UNIVERSITY OF GAZİANTEP
GRADUATE SCHOOL OF
NATURAL & APPLIED SCIENCES

DESIGN OF THE FAST GAUSSIAN FILTERS FOR IMAGE PROCESSING APPLICATIONS

M. Sc. THESIS
IN
ELECTRICAL-ELECTRONICS ENGINEERING

BY
VEYSİ GÖKHAN
January 2018

JANUARY 2018

M. Sc. Thesis - Electrical-Electronics Engineering

VEYSİ GÖKHAN

Design of the Fast Gaussian Filters for Image Processing Applications

M.Sc. Thesis
in
Electrical-Electronics Engineering

University of Gaziantep

Supervisor
Prof. Dr. Arif NACAROĞLU

BY
Veysi GOKHAN
January 2018

REPUBLIC OF TURKEY
UNIVERSITY OF GAZİANTEP
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
ELECTRICAL-ELECTRONICS ENGINEERING
DEPARTMENT

Name of the thesis: Design Of The Fast Gaussian Filters For Image Processing Applications

Name of student: Veysi GÖKHAN

Exam date: 25.01.2018

Approval of the Graduate school of Natural and Applied sciences

Prof.Dr. A. Necmeddin YAZICI
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science

Prof.Dr. Ergün ERÇELEBİ
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Prof. Dr. Arif NACAROĞLU
Supervisor

Examining Committee Members	Signature
Prof. Dr. Sadettin ÖZYAZICI	………………………
Prof. Dr. Arif NACAROGLU	………………………
Assist. Prof. Dr. Nurdal WATSUJI	………………………

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Veysi GÖKHAN

ABSTRACT

DESIGN OF THE FAST GAUSSIAN FILTERS FOR IMAGE PROCESSING APPLICATIONS

GÖKHAN, Veysi

M.Sc. in Electrical-Electronics Engineering

Supervisor: Prof. Dr. Arif NACAROĞLU

January 2018, 49 Pages

Image processing is a method used for execution of some operations on image transfered into the difital medium by transducer. Digital image may be processed in order to increase the quality of the picture as well as to get some information from the Picture such as identification. But the application area of the image processing is not limited only for enhancement. It can be used in many different areas like engineering applications, medical measurements, digitization of the information, space studies, natural sources surveys taking the satellite images, security, film industry, military applications like target detection, otonomous aerial vehicles, night vision, etc.

In this thesis, Fast Gaussian filter is studied for its application to image processing. The coefficients of the Gaussian filter are changed by a specific algorithm, the changes in the images are observed for image processing applications. The attension of this study is focused on to obtain the most suitable coefficient for each image.

Keywords: Filtering, Image Processing, Filter Design, Gaussian Filter, Fast Gauss Filter, Image processing techniques, Matlab.

ÖZET

GÖRÜNTÜ İŞLEME UYGULAMALARINDA HIZLI GAUSS FİLTRE TASARIMI

GÖKHAN, Veysi

Yüksek lisans tezi, Elektrik-Elektronik Mühendisliği

Danışman: Prof. Dr Arif NACAROĞLU

Ocak 2018, 49 Sayfa

Görüntü işleme, bir görüntüyü sayısal yapı haline getirmek ve sayısal ortamdaki görüntü üzerinde bazı işlemleri gerçekleştirmek için bir metod olarak kullanılabilir. Böylece daha gelişmiş bir görüntü elde edebilir veya görüntünün içeriğinden bazı faydalı bilgiler çıkarılabilir. Görüntü işleme yöntemini tıp, uzay çalışmaları, uzak yeryüzü kaynakları araştırmaları, güvenlik, mühendislik, film efektleri, yayıncılık, sanat, spor, belgelerin sayısallaştırılması, askeri uygulamalar, insansız hava araçları, gece görüşü gibi birçok alanda kullanılmaktadır.

Bu çalışmada, görüntü işleme uygulamalarında hızlı Gauss filtre tasarımı ve kullanılması incelenmiştir. Görüntü işleme uygulamaları için Gauss filtrenin tasarımında katsayıları belirli bir metod ile değiştirildiğinde görüntülerdeki değişimler gözlemlenmektedir. Her bir görüntü için en uygun katsayının ne olması gerektiği ile ilgili çalışmalar bu tezin ana konusunu oluşturmuştur.

Anahtar Kelimeler: Filtreleme, Görüntü İşleme, Filtre Tasarımı, Gaussian Filtre, Hızlı Gauss Filtresi, Görüntü işleme teknikleri, Matlab.

To my beloved family…

ACKNOWLEDGEMENTS

I would like to express my highest appreciation to my supervisor Prof. Dr. Arif NACAROĞLU for his continual encouragement and supervision.

I would like to thank my family and my friends for their continuous moral & material support and encouragement.

CONTENTS

Page

ABSTRACT .... **v**

ÖZET .... **vi**

ACKNOWLEDGEMENTS .... **viii**

CONTENTS .... **ix**

LIST OF FIGURES .... **xi**

LIST OF TABLES .... **xiii**

LIST OF SYMBOLS .... **xiv**

CHAPTER 1 .... **1**

INTRODUCTION **1**

1.1 Image filtering 1

1.2 Applying filter to image 1

1.3 Applying gaussian filter to image 1

CHAPTER 2 .... **3**

LITERATURE REVIEW AND BACKGROUND **3**

2.1 Definition of pixel 3

2.2 The meaning of images 3

2.2.1 Picture 3

2.2.2 Disk image 4

2.3 Structure of images 4

2.4 Color 6

2.5 What are signals? 6

2.6 Why use filters? 8

2.7 Classes of filters 8

2.7.1 Frequency choosing filters 8

2.7.2 Statistical filters 8

2.7.3 Adaptive filters 8

2.7.4 Digital filters 9

2.7.5 Analog filters 9

2.8 Two-dimensional filters 9

2.9 IIR and FIR filtering 10
2.9.1 Infinite impulse response (IIR) filtering 10
2.9.2 Finite impulse response filtering 10
2.9.3 Properties of FIR and IIR filters 11
2.10 Mean Filter 11
2.10.1 Arithmetic mean filter 12
2.10.2 Geometric mean filter 12
2.10.3 Median filter 12
2.10.4 Comparison of median and mean filters 13
2.10.5 Frequency filters 14
CHAPTER 3 18
GAUSSIAN FILTERS 18
3.1 Gaussian smoothing 18
3.1.1 Canny edge detection 22
3.1.2 Standart deviation 23
3.1.3 Laplacian of Gaussian 24
3.2 Unsharp filters 26
CHAPTER 4 29
DESIGN OF THE FAST GAUSSIAN FILTER FOR IMAGE PROCESSING APPLICATIONS 29
4.1 Image filtering 29
4.1.1 Contrast improvement 30
4.1.2 Template matching 30
4.1.3 Smoothing 31
4.2 Basic informations about image processing in Matlab 32
4.3 Implementing Gaussian Filter 34
CHAPTER 5 48
CONCLUSIONS 48
5.1 Conclusion 48
REFERENCES 49

LIST OF FIGURES

Page

Figure 1.1 Techniques of image processing 2

Figure 2.1 The first appearance of Picture element [1] 3

Figure 2.2 Digital image structure of the planet Venus [3] 5

Figure 2.3 Matrix structure of a colorful image 7

Figure 2.4 Waveform and spectrum of sinusoidal signal [4] 7

Figure 2.5 Block diagram of FIR and IIR filters[14] 11

Figure 2.6 The median value of a pixel neighborhood is calculated 13

Figure 2.7 Ideal lowpass in frequency and spatial domain 16

Figure 3.1 1-D Gaussian distribution (mean=0 and $\boldsymbol{\sigma}$=1) 18

Figure 3.2 2-D Gaussian circulation while mean (0,0) and $\boldsymbol{\sigma}$ =1 19

Figure 3.3 Discrete approximation to Gaussian function with $\boldsymbol{\sigma}$ = 1.0 20

Figure 3.4 Position of pixeles for Gaussian function [16] 21

Figure 3.5 Distribution of the Gaussian function values 23

Figure 3.6 The 2-D Laplacian of Gaussian (LoG) function [17] 25

Figure 3.7 Spatial sharpening [18] 26

Figure 3.8 Calculating an edge image for unsharp filtering [18] 27

Figure 3.9 Sharpening the original signal using the edge image.[18] 27

Figure 3.10 The complete unsharp filtering operator [18] 28

Figure 4.1 The image used in this study (ferit.jpg) 32

Figure 4.2 Gray image of (ferit.jpg) 33

Figure 4.3 Original image (ferit.jpg) 36

Figure 4.4 The image that convolved with 5X5 Gaussian Kernel 37

Figure 4.5 Gaussian filter applied with built_in functions 38

Figure 4.6 Original image 41

Figure 4.7 Math. algorithms applied, if pixel value > 0.3 change it with 5 41

Figure 4.8 Math. algorithms applied, if pixel value < 0.3 change it with 0.1 42

Figure 4.9 Original image 43

Figure 4.10 Gaussian filter applied image 44

Figure 4.11 After Gaussian filter, if pixel value>0.3 change it with 5 44

Figure 4.12 After Gaussian filter, if pixel value < 0.3 change it with 0.1 45

Figure 4.13 Math. algorithms applied and if pixel value >0.3, change it with 5.........46

Figure 4.14 Gaussian filter applied and if pixel value > 0.3, change it with 5 47

LIST OF TABLES

Page

Table 2.1 General information about mean and median filters 13

Table 4.1 Characteristics of smoothing filters 29

Table 4.2 Details of an image 34

LIST OF SYMBOLS

1-D	One-Dimensional
2-D	Two-Dimensional
3-D	Three-Dimensional
PEL	Picture Element
PX	Pixel
LMS	Least-Mean-Square
RLS	Recursive-Least-Square
BIBO	Bounded Input Bounded Output
IIR	Infinite Impulse Responce
FIR	Finite Impulse Responce
ARMA	Autoregressive Moving Average
Fs	Sampling Frequency (clock frequency of the digital filter)
N	Number of Delay Register
σ	Standard Deviation
x0, y0	Optional Pixel Positions
r	Filter Radius
M	Box Width
L(x,y)	Laplacian of an image
I(x,y)	pixel intensity values
LoG	Laplacian of Gaussian
K	Scaling Constant

CHAPTER 1
INTRODUCTION

1.1 Image filtering

Image filtering is a most important part of the image processing. Images transfered in to digital medium have original and main information and unfortunately noise created by many different sources in envioremment. Human brain has a capacity of identification of the event observing less information than presented by the picture. The summary of the picture which is enough to understand the content without noise may be obtained using generally linear and nonlinear filters in addition to the use of some decision algoritms. The data can include a large amount of noise although the magnitude is low or can contain a low noise with relatively high information magnitude. In both cases, the filter process changes the frequency content of the image.

Image filtering is function that contains interpolation, noise reduction, and resampling. There are a lot of details affecting the function of a filter in application such as noise, dynamic range, color, optical artifacts. In all applications of the image processing such as image enhancement and restoration, morphological operations, edge detection, segmentation, recognition, object tracking, template matching, compression.

1.2 Applying filter to image

Image processing is the operation on the pixel values of the picture (numbers) in the matrix form with some logic (algorithm). Since an image is a two-dimensional (2-D), the mathematical operation in the algorithm should have 2 dimensional domain. The filtering of a signal is either processed in time (convolution) or in frequency (Fourier) domain. Since the image is 2-D, filters also must be 2-D.

1.3 Applying Gaussian filter to image

In Gaussian filtering, each pixel is replaced with a Gaussian shaped blob with the same total weight as the original intensity value. Gaussian filters can be designed easily because of their simple variable dependence on variance. Gaussian filter functions are given in Eq. (1.1) and in Eq.(1.2) for 1-D and 2-D applications, respectively.

$$G(x,y) = \frac{1}{\sqrt{2*\pi*\sigma}} * e^{-\frac{x*x}{2*\sigma*\sigma}} \tag{1.1}$$

$$G(x,y)\frac{1}{2*\pi*\sigma*\sigma} * e^{-\frac{(x*x)+(y*y)}{2*\sigma*\sigma}} \tag{1.2}$$

The value of the variance affects the bandwidth. Smaller value of sigma causes the suppression of more frequencies and vice versa. Filtering can be said to be frequency selection. Selection criteria depends on filter. A low-pass filter removes the sharp changes (edges) and a high pass filter removes the low frequencies from the image. The general image processing applications are summariez in Fig. 1.1.

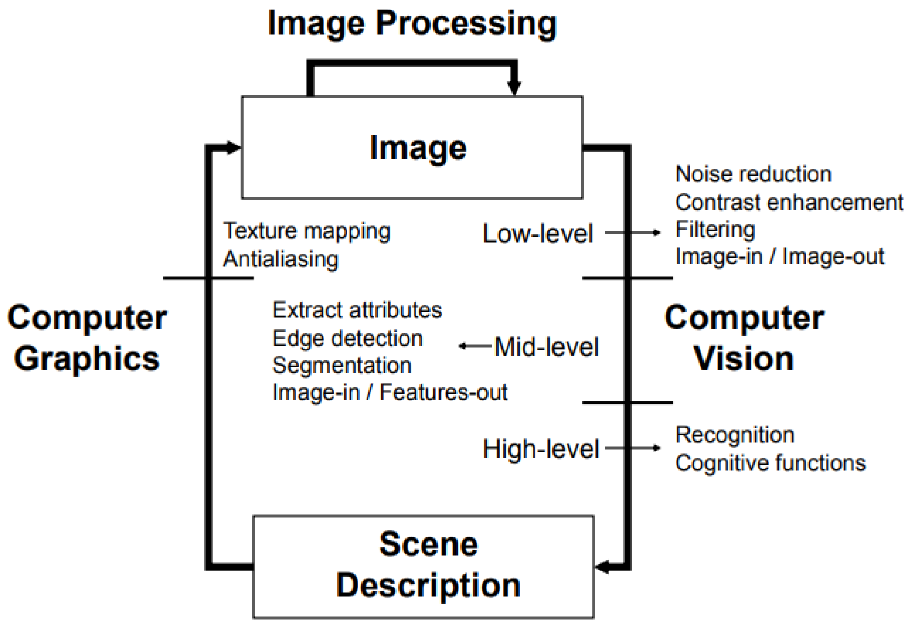


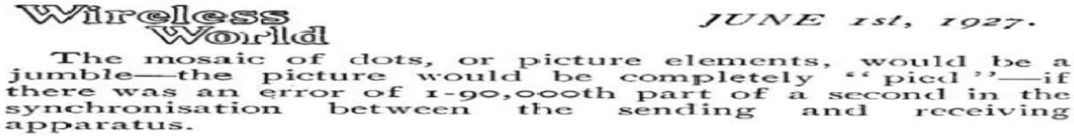
Figure 1.1 Techniques of image processing

Under overall these literature, the main purpose of this study is focused on the determination of the minimum number of coefficient of a Gaussian filter for each image.

