Survey of Various Image Enhancement Techniques in Spatial Domain Using MATLAB

Shailendra Singh Negi
M.Tech Scholar
G. B. Pant Engineering College, Pauri Garhwal
Uttarakhand, India- 246194

Bhumika Gupta
Assistant Professor
G. B. Pant Engineering College, Pauri Garhwal
Uttarakhand, India- 246194

ABSTRACT
Image Enhancement is one of the first steps in Image processing. In this technique, the original image is processed so that the resultant image is more suitable than the original for specific applications i.e. the image is enhanced. Image enhancement is a purely subjective processing technique. An image enhancement technique used to process images might be excellent for a person but the same result might not be good enough for another. Image enhancement is a cosmetic procedure i.e. it does not add any extra information to the original image. It merely improves the subjective quality of the images by working with the existing data. Image enhancement can be done in following two domains: The Spatial domain and The Frequency domain. This paper focuses on spatial domain techniques for image enhancement, with particular reference to point processing methods, histogram processing and spatial filtering.

Keywords
Image enhancement, spatial domain, Point processing, Histogram processing, Neighborhood processing, Filters, Smoothing filters, and Sharpening filters.

1. INTRODUCTION
Image enhancement [8] is used in increasing the interpretability or perception of information in images for humans and providing 'enhanced' input for other image processing applications. Digital image enhancement techniques provide many choices for improving and enhancing the visual quality of images. Appropriate choice of such techniques is greatly influenced by the imaging conditions, task to do and viewing conditions. There exist many techniques that can enhance and improve a digital image without spoiling it. The enhancement techniques can be divided into the following categories and subcategories.

1.1 Spatial domain methods
The term spatial domain [2] means working in the given space i.e. the image. It implies working with the pixel values or in other words, working directly with the raw data. The pixel values are altered to achieve desired enhancement. Image enhancement [8] is applied in every field of science where images are understood and analyzed. For example, medical image analysis, satellite image analysis etc. Let $f(x, y)$ be the original image where f is the grey level value or intensity

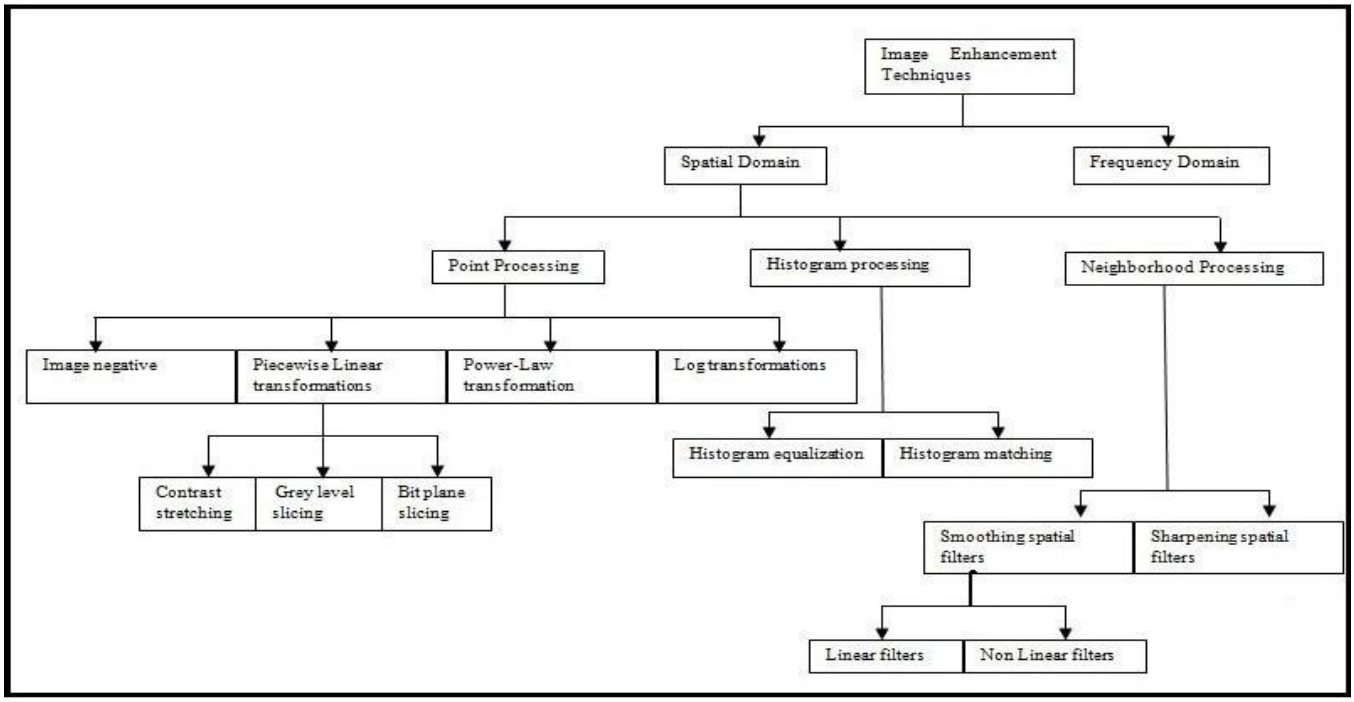

Fig.1 Classification of Image enhancement techniques.

value and (x, y) are the image co-ordinates. For an 8-bit image, 'f' can take values from 0-255, where 0 represents black, 255 represents white and all the intermediate values represent shades of gray. Image enhancement simply means, transforming an image 'f' into image 'g' using 'T'. The modified image can be expressed as:

$$g(x, y) = T[f(x, y)] \quad \text{..... (1).}$$

For all spatial domain [2] techniques it is simply T that changes. The above equation can also be written as:

$$s = T(r) \quad \text{..... (2).}$$

Where 'T' is the transformation that maps a pixel value 'r' into a pixel value 's'. The results of this transformation [8] are mapped back into the gray scale range as we are here dealing only with grey scale digital images. So, the results are mapped back into the range [0, L-1], where $L=2^k$, k being the number of bits in the image being operated on. Here we will only consider gray level images. The same methodology can be extended for the color images too. A digital gray image has pixel values in the range of 0 to 255. In this survey paper basic image enhancement techniques have been discussed with their mathematical application and understanding. This survey paper will provide an understanding of underlying concepts, along with algorithms and mathematical concepts commonly used for image enhancement in spatial domain.

