International Journal of Scientific Engineering and Applied Science (IJSEAS) – Volume-2, Issue-1, January 2016
ISSN: 2395-3470
www.ijseas.com

IMPLEMENTATION OF IMAGE SHARPENING AND SMOOTHING USING FILTERS

K. Siva Praveen[1], G.Hamarnath[2], K. Prasad Babu[3], M. Sreenivasulu[4], K.Sudhakar[5]

[1] M.tech Student 13G31D0604, DSCE branch, SJCET Yemmiganur, Andhra pradesh, India
[2] Assistant Professor, Department of ECE, SJCET Yemmiganur, Andhra pradesh, India
[2] Assistant Professor, Department of ECE, SJCET Yemmiganur, Andhra pradesh, India
[4] Associate Professor, Department of ECE, SJCET Yemmiganur, Andhra pradesh, India
[5] H.O.D & Associate Professor, Department of ECE, SJCET Yemmiganur, Andhra pradesh, India

Abstract
In this project implementation of image sharpening and smoothing on image is done by using filters. The objective of image filtering is to process the image so that the result is more suitable than the original image for a specific application. Image filtering refers to a process that removes the noise, improves the digital image for varied application. A gradient filter highlights diagonal edges. A gradient convolution filter is a first-order derivative filter with kernel values. The input image is sharpened by using weighted kernel of different values. By using matlab we implement the smoothening of an image with Gaussian noise, salt and pepper noise with the median filter. Mean square error and peak signal to noise ratio are compared for the input image and obtained output image.

Keywords: *Sharpening, Smoothening, Filters, MSE, PSNR, Matlab.*

1. Introduction

A digital image is an image f(x,y) that has been discretized both in spatial coordinates and brightness. The elements of such a digital array are called image elements or pixels. A simple image model: To be suitable for computer processing, an image f(x,y) must be digitalized both spatially and in amplitude. Digitization of the spatial coordinates (x,y) is called image sampling. Amplitude digitization is called gray-level quantization.

Fundamental steps in image processing:
1. Image acquisition: to acquire a digital image
2. Image preprocessing: to improve the image in ways that increase the chances for success of the other processes.
3. Image segmentation: to partitions an input image into its constituent parts or objects.
4. Image representation: to convert the input data to a form suitable for computer processing.
5. Image description: to extract features that result in some quantitative information of interest or features that are basic for differentiating one class of objects from another.
6. Image recognition: to assign a label to an object based on the information provided by its descriptors.
7. Image interpretation: to assign meaning to an ensemble of recognized objects.

Knowledge about a problem domain is coded into an image processing system in the form of a knowledge database.

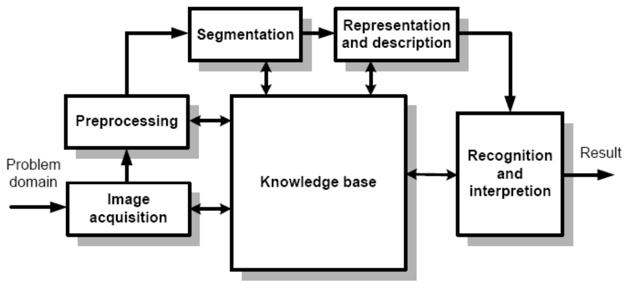


Figure 1: Fundamental steps in Digital image processing

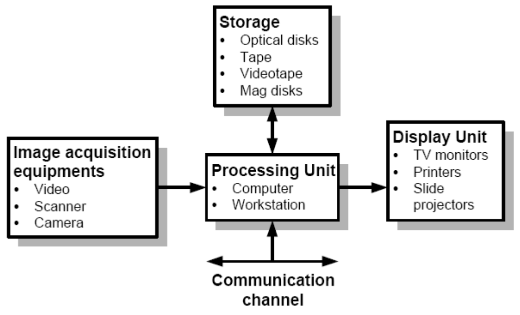


Figure 2: Basic Fundamental elements of an image processing

Define a center point (x; y) Perform an operation that involves only the pixels in a predefined neighborhood Result of the operation response of the process at that point Repeat the process for every pixel in the image.