CHAPTER 2
LITERATURE REVIEW AND BACKGROUND

2.1 Definition of pixel

The term of pixel is the abbreviation of Picture Element (PEL). A pixel (px) is the smallest component for an image. A valuation for the origins of pixel needs some understandings that belongs to sources and meanings of picture element. It was first presented by Alfred Dinsdalein “Wireless World magazine” in 1927 as given in Fig. 2.1. Dinsdale had written the very first English book on Televisionin 1926,but instead of picture element he had used lots of other colorfull language like “a mosaic of selenium cells”, “a great number of small parts”, etc [1].



Wireless World JUNE 1st, 1927.

The mosaic of dots, or picture elements, would be a jumble—the picture would be completely “ pied ”—if there was an error of 1-90,000th part of a second in the synchronisation between the sending and receiving apparatus.

Figure 2.1 The first appearance of Picture element (Wireless World and Radio Review, Ives at Bell Labs of a 50-by-50 element television system)[1].

2.2 The meaning of images

Images are visual representation of some physical information. An image originaly is continuous. Digital image is the binary replica of an image in computer enviorement in the form of picture, graph, logo or individual video frames. Digital images can be stored in any digital storage device.

2.2.1 Picture

An image is a picture made or replicated and put away in electronic frame. An image can be characterized in vector or raster realistic. An image put away in raster shape is called as “bitmap”. An image outline a document which contains data partner distinctive areas on a predetermined picture with hypertext joins

Some image document positions are:

JPEG which is a realistic picture record delivered by a standard from the Joint Photographic Experts Group, an ISO/IEC gathering of specialists that creates and keeps up norms for a suite of pressure calculations for PC picture documents (x.jpg).

GIF is Graphics Interchange Format. The GIF uses the 2-D raster data type and is encoded in binary (x.gif).

GIF89a is a vivified GIF picture, organized by GIF Version 89a. One of the central favorable position arrange is the capacity to make an energized picture that can be played in the wake of transmitting to a watcher page that moves. A GIF89a can likewise be determined for intertwined GIF introduction.

PNG is a document arrange for picture pressure that was intended to give various enhancements over the GIF organize (x.png). SVG, TIFF are other common formats for picture storage.

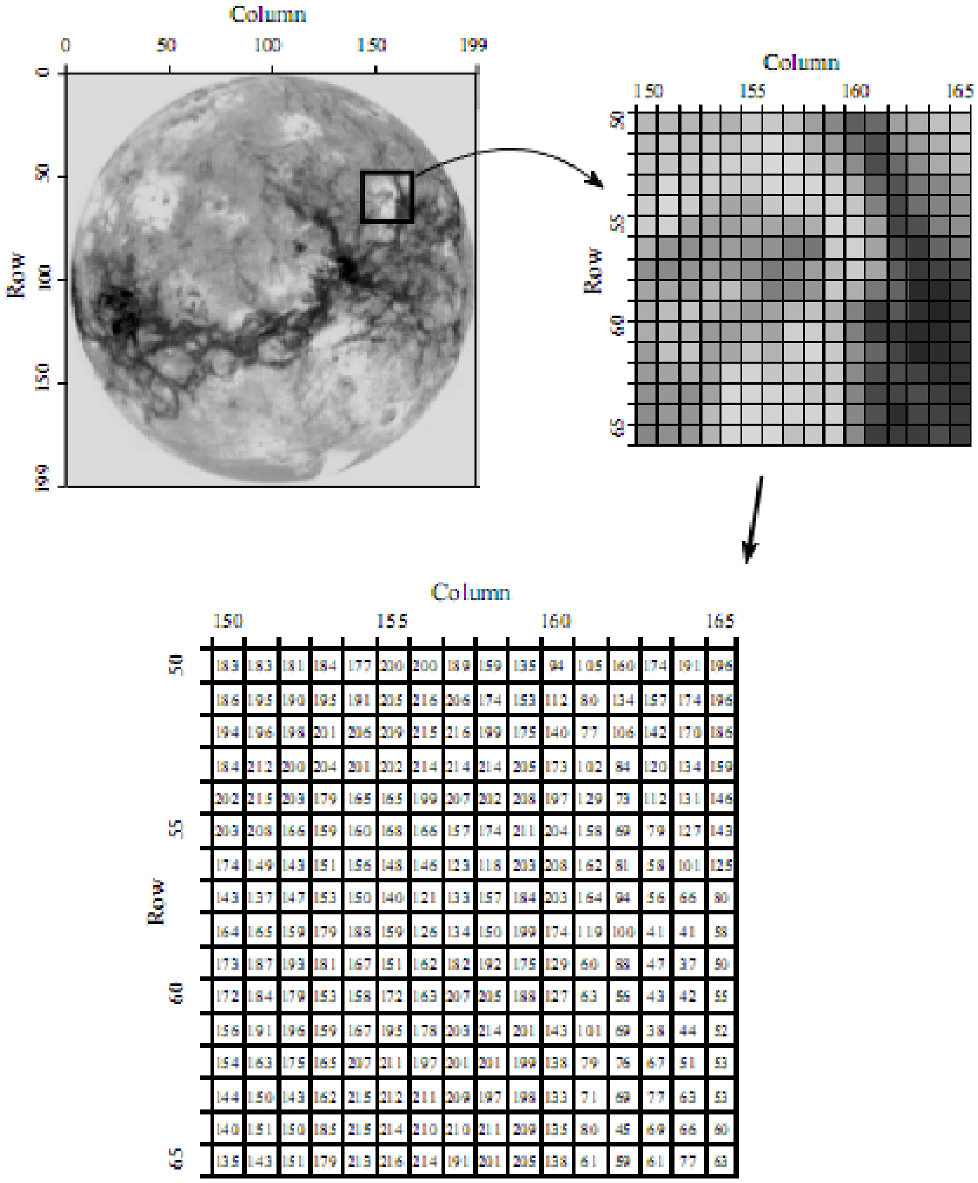
2.2.2 Disk image

Disk image is a duplicate of the whole substance of a capacity gadget, for example, a hard drive or DVD. The circle picture speaks to the substance precisely as it is on the first stockpiling gadget, including the two information and structure data. “Pictures are relative to some input device and a record may be a different picture to different input devices.” [2]

2.3 Structure of images

Images are a definition of how a number varies on a 2-D surface. Normal images result from light (not only but dominant parameter) intensity varieties over a two-dimensional plane. For example, an image can be created by the temperature, blood velocity, x-ray emission, ground motion, etc. These images must be transferred into normal pictures (i.e, light images) to be seenable.

Figure 2. 2 shows the structure of a digital image of the planet Venus obtained by microwave radar. The image in this figure is represented by 200x200 pixels (40,000 samples) in 2-D. Each pixel in this figure is a single number between 0 and 255..Output of the microwave radar converted into visible image replacing the value of each pixel as 0 (black) and 255 (white) and the intermediate values are shades of gray. Features in images are represented by edges. Associating can happen in pictures however it is not a noteworthy issue.



Row \ Column	150					155					160					165
50	183	183	181	184	177	200	200	189	159	135	94	105	160	174	191	196
	186	195	190	195	191	205	216	206	174	153	112	80	134	157	174	196
	194	196	198	201	206	209	215	216	199	175	140	77	106	142	170	186
	184	212	200	204	201	202	214	214	214	205	173	102	84	120	134	159
	202	215	203	179	165	165	199	207	202	208	197	129	73	112	131	146
55	203	208	166	159	160	168	166	157	174	211	204	158	69	79	127	143
	174	149	143	151	156	148	146	123	118	203	208	162	81	58	101	125
	143	137	147	153	150	140	121	133	157	184	203	164	94	56	66	80
	164	165	159	179	188	159	126	134	150	199	174	119	100	41	41	58
	173	187	193	181	167	151	162	182	192	175	129	60	88	47	37	50
60	172	184	179	153	158	172	163	207	205	188	127	63	56	43	42	55
	156	191	196	159	167	195	178	203	214	201	143	101	69	38	44	52
	154	163	175	165	207	211	197	201	201	199	138	79	76	67	51	53
	144	150	143	162	215	212	211	209	197	198	133	71	69	77	63	53
	140	151	150	185	215	214	210	210	211	209	135	80	45	69	66	60
65	135	143	151	179	213	216	214	191	201	205	138	61	59	61	77	63

Figure 2.2 Digital image structure of the planet Venus [3].

A "typical" digital image is made out of around 500 rows by 500 columns. This is the optimum range for television, computer and scientific research. Images that are more than 1000 by 1000 pixels are exceptionally better. Lower resolution is sometimes preferable since there are less pixels to deal. One of the most difficult problem is the managing of huge amount of data. For example, one second of digital audio needs about 8 kb, one second of television needs about 8 Mb. For example, if a pixels in the picture have only the values 145 and 146, the human eye sees the region as a grayness of 145.5. Most important point in these images is a brightness step size of 1/256. It is smaller than eye sensitivity. An image introduced to a human eyewitness won't be enhanced by utilizing more than 256 levels (8 bits). However some images may be stored with more than 8 bits per pixel. The value of each pixel represents a small region in the continuous [3].

2.4 Color

Color is created by using 3 different numbers for every pixel, representing the intensity of green, red and blue. Mixing of these colors generates all possible colors. A solitary byte is much of the time used to store each of the color intensities, enabling the picture to catch a sum of 16.800.000 different colors (256×256×256). The structure and working principle of the eye and camera are very similar. Both are based on lens and imaging sensor. The lens captures a portion of the light reflected from object and focus it onto the imaging sensor. The imaging sensor transforms the pattern of light into a brain or computer. Figure 2.3 shows the matrix structure of the color image.

2.5 What are signals?

Before characterizing filters in detail, it is imperative to comprehend the attributes of signals. A signal can be characterized in the time domain or in the frequency domain. Some signals are periodic. Signals can be measured and shown in time domain by an oscilloscope. Rapid variation in time domain correponds to high frequency and vice versa. The signals can also be defined in the frequency domain in which the magnitude and phase of each harmonic of the signal is shown in the frequency domain. The magnitude of the signal in frequency domain is called as spectrum. The easiest kind of signal is an unadulterated sinusoid which is occasional in the time domain and has vitality at just a single frequency in the frequency range.

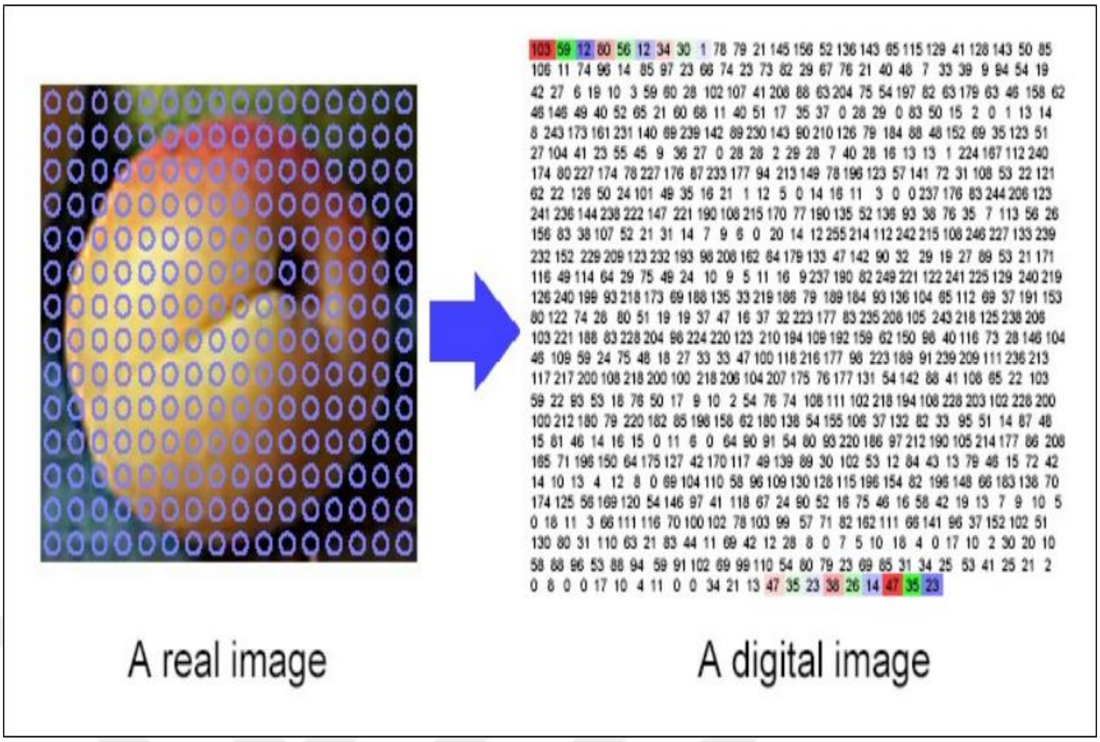


Figure 2.3 Matrix structure of a colorful image

Different signals, for example, human discourse, a square wave, or rash signs, contain vitality at numerous frequencies. Figure 2.4 demonstrates the connection amongst time and frequency domains for a basic sinusoidal signal.

