

Image Enhancement using Sharpen Filters

Mr. Harvinder Singh

M.tech Student, Amity university, Noida(U.P) India

Prof (Dr). J.S. Sodhi

Asstt. Vice President, Amity University, Noida(U.P) India

Abstract - Image processing is basically the use of computer algorithms to perform image processing on digital images. Digital image processing is a part of digital signal processing. Digital image processing has many significant advantages over analog image processing. Image processing allows a much wider range of algorithms to be applied to the input data and can avoid problems such as the build-up of noise and signal distortion during processing of images. In this paper, a new nonlinear filtering technique is introduced for enhancement of images that are highly contaminated by impulse noise. The proposed filtering technique is more effective in eliminating impulse noise and preserving the image features. The filter replaces a corrupted pixel by the median value or by its processed neighboring pixel value.

Keywords – sharpening, Unsharp Masking, Sharpening Filters, Sharpening Parameters

I. INTRODUCTION

Image sharpening is a powerful tool for emphasizing texture and drawing viewer focus. It's also required of any digital photo at some point — whether you're aware it's been applied or not. Digital camera sensors and lenses always blur an image to some degree, for example, and this requires correction. However, not all sharpening techniques are created equal. When performed too aggressively, unsightly sharpening artifacts may appear. On the other hand, when done correctly, sharpening can often improve apparent image quality even more so than upgrading to a high-end camera lens. 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 completely 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.

II. PROPOSED ALGORITHM

2.1 SHARPENING IMAGES AND UNSHARP MASK

One of the most often used and abused post processing tools are the sharpen filters. When applied properly, sharpening can boost the image by making it look crisper and more defined. However, there is a tendency to go overboard with this and the image can come out looking unnatural and ‘oversharpened.’ There is the popular misconception that if you sharpen a blurry image, it will magically appear clearer. What does happen is that the blurry image will just look more terrible.

Sharpening works by exaggerating the contrast of the object’s edges, giving the viewer the impression of distinct delineation. There are many ways to do this. You can sharpen the entire image in one click with the Sharpen filter or you can use the Unsharp Mask for more control over how defined you want the sharpening to be. Some people prefer to use the High Pass Filter or the Smart Sharpen. Whichever method you use, just remember to be light-handed when sharpening since too much can make the image look unnatural with distorted pixels. One tip would be to sharpen only certain areas in the shot instead of the whole image. For example, if your image is of a bird flying against the cloudy sky, sharpen only parts of the bird while leaving the sky untouched to keep it looking smooth.

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 1 illustrates this procedure, often referred to as unsharp masking on a

one-dimensional signal. As shown in Figure 1, 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

$$S_{i,j} = x_{i,j} + \lambda F(x_{i,j})$$

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 than or equal to 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 λ 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.

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.

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

Eq.1

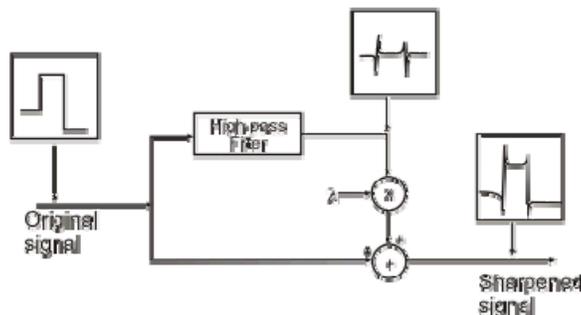


Figure 1.Implementation of the high-pass filter

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 .

Since the filter 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 2 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.

