ADAMA SCIENCE AND TECHNOLOGY UNIVERSITY

SCHOOL OF APPLIED NATURAL SCIENCE
DEPARTMENT OF MATHEMATICS

A THESIS SUBMITED IN PARTIAL FULFILMENT OF THE REQUIRMENT FOR THE DEGREE OF MESTERS IN

HEAT EQUATION AND ITS APPLICATION ON IMAGE PROCESSING

By Dinku Degu
ADIVISOR; Lemi Guta (PhD)

Adama Ethiopia

August 15,2017

Approval Sheet1

We, the undersigned, members of the Board of Examiners of the final open defense by **Dinku Degu Wayecha** have read and evaluated his thesis in titled **"heat equation and its application on image processing "** and Examined the candidate. This is therefore to certify that the thesis has been accepted in partial fulfillment of the requirement of the degree of Master of Science in applied Mathematics.

Name of Chairman	Signature	Date
Name of Advisor	Signature	Date
Name of External Examiner	Signature	Date
Name of internal Examiner	Signature	Date

Acknowledgment

First of all, I would like to thank almighty God for carrying my life and blessing my activities in advance of the completion of my thesis work. My special and heart full gratitude and appreciation goes to my adviser Dr.Lemi Guta for his great assistance, contribution, useful suggestion and constructive criticism on this excellence advising and limitless effort in encouraging me in my work, correcting and giving comments by devoting his time. I greatly appreciate him not only for unlimited constructive advises in every aspect of my thesis work, but also with his professional and personal ethic, that should be an icon for others. My thanks also goes to Mr.Abdo kedir,Mr.lencho Kumbi candidates for their knowledge sharing and cooperation, and all classmates for their cooperative work through our study time and doing the thesis. Lastly but not the least my heartfelt gratitude and appreciation goes to my Lovely wife Buzunesh Awugichew for her continuing love,patience,motivation, encouragement,and constructive advice throughout this long and challenging process.All My friends blessings to all those who supported me in one way or another during my studies and the time of this thesis work.

THANK YOU!

Abstract

In this paper, I tried to find the solution of some image processing problems. In case, I use both isotropic diffusion and Gaussian convolution for de-noising gray scale images. Nowadays there are a number of methods to solve digital image processing problems which employ mathematical algorithms. Isotropic diffusion acting as a linear filter is the most basic approach to regularize or smooth image data. However it blurs fine details of images. For my work I am interested in Isotropic diffusion and the convolution of the given image with Gaussian kernel at specific time. Experimental result shows that both approaches generate almost similar output.

Contents

CHAPTER ONE

1 Introduction

Image restoration and enhancement are important parts of digital image processing (Markowich,2005 ; matevosyan,2005) and belonging to the early visual image processing problems. Image pre-processing is the necessary preliminary work of image analysis such as filtering to reduce image noise and to enhance the image edges.(Yanibian University Chaina,2012) The image enhancement technique plays an important role in improving image quality and is good for image post-processing e.g. image segmentation and image tracking. Image restoration and enhancement have been widely used in military, medical, industrial production and other fields

Image enhancement means a processing method to highlight some information in an image according to the specific needs meanwhile weaken or remove the information unwanted. Its main purpose is to make the processed image more suitable for a particular application than the original image. Therefore such treatments improve image quality for some application purpose. There are two kinds of image enhancement methods: the spatial domain method and the frequency domain method. The spatial domain method is mainly used to do direct operation on the pixel gray values in the space domain, such as the grayscale transformation, histogram modification, spatial domain smoothing, image sharpening and pseudo-color processing (Caselles,1998). The frequency domain method means calculating the image transformation value in a certain image transform domain such as the Fourier transform first then the image frequency domain filtering finally performing inverse transformation of the filtered image transformation value to the spatial domain to obtain the enhanced image. This is an indirect approach

The main goal of this study is the application of heat equation on image processing. The solution to these problems is studied in Heat equation (partial differential) equation. A short description of this thesis is presented in to seven chapters. The first chapter contains background of the study, statement of the problem, objectives of the study delimitation and limitation of the study. The main goal of this chapter is to provide a description over view of the study. The second chapter is review of related literature, which describes how deep the researcher gets information on the issues. The third chapter discuss about the methods and procedures which give a brief over view of the thesis and shows how the researcher interprets the information and goes to his conclusions. The fourth chapter deals with the main point (governing equation) by discritization the method we use and solve the problem. The fifth chapter is about discussion and result of the study that tells us what we are going to learn from this study and the sixth chapter is recommendation about the study and lastly on chapter seven there is reference that tells from where the researcher gets different information on his work.

1.1 statment of the problem

The focus of this study is application of Heat equation on image processing especially by using mathematical convolution this method is important for de-noising and enhancements of an image. The easiest way of removing noise of image is to select the average of the image locally in each pixel. In terms of computational imaging this can be expressed as the heat equation

$$U_t - \Delta U = 0 \tag{1}$$

$$U(0, x) = U_0(x)(noisyimage). \tag{2}$$

According to the basic theory of PDEs, the solution of this equation is expressed using the heat kernel As follows:

$$U(t, x) = \int_{R^d} K(t, x - y)U_0(y)dy \tag{3}$$

$$K(t, x) = \frac{1}{(4\pi t)^{\frac{d}{2}}} \exp \frac{-|x|^2}{4t} \tag{4}$$

In typical image processing techniques we use this method mostly and the most convolution kernel in image processing which is equivalent to that of heat equation is Gaussian convolution. The convolution on a gray image f and Gaussian filter can be represented as follow; For one dimension

$$(G * f)(x) = \int_R G_\sigma(x - y)f(y)dy \tag{5}$$

And for 2D

$$(G_\sigma * f)(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(a, b)G_\sigma(x - a, y - b)dadb \tag{6}$$

Where G_σ represents two dimensional Gaussian kernel of variance σ^2

$$G_\sigma(x) = \frac{1}{2\pi\sigma^2} \exp \frac{-|x|^2}{2\sigma^2} \tag{7}$$

The variance of time field of Gaussian kernel is $\sqrt{2t}$ and smoothing structure of degree σ should be stopped on the time $T = \frac{1}{2}\sigma^2$ Gaussian convolution and linear diffusion process have equivalence relation. For image f the linear diffusion process is as follows

$$\partial_t U = \Delta U \tag{8}$$

$$U(x,t) = \{^{f(x)t=0}_{(G_{\sqrt{2t}} * f)(x)t>0} \tag{9}$$

Even if linear filtering and Gaussian convolution have equivalence relation on image processing it has gaps because of detecting the edges of the image. So the process needs regularization or normalization to preserve the edges
Hence this thesis answers the following questions
1. How we can apply Heat equation in the image processing?
2.How can we normalize the process to preserve the edges of the image?
3.What is the relation between heat equation and Gaussian smoothing on image processing?

1.2 Objectives of the study

1.2.1 General objective

The general objectives of this research is solving problems of image processing by using heat equation

1.2.2 Specific Objectives

The specific objectives of this thesis are
•To Learn how we discretize and numerically approximate solutions of image processing problems by heat equation using MATLAB
•To Knows the method of preserving edges of the image on image processing
•To Learn the similarities and the difference of image processing by heat equation and Gaussian convolution
•To Develop the knowledge of how much smoothing is important in image processing.

1.3 Significance of the study

The main purposes of this study are
• We can see how much heat equation is important in image processing
• It is possible to use isotropic diffusion (linear filters) and Gaussian filters interchangeably

• Longer evaluation corresponding to convolution with Gaussian of higher variance
• It gives us firm footing for studying heat equation in context of image processing Moreover it provide to the researchers own contribution to all natural science students especially for (mathematics and physics) researchers(for post graduate students) who may use as additional reference on the case (i.e. application of smoothing on image processing) and for similar work in the university

1.4 Delimitation (scope) of the study

This study is mainly delaminated on solving problems of image processing by using heat equation especially (isotropic diffusion)

1.5 Limitation of the study

The limitations of the study are due to
• Shortage of reference material
• Lack of internet service on work places
• Shortage on review of related literature on the case

CHAPTER TOW

2 REVIEW OF RELATED LITERATURE

2.1 Application of heat equation (PDE) on image processing

In image processing researchers have been quite interested in PDE-based frame work in recent years. There has been a lot of research based on heat equation concerning image processing. Among this the Gaussian smoothing and the Fourier transformation with convolution (Bromilely,2014) is commonly used PDE based problems. In the past 20 years because human vision has high-level pursuit of image restoration and enhancement, doing image processing by mathematical methods has became an important project. Among those methods heat equations and functional analysis are preferred. Thus a series of mathematical methods are posed including heat equation model used in image processing and solving partial differential equations (INISTA et al.,2009) with computer. The idea of doing image processing with Heat equation could date back to (Gabor and Jian,1965) but this method was not established until (Koenderind and Witikin,1983. They introduced the concept of Scale Space which represented a group of images in different scales simultaneously. Their distribution constituted the basis of doing image processing with heat equation to a great degree. The scales of an image are obtained by Gaussian smoothing using the classical heat conduction equation for the image evolution can also bring the Scale Space

