

# Matlab Analysis of Image Smoothing By Butterworth and Gaussian Low Pass Filter

Aditi Sharma  
Shobhit University, Meerut  
Department of Computer Engineering

Vijay Maheshwari, PhD  
Shobhit University, Meerut  
Department of Computer Engineering

## ABSTRACT

Most images have noise which creates blur and degrade the quality of image. Image analysis can be obtained by using various filtering techniques. Generally, filtering can be done in frequency domain where we use Fourier transform. In this paper we have analyzed smoothing functions like using Butterworth low pass filter Gaussian low pass filter and their adaptation function.

## Keywords

Smoothing, Butterworth, Gaussian, Fourier Transform, Frequency Domain

## 1. INTRODUCTION

Any image can be represented in the form of  $f(x, y)$  in spatial domain where  $(x, y)$  are spatial coordinates and  $f$  is the amplitude or intensity level value. The image captured by camera is of low quality i.e. it contain blur or noise. This noise can be of various types like Gaussian noise, Salt and Pepper noise, Rayleigh noise etc. which hampers the quality of the image. The quality can be improved by using various filters depending on the requirement. In this paper we have discussed the two types of filters for smoothing Butter worth low pass filter and Gaussian low pass filter. Filtering is the process of removing unwanted noise from the image. The motive of image filtering is to process the image so that the result is more suitable then the original image for a specific applications. Image filtering can be given as a process that removes the noise, improves the digital image for varied applications like bio-medical imaging.

### 1.1 Smoothing

In smoothing what we do is we actually perform low pass filtering. Low pass filtering means allowing the low frequency components of the image to pass through filter. For any image low frequency components are background data shaded region etc.

Frequency domain techniques are all about working in the Fourier transform. Smoothing can be done by using three types of filters Ideal, Butterworth and Gaussian filters. In this paper we will discuss about Butterworth and Gaussian low pass filters.

### 1.2 Fourier Transform

Fourier transform named after the French mathematician Jean Baptiste Joseph Fourier is one of the most important transforms that is used in Image Processing. Fourier transform is understood with the help of this real world example:

'White light when passes through prism get divided into its spectrum VIBGYOR (Violet, Indigo, Blue, Green, Yellow, Orange, and Red) this phenomenon of splitting white light into its constituents colors is called Fourier transform when an infinite domain gets converted into a finite domain and when we place an inverted prism in front of prism we get back again

the white light on the screen it is called Inverse Fourier Transform.

## 2. FREQUENCY DOMAIN FILTERING

In frequency domain filtering we have the image as  $F(u, v)$ . This can be achieved by performing Discrete Fourier transform  $f(x, y)$  can be converted into  $F(u, v)$  by using the formula:

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(\frac{ux}{M} + \frac{vy}{N})}$$

Where  $(x, y)$  are spatial domain coordinates,  $(u, v)$  are frequency domain coordinates and  $M * N$  is image matrix size for calculation  $M=N$ .

Similarly  $F(u, v)$  can be converted into  $f(x, y)$  by using the formula:

$$f(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi(\frac{ux}{M} + \frac{vy}{N})}$$

### 2.1 Steps for frequency domain filtering

1. We take the input image  $f(x, y)$  and apply preprocessing.
2. After preprocessing we apply Fourier transform to convert  $f(x, y) \rightarrow F(u, v)$ .
3. When we have obtained image in frequency domain then we apply the respective filter function  $H(u, v)$  in this case it will be Butterworth Low pass filter or Gaussian Low pass filter.
4. We get  $G(u, v)$  but we require filtered image in spatial domain because original image is in spatial so we do Inverse Fourier transform.
5. At last we apply post processing i.e. opposite of preprocessing.

Figure 1 describes how we perform frequency domain filtering.

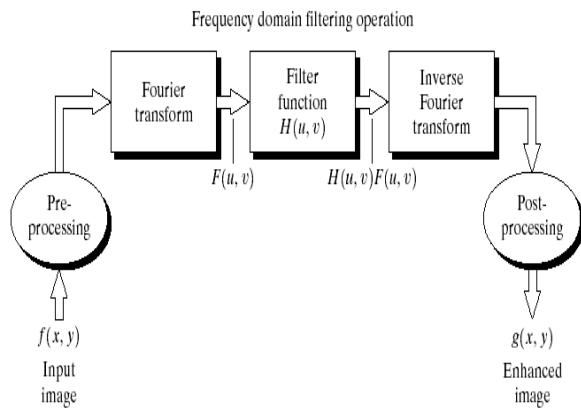


Figure 1: Various Steps Involved in Frequency Domain Process

### 3. LOW PASS FILTERING

This process is also known as smoothing. Low pass filter in frequency domain are of three types: Ideal Low pass Filter, Butterworth Low pass Filter and Gaussian Low pass Filter. Firstly we used Ideal low pass filter then it does not produces desired result for all cut off frequency, so Butterworth low pass filter was introduced but it produces desired result but up to certain limit at last we used Gaussian low pass filter which gives best result.