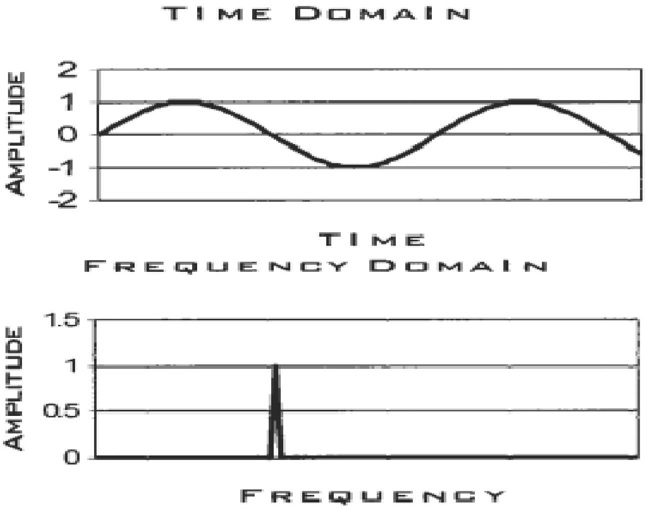


Figure 2.4 Waveform and spectrum of sinusoidal signal [4].

2.6 Why use filters?

When an image is recorded by a camera, often the vision system is unable to use it directly. There are a lot of methods for image increment eliminating undesirable characteristics. Image filtering has many useful applications such as sharpening, edge detection, removing noise, smoothing. Most images are affected by noise in different sizes. Image analysis is often easier if this noise is filtered out.

2.7 Classes of filters

Filtering is a huge meaning and grouping, supplies a whole viewpoint. Types of filtering are mentioned in the follwing sections.

2.7.1 Frequency choosing filters

One of the class of filters is frequency selective filter. Frequency selective filters have different responses to the different harmonics affacting their magnitude and angle. Frequency selective filters have certain parameters such as pass band and stop band attenuations and frequency edges. Once the definitions have been decided, the filter is designed to meet given specifications. Although the gain characteristics of such filters are important, their phases (angle) should also be known in many applications. Owing to the structure of the signal (image) to be filtered analog or digital filters are designed using exactly the different mathematical approaches [5].

2.7.2 Statistical filters

Statistical filters have a consistent frequency reaction however the frequency reaction isn't earlier and for the most part not level qualities. In fact, it is even be not easy to say a passband, a stopband. These filters are used to optimize some statistical design scale. The most well-known statistical filter is the Wiener filter.[6]

2.7.3 Adaptive filters

The transfer functions of the analog or digital filters are generally in the form of polynomial ratio of the s or z values, respectively. The cofficients of these polynomials affect the magnite and phase charateristics of these filters. The values of these coefficients are normally the function of the components of the analog or digital systems and they can be changed and controlled by user in time. These filters are known as adaptive filters. Their characteristics are controlled by some well known algorithms such as Least-Mean-Square (LMS) and Recursive-Least-Square (RLS)[7]. Kalman filters are one of the example of this kind of filters [8,9].

2.7.4 Digital filters

Statistical and adaptive filters are ordinarily advanced channels. However the computerized channels typically allude to advanced frequency-selective filters. Since the hardware of the digital filters contains registers, multipliers and adders, all computers, microprocessors can be programmed as digital filters with suitable algorithms. Since the original information carrying signals are analog, digital filters can be used to process these signals by first converting the signals into digital form. When processing ended, the filter's digital output can be converted back to analog form. Many recursive digital filter design procedures are based on analog prototypes [10].

2.7.5 Analog filters

Analog filters are continuous time domain filters which contain generally LC structures (passive) and/or equivalent active components. Their magnitude characteristics are approximated by the realizable function such as Butterworth, Chebyshev, Elliptic, Bessel and possible others. Every design method has both advantages and disadvantages [11].

2.8 Two-dimensional filters

Two dimensional filters are generally used for filtering the images. In the 2-D case the situation is quite different from 1-D. The transfer function of 2-D filters have two different variables one for each domain. Therefore their magnitude and phase characteristics are surface in 3-D space one for frequency and other two for gain for each dimension. The analog signals have continuous feature in time. These signals can be 1-D, 2-D or Multi-D. In most cases, the variable of 1-D analog filters are time or frequency. After the time sampling and discretization, analog signal becomes a 1-D digital (discrete) signal.

An example of a two-dimensional signal is a picture. A filter generally changes the shape of the signal. Examples of such changes include noise elimination, amplifying frequency additives, phase shifting, etc. The design and implementation of these filters is a main task in signal analysis and processing technology.

The filter normally comprises attractive algorithm that simulates 2-D difference equation. The logic of filtering is to process the coefficients of the filter with the samples and to generate the new signal owing to the proposed algorithm. Generally, scalar systems, linear, recursive, two-dimensional filters are designed as stable filters which mean that they produce bounded output for applied bounded input (BIBO). The conditions for BIBO stability are well known and their output is computable from previous input and output values. Filters that can be identified in this way are said to be causal [12].

2.9 IIR and FIR filtering

2.9.1 Infinite Impulse Response (IIR) filtering

Infinite Impulse Responce (IIR) filters are one of two types of digital filters. The output of these filters are converging functions for impulsive inputs. This is satisfied by the feedbacks in these kinds of filters. Digital filters with an IIR characteristics are very useful in many applications. Because of the feedback necessary in an implementation, the IIR filter is also called as recursive filter or sometimes, an autoregressive moving average filter (ARMA). The transfer function of the IIR filters are in the form of ratio of two z domain polynomials. The transfer function being a ratio of polynomials means it has finite poles as well as zeros. This gives considerably more flexibility and power, but brings with it certain problems in both design and implementation [13].

2.9.2 Finite Impulse Response filtering

Most FIR's are linear-phase filters. "Linear Phase" means the angle of the filter is linear function of the frequency and all harmonics processed by the filter is delayed in the same duration. This is very important criteria in signal filtering if the main purpose is to change only the magnitudes of the harmonics without any delay distortion. FIR filters are also called as "Constant delay filters". The lack of delay distortion is a critical advantage of FIR filters over IIR. A FIR filter has linear-phase if it's coefficients are symmetrical around the center coefficient. The constant delay of the odd order FIR filters can be calculaed by using Eq. 2.4.

$$\text{Delay} = (N - 1) / (2 * Fs) \qquad (2.4)$$

where Fs is the sampling frequency (clock frequency of the digital filter) and N is its order (number of delay registers).

2.9.3 Properties of FIR and IIR filters

The two types of digital filters, FIR and IIR, are commonly used in application. They have very different characteristics. The block diagrams of FIR and IIR filters are given in Fig. 2.5.

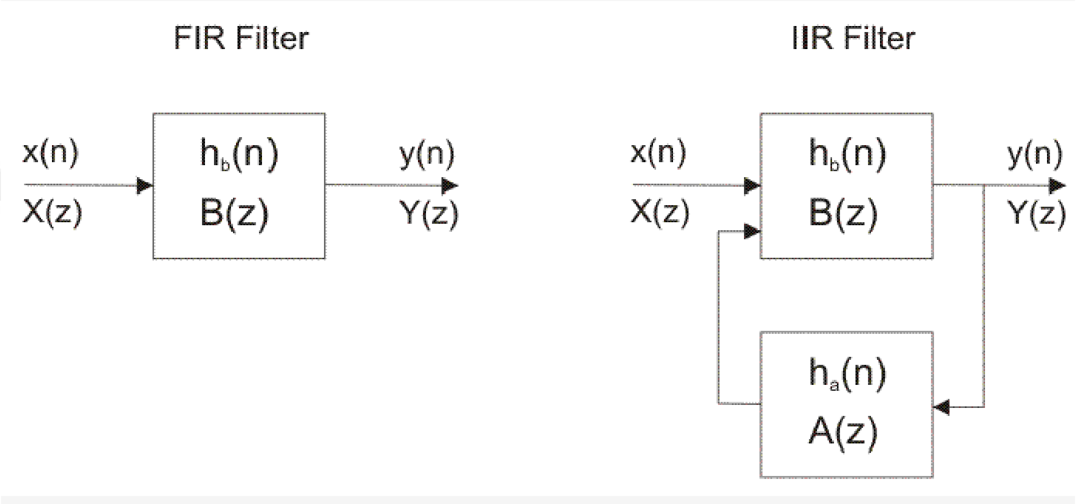


Figure 2.5 Block diagram of FIR and IIR filters[14]

where H(z)=B(z) and H(z)=B(z)/(1+A(z)) for FIR and IIR filters, respectively.

2.10 Mean Filter

Mean filters are basic and simple procedure of filtering the images for smoothing (decreasing the change of density amongthe pixels). It always uses for reducing noise in images. The idea of mean filter is replacing each pixel value in an image with the mean value of its nearby pixels. Often a 3×3 square kernel is used. A small kernel can be applied repetitively for producing a similar but not equal effect.

2.10.1 Arithmetic mean filter

This is the least difficult variant of the mean filters. The number juggling mean filter registers the normal estimation of the ruined picture g(x, y) in the territory characterized by Sxy. The estimation of the reestablished picture anytime (x, y) is basically the number-crunching mean registered utilizing the pixels in the area characterized by S. Expression is given by Eq. (2.7).

$$f(x,y) = \frac{1}{m*n}\sum_{(s,t)\in Sxy} g(s,t) \qquad (2.7)$$

The expression given above which is implemented for using a convolution mask as coefficients have value 1/m*n. Consequently, noise can be reduced as blurring.

2.10.2 Geometric mean filter

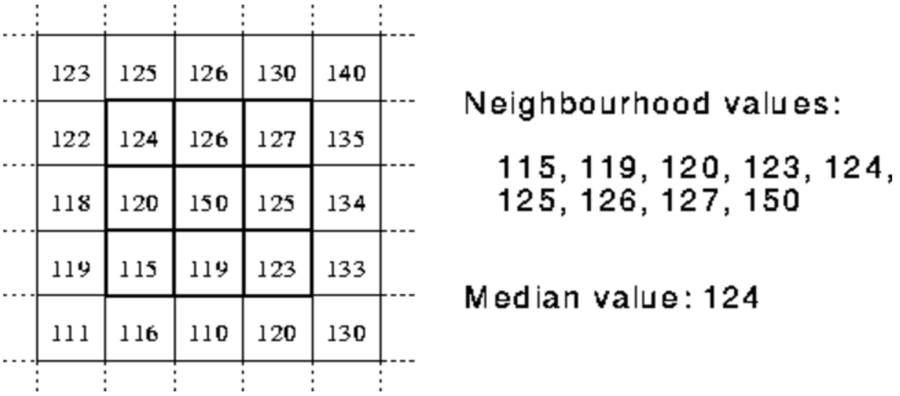
An image reconstruction using a geometric mean filter is given by Eq. (2.8).

$$f(x,y) = \left[\prod_{(s,t)\in Sxy} g(s,t)\right]^{1/m*n} \qquad (2.8)$$

Each reconstructed pixel is represented by the product of the pixels in the sub-image increased to the power 1/m*n. Geometric mean filter provides smoothing to compare with arithmetic mean filter.

2.10.3 Median filter

Median filter is the best known 2-D filter. Its name suggests the substitution of the estimation of a pixel by the gray levels in the adjacent of related pixel. The median filter is constantly utilized for enhancing the nature of the picture by diminishing the noise in it and the productivity of median filter is regularly superior to the mean filter. Median filter takes each pixel in the picture one by one, checks with its close pixels for finding on the off chance that it is illustrative of its environment or not and it changes with the median of that values. The median can be figured by first masterminding the pixel values that adjacent into numerical course of action and replacing the pixel inspects with the center pixel values. (In the event that the territory under significance incorporates a considerably number of pixels, the normal estimation of the two center pixel employments.) A sample of calculation is shown in Fig. 2.6.



123	125	126	130	140
122	124	126	127	135
118	120	150	125	134
119	115	119	123	133
111	116	110	120	130

Neighbourhood values:

115, 119, 120, 123, 124,
125, 126, 127, 150

Median value: 124

Figure 2.6 The median value of a pixel neighborhood is calculated. (Central pixel value is 150) [15].

2.10.4 Comparison of median and mean filters

Table 2.1 General information about mean and median filters

PRINCIPLE FOR COMPRASION	**MEAN FILTERS**	**MEDIAN FILTERS**
Meaning of filter	Mean alludes to the basic normal of the given arrangement of qualities or amounts.	Median is characterized as the center number in a requested of qualities.
What is the filter?	Arithmetic average.	Positional average.
What is the filter represent?	Data set's gravity Center	Data set's Mid-point's Center of gravity.
Applicability of filter	Normal dealing out.	Skewed dealing out.
Outlier of filter	Sensitive to outliers.	Not sensitive to outliers
Calculation of filter	Mean is figured by including every one of the perceptions and after that separating the esteem got with the quantity of perceptions.	To figure median, the informational collection is organized in rising or sliding request, at that point the esteem that falls in the correct center of the new informational collection, is median.

Essential contrasts amongst mean and median filters:

- A mean is characterized as the straightforward normal of the amounts. The middle is the center number in requested esteems.
- Mean is the number-crunching normal, the middle is positional normal.
- Mean traces the focal point of gravity of informational index. Median calls attention to the center estimation of the informational collection.
- The mean is useful for typically disseminated information. The median is better for skewed disseminated information.
- The mean is very influenced by the extreme esteem however median is not.

With the enlightenment of the past sentence, it might be said that these two ideas are very unique. Math mean, or just "mean", is acknowledged as the best measure of focal esteem. It comprises every one of the attributes of a perfect measure. In any case, it has one burden. The testing varieties influences the mean esteem. The middle is additionally depicted and straightforward and compute. The best feature of this filter is, it's insensitivity to sampling varieties. The one weakness of the median flter is that it did not depend on all estimations.

2.10.5 Frequency filters

An image can be processed in the frequency domain by frequency filters. Since original image is in the time domain, first, it’s Fourier transform is found and harmonic magnitudes are multiplied by the filter coefficients pixel-by-pixel as in Eq. (2.11) and the original filtered time domain image is re-obtained using inverse transformation. Reducing high frequencies (sharp color changes in the picture) results in a smoother image while reducing small frequencies increase the edges.