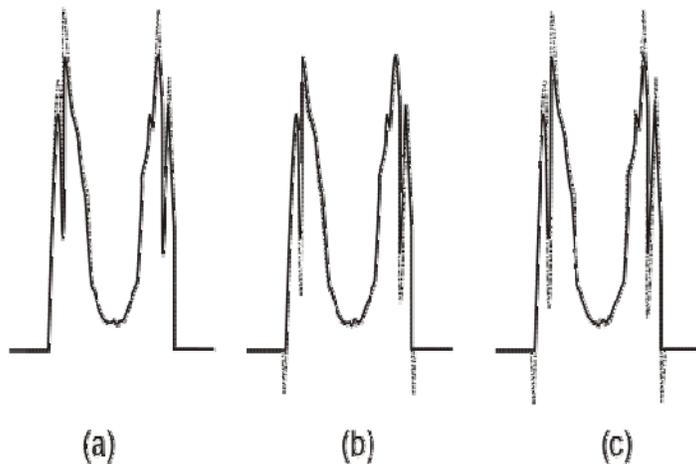


Figure 2. basic image sharpening structure

Thus both positive-slope edges and negative-slope are equally highlighted. This procedure is illustrated in Figure 2, 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 2 together with a row of the sharpened image when only the positive-slope edges are highlighted, only the negative-slope edges are highlighted, and both positive-slope and negative-slope edges are jointly highlighted.

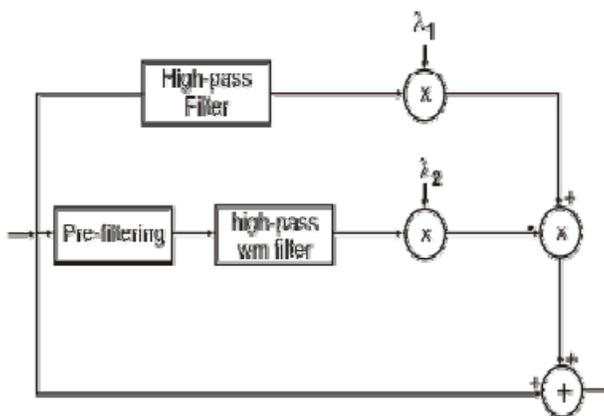


Figure 3. prefiltering operation

In Figure 3 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

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

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.

2.2 SHARPENING: UNSHARP MASK

2.2.1 Types of Sharpening

In an attempt to control the variables in image sharpening, many photographers adopted a three-step sharpening process where separate passes of sharpening are applied at different stages in the workflow. The first stage, often called **Capture sharpening**, adds a light amount of sharpening in the raw conversion process to compensate for the slight amount of image sharpness lost as the photo is digitized.

Note: The term capture sharpening is a bit of a misnomer as it is not a good idea to add sharpening in-camera using your camera's menu. More often than not, sharpening added in-camera will cause problems later in your workflow. The first pass of sharpening should happen in your raw converter, not your camera.

The second round of sharpening, called **Creative sharpening**, selectively sharpens key lines and focus areas within an image. For example, in a portrait, a photographer may add sharpening to the eyes, smile and jewelry to grab the viewer's attention.

The final pass of sharpening, **Output Sharpening**, aims to optimize image sharpness based on the final image size, output medium (print or web; glossy or watercolor paper) and anticipated viewing distance. Of the three passes, this stage requires the greatest attention on the part of the photographer. It is very easy to over or undersharpen images and miss the mistake until the finished print rolls off the printer or appears in the magazine. To the uninitiated, this may feel like a bit of a high-wire act, but never fear, this article will help walk you through the process and provide plenty of guidance on what clues to look for to make sure you never oversharpen your images. It is far better to undersharpen images, than to oversharpen them.

An "unsharp mask" is actually used to sharpen an image, contrary to what its name might lead you to believe. Sharpening can help you emphasize texture and detail, and is critical when post-processing most digital images. Unsharp masks are probably the most common type of sharpening, and can be performed with nearly any image editing software (such as Photoshop). An unsharp mask cannot create additional detail, but it can greatly enhance the appearance of detail by increasing small-scale acutance.

2.3 HOW IT WORKS

Most image sharpening software tools work by applying something called an "unsharp mask," which despite its name, actually acts to sharpen an image.

The sharpening process works by utilizing a slightly blurred version of the original image. This is then subtracted from the original to detect the presence of edges, creating the *unsharp mask* (effectively a high-pass filter). Contrast is then selectively increased along these edges using this mask — leaving behind a sharper final image.