This paper includes discussion about Butterworth and Gaussian filter and changes that occur on filtering if we change certain values of filter.

If it is required to put the average value of an image  $f(x, y)$  to zero. This can be achieved by multiplying each value of  $F(u, v)$  by filter function given below:

$$H(u, v) = \begin{cases} 0 & \text{if } (u, v) = (\frac{M}{2}, \frac{N}{2}) \\ 1 & \text{else where} \end{cases}$$

Such a filter is referred to as Notch filter because it is constant filter with a hole at the origin

#### 3.1 Ideal Low Pass Filter

This is the simplest of three filters. This filter completely depends upon cut off frequency. The filter cuts off all high frequency components of image those are greater than a specified distance  $D_0$  the cut off frequency.

The filter is given by the function  $H(u, v)$  as:

$$H(u, v) = \begin{cases} 0 & \text{if } D(u, v) \leq D_0 \\ 1 & \text{if } D(u, v) > D_0 \end{cases}$$

Where  $D_0$  = cut off frequency given as input by user and  $D(u, v)$  = distance from the point  $(u, v)$  to the origin.

$$D(u, v) = \sqrt{[(u - M/2)^2 + (v - N/2)^2]}$$

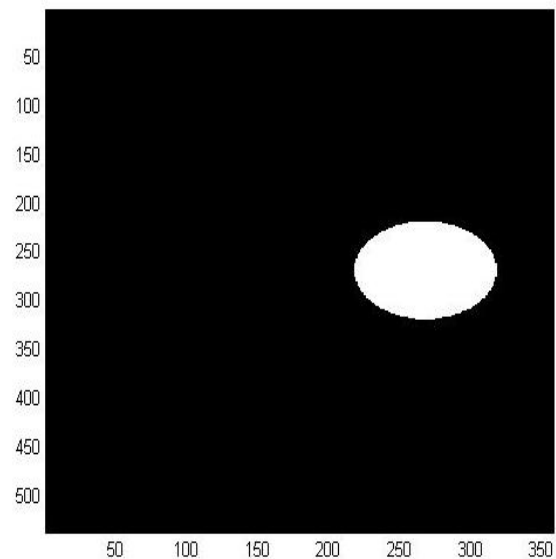


Figure 2: 2 – D ideal low pass filter

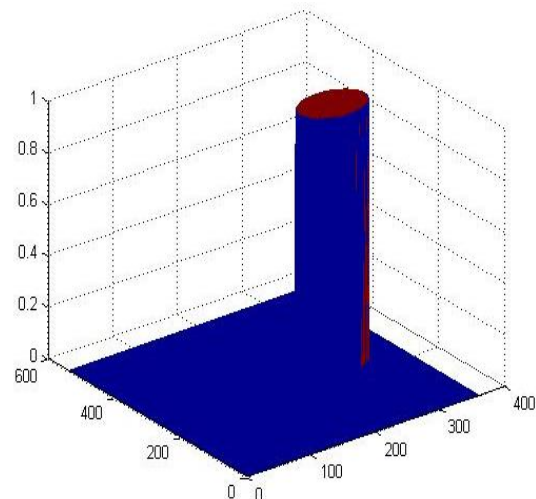


Figure 3: Ideal Low Pass Filter with Frequency Response

Though ideal filters can be implemented easily but are not used normally. The reason behind it is severe ringing effect and blurring of the image. In ringing effects the filter function  $H(u, v)$  is convolve to obtain  $h(x, y)$  and when  $h(x, y)$  is displayed as an image we see dominant object at origin and concentric components about the center. So we get ringing effect and blurring of the image.

The study establishes standard cut-off frequency loci using circles

Though ideal filters can be implemented easily but are not used normally. The reason behind it is severe ringing effect and blurring of the image. In ringing effects the filter function  $H(u, v)$  is convolve to obtain  $h(x, y)$  and when  $h(x, y)$  is displayed as an image we see dominant object at origin and concentric components about the center. So we get ringing effect and blurring of the image.

The study establishes standard cut-off frequency loci using circles

#### 3.2 Butterworth Low Pass Filter

To eliminate ringing effect we have to remove the sharp cut offs in frequency domain. So Butterworth filter are used they

remove the ringing effect but up to a certain limit. Let us know discuss this filter and what happens if we modify some values in this filter

$$H(u, v) = \frac{1}{1 + \left[\frac{D(u, v)}{D_0}\right]^{2n}}$$

Where  $D_0$  = cut off frequency given as input by user and  $n$  = order of filter

$D(u, v)$  = distance from the point  $(u, v)$  to the origin.

$$D(u, v) = \sqrt{[(u - M/2)^2 + (v - N/2)^2]}$$

Where  $M * N$  is the image matrix.