All frequency filters are also used in the time (spatial) domain. Computationally using spatial domain is less expensive to run the filtering algorithm. Frequency filtering is reasonable if there is no main kernel that found in the spatial domain.

$$\left(\mathrm{Y}(\mathrm{k},\mathrm{l})\right) = \left(\mathrm{X}(\mathrm{k},\mathrm{l}) * \mathrm{H}(\mathrm{k},\mathrm{l})\right) \qquad (2.11)$$

where X(k,l) is the input image, H(k,l) is filter function and Y(k,l) is the filtered image. For obtaining the resulted image in the spatial domain, Y(k,l) is re-transformed in time domain (Inverse Fourier Transform). The similar process can be achieved in spatial domain using convolution.

There are four main different types of filters: lowpass, highpass. bandstop and bandpass filters which have passband interval (very low attenuation) for low, high and band of frequencies. The result of spatial domain is equal to which of a smoothing filter.

The simple lowpass filter is a kind of suitable filter. It attenuates all harmonics with higher frequencies and than the cut-off frequency (W_c) and passes all harmonics with smaller frequencies. The mathematical representation of such filters is given in Eq. (2.12).

$$H(k,l) = \begin{cases} 1\ if\ \sqrt{(k*k)+(l*l)} < \ \mathrm{Wc} \\ 0\ if\ \sqrt{(k*k)+(l*l)} > Wc \end{cases} \tag{2.12}$$

There is a disadvantage of this filter, it is a ringing effect which happens on the edges of the filtered image. The filtering process is shown in Fig. 2.10. Related figure shows the shape of the 1-Dimensional filter in the frequency domain and spatial domain for two different cut off frequencies W_c. Two-dimensional filter is obtained with rotating these functions about the vertical (gain) axis. Convolution in the spatial domain corresponds to the multiplication in the Fourier domain. Spatial domain is ideal because of the multiple peaks of the ideal low pass filter, the processed image generates ringing along the intensity edges.

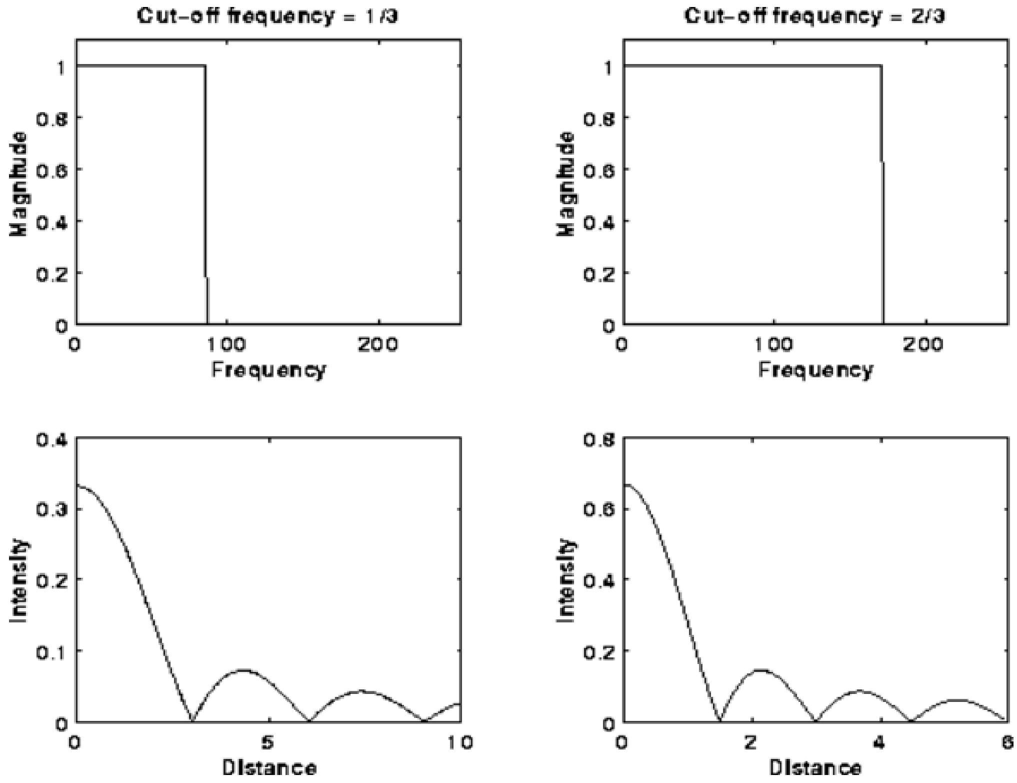


Figure 2.7 Ideal lowpass in frequency and spatial domain.

The shape of the Gaussian is same shape in the time (spatial) and frequency domains and so doesn't attract the ringing effect in the spatial domain for filtered image, the better results can be obtained with Gaussian filters. The Butterworth filter is anusual discrete estimation to the Gaussian. Application of Gaussian filter for the frequency domain gives same result to one obtained using the Gaussian filter in the spatial domain. Main difference is that the computational time (cost) of the spatial filter rises with standard deviation (because of the size of the filter kernel), but the costs are independent from the filter function in frequency filter. Hence, the spatial Gaussian filter is better than frequency filter about narrow passband regions but using Butterworth filter is better for large lowpass filters.

The equal methods are applied for highpass filters. Function of highpass filter is obtained by inverting the relating lowpass filter, for example ideal highpass filters block all frequencies which are smaller than Wc.

Bandpass filters can be used as a cascade lowpass-highpass filters with well adjusted cut off frequencies(cut-off frequency of the lowpass filter is higher than the cut off frequency of highpass filter). They reduce all harmonics with which frequencies are smaller than a frequency W_{c1} and higher than frequency W_{c2}, and frequencies which are between these cut offs. The filter function of bandpass is obtained of by multiplying (cascade connected) lowpass and highpass filter transfer function in the frequency domain.

Beside of using standard filter functions, our own filter mask can be designed by increasing or decreasing main frequencies.

CHAPTER 3
GAUSSIAN FILTERS

3.1 Gaussian smoothing

The Gaussian smoothing process is 2-D convolution operation. It is applied to `blur' the image and to remove the details and noise from the picture. Although it is same like Mean Filter in this sence, but a different kernel is used to represent the shape of a Gaussian (`bell-shaped') curve. This kernel has some special properties which are given in details. 1-D Gaussian distribution in with zero mean value has the form as given in Eq. (3.1).

$$G(x,y) = \frac{1}{\sqrt{2*\pi*\sigma}} * e^{-\frac{x*x}{2*\sigma*\sigma}} \tag{3.1}$$

where $\sigma$is the standard deviation of the distribution. If mean vaule is not zero all x in the equation (3.1) should be replaced by (x-x$_0$), where x$_0$ isthe mean value. The distribution is shown in Figure 3.1.

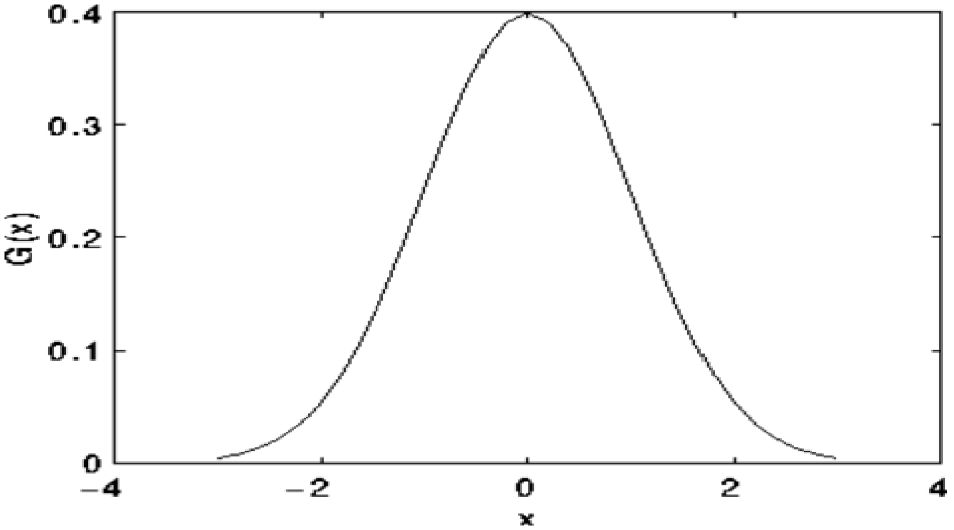


Figure 3.1 1-D Gaussian distribution (mean=0 and σ=1).

For 2-D, the isotropic Gaussian form is given in Eq. (3.2) and shown in Fig. 3.2.

$$G(x,y) = \frac{1}{2*\pi*\sigma*\sigma} * e^{-\frac{(x*x)+(y*y)}{2*\sigma*\sigma}} \qquad (3.2)$$

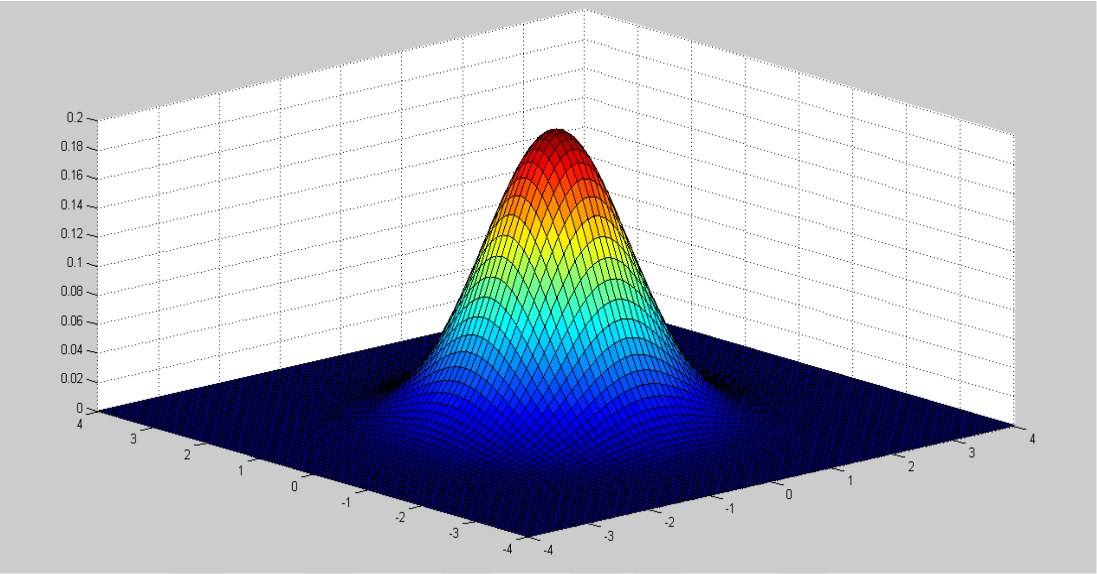
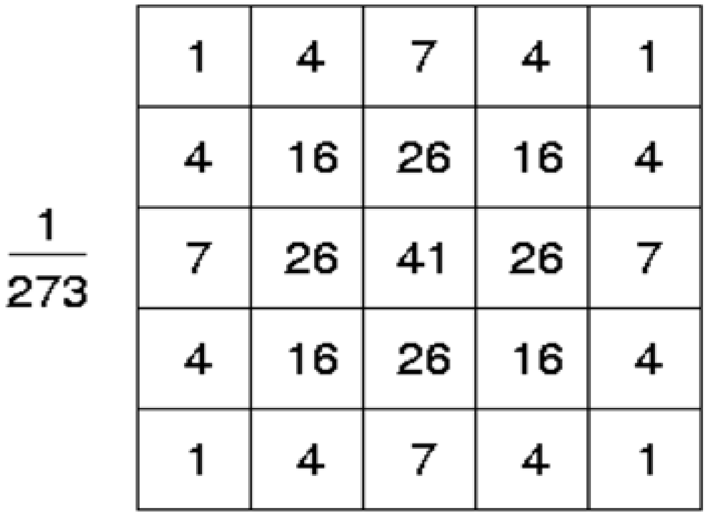


Figure 3.2 2-D Gaussian circulation while mean (0,0) and σ =1

The point of Gaussian smoothing is utilizing this 2-D dissemination work as a `point - spread' capacity. This happen is prevailing by convolution, and picture is the union of particular pixels, a discrete estimation to the Gaussian capacity ought to be characterized before convolution. In principle, the Gaussian circulation is nonzero all around ((zero at infinity) which requires a vast convolution piece yet in application it is thought to be zero after x and y are more prominent than three standard deviations from the mean. Figure 3.3 demonstrates a worthy whole number esteemed convolution part that approximates a Gaussian with a Standard deviation σ of 1.0. It isn't clear how to gather the estimations of the mask to inexact a Gaussian. The estimation of the Gaussian at the center of a pixel in the mask be utilized yet this isn't right in light of the fact that the estimation of the Gaussian differs nonlinearly across the pixel. Along these lines the estimation of the Gaussian is incorporated over the entire pixel (by including the Gaussian at 0.001 stages). The integrals are not numbers, in this manner the array is rescaled with the goal that the corners had the value 1. At last, the 273 is the whole of the considerable number of qualities in the mask.



$\frac{1}{273}$

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

Figure 3.3 Discrete approximation to Gaussian function with σ = 1.0

Thus, firstly an acceptable kernel is, at that point the Gaussian smoothing is performed utilizing standard convolution strategies. The convolution can be performed decently fast since the condition for the 2-D isotropic. Gaussian appeared above is distinct into x and y segments. In this manner the 2-D convolution should be possible by convolving 1-D Gaussian in the x and y headings independently. (The Gaussian is the main totally circularly symmetric operator which can be decomposed in such a way.)

Another method for figuring the Gaussian smoothing with an extensive standard deviation is to convolve a picture a few times with a smaller Gaussian. This technique is computationally complex however it can has applicability if the preparing is completed utilizing an hardware pipeline.