2. POINT PROCESSING

In point processing [2], we work with single pixels i.e. 'T' is 1×1 operator. It means that the new value $g(x, y)$ depends on the operator 'T' and the present $f(x, y)$. Point processing operations take the form of equation (1) and equation (2). Figure below shows basic grey level transformation curves.

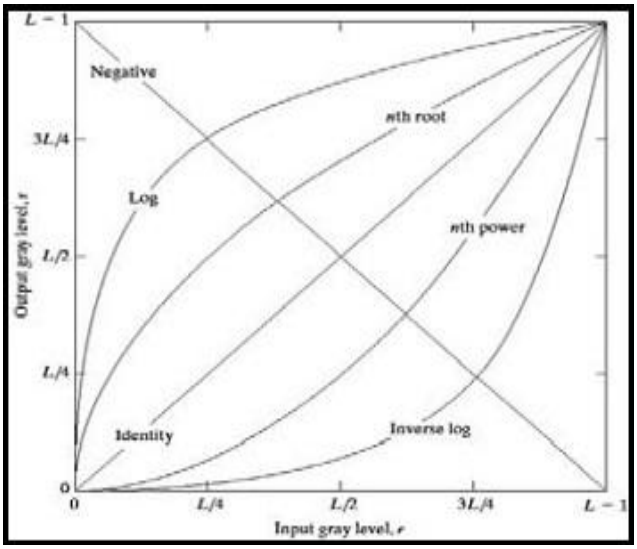

Fig.2: Basic grey level transformation curves

2.1 Digital Negative

Digital Negatives [13] are useful in many applications. A common example of digital negative is the displaying of an X-ray image. The pixel gray values are inverted to compute and find the negative of an image i.e. black in the original image is converted into white and vice versa. Digital Negative images are particularly useful for enhancing white detail embedded in dark regions of an image. The negative of gray image can be obtained by using a simple transformation given by:

$$s = 255 - r \quad \text{..... (3).}$$

Hence when r =0, s =255 and when r =255, s =0. Thus

$$s = (L-1) - r \quad \text{..... (4).}$$

Where 'L' is the number of gray levels. Figure below shows the negation concept.

Fig.3 (a): Input image for Negation 3 (b): Output Negative image

2.2 Logarithmic transformations (dynamic range compression)

The log transformation [8] maps a narrow range of low input gray level values into a wider range of output values. The inverse log transformation performs the opposite function. Log functions are particularly useful when the input gray level values in an image have an extremely large range of values as the log operator is an excellent compressing function. The general equation of log transformation is:

$$s = c \times \log(1 + r) \quad \text{..... (5).}$$

Where 'c' is the normalization constant.

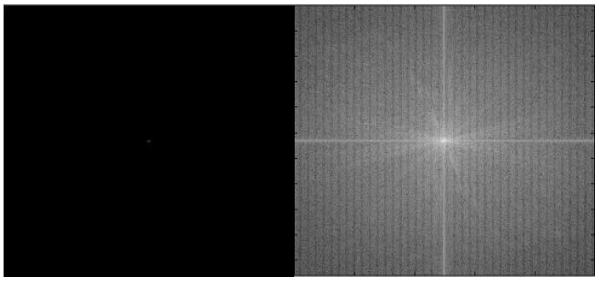

Fig.4 (a): Input image for log function 4(b): Output from log function

2.3 Power-Law transformation

The basic formula for power law transformation [13] is:

$$g(x, y) = c \times f(x, y)^{\gamma} \quad \text{..... (6).}$$

Or $s = c \times r^{\gamma}$ ..... (7).

Here 'c' and 'γ' are positive constants. This transformation function is also called as gamma correction since 'γ' is also called as the gamma correction factor. For various values of 'γ' different levels of enhancement can be obtained. Non linearity's encountered during image capturing, printing and displaying can be corrected using gamma correction. Hence gamma correction is important if the image is to be displayed on the computer screen. The power law

transformation can be used to improve the dynamic range of an image. The minute difference between the log transformation and the power law transformation is that by using the power law transformation function a large family of possible transformation curves can be obtained just by varying the 'γ'.

Fig.5 (a): Input image Fig. 5(b) and 5(c): Output image for γ =0.5 and 1.5

2.4 Piecewise-linear transformation functions

Instead of using a well defined mathematical function we can use arbitrary user defined transforms [8]. The principal advantage of piecewise linear functions over the above types of functions is that the form of piecewise functions can be arbitrarily complex. The main disadvantage of piecewise-linear transformation functions is that their specification requires more user input. Following are the three approaches under this transformation.

2.4.1 Contrast Stretching

We get low contrast images because of insufficient illumination, lack of dynamic range in the image sensors and wrong setting of a lens aperture during image acquisition. Contrast stretching [8][14] is a process that expands the range of intensity levels in an image so that it spans the full intensity range of the recording medium or display device. The reason behind this is to increase the contrast of the images by making the dark portions darker and the bright portions brighter. The figure below shows the transformation used to achieve contrast stretching**.**

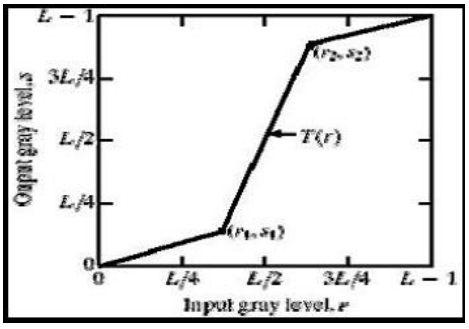

Fig.6: Contrast stretching curve

$$s = \begin{cases} l * r & 0 \le r < a \\ m*(r-a)+v & a \le r < b \\ n*(r-b)+w & b \le r < L-1 \end{cases} \quad \ldots\ldots (8).$$

As is evident from the figure above and by using the above formula, we make the dark grey levels darker by assigning a slope of less than one and make the bright grey levels brighter by assigning a slope greater than one.