For a particular code and image the 2 – D filter and filter with frequency response are given below in figure 4 and 5

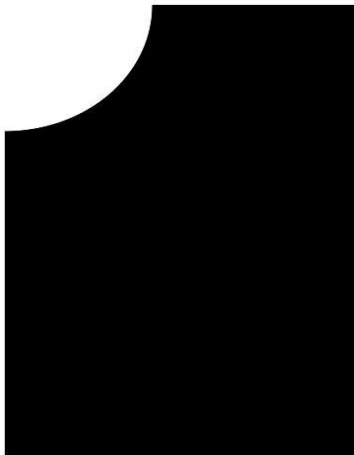


Figure 4: 2 –D Butterworth filter

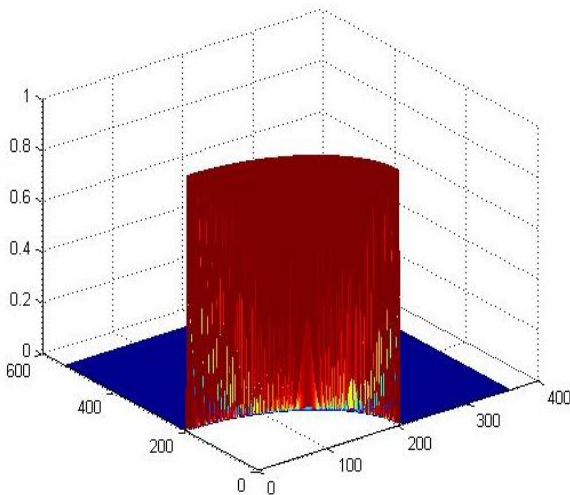


Figure 5: Butterworth filter with frequency response



Figure 6: Original Image without filtering



Figure 7: Image after filtering

### 3.3 Adaptation in Butterworth Low pass filter

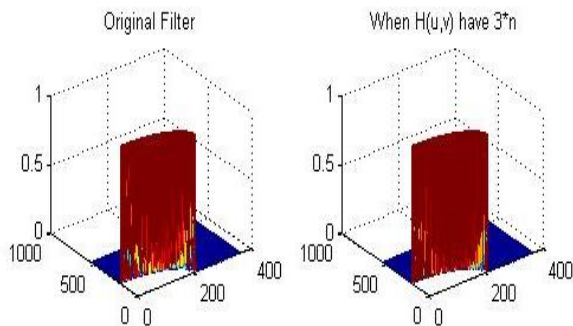
Original Butterworth low pass filter have  $2*n$  in the formula of  $H(u, v)$  where  $n$  is order what happens if we change the formula in to  $3*n$ ,  $4*n$  and  $5*n$  respectively. Changes that occur in  $H(u, v)$  and modifications that happen to the result all this will be discussed below:

When order is taken 2 and cut off frequency is 250 the results are as follows:

Image when H(u,v) have 2\*n



Image when H(u,v) have 3\*n



**Figure 8: Order of filter is 2 and cut off frequency is 250 this show what happens to result and the spreadness of the curve**

Original Filter



When H(u,v) have 3\*n



**Figure 9: Show the 2 –D view of Butterworth low pass filter for original function and for 3\*n value.**

From the above computation we observed that when we take 3\*n instead of 2\*n the spreadness of curve decreases, i.e. area of computation over the image decreases which will keep on decreasing if we take 4\*n, 5\*n.... etc. In Butterworth low pass filter does not have sharp discontinuities so ringing effect is not present, but if we keep on increasing the order to 6, 7, 8, 9 the ringing effect do creeps in so we introduce third filter i.e. Gaussian Low Pass Filter.

### 3.4 Gaussian Low Pass Filter

The disadvantages of Butterworth Low Pass Filter are removed in Gaussian filter i.e. no ringing effect at all. Here we do not consider the order of filter but we use 'σ' i.e. standard deviation.

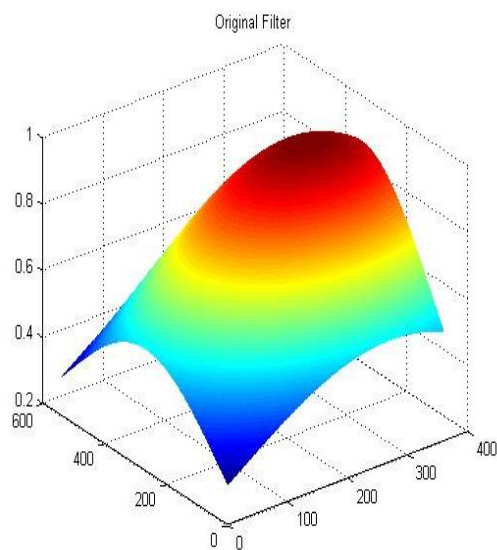
$$H(u, v) = e^{-\frac{D^2(u, v)}{2D_0^2}}$$

Where  $D_0$  = cut off frequency given as input by user  $D(u, v)$  = distance from the point  $(u, v)$  to the origin.

$$D(u, v) = \sqrt{[(u - M/2)^2 + (v - N/2)^2]} \quad \text{Where } M * N \text{ is the image matrix.}$$

If  $D_0 = \sigma$  then the formula becomes  $H(u, v) = e^{-D^2(u, v)/2\sigma^2}$  where  $\sigma$  is the standard deviation and measure of spreadness of the Gaussian curve.