The Gaussian filter has utility in designing applications as well as in a wide range of territories, for example, computational science, arithmetic, any smoothing operations, and so on.

The Gaussian filter is additionally said to be a non-uniform low pass filter. The kernel coefficients diminish with expanding separation from the kernels center. Focal pixels have a higher weighting than the surrounding pixels. Bigger estimations of standard deviations σ make bigger pinnacle. Part estimate increments with expanding σ to set the Gaussian idea of the filter and its bit coefficients rely upon the estimation of σ. At the edge of the mask, coefficients must be near 0. The kernel is rotationally symmetric with no directional inclination. Gaussian kernel is distinct. It permits quick calculation. But as a disadvantage, Gaussian filters might not save the image brightness. Figure 3.4 shows the position of pixels for Gaussian function.

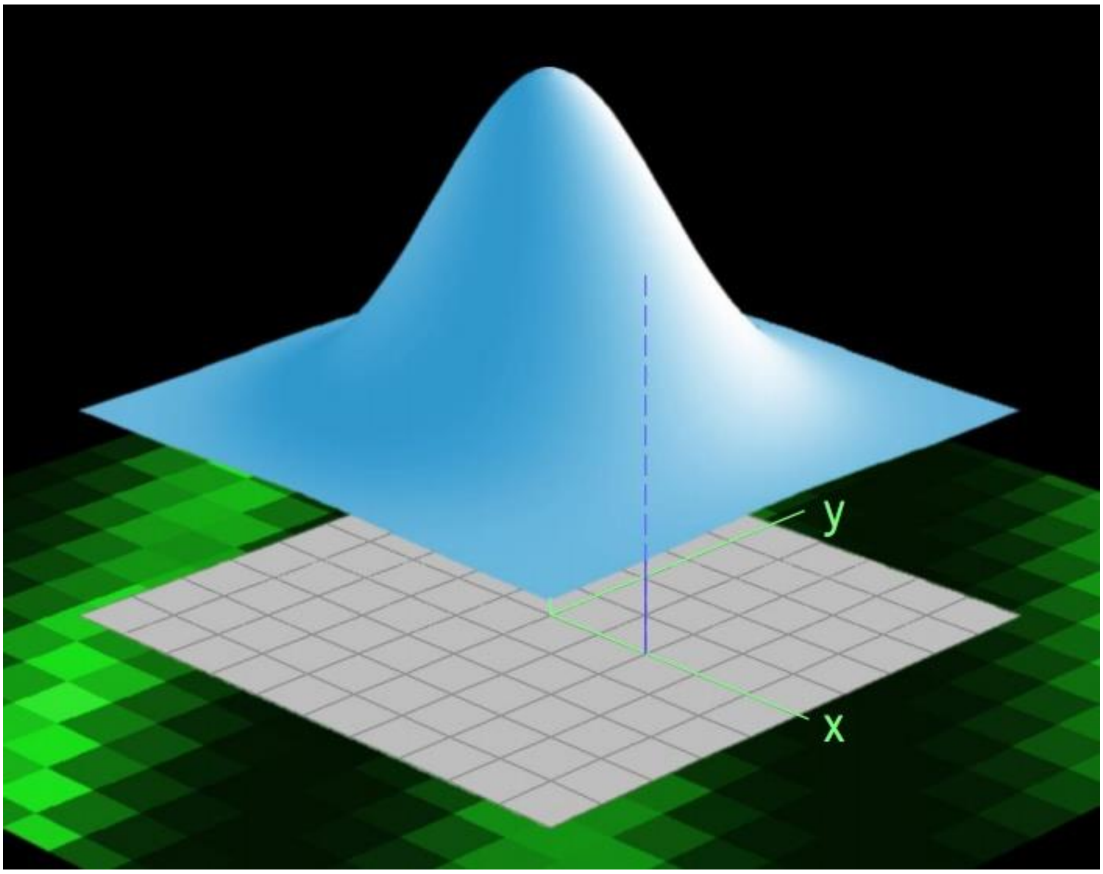


Figure 3.4 Position of pixeles for Gaussian function [16]

Coordinates x_0, y_0 are optional pixel positions in a bitmap image and x, y is a local coordinate system centered in x_0, y_0. The gray area is a filter box with m*m nodes and x and y reach from -n to +n. The box width m (= 2n + 1) is assumed to be odd.

Weighting factors are calculated for a Gaussian bell using the equations (3.3) and (3.4)

$$w(x, y) = e^{-a} \tag{3.3}$$

$$a = (x^2 + y^2)/(2 * r^2) \tag{3.4}$$

The filter radius r denotes the standard deviation sigma in Gaussian distribution. n is selected as (2...3) times of r for a reasonable reproduction without clipping. For example, for x = r, y = 0 we find w = $e^{-0.5}$= 0.6065.

The image shows the function relative to the filter box vertically shifted. The radius is r = 2 in the image.

Kernel size and standard deviation are two basic parameters in this filter. Standard deviation controls the smoothing result (larger standard deviation requires larger convolution kernel). This value attenuates high frequency harmonics in the image (low pass filters). As opposite to mean filter, it uses weighted average in the kernel changing the center pixel value. Gaussian filter is considered as a first step in edge detection for noise elimination and is well used for analysis in the frequency domain[16].

3.1.1 Canny edge detection

Canny edge detection is a technique to get some valuable data from various vision objects and to diminish the measure of information to be prepared, drastically. It is broadly utilized as a part of PC vision frameworks. Canny has discovered that the necessities for the use of edge discovery on different vision frameworks are moderately comparable. In this way, an edge discovery answer for address these necessities can be connected in an extensive variety of circumstances. The general criteria for edge discovery incorporates:

- Detection of edge with low error rate, which implies that the location ought to precisely get whatever number edges appeared in the picture as could reasonably be expected

- The edge point distinguished from the operator ought to, precisely confine on the focal point of the edge.
- A given edge in the image should just be checked once, and where conceivable, image noise ought not make false edges.

To satisfy these prerequisites Canny utilized the analytics of varieties a strategy which finds the capacity which upgrades a given practical. The ideal capacity in Canny's detector is described by the sum of four exponential terms, yet it can be approximated by the first derivative of a Gaussian.

Among the edge location techniques grew up until this point, watchful edge identification calculation is a standout amongst the most unquestionably characterized strategies that supplies integrity and reliable recognition. Due to its optimality to meet with the three criteria for edge recognition and the straight forwardness of process for usage, it ended up plainly a standout amongst the most well known calculations for edge discovery.

3.1.2 Standart deviation

The standard deviation of the Gaussian function is the main parameter and plays an important role in its characteristics. The values of variable between $-\sigma < value < +\sigma$ account for 68% of the set and while two standard deviation from the mean (purple and green) corresponds to 95%. Furthermore, three standard deviations (purple, green and brown) occupies 99.7%. This creteria is very important in the design of Gaussian kernel of fixed length (see the figure 3.5).

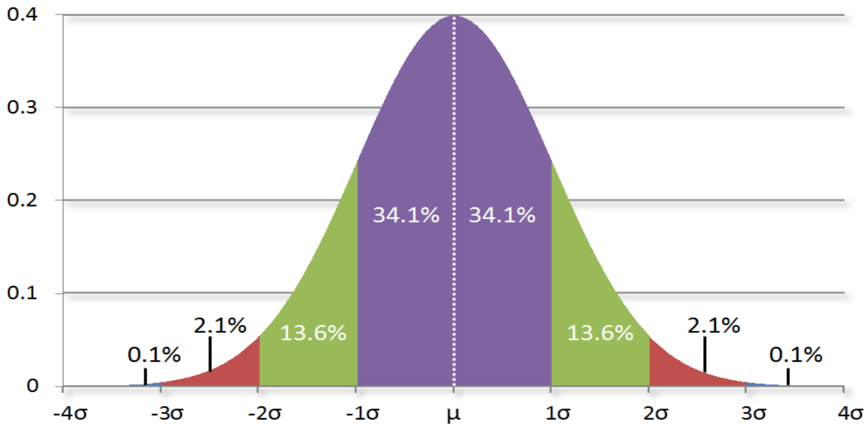


Figure 3.5 Distribution of the Gaussian function values

3.1.3 Laplacian of Gaussian

The Laplacian is two dimensional isotropic measure of the second order spatial derivative of a image. The Laplacian of an image concentrates on locals of quick intensity change and is hence regularly utilized for edge discovery. To apply the Laplacian, for the most part, the picture is first smoothed by utilizing Gaussian smoothing filter to lessen its affectability to noise. Hence the two factors will be depicted together here. The contribution of the operator is typically single gray level image and its yield turns into another gray level picture.

The Laplacian L(x,y) of an image with pixel intensity values I(x,y) is given by Eq. (3.5).

$$L(x,y) = \frac{d^2I}{dx^2} + \frac{d^2I}{dy^2} \tag{3.5}$$

This can be calculated using a convolution filter.

Since the input image is represented as a set of discrete pixels, first, a discrete convolution kernel has to be found that can approach the second derivatives in the definition of the Laplacian. Two commonly used discrete estimations as small kernels to Laplacian filter is shown in Eq. (3.6).

$$\begin{matrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{matrix} \qquad \begin{matrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{matrix} \tag{3.6}$$

Note that the Laplacian is defined using a negative peak. This is more common, however it is equally valid to use the opposite sign convention.

Using one of these kernels, the Laplacian can be computed using standard convolution methods.

Since these kernels are assessing a second derivative estimation on the image, which causes to build the high frequency parasitics (noise) amplitudes in the image, they are extremely delicate to noise. To keep the noise enhancement, the image is first Gaussian smoothed before executing the Laplacian filter. This pre-preparing operation diminishes the high frequency noise segments past to the separation step.

Moreover, since the convolution operation is cooperative, the Gaussian smoothing filter and the Laplacian filter initially are convoluted as though they are single structure and after that this half breed filter is convolved with the picture to acquire the vital outcome. There are basically two points of interest of this operation.

- Since both the Gaussian and the Laplacian kernels are for the most part littler than the image, this procedure normally requires considerably less number-crunching operations.
- The LoG (`Laplacian of Gaussian') kernel can be precalculated ahead of time so just a single convolution should be performed at run-time on the image.

The 2-D LoG function centered on zero and with Gaussian standard deviation σ has the form given in Eq. 3.7and this function is sketched as in Fig. 3.6.

$$\log(x, y) = -\frac{1}{\pi*\rho*\rho*\rho*\rho}\left[1-\frac{(x*x)+(y*y)}{2*\sigma*\sigma}\right]*e^{-\frac{(x*x)+(y*y)}{2*\sigma*\sigma}} \qquad (3.7)$$

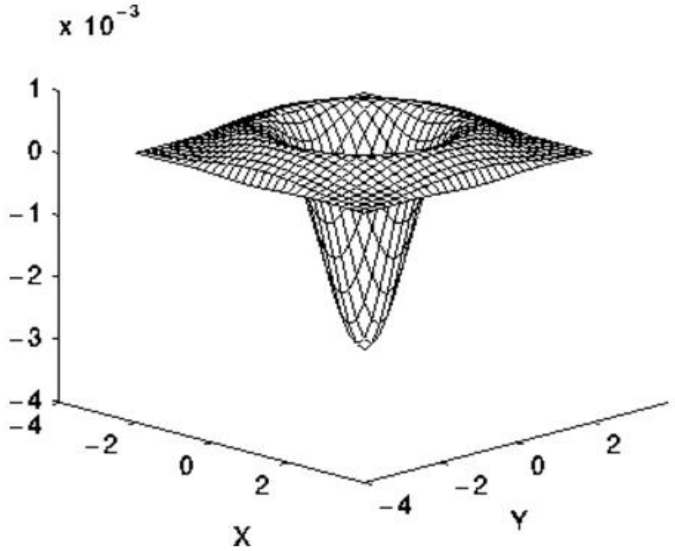


Figure 3.6 The 2-D Laplacian of Gaussian (LoG) function [17].

3.2 Unsharp filters

The unsharp filter is a simple sharpening operator. Its name comes from the fact that it improves the quality (enhancement) of the edges and other high frequency components in the image using a procedure which subtracts an unsharp, or smoothed, version of an image from the original image.

Input-output relation of the unsharp masking procedure is given in Eq. (3.8) in which edge image g(x,y) is output and f(x,y) is the input.

$$g(x, y) = f(x, y) - fsmooth(x, y) \tag{3.8}$$

where $\mathrm{fsmooth}(x, y)$ is a smoothed version of $\mathrm{f}(x, y)$. The block diagram of the process is given in Fig. 3.7.

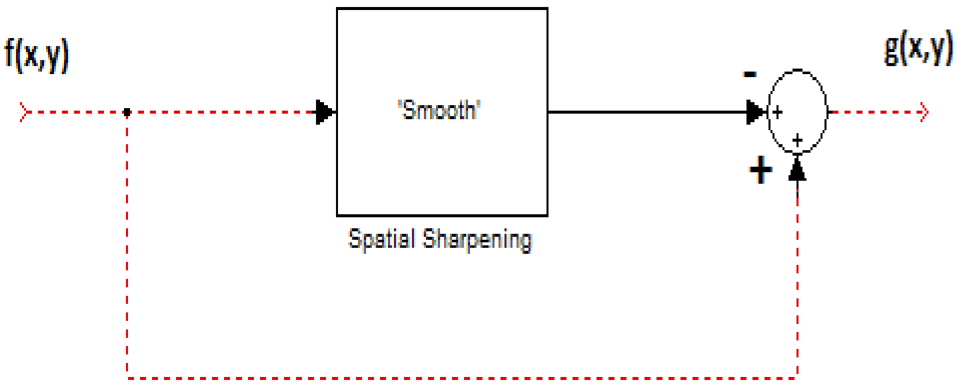


Figure 3.7 Spatial sharpening [18].

The process of the unsharp sharpening filter can be better understood by analyzing its frequency response characteristics. If we have a signal as shown in Figure 3.8(a), subtracting away the lowpass component of that signal (as in Figure 3.8(b)), yields the highpass, or `edge', representation shown in Figure 3.8(c).