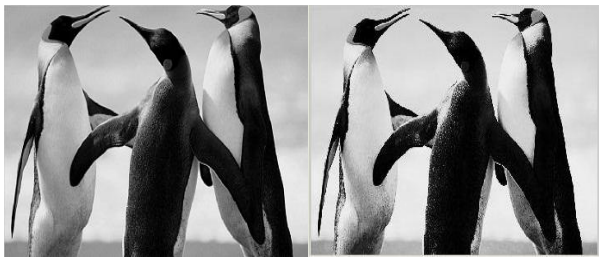

Fig.7 (a): Input image for contrast stretching 7(b): Output image after contrast stretching

One can assign different slopes depending on the input image and the application. As we know that image enhancement is a subjective technique and hence there is no set of slope values that would yield the desired result. The contrast stretching transformation increases the dynamic range of the modified image. From the above graph we can see that the location of points (r_1, s_1) and (r_2, s_2) control the shape of the transformation function. If $r_1 = s_1$ and $r_2 = s_2$, the transformation is a linear function that produces no changes in intensity levels. If $r_1 = r_2$, $s_1 = 0$ and $s_2 = L-1$, the transformation becomes a threshold function [5] that creates a binary image as shown below. The general equation for threshold image is $s = 0$ if $r \le a$ and $s = L-1$ if $r > a$.

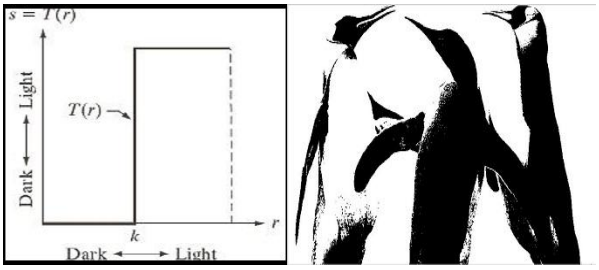

Fig.8 (a): Thresholding function 8(b): Threshold image for fig.7 (a)

Intermediate values of (r_1, s_1) and (r_2, s_2) produce various degrees of spread in the intensity levels of the output image. In general $r_1 \le r_2$ and $s_1 \le s_2$ is assumed, so that the function is single valued and monotonically increasing. This condition preserves the order of intensity levels, thus preventing the creation of intensity artifacts in the processed image.

2.4.2 Intensity (gray) level slicing

Gray level slicing [8] is the spatial domain equivalent to band-pass filtering in the frequency domain. A grey level slicing function can either, highlight a group of intensities and diminish all others or it can highlight a group of intensities

and leave the rest alone. Applications include enhancing features such as masses of water in satellite imagery and enhancing flaws in X-ray images. The transformation function looks similar to the threshold function except that here we select a band of grey level values. It is of following two types: grey level slicing without background and grey level slicing with background.

2.4.2.1 Gray level slicing without background

This can be implemented using the below formulation: $s = L-1$ if $a \le r \le b$ and $s = 0$, otherwise. This method is called as grey level slicing without background [13]. This is because in this process, we have completely lost the background.

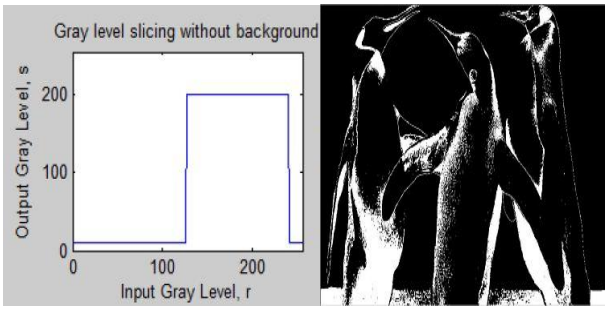

Fig.9 (a): Grey level without background function 9(b): Output image

2.4.2.2 Gray level slicing with background

In some applications, there is a need to enhance a band of grey levels as well as to retain the background. This technique of retaining the background is called as grey level slicing with background [13]. This can be implemented by using the formula below: $s = L-1$ if $a \le r \le b$ and $s = r$, otherwise.

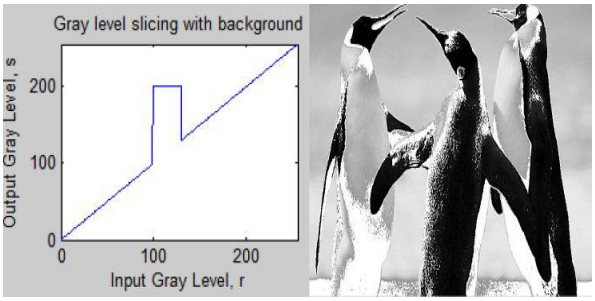

Fig.10 (a): Grey level with background function 10(b): Output image

2.4.3 Bit plane slicing

Pixels are digital numbers composed of bits. For example, the intensity or grey level value of each pixel in a 256-level grey-scale is composed of 8 bits. Instead of showing intensity level ranges, we could highlight the contribution made to total image appearance by specific bits. An 8-bit image may be considered as being composed of eight 1-bit planes, with plane 1 containing the lowest-order bit of all pixels in the image and plane 8 all the highest-order bits. Decomposing an image into its bit planes is useful for analyzing the relative importance of each bit in the image. Also, this kind of decomposition into bit planes is useful for image compression [8]. Let an image is defined as an $256 \times 256 \times 8$ image. In this, 256×256 is the total number of pixels in the image and 8 is the number of bits required to represent each pixel. 8 bits simply means 2^8 or 256 grey levels. In bit plane slicing, we see the importance of each bit in the final image. The higher order bits contain majority of the visually significant data, while the lower order bits contain less details. It is also used in Steganography [10], which is the art of hiding information.

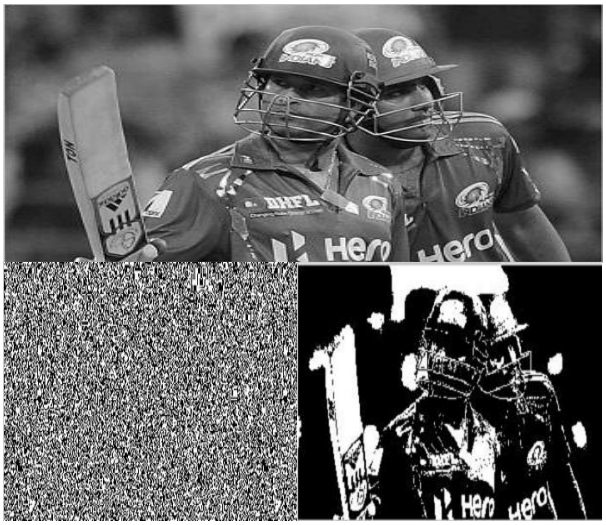

Fig.11 (a): Input image 11(b): Image formed with LSB and 11(c): Image formed with MSB

3. HISTOGRAM PROCESSING

Histogram [2] of images provides a global description of the appearance of an image. Histogram of an image represents the relative frequency of occurrence of the various grey levels in an image. Just by looking at the histogram of an image, a great deal of information can be obtained. It can be plotted in two ways. It can be plotted in two ways. In the first case, the x-axis has the grey levels and the y-axis has the number of pixels in each grey level i.e.

$$h(r_k) = n_k \quad \text{..... (9),}$$

where $r_k = k^{th}$ intensity value and n_k = number of pixels in the image with intensity r_k. In the second case, the x-axis has the grey levels and the y-axis has the probability of the occurrence of that grey level i.e.

$$p(r_k) = n_k / MN \quad \text{..... (10),}$$

for k=1, 2… L-1, where M and N are the row and column dimensions of the image. This is known as normalized histogram. The advantage of this method is that the maximum value to be plotted will always be 1.