In the late (Humme,1989) proposed that the heat conduction equation is not the only one for Scale Space constitution and suggested some principles instead. The an-isotropic diffusion model proposed by (perona and Malik,1983) is the most influential one in this field. They suggested replacing the Gaussian diffusion with a diffusion which can keep the selectivity of edge. This incurred large amounts of researches on theoretical and practical problems. Under the same framework the Shock filter proposed by (Osher and Rudin,1990) and the method of total variation (TV) reduction proposed by (Rudin,1990) further stressed the importance of the heat equation equations in image processing. In the fields of image processing and computer vision some other partial differential equations are based on the curve and surface of curve linear motion.(Rudin,1990) Developed the level set numerical algorithm. They tried to describe the deformation of curves surfaces or images with a higher dimensional and hyper surface level set. This technique not only made the numerical results more accurate but also solved the topology problem difficult to solve previously. Heat equation can also be used in image segmentation. The model proposed by (Muford and shah,1989) combined a variety of image segmentation algorithms incurring many new theoretical and practical problems. Proposed the image segmentation algorithm based on moving boundary also had a great influence and later many scholars expanded their work with the geometric partial differential equations

Heat equation can also be used in image in painting (Chan and Shcen,2001) and (Muford and Shah,1989). It is synonymous with the image interpolation originally coming down from the artists who restored broken works of art by hand in museums.(Gonzalez,2004) Currently the digital image in painting technique is widely applied to the image processing visual analysis and digital technology. for example image restoration, image enlargement, image super resolution analysis and error concealment in the wireless image transmission etc

As the models for image processing emerging multiple mathematical methods that solve the continual models represented with Heat equation of partial differential equations also appeared. In terms of speeding up the calculating speed as describe complexity calculation should be taken into account when calculating the continual integration with models based on partial differential equations. There are two kinds of methods to cope with. One kind of methods is to find a time integration algorithm without solving the an-isotropic diffusion equation the calculation speed of which is faster than solving the an-isotropic diffusion equation. Meanwhile regularization can eliminate required demands of algorithms like of the sensibility to noise instability and consistency of calculation which means getting a parallel result to that of the an-isotropic diffusion method by less calculated amount. Common methods are finite difference method finite element finite volume or spectral method

(Foyer and Zou,2006); (Caselles,1998). The second kind of methods is to reduce the frequency of continuous integration equations that an-isotropic diffusion equation needs by the adaptive grid algorithm. researches about the imaging of human retina to geometry structures indicate that it can result in enhanced image of large size by utilizing variable grid integrating algorithm in relatively less time. It shows that the basic equations for image processing are (Osher and Rudin,1990), total variation, an-isotropic diffusion equation, moving boundary and so on. They can realize the recovery enhancement and segmentation of images

Heat equation is a sophisticated method of image analysis and processing of great values of research and application which needs a deep study. As both the variational model and the an-isotropic diffusion model have a complete theoretical framework. (Weickert,1998) A non linear diffusion which is a variety of models and sophisticated numerical schemes is the introduction of the fields of digital image processing and computer vision provides a powerful tool to solve problems undoubtedly (Morris et al.,N,27112). This paper concerns about the applications of the heat equation in image restoration and image enhancement. We mainly assay equations in image restoration as well as their improved algorithm for image enhancement

2.2 The basic theory related to image enhancement algorithm

The image enhancement algorithms that based on heat equation mainly are diffusion method and variation method. Compared to other methods there are some advantages in calculation. Firstly they can process important geometric characters directly which affects visual effect. Such as gradient angle curvature and so on. The diffusion method (Weickert,1998) can do simulation effectively on linear and nonlinear diffusion. Secondly the diffusion method and variation method get well development on calculation of current relative mathematics analysis theory and partial differential equation.

2.3 Evaluation methods of image processing quality

The research of evaluation of image processing quality is one of the basic studies in image information science. The image is the mainstay of information for image processing or image communication morphology while the quality of an image is the important indicator when measuring the system. Image enhancement is to improve the subjective visual display quality While image restoration is used to compensate for image degradation so that the quality of restored image can keep the same as the original one as possible. All these call for an appropriate evaluation method of image processing quality. Image processing quality encompasses two aspects one is the fidelity of an image that is the degree of the deviation between evaluated image and the original standard image the other one is the intelligibility of an image which means the ability of the image providing information for mankind or machines. The ideal situation is to find a quantitative description method of fidelity and intelligibility of an image that can be seen as the basis of the image evaluation and the image system design. However, due to the lack of full understanding of both the characteristics of human vision, system and quantitative description methods of human psychological factors are the most frequently used and most authoritative method is the so-called subjective evaluation method

2.4 Gaussian function

In order to know the structure of the image one should analyze the deviation of gray value in the neighborhood of every pixel. That is calculating the gradient information of the image. However, affected by noise and so on a small disturbance of the initial image can cause arbitrarily large deviation of derivative. So the regularizing method is needed. One available regularizing method is making convolution between the image and Gaussian kernel firstly (KIM,2013). We can find that all images seeking derivative experience the same Gaussian smoothing process which is equivalent to the convolution between an image and a Gaussian function from equation

$$\partial_{x1}^{n}\partial_{x2}^{m}(K_\sigma * f) = K_\sigma * (\partial_{x1}^{n}\partial_{x2}^{m}K_\sigma * f) \tag{10}$$

Gaussian derivative can also show differential invariance after rotation transmission. Such as

$$|K_\sigma * U| and K_\sigma * U \tag{11}$$

It is very useful in checking the boundary and other structures of the image.

CHAPTER THREE

3 METHODS AND PROCEDURES

3.1 Design

Now a day there are a number of methods in image processing these are linear and non linear filtering, edge detection, enhancement, Deblurring, restoration, image in painting shape extraction and analysis, image segmentation and so on. The design of this study focuses on solving problems of image processing by using heat equation especially linear filtering and Gaussian convolution on my work

3.2 linear filtering

3.2.1 Correlation and convolution

Correlation and Convolution are basic operations that we will perform to extract information from images. They are in some sense the simplest operations that they can be analyzed and understood very well and they are also easy to implement and can be computed very efficiently. Our main goal is to understand exactly what correlation performs on an image but they are extremely useful. Moreover because they are simple and convolution do and why they are useful. We will also touch on some of their interesting theoretical properties; though developing a full understanding of them would take more time than we have.

These operations have two key features they are shift-invariant and they are linear. Shift-invariant means that we perform the same operation at every point in the image. Linear means that this operation is linear that is we replace every pixel with a linear combination of its neighbors. These two properties make these operations very simple its simpler if we do the same thing everywhere and linear operations are always the simplest ones. We will first consider the easiest versions of these operations and then generalize. Well make things easier in a couple of ways. First convolution and correlation are almost identical operations. So we will begin by only speaking of correlation, and then later describe convolution. Second we will start out by discussing 1D image. We can think of a 1D image as just a single row of pixels. Sometimes things become much more complicated in 2D than 1D but luckily correlation and convolution do not change much with the dimension of the image so understanding things in 1D will help a lot. Also later we will find that in some cases it is enlightening to think of an image as a continuous function but we will begin by considering an image as discrete meaning as composed of a collection of pixels.

3.2.2 Notation

We will use uppercase letters such as I and J to denote an image. An image may be either 2D as it is in real life or 1D. We will use lowercase letters like i and j to denote indices or positions in the image. When we index into an image we will use the same conventions as Mat lab. That means the first element of an image is indicated by 1. So if I is a 1D image I (1) is its first element. For 2D images we give first the row then the column. So I (i, j) is the pixel in the ith row and the jth column of the image.

3.2.3 A Mathematical Definition for Correlation

Its helpful to write this all down more formally suppose F is a correlation filter it will be convenient notationally to suppose that F has an odd number of elements so we can suppose that as it shifts its center is right on top of an element of I. So we say that F has 2N+1 element and that these are indexed from -N to N so that the center element of F is F (0). Then we can write

$$FoI(x) = \sum_{i=-N}^{N} F(i)I(x+i) \tag{12}$$

Where the circle denotes correlation with this notation we can define a simple box filter as:

$$F(i) = \{^{\frac{1}{3} fori=-1,0,1}_{0 fori\neq -1,0,1} \tag{13}$$

We can think of it as a 1x3 structure that we slide along the image. At each position we multiply each number of the filter by the image number that lies underneath it and add these all up. The result is a new number corresponding to the pixel that is underneath the center of the filter.

