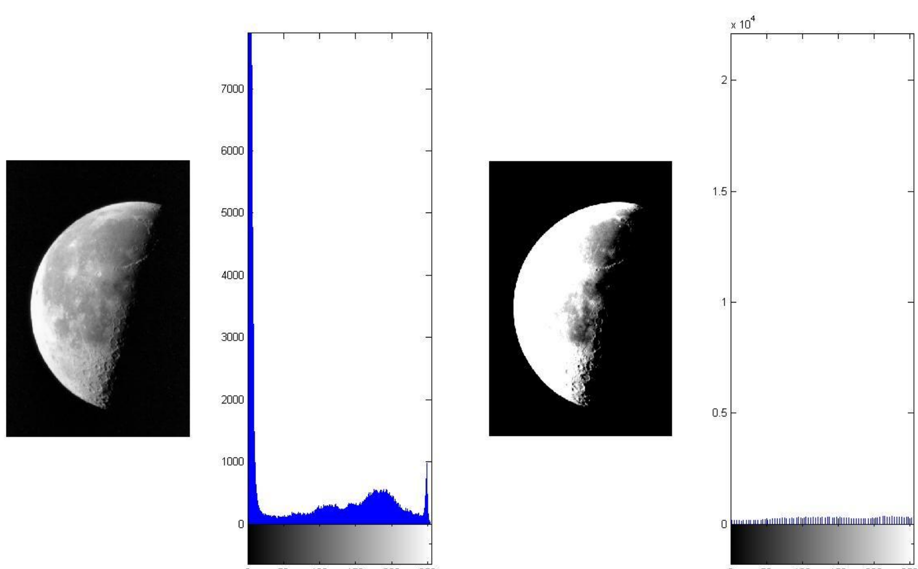

Figure 24: image adjusting where gamma = 1

5.2.2 Color Image Adjusting Histogram

Here I just applied the function of 'imadjust' for the color image, and as we see the difference in the colors berween the two pictures.

Figure 25:Original Image

Figure 26:imadjust effect on the image

5.2.3 F-special filters:

Create 2-D special filters.

Description: h = fspecial (*type*) creates a two-dimensional filter, h, of the specified *type*. fspecial returns h as a correlation kernel.

Applying for each one a different filter, the results are noticed in the picture below, so I used the Laplacian of Gaussian (LOG) filter, then the Sobel horizontal edge-emphasizing filter, and the last one was the averaging filter.

Figure 27: color image using Laplace transform

5.2.4 Color Image Enhancement:

Here is an example of color image engancement. The picture below is the input image that i used in this implementation.

Figure 28: input color image

The process of color image algorithms works in three different components and never works as all components in once. So I had to separate the RGB into R component, G component and B component. We can see the result in figure below.

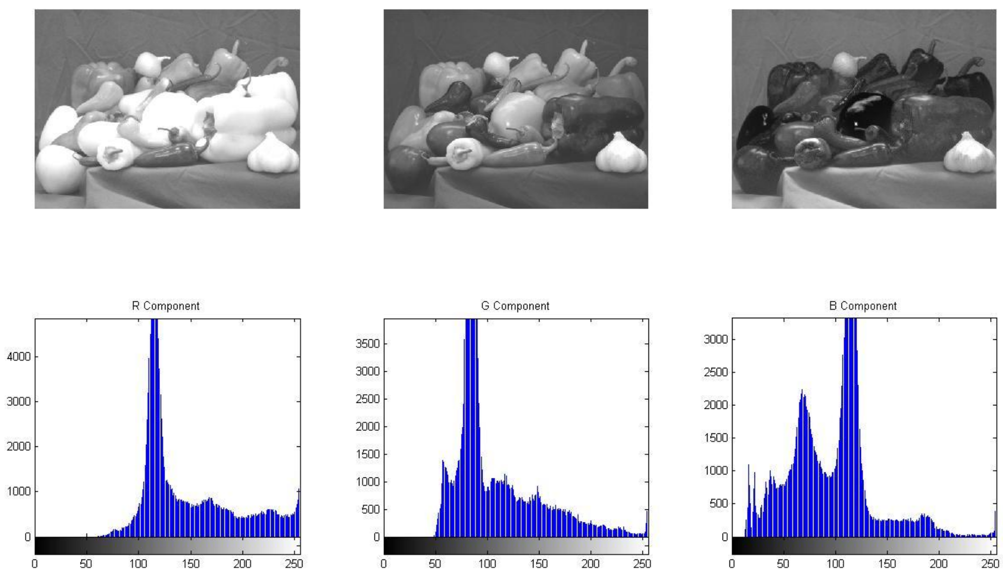

Figure 29: RGB components

Later I had to use some filters to make sure that the picture is saturated or at least to be perfect for the human eyes, and we can see how it is saturated in the result below.

As we see in the right corner the 4th imgae that I used the Unsharp masking filter for the saturated image, so the details of the images colors and edges and well enhanced and better results for the human eye.

6 Coclusion

My prnoject is about image enhancement, there are 2 parts, spatial domain and frequency domain. I focused on edge and color enhancement, especially on the sharpening techniques and I used its different types. And I had to use different examples of filters based on sharpening technique, and I compared them, and I got results of how the effect to enhance edge and color in the digital image. I have created GUI and done some other implantation on MATLAB for image processing.

During my work I have learn and gain a lot of information of image enhancement. However, like each student I met with some problems especially in programming, so I had to learn to program in Matlab and it takes time to learn.

I believe I fill task and I would like to focus and work more on this issue in the future and I hope I will improve more.

According to the result in Matlab, it seems that the sharpening techniques were the best, but it does not mean it is the best at all, because maybe using another filters or another methods will give a completely different result.

7 LITERATURE

[1] BOVIK, A. Handbook of Image and Video Processing, Elsevier Academic Press, New York, 2005.

[2] Matlab HELP [User's Guide - Image Processing Toolbox]. Natick (Massachusetts, USA): The MathWorks Inc., www.mathworks.com.

[3] DSP Book_ - Gonzalez e Woods - Digital Image Processing _2nd ed_ - Prentice Hall 2002.

[4] Bernd Jahne, Digital image processing, 5th revised and extended edition, Springer, Berlin 2002

[5] William k. Pratt, Digital Image Processing, fourth edition, wiley 2007.

[6] Prof. Sumana Gupta, Digital image processing, Indian Institute of technology., June 2009, http://nptel.iitm.ac.in/courses.php?branch=Ece

[7] Prof. P.K.Biswas, Department of Electronics and Communication Engineering, July 2009, http://nptel.iitm.ac.in/video.php?courseId=1079

[8] Russ, John C, The image Processing Handbook, Taylor and Francis LTD, Edition 5, 2007.