3.1 Histogram Stretching

It is a technique used to increase the dynamic range [11] of an image. In this method, we do not change the basic shape of the original histogram, but we spread it to the entire dynamic range. We achieve this by using a straight line equation as shown.

$$s = T(r) = m \times (r - r_{\min}) + s_{\min} \quad \text{..... (11),}$$

Where $m = s_{\max} - s_{\min} / r_{\max} - r_{\min}$. $r_{\max}$ = maximum

grey level of input image, $r_{\min}$ = minimum grey level of input image, $s_{\max}$ = maximum grey level of output image, $s_{\min}$ = minimum grey level of output image. This transformation function shifts and stretches the grey level range of the input image to occupy the entire dynamic range $(s_{\min}, s_{\max})$. Figure below explains this thing clearly.

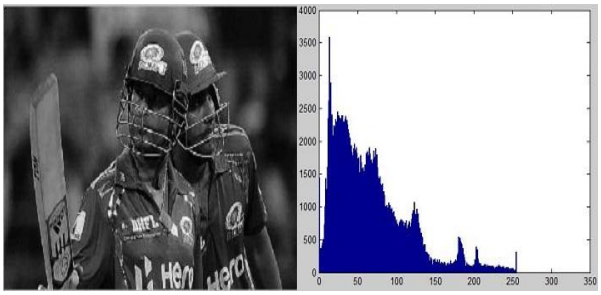

Fig.12 (a): Original Image 12(b): Original Image Histogram

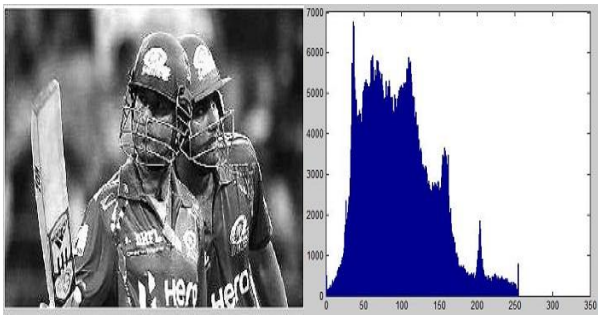

Fig.13 (a): Processed image 13(b): Processed image histogram

3.2 Histogram Equalization

There are many applications, wherein we need a flat histogram; this cannot be achieved by histogram stretching [12]. A perfect image is one which has equal number of pixels in all its grey levels. Hence our objective is not only to spread the dynamic range, but also to have equal pixels in all the grey levels. This is called as histogram equalization [12]. So we need a transformation that would transform a bad histogram to a flat histogram. The transformation that we need must satisfy the below two conditions. We know $s = T(r)$ for $0 \le r \le 1$.

(a) $T(r)$ Must be single valued and strictly monotonically increasing function in the interval $0 \le r \le 1$.

(b) $0 \le T(r) \le 1$ For $0 \le r \le 1$ i.e. $0 \le s \le 1$ for $0 \le r \le 1$, here the range of r is taken as [0, 1]. This is called the normalized range [8]. This range is taken for simplicity. Since the transformation is single valued and monotonically increasing, the inverse transformation exists. i.e. $r = T^{-1}(s); 0 \le s \le 1$. The grey levels for continuous variables can be characterized by their probability density function $p_r(r)$ and $p_s(s)$. From probability theory we know that if $p_r(r)$ and $T(r)$ are known and if $T^{-1}(s)$ satisfies condition (a), then the probability density of the transformed grey level is

$$p_s(s) = [p_r(r)dr / ds]_{r=T^{-1}(s)} \quad \text{..... (12).}$$

we now need to find a transformation which would give us a flat histogram. Let us consider the cumulative density function (CDF) [11]. CDF is obtained by simply adding up all the PDF. i.e.

$$s = T(r) = \int_0^r p_r(r)dr \; ; 0 \le r \le 1, \text{..... (13).}$$

Differentiating with respect to r, we get $\frac{ds}{dr} = p_r(r)$. Hence $p_s(s) = [1] = 1; \; 0 \le s \le 1$. This is nothing but a uniform density function. A bad histogram becomes a flat histogram when we find the CDF.

For discrete values, we deal with probabilities and summations instead of PDFs and integrals. Thus

$$p_r(r) = \frac{n_k}{MN} \quad \text{..... (14),}$$

for k=0, 1, 2…..L-1, where MN=total number of pixels in the image, n_k = number of pixels that have intensity r_k. The discrete form of the transformation is

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j) = \frac{(L-1)}{MN}\sum_{j=0}^{k} n_j \quad \text{... (15),}$$

for k=0, 1, 2…..L-1. Thus a processed image is obtained by mapping each pixel in the input image with intensity r_k into a corresponding pixel with level s_k in the output image using above equation. The example given below explains this procedure.