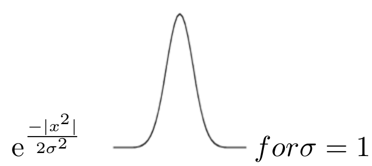
3.2.4 Constructing a Filter from a Continuous Function

It is pretty intuitive what a reasonable averaging filter should look like. Now we want to start to consider more general strategies for constructing filters. It commonly occurs that we have in mind a continuous function that would make a good filter and we want to come up with a discrete filter that approximates this continuous function. Some reasons for thinking of filters first as continuous functions will be given when we talk about the Fourier transform but in the mean time well give an example of an important continuous function used for image smoothing by Gaussian. A one-dimensional Gaussian filter is:-

$$G(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp \frac{(-x-\mu)^2}{2\sigma^2} \tag{14}$$

This is also known as a normal distribution. μ is the mean value and σ is the variance. Here is a plot of a Gaussian

Figure 1: 1D model of Gaussian noise



This means μ gives the location of the peak of the function. The parameter σ controls how wide the peak is. As σgets smaller the peak becomes narrower. The Gaussian has a lot of very useful properties some of which well mention later. When we use it as a filter we can think of it as a nice way to average. The averaging filter that we introduced above replaces each pixel with the average of its neighbors. This means that nearby pixels all play an equal role in the average and more distant pixels play no role. It is more appealing to use the Gaussian to replace each pixel with a weighted average of its neighbors. In this way the nearest pixels influence the average more and more distant pixels play a smaller and smaller role. This is more elegant because we have a smooth and continuous drop-off in the influence of pixels on the result instead of a sudden discontinuous change. We will show this more rigorously when we discuss the Fourier transform. When we use a Gaussian for smoothing we will set μ=0 because we want each pixel to be the one that has the biggest effect on its new smoothed value. So our Gaussian has the form:

$$G_\sigma(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp(\frac{-|x|^2}{2\sigma^2}) \tag{15}$$

σwill serve as a parameter that allows us to control how much we smooth the image that is how big a neighborhood we use for averaging. The bigger the more we smooth the image. We now want to build a discrete filter that looks like a Gaussian. We can do this by just evaluating the Gaussian at discrete locations. That is although G is defined for any continuous value of x we will just use its values at discrete locations (-3, -2, -1, 0, 1, 2, 3). One way to think about this is that we are approximating our original continuous function with a piecewise constant function where the pieces are each one pixel wide. However, we cannot evaluate G at every integer location because this would give us a filter that is infinitely long which is impractical. Fortunately as x gets away from 0 the value of G(x) will approach 0 pretty quickly so it will be safe to ignore values of x with a high absolute value. There is no hard and fast rule about when it is safe to ignore some values of G(x) but a reasonable rule of thumb is to make sure we capture 99 percent of the function. What we mean by this is that the sum of values in our filter will contain at least 99 percent of the function. What we mean by this is that the sum of values in our filter will contain at least 99 percent of what we would get in an infinitely long filter that is we choose N so that:

$$\frac{\sum_{-N}^{N} G(x)}{\sum_{-\infty}^{\infty} G(x)} > .99 \tag{16}$$

The other point is an averaging filter should have all its elements add up to 1. This is what averaging means if the filter elements do not add up to one then we are not only smoothing the image we are making it dimmer or brighter. The continuous Gaussian integrates to one after all it is a probability density function and probabilities must add up to one but when we sample it and truncate it there is no guarantee that the values we get will still add up to one. So we must normalize our filter. Meaning the elements of our filter will be calculated as:-

$$F(i) = \frac{G(i)}{\sum_{-N}^{N} G(x)} fori = -N - - - - N \tag{17}$$

3.3 Convolution

Convolution can be intuitively described as a function that is the integral or summation of two component functions and that measures the amount of overlap as one function is shifted over the other. An easy way to think of convolution with respect to one variable is to picture a square pulse sliding across the x-axis towards a second square pulse. The convolution at a point is the product of the two functions that occurs when the leading edge of the moving pulse is at that point. When actually taking the convolution of two functions one function is flipped with respect to the independent variable before shifting and a change of variables from x to y is used to facilitate the shifting operation. In one dimension the mathematical definitions of convolution in discrete and continuous time are indicated by the * operator If f and g are functions in x then the convolution of f and g over an infinite interval is an integral given by:

$$F * g(x) = \int_{-\infty}^{\infty} F(y)g(x-y)dy \tag{18}$$

If the convolution is performed over a finite range [0, t], then the convolution is:

$$F * G(x) = \int_{0}^{t} F(y)g(x-y)dy \tag{19}$$

To understand these equations we can make some simple observations. Notice that because of the change-of-variables f and g are functions of y under the integral but f * g is still a function in x. Since the reflection of function p (z) is given by p (-z) it is clear that g (x , -y) is a reflection of g (y) shifted by an amount x on the y axis. Hence as the integral with respect to y is evaluated the amount that the function g (x ,-y) changes rather the function g slides from $-\infty$ to ∞ or from 0 to t. Convolution is just like correlation except that we flip over the filter before correlating. For example convolution of a 1D image with the filter (a,b,c) is exactly the same as correlation with the filter (c,b,a). We can write the formula for this as

$$F * I(x) = F(i)I(x-i) \tag{20}$$

The one-dimensional versions are given here as they are the simplest presentation of the convolution operation. However since images are intrinsically two dimensional the 2-D extension of discrete convolution is required to perform convolution with images. The two-dimensional and one-dimensional versions are very similar in that the two are identical save for an additional set of indices:

$$F * I(x,y) = \sum_{j=-N-i=N}^{N} \sum^{N} F(i,j) I(x-i, y-j) \tag{21}$$

We must imagine that the two-dimensional case as one matrix sliding over the other one unit at a time with the sum of the element-wise products of the two matrices as the result the complete convolution is found by repeating the process until the kernel has passed over every possible pixel of the source matrix. In the case where the two matrices are a source image and a filter kernel the result of convolution is a filtered version of the source image.

3.4 Correlation compared to Convolution

Definition:-correlation

$$I(x,y) = \sum_{j=-k}^{k} \sum_{i=-k}^{k} F(i,j) I(x+i, y+j) \tag{22}$$

Definition:-Convolution

$$I(x,y) = \sum_{j=-k}^{k} \sum_{i=-k}^{k} F(i,j) I(x-i, y-j) \tag{23}$$

$$I(x,y) = \sum_{j=-k}^{k} \sum_{i=-k}^{k} F(-i,-j) I(x+i, y+j) \tag{24}$$

Note:-From this formula we conclude that, if F (x, y) =F (-x, -y) then correlation = convolution
Convolution is an important operation in signal and image processing. Convolution operates on two signals in 1D or two images in 2D you can think of one as the input signal or image and the other called the kernel as a filter on the input image producing an output image so convolution takes two images as input and produces a third as output. Convolution is an incredibly important concept in many areas of Mathematics and Engineering including computer vision, as we'll see later. Definition: - Let's start with 1D convolution. A 1D image is also known as a signal

and can be represented by a regular 1D vector in Mat lab. Let's call our input vector f and our kernel g and say that f has length n, and g has length m. The convolution f * g of f and g is defined as:-

$$(f * g)(i) = \sum_{j=1}^{m} G(j)F(i - j + \frac{m}{2}) \tag{25}$$

One way to think of this operation is that we're sliding the kernel over the input image. For each position of the kernel we multiply the overlapping values of the kernel and image together and add up the results. This sum of products will be the value of the output image at the point in the input image where the kernel is centered. Let's look at a simple example. Suppose our input 1D image is:

F=

20	10	60	50	30	40

and the kernel is

g=

1/3	1/3	1/3

Let's call the output image h. What is the value of h (3)? To compute this we slide the kernel so that it is centered around f(3):

20	10	60	50	30	40
	1/3	1/3	1/3		

For now we'll assume that the value of the input image and kernel is 0 everywhere outside the boundary so we could rewrite this as:

20	10	60	50	30	40
0	1/3	1/3	1/3	0	0

We now multiply the corresponding lined-up values of f and g then add up the products. Most of these products will be 0, except for at the three non-zero entries of the kernel. So the product is

$$\tfrac{1}{3}10 + \tfrac{1}{3}60 + \tfrac{1}{3}50 = \tfrac{10}{3} + \tfrac{60}{3} + \tfrac{50}{3} = 40$$

Thus, h (3) = 40. From the above it should be clear that what this kernel is doing is computing a windowed average of the image i.e., replacing each entry with the average of that entry and its left and right neighbor. Using this intuition we can compute the other values of h as well.

h=	10	30	40	46.66	40	23.33

Remember that we're assuming all entries outside the image boundaries are 0 this is important because when we slide the kernel to the edges of the image the kernel will spill out over the image boundaries. We can also apply convolution in 2D our images and kernels are now 2D functions or matrices.
Example 2 let 2D image I is

$$\text{I}=\begin{bmatrix} 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \\ 10 & 10 & 20 & 20 & 20 \end{bmatrix}$$

and derivatives mask or kernel is

$$F_x = 1/3\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

$$\text{F}_y = 1/3\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

$$\text{I}_x = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 10 & 10 & 0 & 0 \\ 0 & 10 & 10 & 0 & 0 \\ 0 & 10 & 10 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \text{I}_y = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

In Mat lab Ill stick with intensity images for now and leave color for another time. For 2D convolution just as before we slide the kernel over each pixel of the image multiply the corresponding entries of the input image and kernel and add them up the result is the new value of the image.