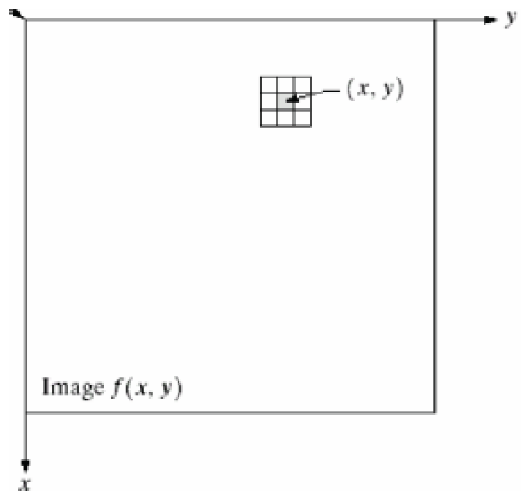
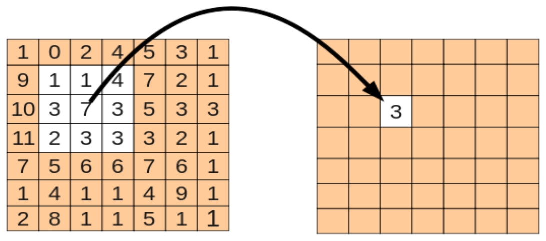


Figure 3: Basic representation of an image in terms of pixels

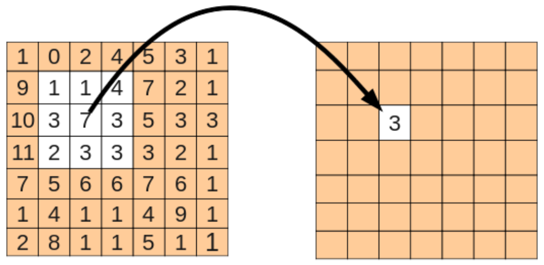
Output is a function of a pixel value and its neighbors Possible operations are: sum, weighted sum, average, weighted average, min, max, median,
Example: 3x 3 neighbourhood

$$OPERATION\left(\begin{bmatrix} f(x-1,y-1) & f(x-1,y) & f(x-1,y+1) \\ f(x,y-1) & f(x,y) & f(x,y+1) \\ f(x+1,y-1) & f(x+1,y) & f(x+1,y+1) \end{bmatrix}\right)$$

Example 1



Example 2



The most common neighbourhood operation is to multiply each of the pixels in the neighbourhood by a weight and add them together. The local weights are sometimes called a mask or kernel

f(x-1,y-1)	f(x-1,y)	f(x-1,y+1)
f(x,y-1)	f(x,y)	f(x,y+1)
f(x+1,y-1)	f(x+1,y)	f(x+1,y+1)

W=

w(-1,-1)	w(-1,0)	w(-1,1)
w(0,-1)	w(0,0)	w(0,1)
w(1,-1)	w(1,0)	w(1,1)

$$g(x,y)=\sum_{j=-k}^{k}\sum_{i=-k}^{k} w(i,j)\cdot f(x+i,y+j)$$

Convolution - the same as correlation except that filter kernel W is rotated for 180 degrees

$$g(x,y)=\sum_{j=-k}^{k}\sum_{i=-k}^{k} w(i,j)\cdot f(x-i,y-j)$$

Convolution (*) g = W* f
For spatial filtering using correlation or convolution is a matter of preferences

Pixels in the neighbourhood are correlated
Rectangular neighbourhoods are often used with an odd number of pixels in rows and columns, enabling specification of the central pixel of the neighbourhood
Different neighbourhood sizes: 3_3, 5_5, 1_3,...,2n+1_2m+1
The choice of size and shape (rectangular, circular,...) of the neighbourhood depends on the size of the objects in the image

2. Smoothening and Sharpening Filters

Smoothing filters are those which remove fine detail in an image. Whereas the Sharpening spatial filters seek to highlight fine detail, Remove blurring from images, Highlight edges
Sharpening filters are based on spatial differentiation
Differentiation measures the rate of change of a function
.
Let's consider a simple 1 dimensional example

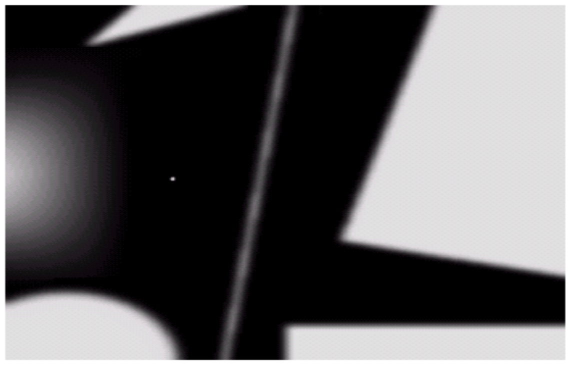


Figure 4: One dimensional figure

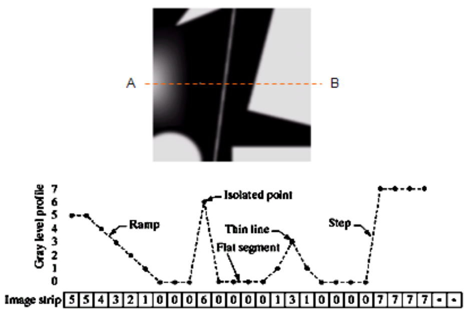


Figure 5: Selection of for derivative

The formula for the 1st derivative of a function is as follows:

$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

It's just the difference between subsequent values and measures the rate of change of the function

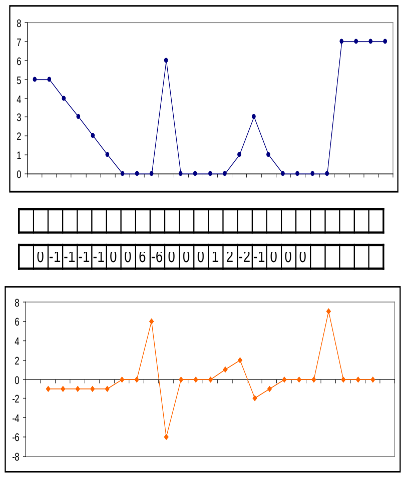
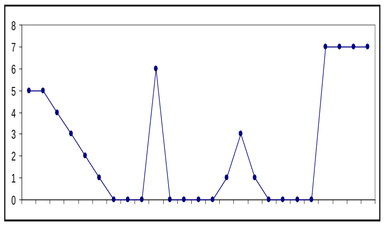


Figure6: 1st derivate

The formula for the 2nd derivative of a function is as follows:

$$\frac{\partial^2 f}{\partial^2 x} = f(x+1) + f(x-1) - 2f(x)$$

Simply takes into account the values both before and after the current value



	0	-1	-1	-1	-1	0	0	6	-6	0	0	0	1	2	-2	-1	0	0	0					

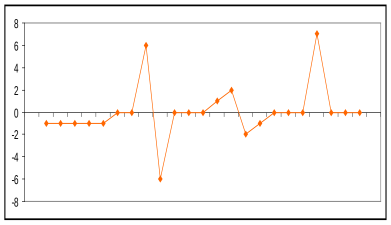


Figure7: 2nd derivate

The 2nd derivative is more useful for image enhancement than the 1st derivative

Stronger response to fine detail

Simpler implementation

We will come back to the 1st order derivative later on

The first sharpening filter we will look at is the Laplacian Isotropic One of the simplest sharpening filters We will look at a digital implementation

The Laplacian is defined as follows:

$$\nabla^2 f = \frac{\partial^2 f}{\partial^2 x} + \frac{\partial^2 f}{\partial^2 y}$$

where the partial 1st order derivative in the x direction is defined as follows:

$$\frac{\partial^2 f}{\partial^2 x} = f(x+1,y) + f(x-1,y) - 2f(x,y)$$

and in the y direction as follows:

$$\frac{\partial^2 f}{\partial^2 y} = f(x,y+1) + f(x,y-1) - 2f(x,y)$$

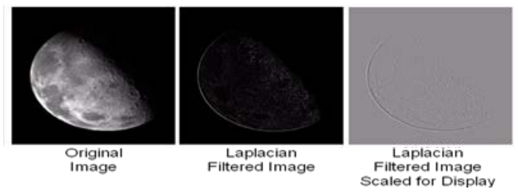
So, the Laplacian can be given as follows:

$$\nabla^2 f = [f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1)] - 4f(x,y)$$

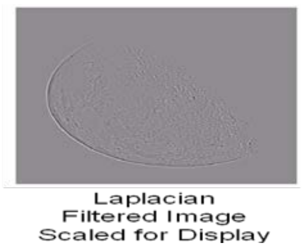
We can easily build a filter based on this

0	1	0
1	-4	1
0	1	0

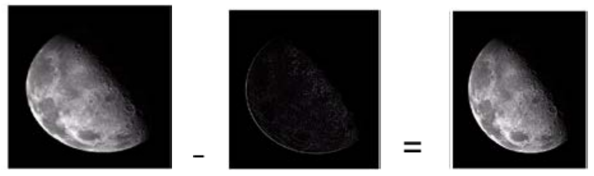
Applying the Laplacian to an image we get a new image that highlights edges and other discontinuities