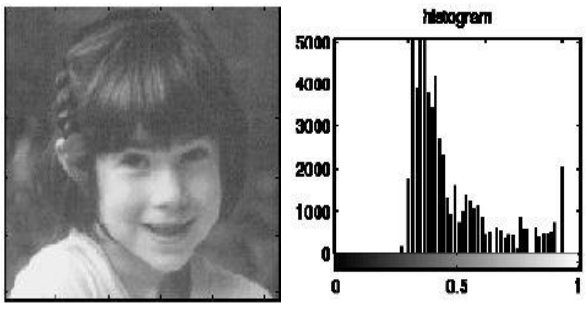

Fig.14 (a): Original image 14(b): Original image Histogram

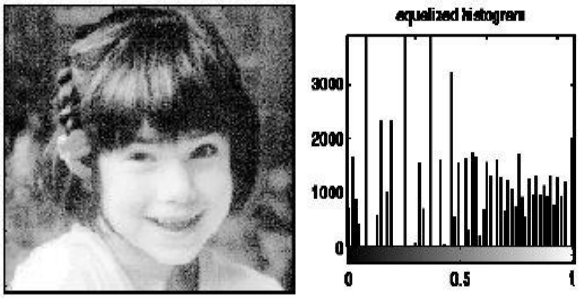

Fig. 15 (a): Equalized image 15(b): Equalized image histogram

3.3 Histogram Specification (Matching)

Histogram equalization [11] automatically determines a transformation function that seeks to produce an output image that has a uniform histogram. It is not interactive, i.e., it always gives one result-an approximation to a uniform histogram. It is at times desirable to have an interactive method in which certain grey levels are highlighted. In particular, it is useful some times to be able to specify the shape of the histogram that we wish the processed image or output image to have. The method used to generate a processed image that has a specified histogram [13] is called histogram matching or histogram specification. Let r and z are continuous intensities and let $p_r(r)$ and $p_z(z)$ denote their corresponding continuous PDFs, respectively. Here r and z denote the intensity levels of the input and output images respectively. We can estimate $p_r(r)$ from the given input image, while $p_z(z)$ is the specified probability density function that we wish the output image to have. Let s be a random variable with the property:

$$s = T(r) = (L-1)\int_o^r p_r(w)dw \quad \cdots\cdots\cdots\cdots$$

(16), where w is a dummy variable of integration. Suppose z be a random variable with the property

$$G(z) = (L-1)\int_o^z p_z(t)dt = s \quad \cdots\cdots\cdots\cdots \quad (17),$$

where t is a dummy variable of integration. It then follows from these two equations that $G(z) = T(r)$ and, therefore, that z must satisfy the condition

$$z = G^{-1}[T(r)] = G^{-1}(s) \quad \cdots\cdots\cdots\cdots \quad (18).$$

the equations (16), (17), (18) above show that an image whose intensity levels have a specified PDF can be obtained from a given image by using the following procedure.

(a) Find the histogram $p_r(r)$ of the input image and determine its equalization transformation using equation (16).

(b) Use the specified PDF $p_z(z)$ of the output image to obtain the transformation function using equation (17).

(c) Find the inverse transformation $z = G^{-1}(s)$; because z is obtained from s, this process is a mapping from s to z, the latter being the desired values.

(d) Obtain the output image by equalizing the input image first; the pixel values in this image are the s values. Then for each pixel in the equalized image [8], perform the inverse mapping to obtain the corresponding pixel of the output image. Histogram matching [11] enables us to 'match' the grey scale distribution in one image to the grey scale distribution in another image. The discrete formulation for histogram matching [11] is shown below. The discrete formulation of equation (16) is

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j) = \frac{(L-1)}{MN}\sum_{j=0}^{k} n_j \quad \cdots \quad (19),$$

for k=0, 1, 2…..L-1. Similarly, given a specific value of s_k, the discrete formulation of equation (17) involves computing the transformation function

$$G(z_q) = (L-1)\sum_{i=0}^{q} p_z(z_i) \quad \cdots\cdots\cdots\cdots \quad (20),$$

for a value of q, so that

$$G(z_q) = G^{-1}(s_k) \quad \cdots\cdots\cdots\cdots \quad (21).$$

Thus, it performs a mapping from s to z. We may summarize the histogram-specification procedure for discrete case as follows: -

(a) Find the histogram $p_r(r)$ of the given image and use it to find the histogram equalization transformation. Round off the resulting values of s_k to the integer range [0, L-1].

(b) Compute all values of the transformation function G for q=0, 1, 2….L-1, where $p_z(z_i)$ are the values of the specified histogram. Round off the values of G to integers in the range [0, L-1]. Store the values of G in a table.

(c) For every value of s_k, k=0, 1, 2…….L-1. Use the stored values of G from previous step to find the corresponding value of z_q so that $G(z_q)$ is closest to s_k and store these mappings from s to z. when more than one values of z_q satisfies the given s_k, choose the smallest value by convention.

(d) Form the histogram-specified image by first histogram-equalizing the input image and then mapping every equalized pixel value, s_k of this image to the corresponding value z_q in the histogram-specified image using the mappings found in step (c).

4. NEIGHBOURHOOD PROCESSING OR SPATIAL FILTERING

The name filter [13] is taken from frequency domain processing, where 'filtering' refers to accepting (passing) or rejecting certain frequency components. A filter that passes low frequencies is called a low pass filter [13]. The net effect produced by a low pass filter is to blur (smooth) an image. We can obtain a similar smoothing directly on the image itself by using spatial filters. A spatial filter consists of

(a) A neighbourhood

(b) A predefined operation that is performed on the image pixels encompassed by the neighbourhood.

Filtering [2] creates a new pixel with co-ordinates equal to the co-ordinates of the centre of the neighbourhood, and whose value is the result of the filtering operation. A processed image is generated as the centre of the filter visits each pixel in the input image. If the operation performed on the input image pixels is linear, then the filter applied is called as a linear spatial filter. Otherwise, the filter is non-linear [2].

4.1 Smoothing Spatial Filters

Smoothing filters are basically used for blurring and for reduction of noise in images. Blurring is used in pre-processing tasks, such as removal of fine details from an image prior to object extraction and various other applications, and bridging of small gaps in lines or curves.

Noise reduction can be accomplished by blurring with a linear filter and also by non-linear filter.

4.1.1 Smoothing Linear Filters

The output of a smoothing, linear spatial filter [13] is simply the average of the pixels contained in the neighbourhood of the filter mask. These filters sometimes are called averaging filters, they are also called as low pass filters. Averaging filters [2] work on the principal of replacing the value of every pixel in an image by the average of the intensity levels in the neighbourhood defined by the mask. This process results in an image with reduced 'sharp' transitions in intensities. Since random noise shows sharp transitions in intensity levels, the most obvious application of smoothing is Gaussian noise reduction. However edges also are characterized by sharp intensity transitions, so averaging filters have the undesirable side effect that they blur edges along with noise reduction. Averaging filters works well for Gaussian and salt noise but fails for pepper noise. Figure below shows two common 3×3 filters.