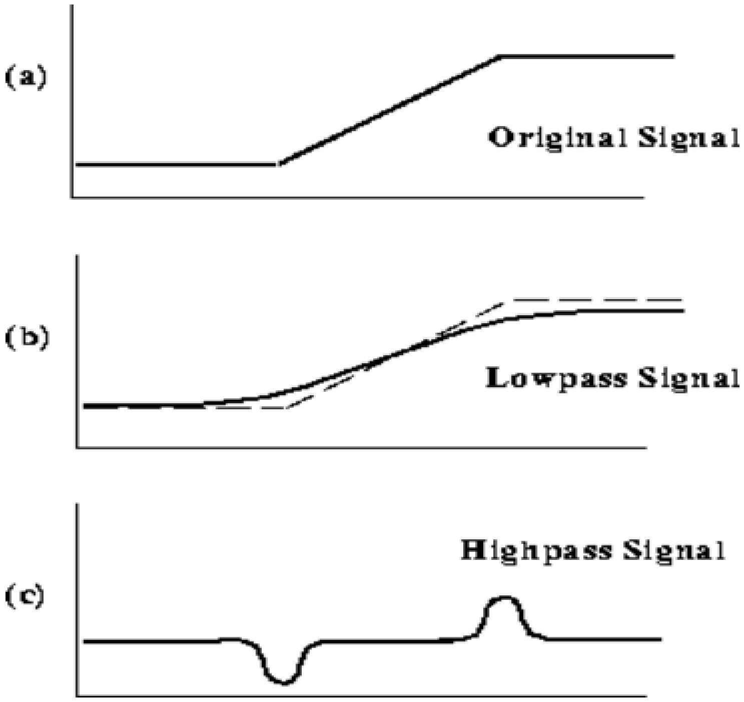


Figure 3.8 Calculating an edge image for unsharp filtering [18].

This edge image can be used for sharpening if we add it back into the original signal, as shown in Figure 3.9

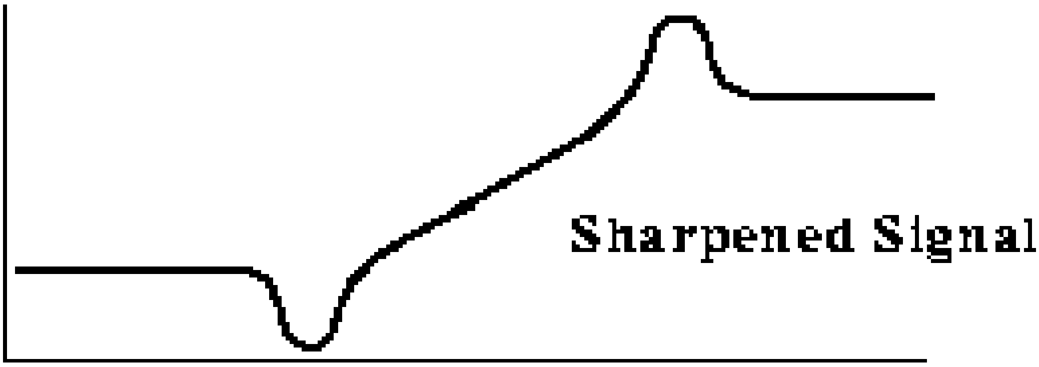


Figure 3.9 Sharpening the original signal using the edge image.[18]

Thus, the complete unsharp sharpening operator is shown in Figure 3.10

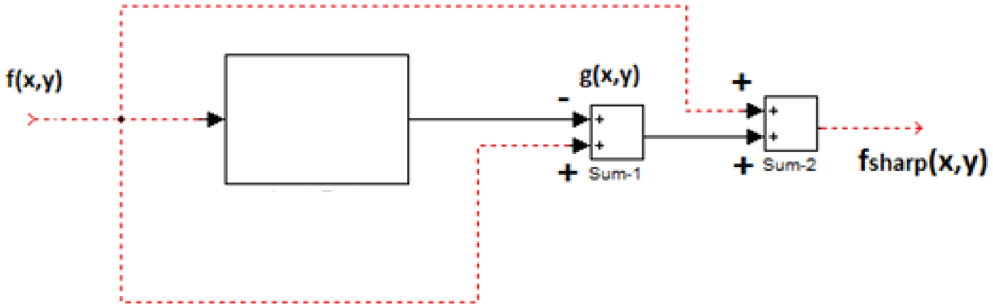


Figure 3.10 The complete unsharp filtering operator [18].

Eq. (3.8) covers all of these in one equation as

$$fsharp(x,y) = f(x,y) + k * g(x,y) \tag{3.8}$$

where k is a scaling constant. Reasonable values for k change between 0.2 and 0.7. Amount of sharpening may be more increase chosing the bigger values for k.

CHAPTER 4

DESIGN OF THE FAST GAUSSIAN FILTER FOR IMAGE PROCESSING APPLICATIONS

4.1 Image filtering

Image processing is the way toward changing the input image and to get the yield as another image. An image is a two-dimensional function f(x,y), x and y symbolizes to the coordinates of every pixel in the image that in the spatial (plane) domain. The size of the incentive at any couple of coordinates (x,y) is known as the intensity of the image at that position of the pixel. For computerized images, the coordinates x and y, and the magnitude estimations of f for every pixel position are discrete. An advanced image is restricted by a limited number of components called pixels where each of pixels has a specific area and value. Advanced picture preparing systems help in control of the computerized(digital) pictures by utilizing microprocessors. The three general application of the digital image processing is said to be pre-processing (to prepare the image to be processed), enhancement (to improve the quality of the picture) and display, information extraction (identification of the picture like face recognition).

Images are often disrupted by random variations in density, irradiation, or have deficient contrast and can not be used directly and they have to be processed by some filter application listed in Table 4.1.

Table 4.1 Characteristics of smoothing filters

Algorithm	Domain	Type
Uniform	Space	Linear
Triangle	Space	Linear
Gaussian	Space	Linear
Median	Space	Non-Linear
Kuwahara	Space	Non-Linear
Other	Frequency	Linear

4.1.1 Contrast improvement

Contrasts changes improve the detectable quality of articles in the scene. It is fulfilled by expanding the shine of the articles in the photo contrasting with the brilliance of its experiences. Complexity upgrades are commonly executed as a differentiation strain took after by a tonal improvement in spite of the fact that these could both be connected in one stage. A differentiation pressure upgrades the brilliance contrasts similarly over the dynamic scope of the picture though tonal changes improve the shine contrasts in the shadow (dark), midtone (grays), or feature (bright) zones at the consumption of the magnificence contrasts in alternate ranges.

Most contrast improvement techniques utilize the dark level histogram made by count the quantity of times each gray-level value be shapes in the picture at that point isolating by the total number of pixels in the picture to deliver a distribution of the level of each gray level. The dark level histogram clarifies the real circulation of the gray levels in the image however contains no spatial data about the image.

4.1.2 Template matching

Template Matching is a high-level machine vision process which defines the parts of a figure which matches a predefined template. Template Matching can be used as a methodology in digital image processing for defining small components of a picture which match a template image. Our aim is matching a template to an image wheresoever the template is a sub image which contains the form which we want to find out. This process is repeated for the whole image, and the point which spearheads to a best match, the extreme count, is described to be the point wherever the shape (given by the template) contains in the image. Templates are usually utilized to print characters, define numbers, and other small, simple objects. We can use for detecting of edges in figures, in manufacturing as a part of quality control and a way to explore a versatile robot. Template matching is a strategy for discovering areas of an image which matches (are indistinct) a template image (patch).

4.1.3 Smoothing

Smoothing is utilizing for diminishing noise inside an image or to deliver a less pixelated image. Image smoothing is a principle innovation of image change which can remove noise in images. Along these lines, it is a fundamental utilitarian module in different image handling strategies. Magnificent smoothing calculation can both evacuate different noises and secure points of interest. This examination researched principle image smoothing strategies. These strategies have the capacity of ensuring subtle elements, for example, gradient weighting filtering, self-adaptive median filtering, strong smoothing and edge protecting filtering. Appropriate decision of such procedures is extraordinarily influenced by the imaging technique, current task and watching conditions. In this examination we will see a diagram of fundamental ideas, alongside calculations normally utilized for picture smoothing.

Evacuating noise is primary focus of picture smoothing in computerized pictures. It is an established thing in advanced picture handling for smoothing image. Also, it has been broadly utilized as a part of many fields, for example, image show, picture transmission and picture examination, and so forth. Image smoothing has been an essential module in all the image preparing techniques. Hence, it merits considering all the more profoundly. Picture smoothing is a strategy for improving the nature of images. The picture quality is an essential factor for the perspective of human. The picture for the most part has noise which isn't effectively distributed with in image handling. The nature of the picture is influenced by the presence of noise. There are a ton of strategies for removing noise from images. Many picture handling calculations can not function admirably in noisy condition, so image filtering is acknowledged as a preprocessing module.

4.2 Basic informations about image processing in Matlab

We can evaluate the dimensions and type of images shown in the figure 4.1 with the codes given below.

```
img = imread('ferit.jpg');
imshow(img);
sz = size(img)
imgType = class(img)
sz = 1040        780         3
imgType = uint8
```

Array of class uint8 values differ from 0 to $2^8 – 1$.

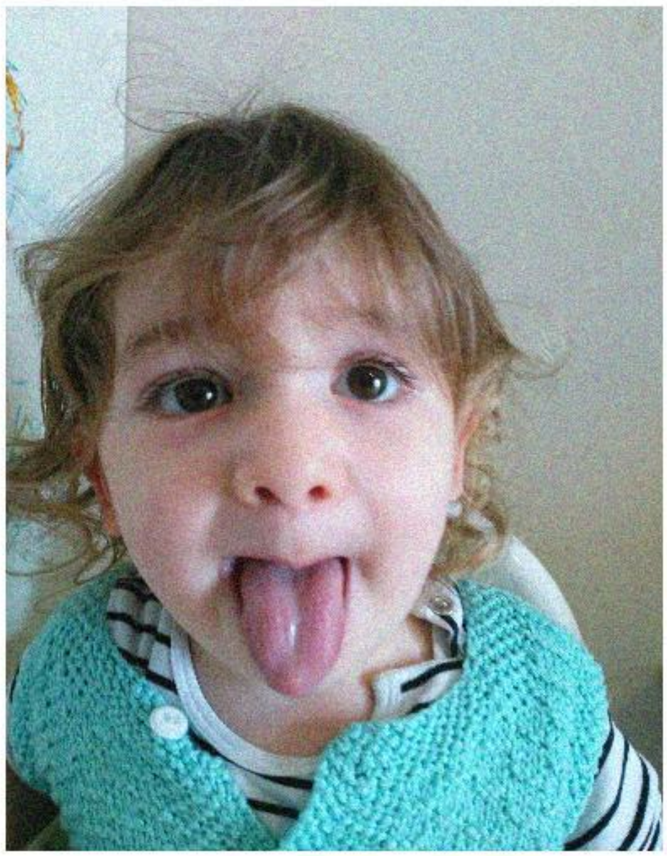


Figure 4.1 The image used in this study (ferit.jpg)

The picture is transfered in to MATLAB using the commands

```
img = imread('ferit.jpg');
unique(img)
```

In picture given in Fig. 4.1 each pixel is represented by 256 different values and the whole image can be converted in to the gray form using the below command and gray picture is obtained as in Fig. 4.2.

```
rgbImg = imread('ferit.jpg');
img = rgb2gray(rgbImg);
imshow(img)
truesize
title('Original image')
```

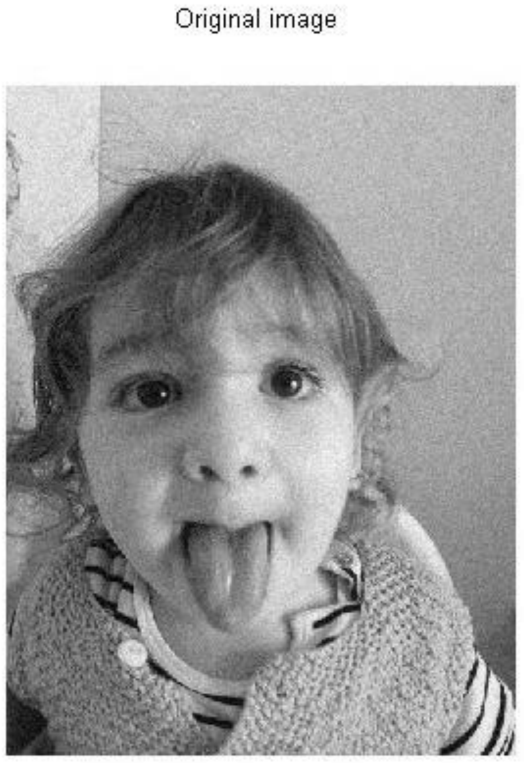


Figure 4.2 Gray image of (ferit.jpg)

The size of colorful image (Fig. 4.1) is “1040-780-3” and after conversion into gray level the size is

```
size(img)
ans = 1040      780
```

The type of our image is now “uint8” but sometimes we have to use “double” type so we can use the command is given below:

```
img = im2double(img);
class(img)
ans = double
```

We can obtain almost all important information with ‘whos’ command.

```
img = imread('ferit.jpg');

whos
```

and the answer of the codes becomes

Table 4.2 Details of an image

Name	**Size**	**Bytes**	**Class**
ans	1x5	10	char
Img	1040x780x3	2433600	uint8
ImgType	1x4	8	char
Rgb	455x642x3	876330	uint8

4.3 Implementing Gaussian Filter

First we have to design a kernel using the equation 4.1 for 2D Gaussian kernel.

$$G(x,y) = \frac{1}{2*\pi*\sigma*\sigma} * e^{-\frac{(x*x)+(y*y)}{2*\sigma*\sigma}} \tag{4.1}$$