The result of a Laplacian filtering is not an enhanced image. We have to do more work in order to get our final image Subtract the Laplacian result from the original image to generate our final sharpened enhanced image $g(x,y) = f(x,y) - \nabla^2 f$



In the final sharpened image edges and fine detail are much more obvious



Original - Laplacian = Sharpened image

The key point in the effective sharpening process lies in the choice of the high-pass filtering operation. Traditionally, linear filters have been used to implement the high-pass filter, however, linear techniques can lead to unacceptable results if the original image is corrupted with noise. A tradeoff between noise attenuation and edge highlighting can be obtained if a weighted median filter with appropriated weights is used. To illustrate this, consider a WM filter applied to a gray-scale image where the following filter mask is used.

Human perception is highly sensitive to edges and fine details of an image, and since they are composed primarily by high frequency components, the visual quality of an image can be enormously degraded if the high frequencies are attenuated or completed removed. In contrast, enhancing the high-frequency components of an image leads to an improvement in the visual quality. Image sharpening refers to any enhancement technique that highlights edges and fine details in an image. Image sharpening is widely used in printing and photographic industries for increasing the local contrast and sharpening the images.

In principle, image sharpening consists of adding to the original image a signal that is proportional to a high-pass filtered version of the original image. Figure (8) illustrates this procedure, often referred to an unsharp masking on a one-dimensional signal. As shown in Fig (8), the original image is first filtered by a high-pass filter that extracts the high-frequency components, and then a scaled version of the high-pass filter output is added to the original image, thus producing a sharpened image of the original. Note that the homogeneous regions of the signal, i.e., where the signal is constant, remain unchanged. The sharpening operation can be represented by

$$S_{i,j} = x_{i,j} + \lambda F(x_{i,j}) \quad \text{equation}\dashrightarrow 1$$

where $x_{i,j}$ is the original pixel value at the coordinate (i,j), $F(.)$ is the high-pass filter, λ is a tuning parameter greater that or equal zero, and $S_{i,j}$ is the sharpened pixel at the coordinate (i,j). The value taken by λ depends on the grade of sharpness desired. Increasing λ yields a more sharpened image.

If color images are used $x_{i,j}, S_{i,j},$ and λ are three-component vectors, whereas if gray-scale images are used $x_{i,j}, S_{i,j},$ and are single-component vectors. Thus the process described here can be applied to either gray-scale or color images, with the only difference being that vector filters have to be used in sharpening color images whereas single-component filters are used with gray-scale images.

$$W = \frac{1}{3}\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

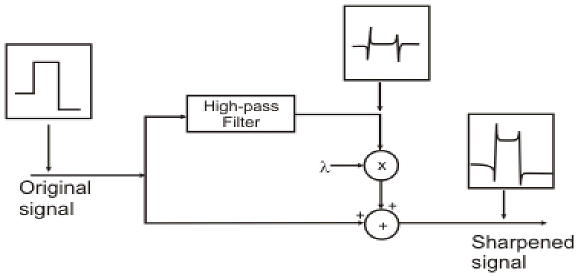


Figure 8: Image sharpening block diagram representation

Because of the weight coefficients in Eq. 1, for each position of the moving window, the output is proportional to the difference between the center pixel and the smallest pixel around the center pixel. Thus, the filter output takes relatively large values for prominent edges in an image, and small values in regions that are fairly smooth, being zero only in regions that have a constant gray level.

Although this filter can effectively extract the edges contained in an image, the effect that this filtering operation has over negative-slope is different from that obtained for positive-Slope edges .A change from a gray level to a lower gray level is referred to as a negative-slope edge, whereas a change from a gray level to a higher gray level is referred to as a positive-slop edge

Since the filer output is proportional to the difference between the center pixel and the small pixel around the center, for negative-slope edges, the center pixel small values producing small values at the filter output. Moreover, the filter output is zero if the smallest pixel around the center pixel and the center pixel have the same values.

This implies that negative-slope edges are not extracted in the same way as positive-slope edges. To overcome this limitation the basic image sharpening structure shown in Figure 8 must be modified such that positive-slope edges as well as negative-slope edges are highlighted in the same proportion.

A simple way to accomplish that is:

(a) extract the positive-slope edges, and then filter the preprocessed image with the filter described above; (c) combine appropriately the original image, the filtered version of the original image, and the filtered version of the preprocessed image to form the sharpened image.