1/9

1	1	1
1	1	1
1	1	1

1/16

1	2	1
2	4	2
1	2	1

Fig.16 (a): Averaging Filter 16(b): Weighted Averaging Filter

Use of the first filter gives the standard average of the pixels under the mask. A $m\times n$ mask would have a normalizing constant equal to $1/nm$. The second mask gives us a weighted average of the pixels under its neighborhood. In this mask the center of the mask's pixel is multiplied by a higher value than any other, thus giving this pixel more importance in the calculation of the average. The main motive behind weighing the center point the highest and then reducing the value of the coefficients as a function of increasing distance from the origin is simply an effort to reduce blurring in the smoothing process. The general implementation for filtering an $M\times N$ image with a weighted averaging filter of size $m\times n$ is given by

$$g(x,y)=\frac{\sum_{s=-a}^{a}\sum_{t=-b}^{b}w(s,t)\,f(x+s,y+t)}{\sum_{s=-a}^{a}\sum_{t=-b}^{b}w(s,t)} \quad \text{..... (22),}$$

for x=0, 1, 2…..M-1, y=0, 1, 2……N-1. The images below show the result of both the above mask.

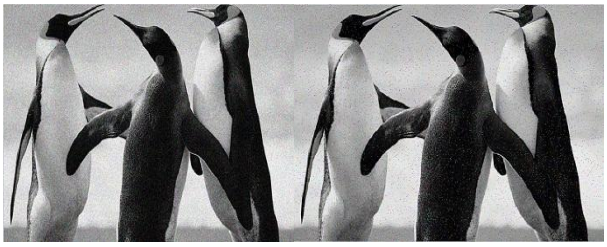

Fig.17 (a): Image with Gaussian noise 17(b): Image with Salt and pepper noise

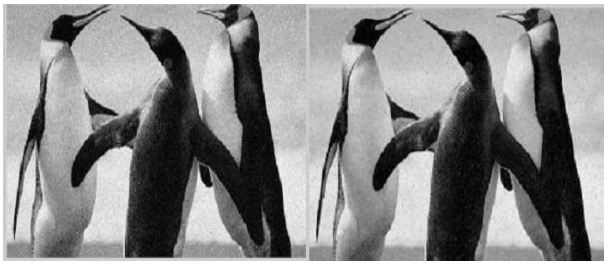

Fig.18: Output of fig.17 (a) and (b) after applying averaging filter

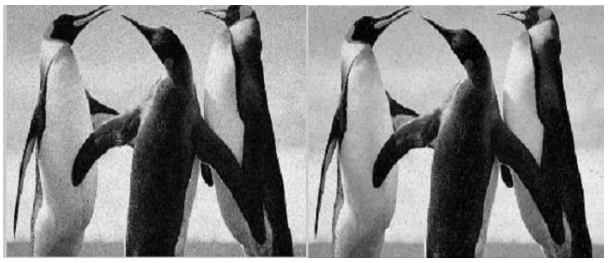

Fig.19: Output of fig.17 (a) and (b) on applying weighted averaging filter

4.1.2 Order-Static (non Linear) filters

The response or result of these filters depends on ordering (ranking) the pixels contained in the image area encompassed by the filter or mask, and then replacing the value of the centre pixel with the value determined by the ranking operation result. Median filter [2], max filter [2] and min filter [2] fall under this category. Median filter replaces the value of a pixel by the median of the intensity values in the neighborhood of that pixel. Median filters are very effective in the removal of salt and pepper noise (impulsive noise) because of its appearance as white and black dots superimposed on an image. The median operator requires an ordering of the values in the pixel neighborhood at every pixel location. This increases the computational cost of the median operator.

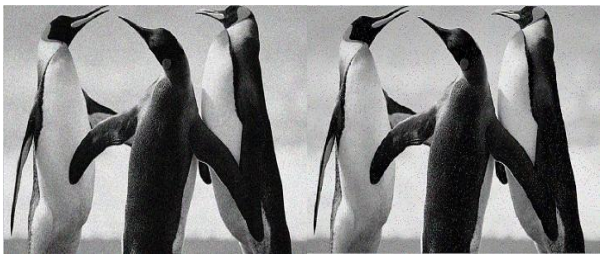

Fig.20 (a): Image with Gaussian noise 20(b): Image with Salt and pepper noise

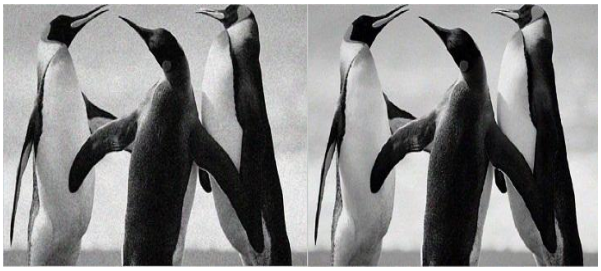

Fig.21 (a): Output of Median filter on Gaussian noise 21(b): Output of Median filter on salt and pepper noise

4.2 Sharpening Spatial filters

The main objective of sharpening filters is to highlight transitions in intensity. Sharpening [2] can be accomplished by spatial differentiation [6]. The derivatives of a digital function [2] are defined in terms of differences. We require that any definition we use for a first derivative must be-

(a) Zero in area of constant intensity.

(b) Non zero at the onset of an intensity step or ramp.

(c) Non zero along ramps.

Similarly, any definition of a second derivative must be-

(a) Zero in areas of constant intensity.

(b) Non zero at the onset and end of an intensity step or ramp.

(c) Zero along ramps of constant slope.

4.2.1 The Laplacian

This is the simplest isotropic filter [13], whose response is independent of the direction of the discontinuities in the image. It is defined for a function (image) $f(x, y)$ of two variables as

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad \text{.....} \quad (23).$$

Because derivatives of any order are linear operations, the laplacian is a linear operator. To express this equation in discrete form, we have

$$\frac{\partial^2 f}{\partial x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y) \quad \text{.....} \quad (24).$$

$$\frac{\partial^2 f}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y) \quad \text{.....} \quad (25).$$

Therefore, discrete laplacian of two variables is as below:

$$\nabla^2 f(x,y) = f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y) \quad \text{..(26).}$$

this equation can be implemented using the filter mask (a) shown below. It gives an isotropic result for rotations in increments of 90 degree. Filter mask (b) is used to implement isotropic results in increments of 45 degree. Their implementation on images is shown below.