The standart deviation is 0.6. Kernel size is taken as

$$\frac{1}{2*\pi*\sigma*\sigma} = \frac{1}{2*3,14*0,6*0,6} = \frac{1}{2.2619}$$

If we choose size of kernel as 3x3, the width of the kernel is x = 3, the height of the kernel is y = 3 as

$$\text{X} = \begin{matrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{matrix} \qquad \text{and} \qquad \text{Y} = \begin{matrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{matrix}$$

$$e^{-\frac{(x*x)+(y*y)}{2*\sigma*\sigma}} = \begin{matrix} -2.7778 & -1.3889 & -2.7778 \\ -1.3889 & 0 & -1.3889 \\ -2.7778 & -1.3889 & -2.7778 \end{matrix} \qquad (4.2)$$

And than, Gaussian kernel is as given below.

$$\text{G(x, y)} = \frac{1}{2*\pi*\sigma*\sigma} * e^{-\frac{(x*x)+(y*y)}{2*\sigma*\sigma}} = \begin{matrix} 0.0275 & 0.1102 & 0.0275 \\ 0.1102 & 0.4421 & 0.1102 \\ 0.0275 & 0.1102 & 0.0275 \end{matrix} \qquad (4.3)$$

The biggest value is 0.4421 and it is centered at the Gaussian kernel's. Density of other pixels decrease when the distance from the center value increases. Finally, we can convolve the Gaussian kernel and any image. There are a lot of functions that are frequently used each takes different time for execution. These special functions provide us faster executable files. The name of these special functions are **built-in**s. First, matlab code for Gaussian smoothing sample with“**built_in**”functions are given below:

```
img=imread('ferit.jpg');  % Picture Uploaded.
hGaussionFiltresi=fspecial('gaussian',5,2);  % kernel 5x5.
gaussianFiltresi=imfilter(img,hGaussionFiltresi,'replicate');
figure(1);
imshow(img);
title('Original image');
figure(2);
imshow(gaussianFiltresi,[]);
title('Gaussian Filter');
```

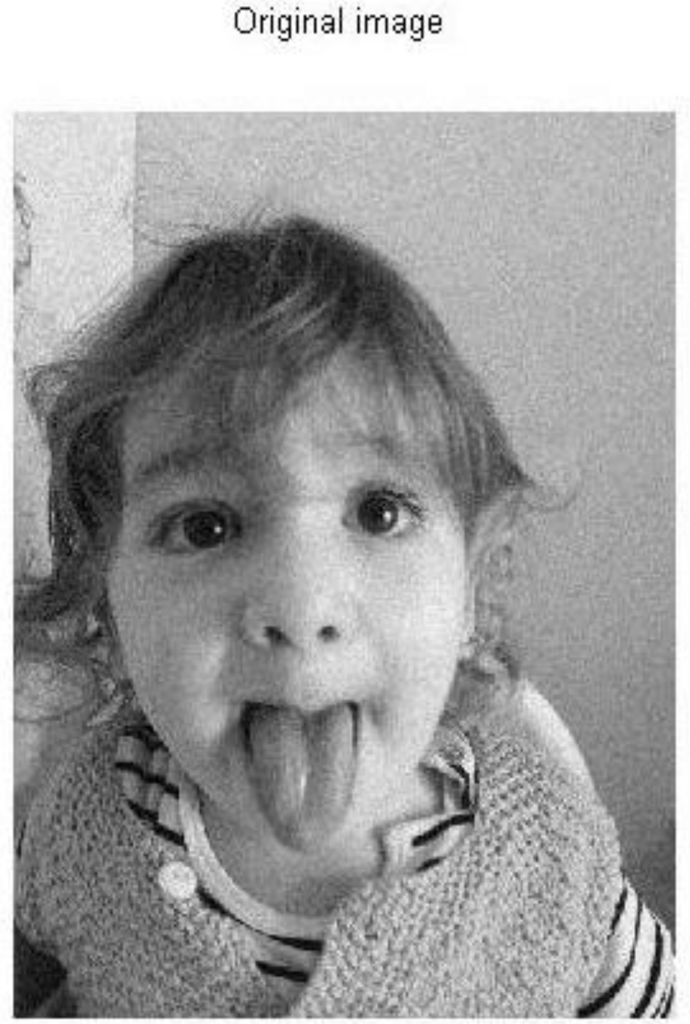


Figure 4.3 Original image (ferit.jpg)

5x5 Gaussian Kernel is taken for the convolution of the image and the result is given in Fig. 4. 4.

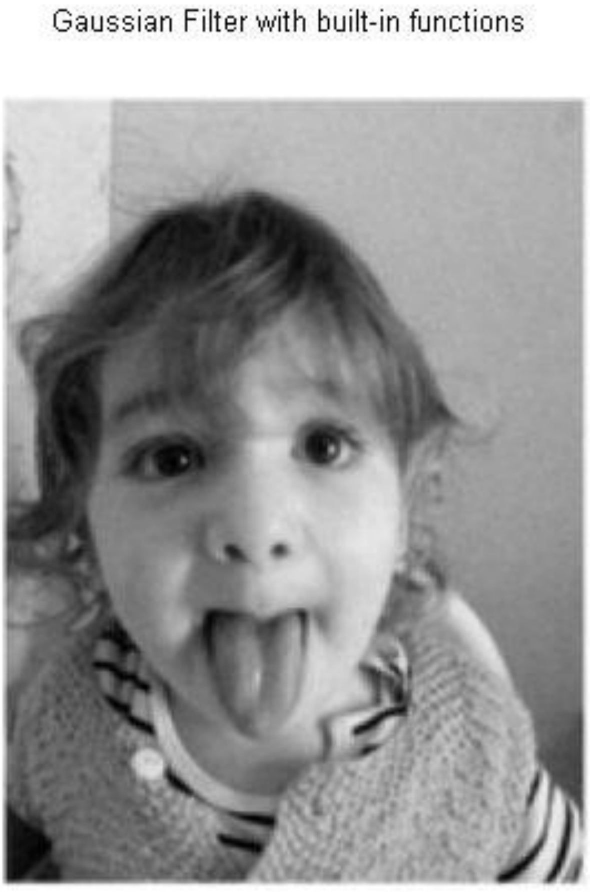


Figure 4.4 The image that convolved with 5X5 Gaussian Kernel

Than matlab codes designed for gaussian smoothing sample without built_in functions are given below:

```
a=im2double(rgb2gray(imread('ferit.jpg'))); % convert gray and read the image
figure(1)
imshow(a)
title('Original image')
[m n]=size(a); % compute size of image
f_transform=fft2(a); %Apply Fast Fourier Transform algorithm
f_shift=fftshift(f_transform); % shift to center and rearranges the outputs of fft2
p=m/2; %center of m
q=n/2; %center of n
d0=70; % standard deviation
for i=1:m
for j=1:n
distance=sqrt((i-p)^2+(j-q)^2);
low_filter(i,j)=exp(-(distance)^2/(2*(d0^2)));
end
end
filter_apply=f_shift.*low_filter;
image_orignal=ifftshift(filter_apply); %undoes the result of Fast
Fourier Transform
image_filter_apply=abs(ifft2(image_orignal)); %return the absolute
value of each element
figure(2)
imshow(image_filter_apply,[])
title('Gaussian filter applied')
```

The result of execution algorithms which are for gaussian smoothing sample without built_in functions is given in Figure 4.5

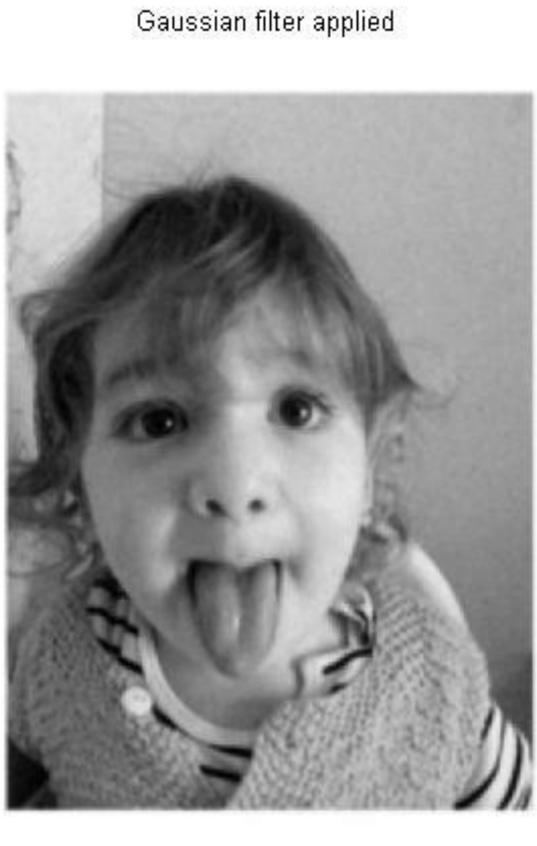


Figure 4.5 Gaussian filter applied with built_in functions

In above statements, putting forward and developing a method to identify the specific lines and shadows of colorful images (x * x) pixels are represented. The obtained special lines and shadows are used for analysis and developments. When the pictures are caricatured necessary and unnecessary all lines and shadows are seen and we want to have basic and specific ideas about image by giving particular importance to dominant lines. These dominant lines provide us evaluating specific features of images with less and clear information. There are a lot of methods for obtaining less and clear information one of important method is Gaussian Blur filtering.

As explained in the previous sections, almost all images are affected by a kind of noise. In order to analyze and extract the features of images with less and clear information, firstly we have to eliminate the noises of image.

As seen in Figure 4.4 and 4.5, Gaussian filtering minimizes the noise in images. We applied a standard or a new created Gaussian algorithm which gives same useful and efficient results for evaluating specific features of images with less and clear information. The dominant lines provide us evaluating specific features of images with less and clear information but obtaining these dominant lines is very important and critical.

Firstly, with Gaussian filter we filter out of noises of image. Secondly, the values of pixels are examined and a reference value is chosen according to spread of line and shadow of image. Related algorithm is prepared with combination of the new constituted Gaussian filtering algorithm and a few loops. Finally, reference value is optional it is chosen according to situation of image and combinated algorithm is a suplementary method because it can determine both values which are smaller and bigger than reference value.

Some new constituted algorithms and the result of these algorithms are given below. Especially the result of situations which are mentioned above will be seen. First, the image is loaded and readed in MATLAB. Second, standard and new design Gaussian filtering algorithms are written. And then, special algorithms and loops are used.

For example, after analyzing pixel values of image, a reference value is chosen and in below algorithm 0.3 is decided as suitable number. Our algorithm shows that if pixel value is bigger than reference value it is replaced by another selected value (5), if pixel value is smaller than reference value then it is replaced by another selected value (0.1).

Our aim is to compare the result and method of extracting the dominant lines by two different ways. One of them is using applied Gaussian filter, other one is using normal mathematical algorithms. Some new constituted algorithms and a few practical loops are combined. The result of this combination, the normal mathematical algorithms and its results, combinated algorithm with applied Gaussian filter and its results are given below.

- The normal mathematical algorithms and its results

```
a=im2double(rgb2gray(imread('ferit.jpg')));
figure(1)
imshow(a)
title('Original image')
[m n]=size(a);
for i=1:m;
for j=1:n;
   if (a(i,j)>0.3);
      a(i,j)=5;
   else
      b(i,j)=0.1;
   end
end
end
figure(2)
imshow(a)
title('Without Gaussian, changing pixel values if a(i,j)>0.3 ')
figure(3)
imshow(b)
title('Without Gaussian, changing pixel values if a(i,j)<0.3 ')
```

The result of these algorithms are given in Figures 4.6, 4.7 and 4.8.

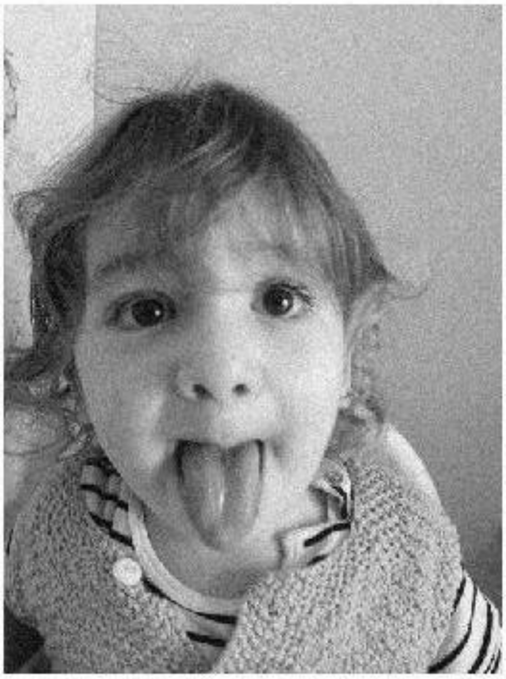
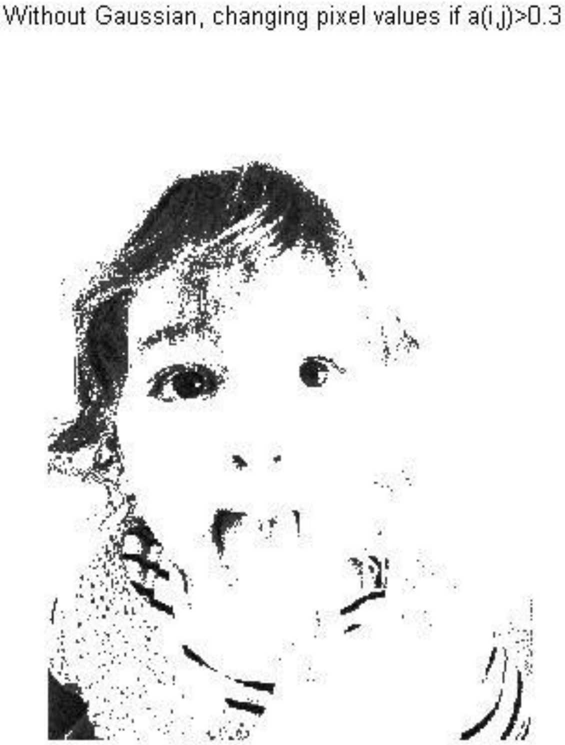


Figure 4.6 Original image



Without Gaussian, changing pixel values if a(i,j)>0.3

Figure 4.7 Math. algorithms applied, if pixel value > 0.3 change it with 5

Figure 4.8 Math. algorithms applied and if pixel value < 0.3 change it with 0.1

- Combinated algorithm with applied Gaussian filter and its results are

```
a=im2double(rgb2gray(imread('ferit.jpg')));
figure(1)
imshow(a)
title('Original image')
[m n]=size(a);
f_transform=fft2(a); % fast Fourier transform algorithm
f_shift=fftshift(f_transform);
p=m/2;
q=n/2;
d0=85;
for i=1:m
for j=1:n
distance=sqrt((i-p)^2+(j-q)^2);
low_filter(i,j)=exp(-(distance)^2/(2*(d0^2)));
end
end
filter_apply=f_shift.*low_filter;
image_orignal=ifftshift(filter_apply);
image_filter_apply=abs(ifft2(image_orignal));
```

```
figure(2)
imshow(image_filter_apply,[])
title('Gaussian filter applied')
for i=1:m;
for j=1:n;
 if (a(i,j)>0.3);
        image_filter_apply(i,j)=5;
    else
        b(i,j)=0.1;
    end
end
end
figure(3);
imshow(image_filter_apply,[]);
title('After Gaussian, changing pixel values if a(i,j)>0.3 ')
figure(4);
imshow(b);
title('After Gaussian, changing pixel values if a(i,j)<0.3 ')
```

The result of these algorithms are given in Figures 4.9, 4.10, 4.11, and 4.12.