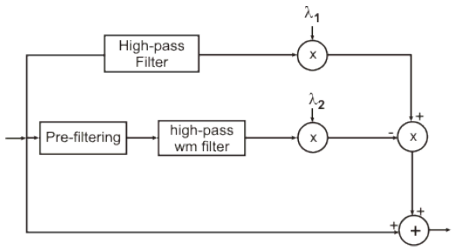


Figure 9: Block diagram representation

Thus both positive-slope edges and negative-slope are equally highlighted. This procedure is illustrated in Figure 9, where the top branch extracts the positive-slope edges and the middle branch extracts the negative-slope edges. In order to understand the effects of edge sharpening, a row of a test image is plotted in Figure 10 together with a row of the sharpened image when only the positive-slope edges are highlighted, Figure 10, only the negative-slope edges are highlighted, Figure 10, and both positive-slope and negative-slope edges are jointly highlighted, Figure 10

In Figure (9). λ_1 and λ_2 are tuning parameters that control the amount of sharpness desired in the positive-slope direction and in the negative-slope direction, respectively. The values of λ_1 and λ_2 are generally selected to be equal. The output of the prefiltering operation is defined as

$$x'_{i,j} = M - x_{i,j},$$ equation -----→2

With M equal to the maximum pixel value of the original image. This prefiltering operation can be thought of as a flipping and a shifting operation of the values of the original image such that the negative-slope edges are converted in positive-slope edges. Since the original image and the pre-filtered image are filtered by the same WM filter, the positive-slope edges and negative-slopes edges are sharpened in the same way.

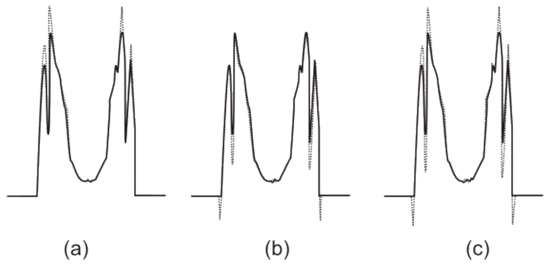
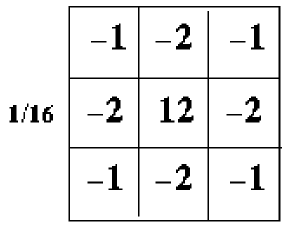


Figure 10

A powerful technique for sharpening images in the presence of low noise levels is via an adaptive filtering algorithm. Here we look at a method of re-defining a highpass filter (such as the one shown in Figure 8) as the sum of a collection of edge directional kernels.

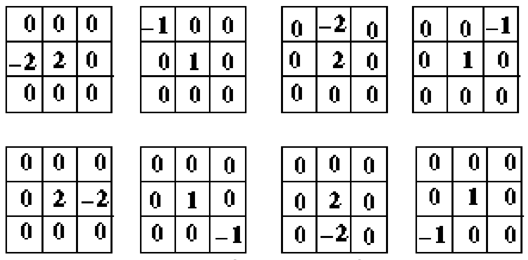


1/16

−1	−2	−1
−2	12	−2
−1	−2	−1

Figure 11: Sharpening filter.

This filter can be re-written as 1/16 times the sum of the eight edge sensitive kernels shown in Figure 9.



0	0	0
−2	2	0
0	0	0

−1	0	0
0	1	0
0	0	0

0	−2	0
0	2	0
0	0	0

0	0	−1
0	1	0
0	0	0

0	0	0
0	2	−2
0	0	0

0	0	0
0	1	0
0	0	−1

0	0	0
0	2	0
0	−2	0

0	0	0
0	1	0
−1	0	0

Figure 12: Sharpening filter re-defined as eight edge directional kernels

Successful image enhancement is typically not achieved using a single operation, rather we combine a range of techniques in order to achieve a final result