0	1	0
1	-4	1
0	1	0

1	1	1
1	-8	1
1	1	1

Fig.22 (a): Laplacian filter 1 **22 (b): Laplacian filter 2**

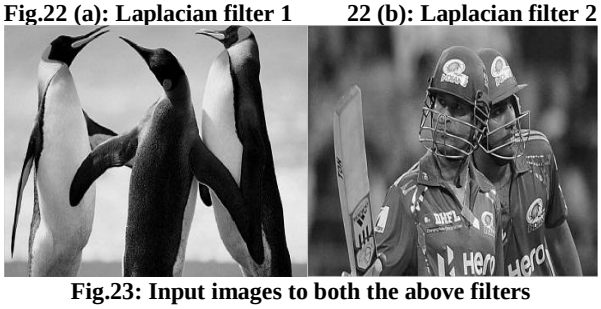

Fig.23: Input images to both the above filters

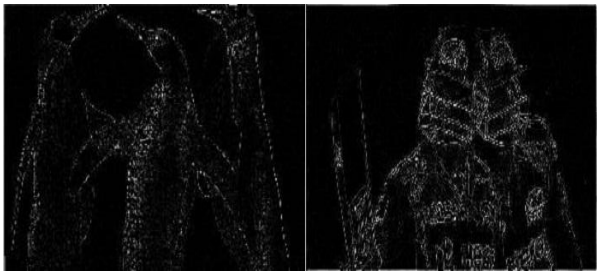

Fig. 24(a): and 24(b): Output on applying Laplacian filter 1

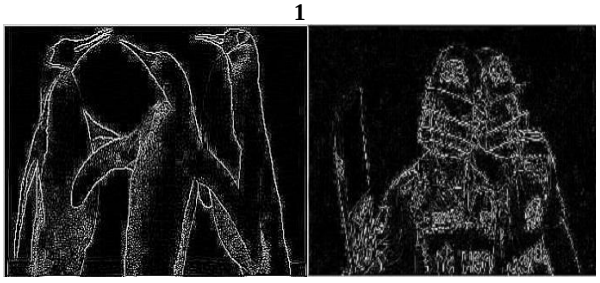

Fig. 25(a): and 25(b): Output on applying Laplacian filter 2

Since the laplacian [15] is a derivative operator, its usage highlights intensity discontinuities in an image and deemphasizes regions with slowly varying intensity levels. This will result in images with grayish edge lines and other discontinuities producing featureless background. Background features can be recovered while still preserving the sharpening effect of the laplacian simply by adding the laplacian image to the original image. The basic ways in which we use the laplacian for image sharpening is given by

$$g(x, y) = f(x, y) + c[\nabla^2 f(x, y)] \quad \text{.....} \quad (27),$$

where $f(x, y)$ and $g(x, y)$ are the input and sharpened images respectively. The constant c=-1, if the above two laplacian filters are used and c=1 if the above laplacian filters coefficient are multiplied by -1.

Fig.26 (a): and 26(b): Input images for Sharpening

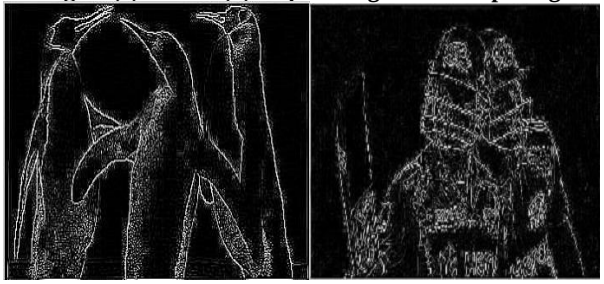

Fig. 27(a) and (b): Images after applying laplacian filter 2

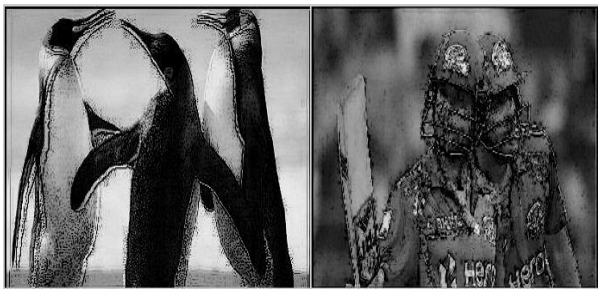

Fig. 28 (a) and (b) sharpened Image after adding Images of Fig.26 (a) with 27 (a) and fig.26 (b) with 27 (b)

4.2.2 The Gradient

First derivatives in image processing are implemented using the magnitude of the gradient [12]. For $f(x,y)$, the gradient of f at coordinates (x,y) is defined as the 2D column vector given by

$$\nabla f \equiv grad(f) \equiv \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad \text{…….....} \quad (28).$$

this vector points in the direction of the greatest rate of change of f at location (x,y). The magnitude of vector ∇f, denoted as $M(x,y)$ is the value at (x,y) of the rate of change in the direction of the gradient vector and is as

$$M(x,y) = mag(\nabla f) = \sqrt{g_x^2 + g_y^2} \quad \text{……..... (29).}$$

$M(x,y)$ is an image of the same size as the original and is also called as the gradient image. Because the components of the gradient vector are derivatives, they are linear operators. However, the magnitude of this vector is not because of the squaring and square root operations. On the other hand, the partial derivatives [2] in eq. (28) are not rotation invariant (isotropic), but the magnitude of the gradient vector is. Equation (29) can be written as

$$M(x,y) \approx |g_x| + |g_y| \quad \text{……..... (30).}$$

We now define discrete approximations to the above equations using the matrix shown below.

	y-1	y	y+1
x-1	Z_1	Z_2	Z_3
x	Z_4	Z_5	Z_6
x+1	Z_7	Z_8	Z_9

Fig.29: Matrix representation for pixels in an image

Discrete approximations to the above equations are: -

$$g_x = \frac{\partial f}{\partial x} = f(x+1,y) - f(x,y) = z_8 - z_5 \quad \text{…….} \quad (31).$$

$$g_y = \frac{\partial f}{\partial y} = f(x,y+1) - f(x,y) = z_6 - z_5 \quad \text{…….} \quad (32).$$

From these equations filter masks are developed. Roberts use cross differences:- $g_x = z_9 - z_5$ and $g_x = z_9 - z_6$. The gradient image using above two gradients is

$$M(x,y) = [(z_9 - z_5)^2 + (z_8 - z_6)^2]^{1/2} \quad \text{……..…..} \quad (33).$$

From equation (30),

$$M(x,y) \approx |z_9 - z_5| + |z_8 - z_6|.$$

The partial derivative terms used in above equation can be implemented using the two linear filter masks below.