CHAPTER FOUR

4 GOVERNING EQUATION AND DISCRITAION

4.1 Isotropic equation

If the conductivity K is constant through the material we can pull it out of the derivative
For one dimension

$$U_t = \frac{\partial}{\partial_x}(KU_x) \tag{26}$$

$$= K\frac{\partial}{\partial_x}(U_x) \tag{27}$$

$$U_{t=KU_{xx}} \tag{28}$$

And for two dimensions it is:-

$$U_t = \bigtriangledown(K \bigtriangledown U) \tag{29}$$

$$= K\lfloor \frac{\partial}{\partial x}U_x + \frac{\partial}{\partial x}U_y \rfloor \tag{30}$$

$$= K\lfloor U_{xx} + U_{yy} \rfloor \tag{31}$$

from Laplactian operator

$$\Delta U = U_{xx+U_{yy}} \tag{32}$$

the isotropic heat equation is:

$$U_t = K\Delta U \tag{33}$$

Since the heat flows constantly at all points through the material we refer it is isotropic diffusion. Isotropic heat diffusion looks just like blurring the image with Gaussian filter. In fact Gaussian blurred image is a solution of heat equation.

$$U(x, y, t) = G_t * F(x, y) \tag{34}$$

Blurring is a nice way to smooth the image and get rid of noise. But we dont want to blur the edge of the image.

4.2 Discritaization

4.2.1 Spatial discritaization

From finite difference method we have:-

$$U_x = \frac{U(x+1,y) - U(x-1,y)}{2} \tag{35}$$

$$U_y = \frac{U(x,y+1) - U(x,y-1)}{2} \tag{36}$$

$$U_{xx} = U(x+1) - 2U(x,y) + U(x-1,y) \tag{37}$$

$$U_{yy} = U(x,y+1) - 2U(x,y) + U(x,y-1) \tag{38}$$

So the idea to approximate in the right side of our PDE is:-

$$U_t = K[U_{xx} + U_{yy}] \tag{39}$$

4.2.2 Discritaization of PDE

Our heat equation (PDE)

$$U_t = K[U_{xx} + U_{yy}] \tag{40}$$

is descretized as:-

$$\frac{U^{n+1} - U^n}{\Delta t} = K[U^n_{xx} + U^n_{yy}] \tag{41}$$

Where the spatial derivatives are approximated as

$$U^n_{xx} = U^n(x+1,y) - 2U^n(x,y) + U(x-1,y) \tag{42}$$

$$U^n_{yy} = U^n(x,y+1) - 2U^n(x,y) + U(x,y-1) \tag{43}$$

4.3 Comparisons of image processing between linear filtering and Gaussian filtering

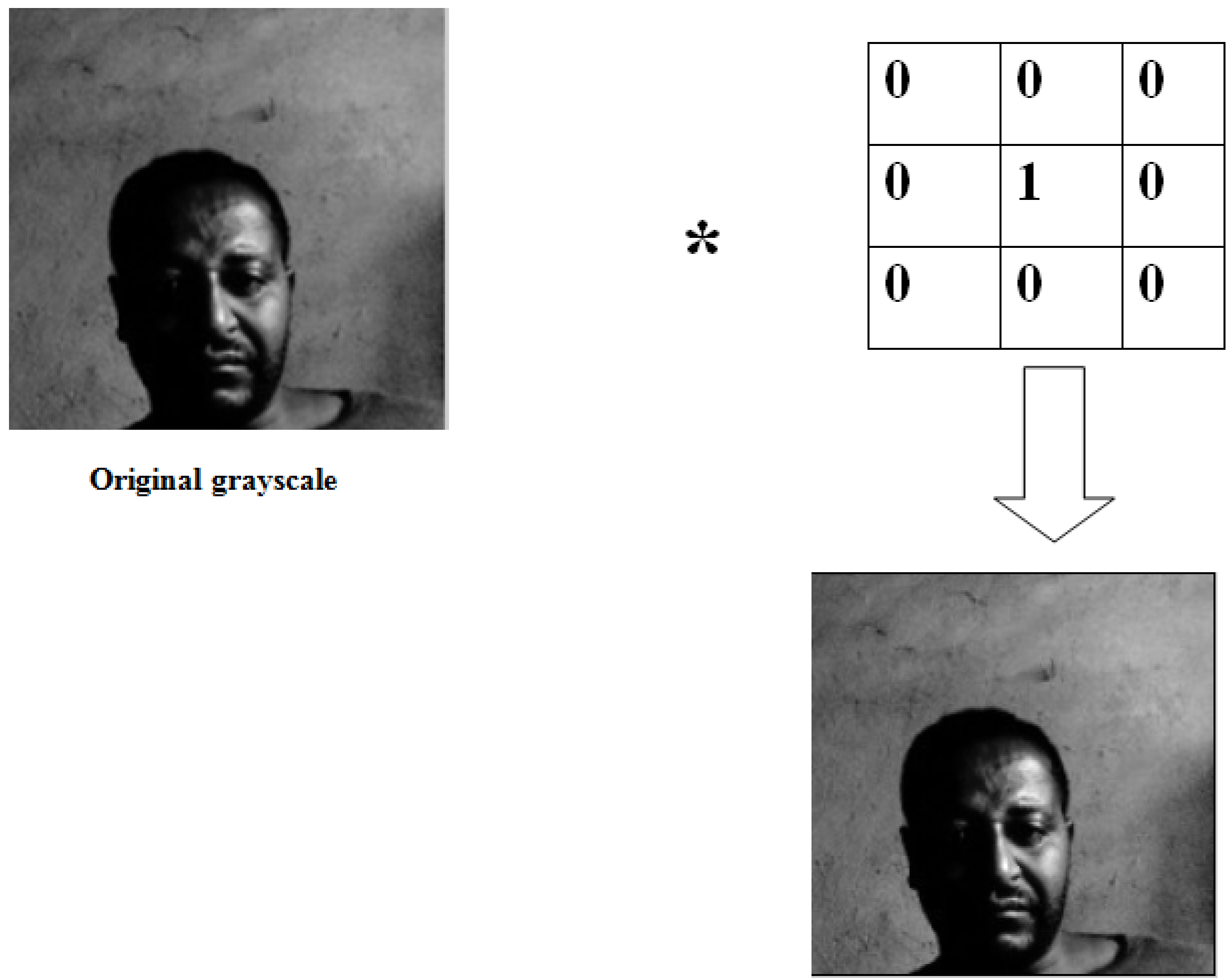
4.3.1 Liner filtering

Solution of the heat equation is a convolution

$$U(x,t) = \{^{f(x)t=0}_{(G_{\sqrt{2t}}*f)(x)t>0} \tag{44}$$

Let's see the result of convolution of an image with some example kernels. We'll use the original image as our input.

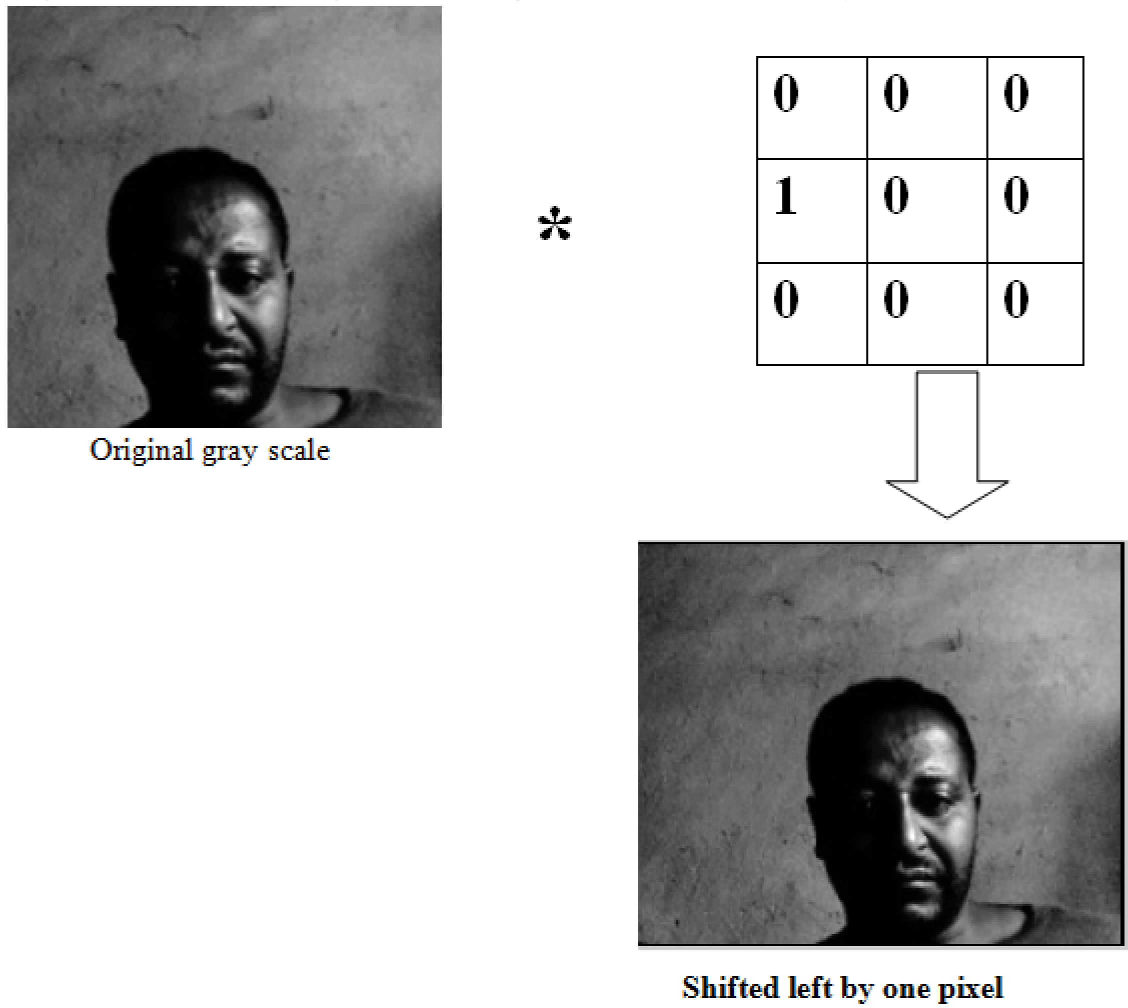
Figure 2: convolution of original gray scale image with identity kernel