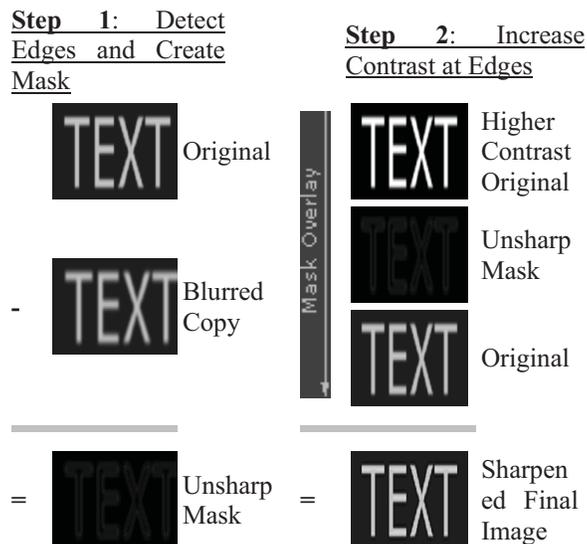
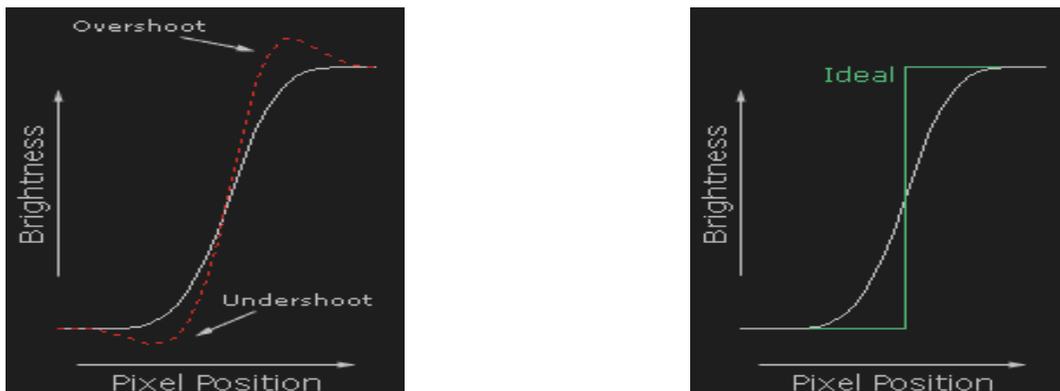


Figure 4. The sharpening process

Note: The "mask overlay" is when image information from the layer above the unsharp mask passes through and replaces the layer below in a way which is proportional to the brightness in that region of the mask. The upper image does not contribute to the final for regions where the mask is black, while it completely replaces the layer below in regions where the unsharp mask is white. If the resolution in the above image is not increasing, then why is the final text so much sharper? We can better see how it works if we magnify and examine the edge of one of these letters as follows:



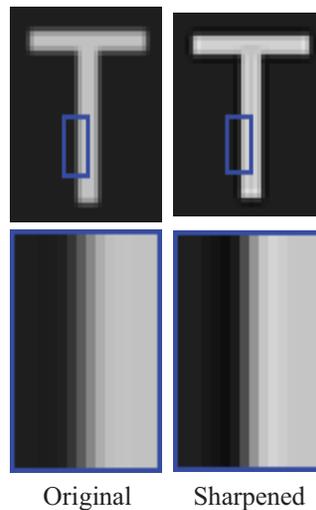


Figure 5. examine the edge of one of the letters

2.4 SHARPENING FILTERS

Sharpening filters are used in order to highlight fine details within an image. They are based on first and second order derivatives. Since in image processing, we deal with discrete quantities, the definitions for the discrete first and second derivatives should be used. The basic definition for a first order derivative of a one-dimensional function is given by:

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

Similarly, the discrete form of a second order derivative in one dimension is given by:

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

First order derivatives are used to produce thicker edges in an image and are usually used for edge extraction. Second order derivatives on the other hand, have a stronger response to fine detail and are usually better for image enhancement than the first order derivatives.