RESULTS

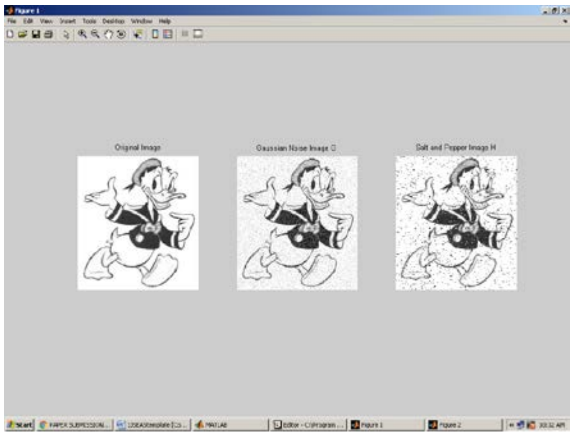


Figure 13: Original image , Gaussian image, salt and pepper noise image

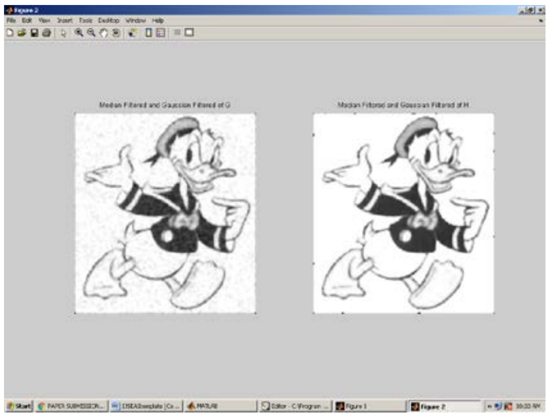


Figure 2
Median Filtered and Gaussian Filtered of G
Median Filtered and Gaussian Filtered of H

Figure 14: Median filtered image for Gaussian image, and salt and pepper noise image

Mean square Error of obtained image is mes = 335.6794

Peak Signal to Noise Ratio

PSNR = 52.6637

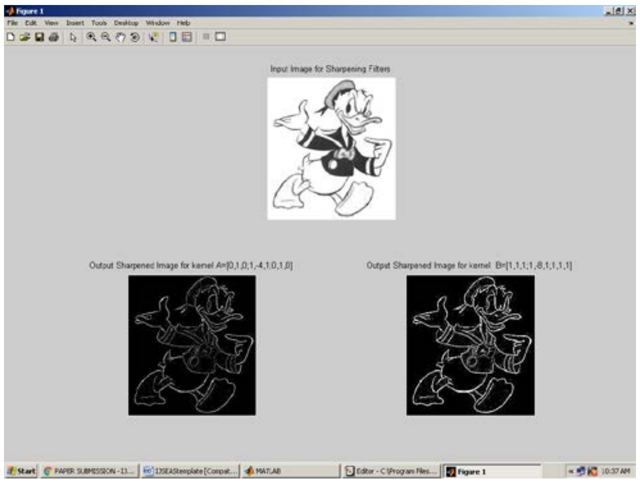


Figure 1
Input Image for Sharpening Filters
Output Sharpened Image for kernel A=[0,1,0;1,-4,1;0,1,0]
Output Sharpened Image for kernel B=[1,1,1;1,-8,1;1,1,1]

Figure 15: Original image , obtained sharpened images with two kernel values

Mean square Error of obtained image is

mes = 5.1584e+004

Peak Signal to Noise Ratio

PSNR = 2.3156

4. Conclusions

From the results we come to conclusion that weighted kernel will have give more sharpened image along with the boundary edge information. The median filtered output image is more smoothened for salt and pepper noise than compared to Gaussian noise.

References

[1] R. Kimmel A. Spira and N. Sochen, "A short time beltrami kernel for smoothing images and manifolds," IEEE Trans. Image Processing, vol. 16, no. 6, pp. 1628–1636, 2007.
[2]. Gonzales R., Woods R., and Eddins S., Digital Image Processing Using Matlab, 2nd Edition, Prentice Hall, USA, 2003.
[3] Geman, S., Geman, D., "Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images," IEEE Trans. Pattern Anal. Machine Intell. Vol.6 (6), 721-741, 1984.

[4] Buades, A., Coll, B., Morel, J, " A non-local algorithm for image denoising," IEEE CVPR, vol. II, pp. 60–65, 2005.
[5] Umbaugh Scot E, Computer Vision and Image Processing, Prentice Hall, NJ, 1988, ISBN 0-13-264599-8
[6]. R.C.Gonzales, R.E.Woods, Digital Image Processing, 2-nd Edition, Prentice Hall, 2002
[7]. R.A. Haddad and A.N. Akansu, "A Class of Fast Gaussian Binomial Filters for Speech and Image Processing," IEEE Transactions on Acoustics, Speech and Signal Processing, vol. 39, pp 723-727, March 1991.
[8]. Shapiro, L. G. & Stockman, G. C: "Computer Vision", page 137, 150. Prentence Hall, 2001
[9]. Mark S. Nixon and Alberto S. Aguado. Feature Extraction and Image Processing. Academic Press, 2008, p. 88.
[10]. Yusuf, Nijad, Sara Tedmory "Exploiting hybrid methods for enhancing digital X-ray images". The International Arab Journal of Information Technology, Vol. 10, No.1, January 2013