One very simple kernel is just a single pixel with a value of 1. This is the identity kernel convolution of an image with identity kernel doesn't change the image and leaves the original image as it is

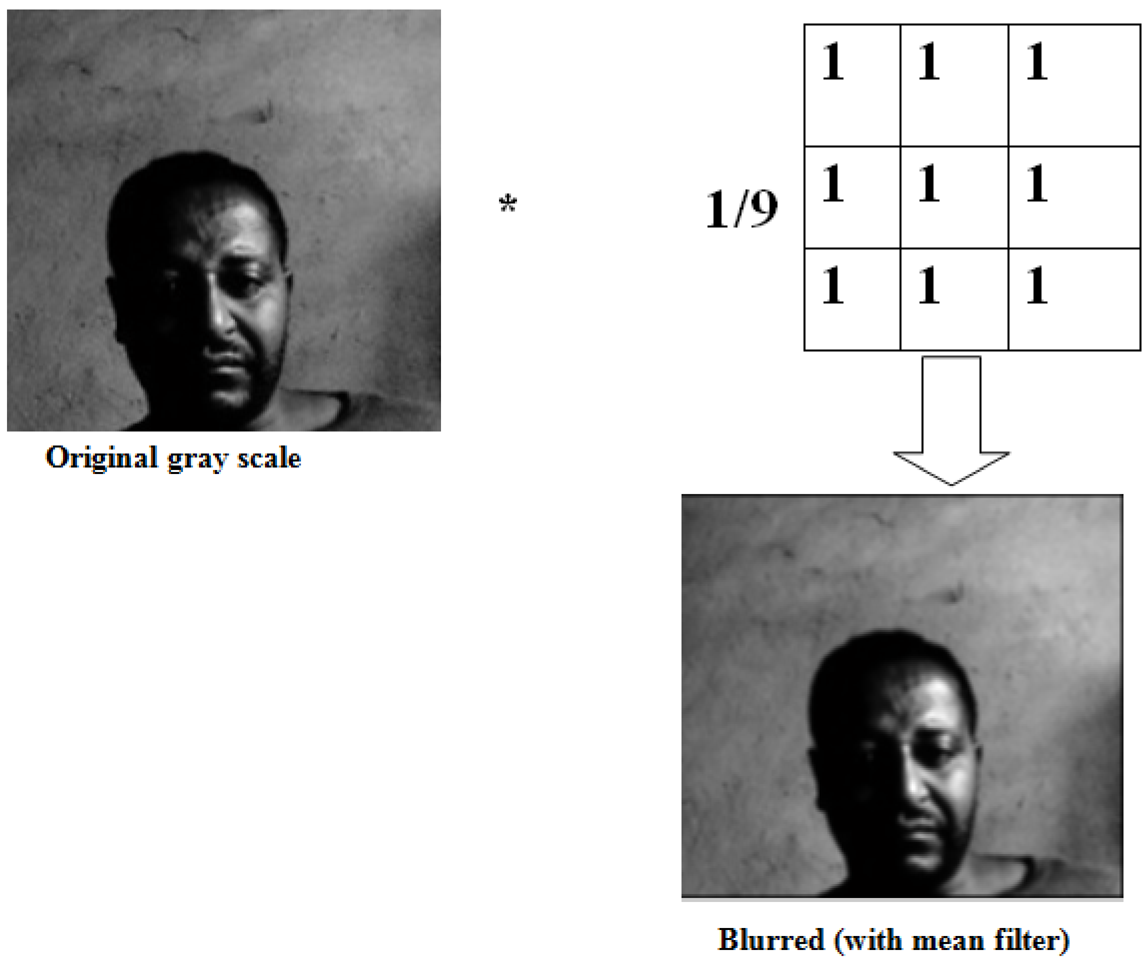
What happens if we convolution the image with an identity kernel but where the one is shifted by one pixel?

Figure 3: convolution of gray scale image with left shifted identity kernel



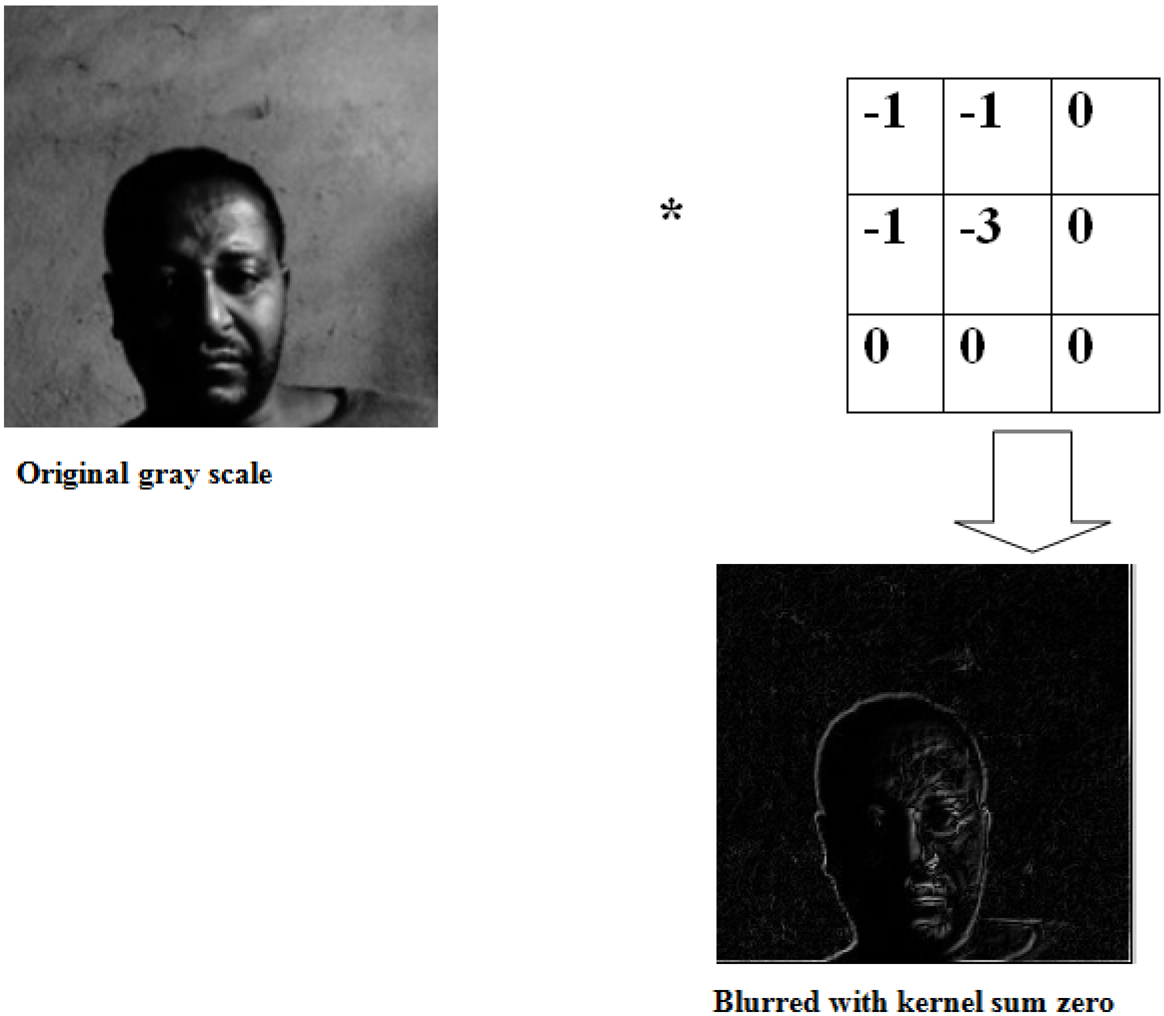
convolution of an image with identity kernel that shift by one pixel to the left again doesn't change the the actual size and behavior of the original image but the output image is shifted by one pixel to the left