One sharpening filter is the Laplacian. The Laplacian is a linear operator and it forms an isotropic filter. The Laplacian highlights grey level discontinuities and will output an image with a black background and grey lines where the edges of an object in the image are. In order to get a sharpened image, typically, the resulting Laplacian filtered image (or a weighted version of it) is added to the original image. In practice, a mask is typically used that will carry out both steps at once.

The Laplacian is given by:

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

The discrete form is:

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

Combining the addition of the Laplacian into the image if:

$$g(x, y) = f(x, y) + \nabla^2 f(x, y)$$

Then,

$$\begin{aligned} g(x, y) &= f(x, y) - [f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1)] + 4f(x, y) \\ &= 5f(x, y) - [f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1)] \end{aligned}$$

Note that while the sharpening process isn't able to reconstruct the ideal image above, it is able to create the appearance of a more pronounced edge. The key to effective sharpening is walking the delicate balance between making edges appear sufficiently pronounced, while also minimizing visible under and overshoots (called "sharpening halos").

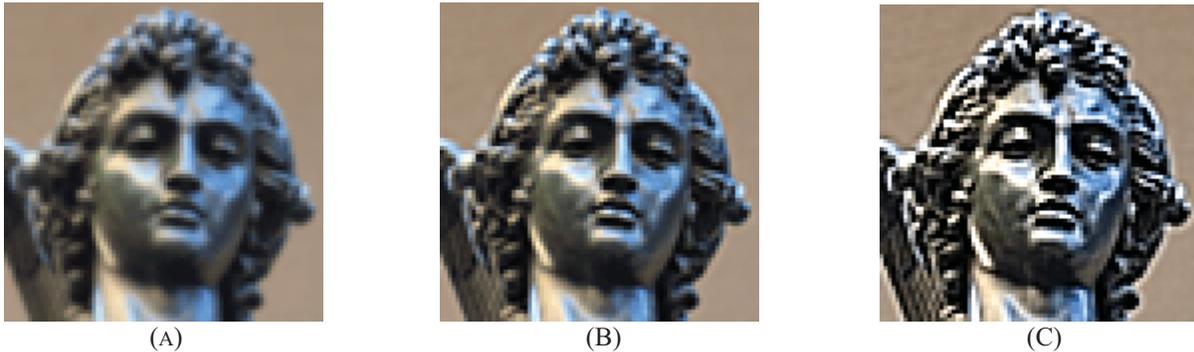


Figure 6. (A) Soft Original (B) Mild Sharpening (C) Over Sharpening

2.5 SHARPENING WORKFLOW

Most photographers now agree that sharpening is most effective and flexible when it's applied more than once during image editing. Each stage of the sharpening process can be categorized as follows:

Capture sharpening aims to address any blurring caused by your image's source, while also taking image noise and detail into consideration. With digital cameras, such blurring is caused by the camera sensor's anti-aliasing filter and demosaicing process, in addition to your camera's lens. Capture sharpening is required for virtually all digital images, and may be applied automatically by the camera for photos which are saved as JPEG files. It also ensures the image will respond well to subsequent rounds of sharpening.

Creative sharpening is usually applied selectively, based on artistic intent and/or image content. For example, you might not want to apply additional sharpening to a smooth sky or a person's skin, but you may want to crank up the sharpness in foliage or a person's eye lashes, respectively. Overall though, its use may vary wildly from photo to photo, so creative sharpening is really a "catch all" category. It's also the least used stage since it can also be the most time-consuming.

Output sharpening uses settings customized for a particular output device, and is applied at the very end of the image editing workflow. This may include special considerations based the size, type and viewing distance of a print, but it can also be used to offset any softening caused by resizing an image for the web or e-mail.

Overall, the above sharpening workflow has the convenience of being able to save edited images at a near-final stage. When printing or sharing one of these images, all that is needed is a quick top-off pass of sharpening for the output device. On the other hand, if all sharpening were applied in a single step, then all image editing would have to be re-done every time you wished share/print the photo using a different output device.