Gaussian filter is a bell shaped function whose value lies in the range of  $-\frac{1}{\sqrt{2\pi}\sigma}$  to  $\frac{1}{\sqrt{2\pi}\sigma}$ . In Gaussian Low Pass Filter there occurs no ringing effect. This filter produces best quality image after low pass filtering. Below are the figures of Gaussian low pass filter:



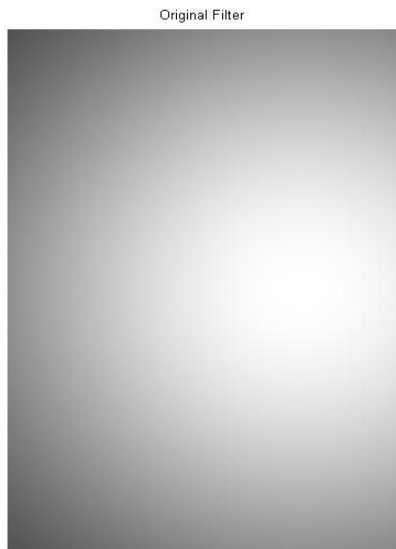


Figure 10: Shows the mesh of Gaussian filter at various values and original Gaussian filter function



Figure 11: Shows the Gaussian Low Pass Filtered Image

### 3.5 Adaptation in Gaussian Low Pass Filter

If we change the function value from  $2 \cdot D_0^2$  to  $3 \cdot D_0^2$  then the result will have slight changes in it. And these changes are listed below:

1. The spreadness of the curve appears to be more in the latter one as compared to the original one.
2. The Gaussian filter of latter is larger in area than the original one.

Images with changes are listed below

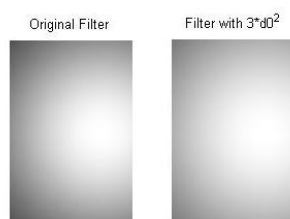


Figure 12: Show filter function of original filter and  $3 \cdot D_0^2$

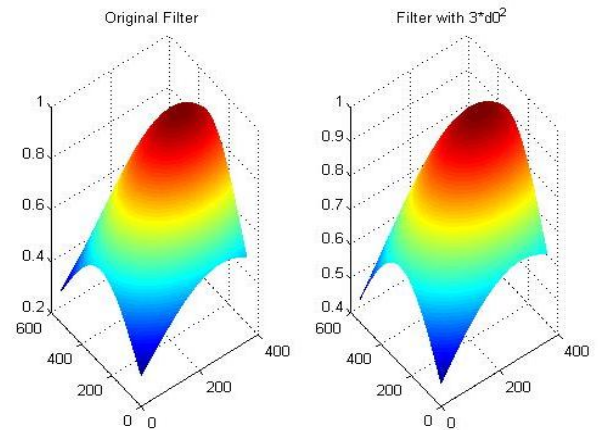


Figure 13: Show various frequency response of original filter and  $3 \cdot D_0^2$

Gaussian Low Pass Image



Gaussian Low Pass Image with  $3 \cdot D_0^2$



Figure 14: Resultant Image when cut off frequency is 250 for both original and  $3 \cdot D_0^2$

## 4. CONCLUSION

In this paper we have studied about various smoothing filter there advantages, disadvantages and working along with slight modification in these filters and their response to image and we conclude that the adaptation in both the filter i.e. Butterworth low pass filter and Gaussian low pass filter it doesn't make any difference in result neither increases the quality nor decreases the quality.

## 5. REFERENCES

- [1] Umbaugh Scot E, Computer Vision and Image Processing, Prentice Hall, NJ, 1988, ISBN 0-13-2645998
- [2] Digital Image Processing, 3<sup>rd</sup> Edition, Prentice Hall, R.C.Gonzales, R.E.Woods.
- [3] 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.
- [4] "Computer Vision", page 130, 137. Prentice Hall, 2001 by Shapiro, L. G. & Stockman, G. C
- [5] 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

[6] Gonzales R., Woods R., and Eddins S., Digital Image Processing Using Matlab, 2<sup>nd</sup> Edition, Prentice Hall, USA, 2003.