Figure 4: convolution of gray scale image with mean filter



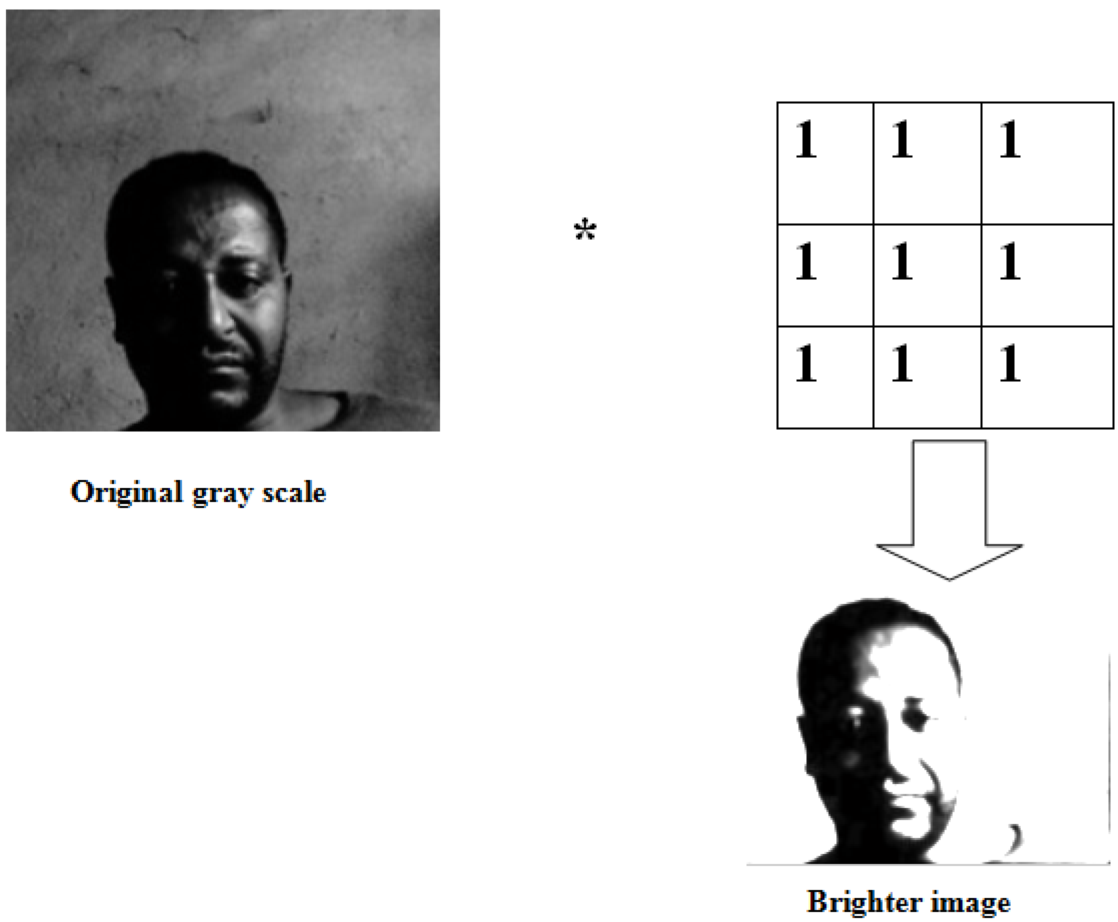
Another useful 2D kernel is an averaging or mean filter. Here's what convolving the image with a 3 x 3 mean filter looks like. In this case the mean kernel is just a 3 x 3 image where every entry is 1/9 because it is a good thing for the entries to sum to one. This kernel has the effect of blurring the images and image where the intensities are smoothed out.

Figure 5: convolution of image with kernel sum zero



convolution of an image with kernel sum zero is black.This is because the intensity of gray scale image ranges between 0 and 255, 0 is for black 255 is for white and the value between 0 and 255 is gray

Figure 6: convolution of image with kernel sum different from zero



Convolution of an image with kernel sum different from 0 and 1 is brighter image. fig:6

4.3.2 Gaussian filtering

Let $U_0$is an image we define

$$U_x = (G_\sigma * U_0)(x) = avecG_\sigma(x) = \frac{1}{2\pi\sigma^2} \exp \frac{-|x|}{2\sigma^2} \qquad (45)$$

Figure 7: Gaussian filtered image

Original Grayscale image   Gaussian filter at $\sigma = 1$

Figure 8: Relation between linear filter on identity kernel and Gaussian filter at parameter=1

Linear filter on identity kernel **Gaussian filter at $\sigma = 1$**

From these two experiments we conclude that linear filters and Gaussian filters have equivalent values on image processing. So it is possible to use one of the methods for further image processing

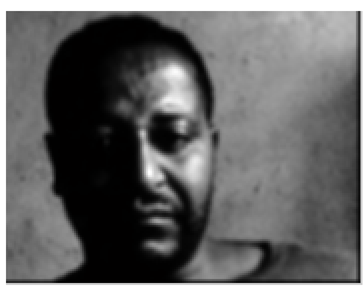
4.4 Image smoothing by different parameters

Figure 9: Image smoothed by different parameters

$\boldsymbol{\sigma = 1}$

$\boldsymbol{\sigma = 5}$

$\boldsymbol{\sigma = 10}$



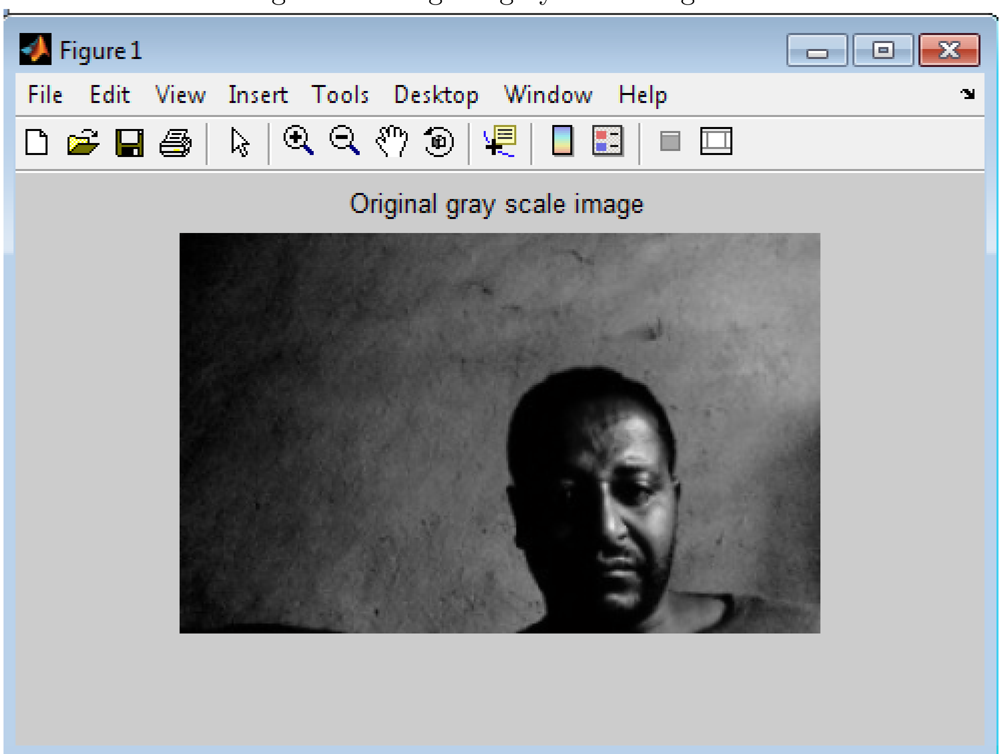
$\boldsymbol{\sigma = 25}$

$\boldsymbol{\sigma = 1oo}$

From this figures we see that if we increase the value of σ the quality of the image is more and more enhance