-1	0
0	1

0	-1
1	0

Fig.30 (a): and (b): Roberts gradient using cross difference

Fig.31 (a): Input image (b): Output image from Roberts mask

Prewitts 3×3 mask depends on following gradients.

$$g_x = \frac{\partial f}{\partial x} = |(z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)| \quad \text{…….} \quad (34).$$

$$g_y = \frac{\partial f}{\partial y} = |(z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)| \quad \text{…..……..} \quad (35).$$

The gradient image using above two gradients is $M(x,y) = |(z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)| + |(z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)| \ldots$ (36).

The partial derivative terms used in above equation can be implemented using the two linear filter masks below.

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

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

Fig.32 (a): Prewitts x-gradient (b): Prewitts y-gradient

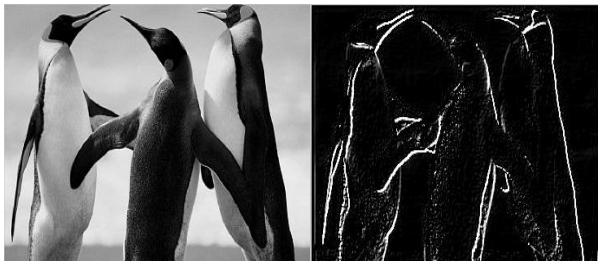

Fig.33 (a): Input image 33(b): Output image on applying Prewitts mask

Sobel 3×3 mask depends on following gradients.

$$g_x = \frac{\partial f}{\partial x} = |(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)| \text{ ..... (37).}$$

$$g_y = \frac{\partial f}{\partial y} = |(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)| \text{ ..... (38).}$$

The gradient image using above two gradients is

$$M(x,y) = |(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)| + |(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)| \text{ .... (39).}$$

The partial derivative terms used in above equation can be implemented using the two linear filter masks below.

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

Fig.34 (a): Sobel x-gradient 34(b): Sobel y-gradient

Fig.35 (a): Input image 35(b): Output image on applying Sobel mask

Implementation of Roberts [3], Prewitts [3] and Sobel [3] operators on the image can be done in two steps. (a) Add the x and y gradient into one gradient mask. (b) Convolve [11] the new mask with the original image.

5. CONCLUSION

Image enhancement [8] algorithms offer an enormous variety of approaches for modifying images to obtain visually acceptable images. The choice of specific methodology is a function of the specific task as an application, content of an image, observer features, and viewing conditions. The survey of various Image enhancement techniques in spatial domain has been successfully accomplished using matlab code on two images 'Penguin.jpg' and 'Sachin.jpg' in this paper. Image Enhancement is one of the most important and difficult component of digital image processing. Based on the type of image and type of noise with which it is corrupted, a slight change in individual method or combination of any methods further improves visual quality. In this paper, we focus on studying the existing techniques of image enhancement in spatial domain. The point processing methods are an important image processing techniques and are used basically for contrast enhancement. Digital Negative is suited and is used for enhancing white detail embedded in dark regions. It has applications in medical imaging. Power-law functions are useful for general purpose contrast manipulation applications. For a dark image with histogram towards the darker region, an expansion of gray levels is obtained using a power-law transformation with a fractional exponent less than 1. Log Transformation is beneficial for expressing and enhancing details in the darker regions of the image at the expense of detail in the brighter regions. For an image having a washed-out or brighter appearance, a compression of gray levels is obtained using a power-law transformation with gamma greater than 1. The histogram of an image (i.e., a plot of the gray level frequencies) provides important information regarding the contrast of an image through the intensity values in an image. Histogram equalization is a transformation that stretches the contrast by redistributing the gray-level values uniformly without producing a flat histogram. Only the global histogram equalization can be done completely automatically. Although in this survey paper we did not discuss the computational cost of enhancement algorithms. it may play an important and critical role in the selection of an algorithm for real-time applications. Despite the effectiveness of each of these techniques when applied separately, in practice one has to devise a combination of such methods to achieve more effective image enhancement.

6. REFERENCES

[1] A. K. Jain, "Fundamentals of Digital Image Processing", Prentice Hall of India, 1989.

[2] Rafael C. Gonzalez and Richard E. woods, "Digital Image Processing", Pearson Education, Second Edition, 2005.

[3] W. K. Pratt, "Digital image processing", Prentice Hall, 1989.

[4] Gajanand Gupta, "Algorithm for Image Processing Using Improved Median Filter and Comparison of Mean, Median and Improved Median Filter", International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-1, Issue-5, November 2011.

[5] Chris Solomon, "Fundamentals of Digital Image Processing", John Wiley & Sons Ltd.

[6] R. Jain, R. Kasturi and B.G. Schunck, "Image Processing Fundamentals", McGraw-Hill International Edition, 1995.

[7] Jafar Ramadhan Mohammed, "An Improved Median Filter Based on Efficient Noise Detection for High Quality Image Restoration", IEEE Int. Conf, PP. 327 – 331, May 2008.

[8] Raman Maini and Himanshu Aggarwal "A Comprehensive Review of Image Enhancement Techniques", Journal of Computing, Volume 2, Issue 3, March 2010, pp.8-13.

[9] Sunita Dhariwal, "Comparative Analysis of Various Image Enhancement Techniques", International Journal of Electronics & Communication Technology (IJECT), Vol. 2, Issue 3, pp. 91-95, Sept. 2011.

[10] Bhabatosh Chanda and Dwijest Dutta Majumder, 2002, Digital Image Processing and Analysis.

[11] R.W.Jr. Weeks, (1996). Fundamental of Electronic Image Processing. Bellingham: SPIE Press.

[12] R Hummel, "Histogram modification techniques", Computer graphics and Image Processing, Vol. 4, pp. 209-224, 1975.

[13] Dhananjay K. Theckedath, 2008. Digital Image Processing. Tech -Max publication, Pune, India.

[14] Balvant Singh, Ravi Shankar Mishra, Puran Gour, "Analysis of Contrast Enhancement Techniques for underwater image", International Journal of Computer Technology and Electronics Engineering (IJCTEE) ISSN: 2249-6343, Vol. 1, Issue 2, pp. 190-194.

[15] Asst. Prof. Dr. Umut Arioz, Lecture notes on Digital Image processing, Lecture 3: Image enhancement in Spatial Domain.