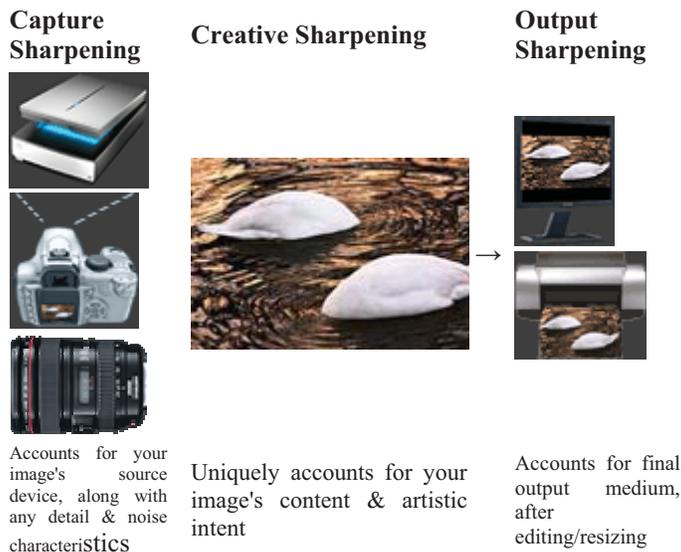


Figure 7. Sharpening Workflow

2.6 SHARPENING PARAMETERS

Most of the sharpening techniques uses at least three parameters:

Radius

Controls the size of the edges you wish to enhance, where a smaller radius enhances smaller-scale detail. You'll usually want a radius setting that is comparable to the size of the smallest detail within your image.

Amount

Controls the overall strength of the sharpening effect, and is usually listed as a percentage. A good starting point is often a value of 100%.

Threshold (Masking)

Controls the minimum brightness change that will be sharpened. This can be used to sharpen more pronounced edges, while leaving more subtle edges untouched. It's especially useful to avoid sharpening noise.

2.7 RATES OF CHANGE

When we are sharpening an image we are trying to enhance the differences. What are the differences? Well, let's take a look at a 1D strip of intensities:

200	180	160	140	120	200	200	200	100	100	100	100
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

If we take a cursory look at these pixels we'll see that the intensities gradually decrease from left to right before jumping back up dramatically then dropping back down. The dramatic 'jumps' are what will ultimately be enhanced when we sharpen the image. Here they are highlighted.

200	180	160	140	120	200	200	200	100	100	100	100
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

We want to know how fast the intensities are changing. If Δ represents the rate of change between adjacent pixels then our rate of change looks like:

<i>i</i>	200	180	160	140	120	200	200	200	100	100	100	100
Δ	-20	-20	-20	-20	80	0	0	-100	0	0	0	

We just subtract the value of the next pixel from the value of the current pixel. A little more formally, this looks like

$$\Delta(x) = f(x+1) - f(x)$$

Under normal conditions, some natural variation will occur pixel by pixel. We want to ignore that natural variation and only concern ourselves when the change is drastic. For that we will have to perform the subtraction one more time.

<i>i</i>	200	180	160	140	120	200	200	200	100	100	100	100
Δ	-20	-20	-20	-20	80	0	0	-100	0	0	0	
Δ^2		0	0	0	100	-80	0	-100	100	0	0	

Now we can see how fast each intensity is changing. Once again, let's formalize this.

$$\Delta^2(x) = \Delta(x+1) - \Delta(x)$$

$$\Delta^2(x) = (f(x+1) - f(x)) - (f(x) - f(x-1))$$

$$\Delta^2(x) = f(x-1) - 2 * f(x) + f(x+1)$$

III. EXPERIMENT AND RESULT

3.1 EXPERIMENTAL RESULTS WITH VARYING PARAMETERS

I.