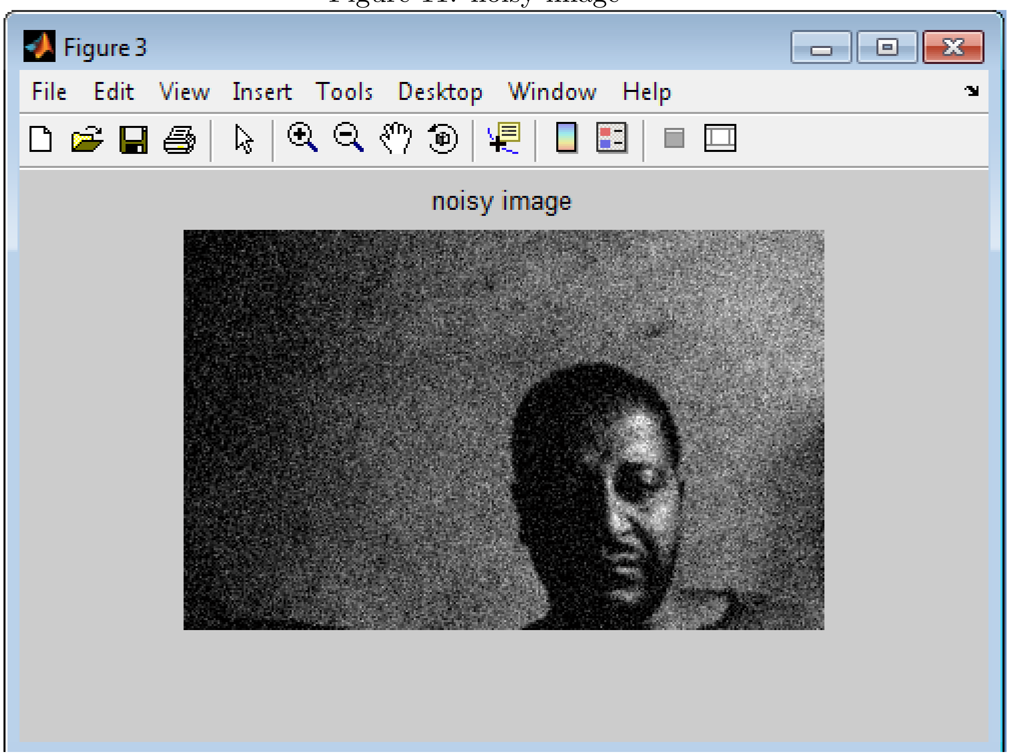
4.5 De-noising of image by removing noise

Figure 10: Original gray scale image



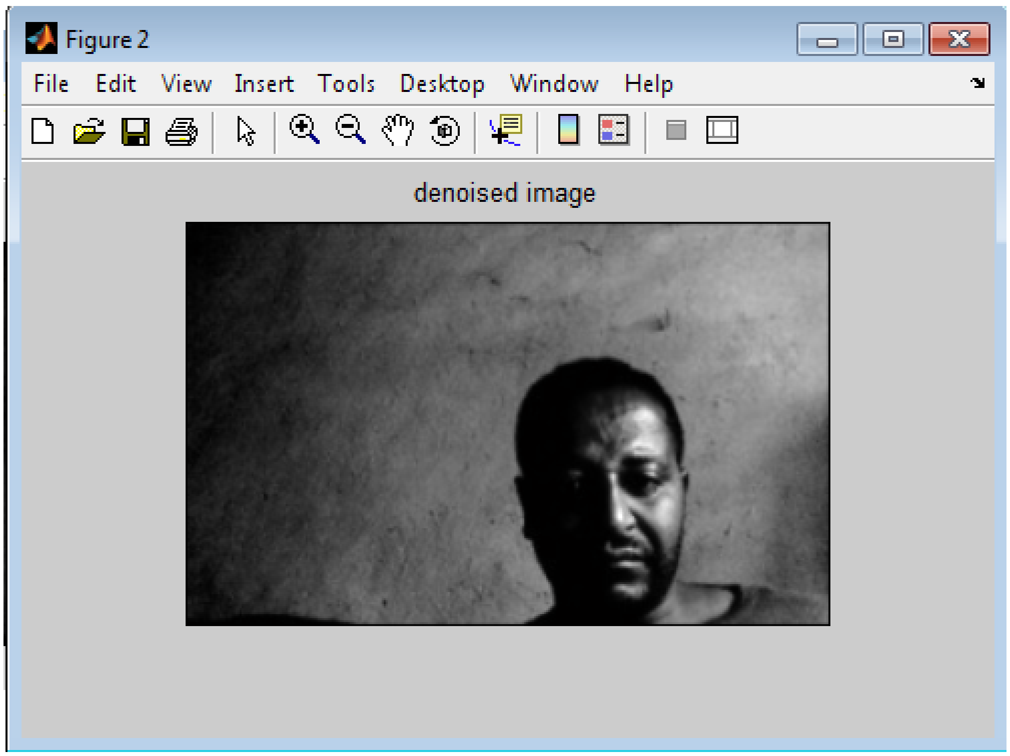
Original gray scale image taken by camera fig:10

Figure 11: noisy image



Every image we take by camera has its own noise.This is do to camera man,Camera lenses,light variation, condition and so on. this nose is shown on fig:11

Figure 12: de-noised image



An image is said to be de-noised if we remove the noise of blured image fig:12

4.6 The side effect of heat equation on an Image

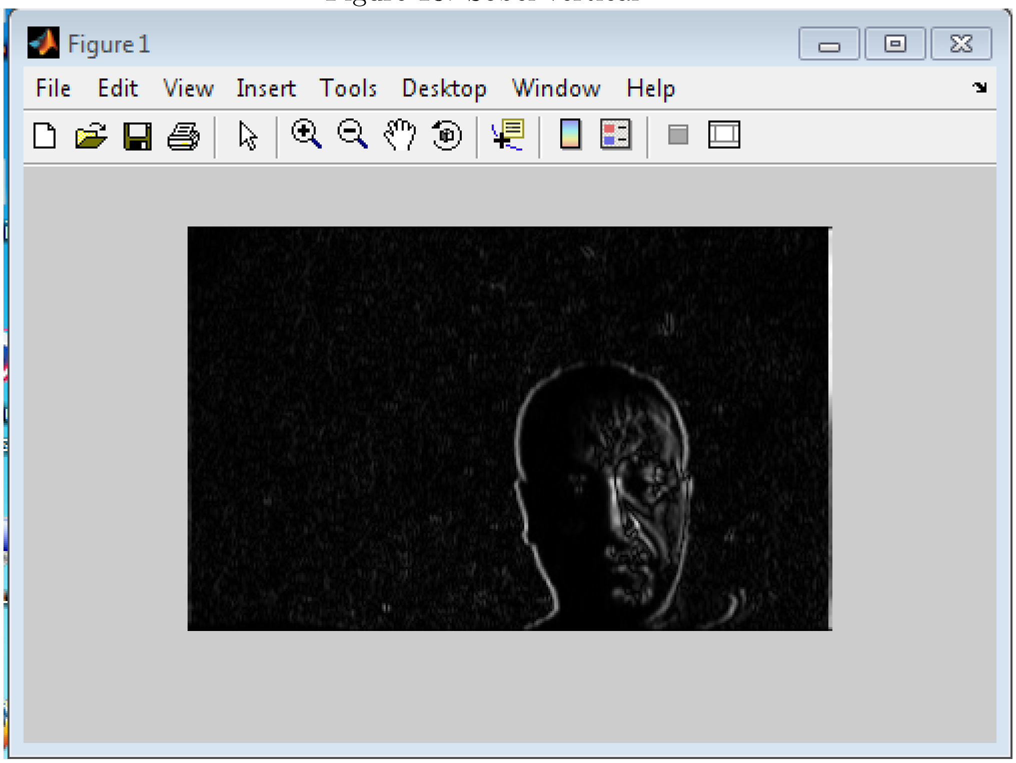
Even though both of these isotropic diffusion and Gaussian convolution methods remove the noise in the image they have a side effect that sharp edges are lost. Many applications in engineering and science require the correct identification of edges. For instance many tools for automated manufacturing processes are equipped with cameras to detect markers (such as a thick black line) that might designate special reference points or physical locations.

An edge-detector can process and extract relevant features in a set of images before they are fed into a pattern recognition algorithm which can result in superior performance. The following are two such kernels for detecting horizontal and vertical edges together called the Sobel operator. By inspection we notice that the kernels are vertically and horizontally symmetric and that the sum of the kernel elements is zero. This means that as the kernel passes over the image vertically or horizontally aligned pixels that differ in intensity value from their neighbors will be multiplied with the non-zero parts of the kernel which results in a non-zero pixel in the result. Since the sum of the kernel elements is zero as soon as the kernel enters a region with uniform pixel values the sum of the element-wise products becomes close to zero. Since we are using UTF-8 which represents black as 0 edges which are nonzero should appear white and non-edges close to black.

$$K_V = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} K_H = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