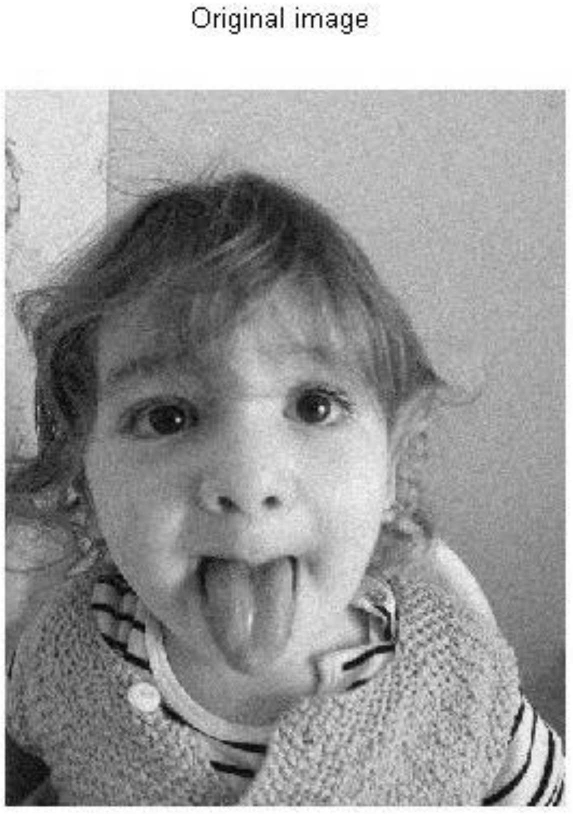


Figure 4.9 Original image

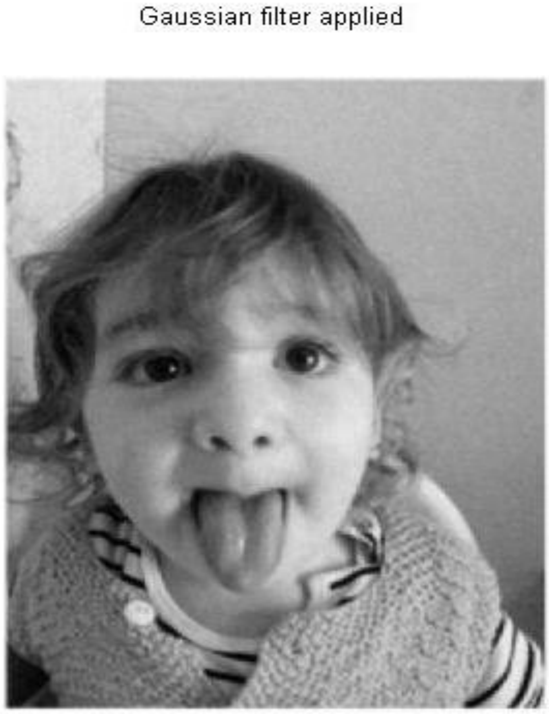


Figure 4.10 Gaussian filter applied image

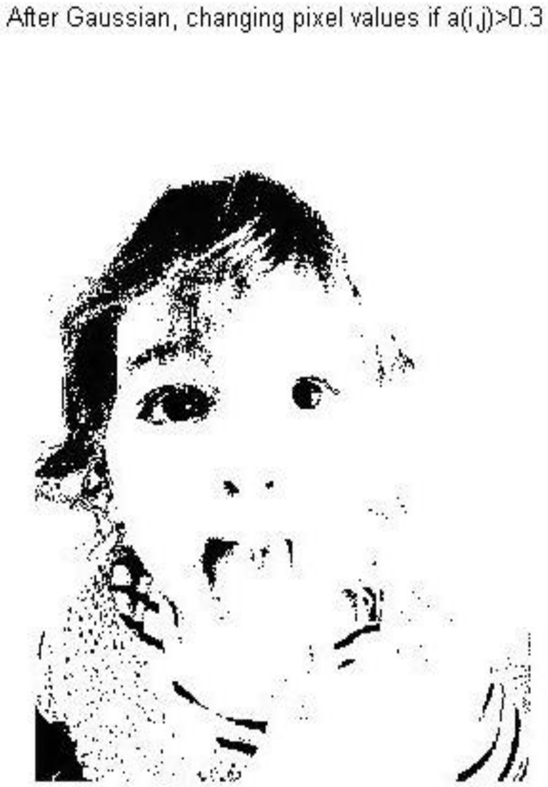


Figure 4.11 After Gaussian filter, if pixel value > 0.3 change it with 5

Figure 4.12 After Gaussian filter, if pixel value < 0.3 change it with 0.1

We can see easly the difference betwen Gaussian Filtering and normal mathematical algorithms results in Figures 4.13 and 4.14. It is obvious that the dominant lines and shadows are seen much better in Figure 4.14 which is the one obtained using Gaussian filter algorithms where Fig. 4.13 is obtained by mathematical aigorithm.

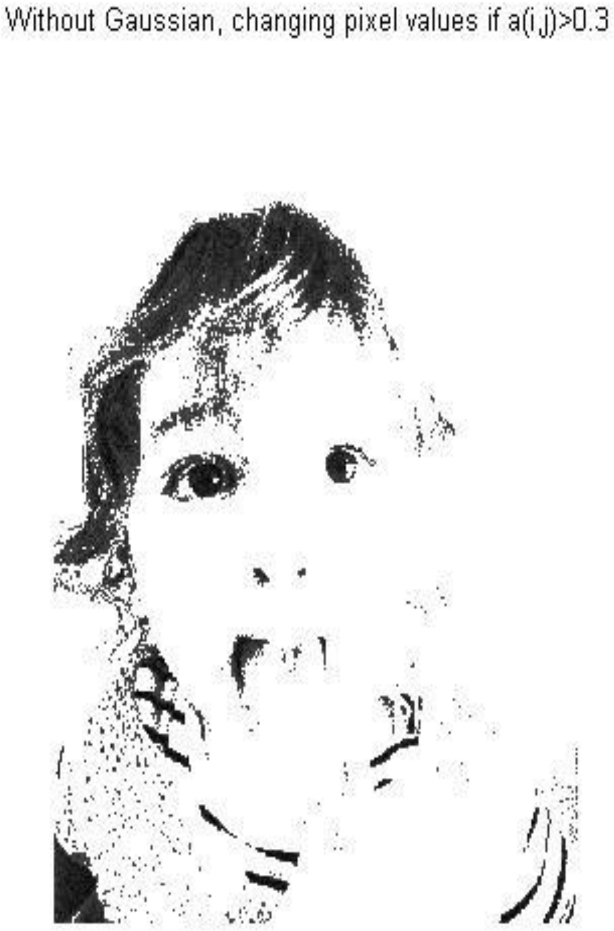


Figure 4.13 Math. algorithms applied and if pixel value > 0.3, change it with 5

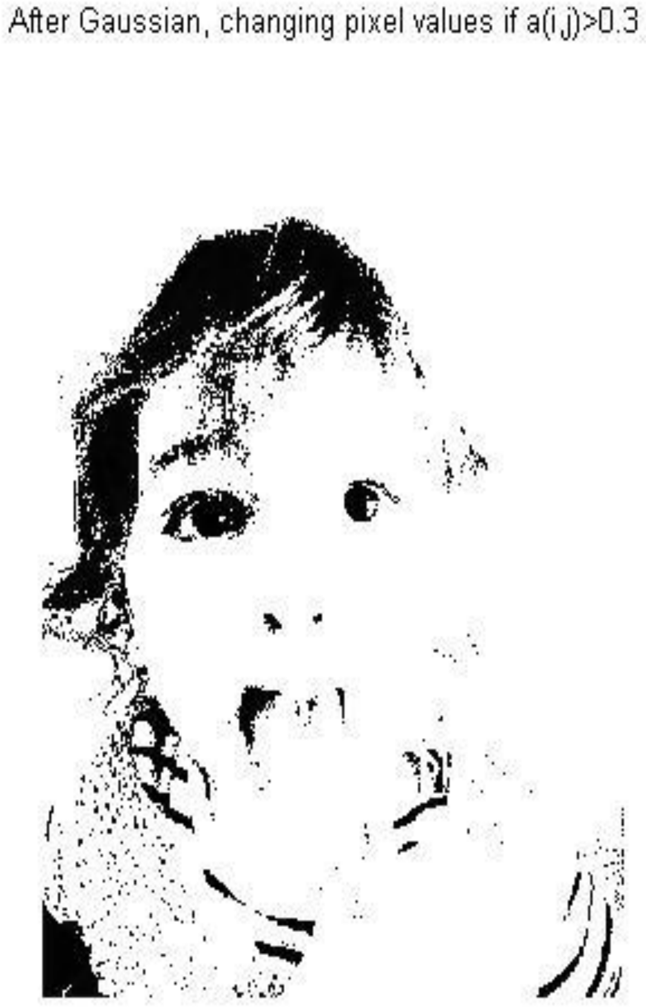


Figure 4.14 Gaussian filter applied and if pixel value > 0.3, change it with 5

CHAPTER 5
CONCLUSIONS

In this study, the main purpose is focused on the determination of how the image is affected by the coefficients of a Gaussian filter and which Gaussian coefficients are most appropriate for each image to obtain the most simplified (minimum information) version of the picture. The dominant lines of an image is obtained and Gaussian filter is applied to provide and evaluate specific features of images with less and clear information. Determination of these dominant lines is very important and critical because the size of the data decreses considerably and it reduces the execution time of the any digital operation on the image.

First, with Gaussian filter, the noise in the image is filtered out. Secondly, the values of pixels are examined and a reference value selected according to spread of line and shadow of image. Later, the related algorithm is prepared with combination of the new constituted Gaussian filtering algorithm and a few loops. Finally, an image has been obtained that cleared from noises and put forward its specific features with less and obvious information.

The slightly modified version of the average filter using the Gaussian distribution is known as the Gaussian filter. Gaussian filtering is also a Fourier Transformation. The Gaussian filter allows an infinite transfer function to filter out a final window (scan window) in the spatial domain. This makes the basic problem of filtering easier to solve.

As a result, the dominant lines in the image (ferit.jpg) is obtained without loosing the dominant information in the picture. This study also covers the comparison of the results and methods of extracting the dominant lines by two different ways.

REFERENCES

[1] Richard F. Lyon, A Brief History of Pixel, 2006, pp.2

[2] Florack Luc, October 2003, pp.47

[3] Steven W. Smith, The Scientist and Engineer's Guide to Digital Signal Processing, 1997-1998, pp.375)

[4] Winder Steve, Analog and digital filter design, 2002, pp.20

[5] Paarmann Larry, Design and Analysis of Analog Filters, 2001, pp.5

[6] Davenport and Root, 1987; Gardner, 1986; Haykin, 1989; Therrien, Design and Analysis of Analog Filters, 1992, pp.6

[7] Alexander, 1986; Haykin, 1996; Honig and Messerschmitt, 1984; Widrow and Stearns, Design and Analysis of Analog Filters, 1985, pp.6-7

[8] Brown and Hwang, 1992; Candy, 1986; Chui and Chen, Design and Analysis of Analog Filters, 1991, pp.7

[9] Haykin, Design and Analysis of Analog Filters, 1996, pp.7-8

[10] Cunningham, 1992; Hamming, 1989; Loy, 1988; Oppenheim and Schafer, 1975; Oppenheim and Schafer, 1989; Parks and Burrus, 1987; Proakis and Manolakis, 1988; Terrell, 1988; Williams, 1986, Design and Analysis of Analog Filters, pp.7-8

[11] Sheahan and Johnson, Design and Analysis of Analog Filters, 1977, pp.7-11

[12] R. Whalley, Two-dimensional digital filters, Appl. Math. Modelling, 1990, **14**, June, pp.304-305

[13] Parks T. W. - Burrus C. Sidney, Digital Filter Design, 1987, pp.153-154

[14] Milivojević Zoran, Digital filter design, 2009, pp.4

[15] Fisher Robert, Perkins Simon, Walker Ashley and Wolfart Erik, Hypermedia Image Processing Reference, 2000, pp.154

[16] Hoffmann Gernot, Gaussian filter, 2002, pp.1

[17] Fisher Robert, Perkins Simon, Walker Ashley and Wolfart Erik, Hypermedia Image Processing Reference, 2003, pp.174

[18] Fisher Robert, Perkins Simon, Walker Ashley and Wolfart Erik, Hypermedia Image Processing Reference, 2000, pp.179