Input image

(a)

(b)

(c)

(d)

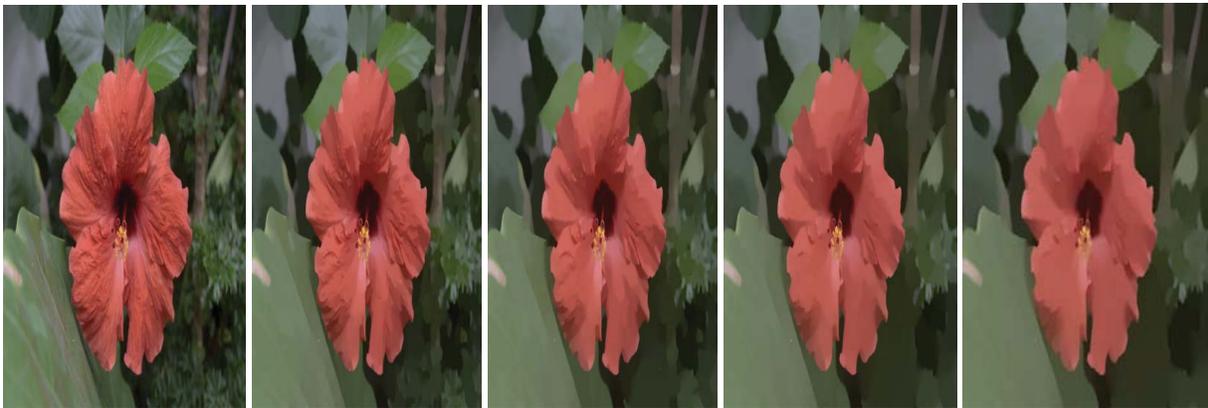
$\lambda=0.006$

$\lambda=0.012$

$\lambda=0.018$

$\lambda=0.024$

II.



Input Image

(a)

(b)

(c)

(d)

$\lambda=0.015$

$\lambda=0.025$

$\lambda=0.035$

$\lambda=0.045$

Table -1 Experiment Result

	(a)	(b)	(c)	(d)
I. Input Image	$\lambda=0.006$	$\lambda=0.012$	$\lambda=0.018$	$\lambda=0.024$
II. Input Image	$\lambda=0.015$	$\lambda=0.025$	$\lambda=0.035$	$\lambda=0.045$

Table 1 show the input image and the other images obtained after applying different parameters to the input image . Two images are taken to show the variations in image quality at different parameters.

IV.CONCLUSION AND FUTURE WORK

There are a couple of areas which we would like to improve on. One area is in improving the de-noising along the edges as the method we used did not perform so well along the edges. Another area of improvement would be to develop a better optimality criterion as the MSE is not always the best optimality criterion.

The future work of research would be to implement Wiener Filter in Wavelet Domain, applying the methods in which the noise variance is known & in which the noise variance is unknown i.e. the MAD method.

REFERENCES

- [1] Image De-noising using Wavelet Thresholding and Model Selection. Shi Zhong Image Processing, 2000, Proceedings, 2000 International Conference on, Volume: 3, 10-13 Sept.2000 Pages: 262.
- [2] James C. Church, Yixin Chen, and Stephen V. Rice Department of Computer and Information Science, University of Mississippi, "A Spatial Median Filter for Noise Removal in Digital Images", IEEE, page(s): 618-623, 2008.
- [3] Image De-noising by Various Filters for Different Noise, Pawan Patidar, Sumit Srivastava, Manoj Gupta, Ashok Kumar Nagawat International Journal of Computer Applications (0975 – 8887) Volume 9– No.4, November 2010
- [4] HUANG, T.S. 1981. Two-Dimensional Signal Processing II: Transforms and Median Filters. Berlin: Springer-Verlag, pp. 209-211.
- [5] GIL, J. AND WERMAN, M. 1993. Computing 2-D Min, Median, and Max Filters. IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 15 No. 5, pp. 504-507.
- [6] KABIR, I. 1996. High Performance Computer Imaging. Greenwich, CT. Manning Publications. pp. 181-192.
- [7] PARIS, S. AND DURAND, F. 2006. A Fast Approximation of the Bilateral Filter using a Signal Processing Approach. ECCV 2006.
- [8] PHA, T. Q. AND VLIET, L. J. V. 2005. Separable bilateral filtering for fast video preprocessing. IEEE Int. Conf. on Multimedia & Expo. CD1-4.
- [9] RANKA, S. AND SAHNI, S. 1989. Efficient Serial and Parallel Algorithms for Median Filtering. Proceeding 1989 International Conference on Parallel Processing, III-56 -- III-62.