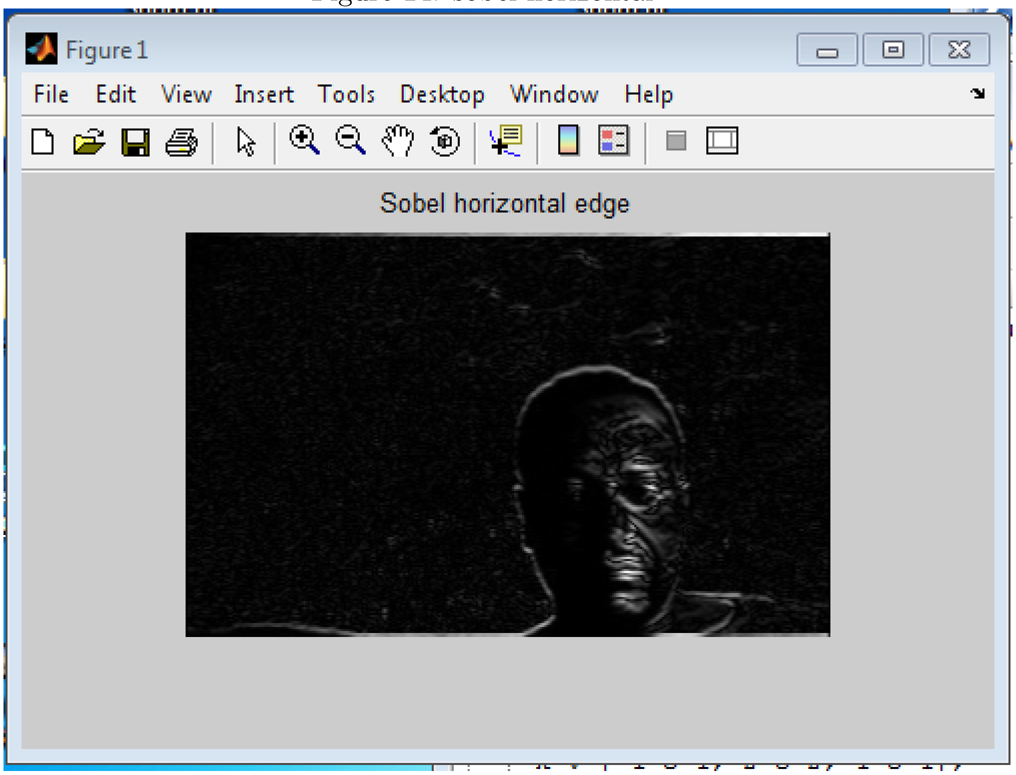
The convolution of original gray scale image with this two kernel is as follows

Figure 13: Sobel vertical



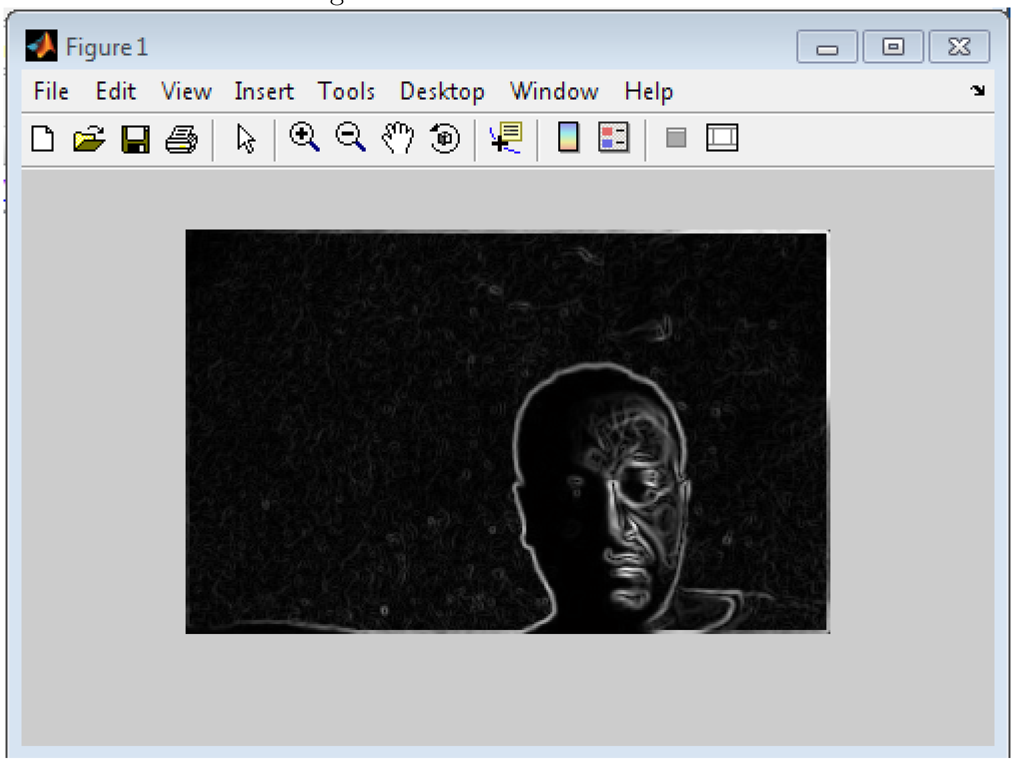
convolution of image with vertical sobel kernel preserve vertical edges of the image fig:13

Figure 14: sobel horizontal



Convolution of an image with Sobel horizontal kernels preserve horizontal edges of the image fig:14

Figure 15: Sobel normalized



Sobel normalized is the combination of sobel vertical and sobel horizontal.Its main purpose is to preserve both horizontal and vertical edges of the image at the same time fig:15

CHAPTER FIVE

5 DISCUSSION AND RESULTS

Heat equation is the best way of image processing in the world. In the past 20 years because of human have high interest on image restoration and enhancement, doing image processing by heat equation is an interesting project. Now a day there are a number of methods on image processing but compared to other methods using heat equation on image processing have some advantages on calculation. These are
•The process is important on geometric characters directly which affect visual effect
• The diffusion method can do simulation effectively on linear and non linear diffusion
• It is fast and easy to implement
• It replaces pixel by the averaging of the neighborhoods and so on

So in this study I want to explain some solution of heat equation problem for noise removal. In case I use isotropic (linear filtering) and Gaussian convolution. Linear filtering is a method for finding a new image whose pixels are a weighted sum of original pixel values. It is similar to that of Smoothing with a box filter and Smoothing with a Gaussian kernel. The overall property of this method is shift invariant function of the input. Means we use the same procedure at each pixel or image location. These approaches reduce noise in gray scale image. However image structures such as edge and texture are quickly blurred do to the fact that these are isotropic approaches. For this reason I use Sobel edge detector to preserve edges from blurring. For the future other mathematical solutions such as anisotropic diffusion that preserve edges and other methods will be needed

CHAPTER SIX

6 RECOMMENDATION

This thesis shows the applications of heat equations for image processing in detail. The motivation of this thesis is two-folds. Firstly image smoothing is an important preprocessing technique in image processing area and it is worthy to do research for it. Secondly heat equations are relatively new techniques which are superior to traditional methods and at the end of this thesis a comprehensive discussion is presented. I believe that this thesis can contribute to the researchers who are interested to apply heat equation (PDE) for image processing

7 BIBILOGRAPHY

Caselles (1998).Introduction to the special issues on partial differential equations and geometric driven diffusion in image processing and analysis

Chan and Shcen (2001). Mathematical models for local non texture in paintings.

Dr.Tim Morris (27112).Image processing with Matlab supporting material for comp

Foyer and Zou (2006).Global approach for solving evolutive heat transfer for image denoisng and in painting.

Gabor and Jian (1965). Information theory in electros copy Lab.

Humme (1989).Reconstructions from zero crossing in scale space.

INISTA (2009). based denosing of gray scale image using fuzzy sets

Joachim Weickert Journal,(E.01.334). Non linear diffusion filtering.

Joachim Weickert (1998).an isotropic diffusion in image processing

.

Koenderind and Witikin (1983).Scale space filtering.

Muford and Shah (1989).Optimal approximations by piecewise smooth function and variation problems

Osher and Rudin (1990).Feature oriented image enhancement using shock filters.

Perona and malik (1983).Scale space and edge detection using isotropic diffusion.

P.A.Bromiley last updated (2014). Products and convolution Gaussian probability density function.

P.Markowich N.Matevosyan. (2005). Non linear diffusion equation in image processing.

R.C Gonzalez,R.E.woods,S.L.Eddins,(2004). Digital image processing using Matlab

Slovak University of technology Journal,(81369). Image processing with partial differential equation.Department of Mathematics and discriptive Geometry

Sung.KIM august (20,2013). Application of convolution in image processing with Matlab

Yanibian University China (2012).Application of partial differential equation for image inhancement. department of Mathematics

.

APPENDICES

Appendix A: Matlab code used to smooth original gray image with identity matrix

```
f=im2double (imread('zoof.jpg'));
y=rgb2gray (f);
Figure (1);
Imshow(y);
h= [0 0 0:0 1 0:0 0 0];
d=conv2(y, h);
Figure (2);
imshow(d);
```

Appendix B: Matlab code used to smooth original gray image with Gaussian kernel

```
Original=imread('zoof.jpg');
grayscale=rgb2gray(original);
figure(1);
imshow(grayscale);
noisyimage=imnoise(grayscale,('gaussian'));
figure(3);
imshow(noisyimage);
h=fspecial('gaussian',[3 3], .5);
m=conv2 (double (grayscale), double (h));
figure(2);
```

imshow((m.^{2}).^{0}.5, []);

Apendix C : $Matlabcodeshowsedgedetection$

$grayscale = rgb2gray(orig);$

$figure(1);$

$imshow(grayscale);$

$k_v = [-101; -202; -101];$

$k_h = [121; 000; -1-2-1];$

$m1 = conv2(double(grayscale), double(k_v));$

$m2 = conv2(double(grayscale), double(k_h));$

$figure(2);$

$imshow(abs(m1), []);$

$figure(3);$

$imshow(abs(m2), []);$

$figure(4);$

$imshow((m1.^2 + m2.^2).^0.5, []);$