Chapter 04: Image Enhancement in the Frequency Domain

Table of Contents

[Fourier Transform 2](#_Toc138876565)

[Discrete Fourier Transform 5](#_Toc138876566)

[Periodicity 7](#_Toc138876567)

[Conjugate Symmetry 9](#_Toc138876568)

[Filtering 13](#_Toc138876569)

[Notch Filter 14](#_Toc138876570)

[Lowpass and Highpass Filtering 15](#_Toc138876571)

[Converting Between Spatial and Frequency Domain Filters 17](#_Toc138876572)

[Low Pass Filters 19](#_Toc138876573)

[Ideal Filters 19](#_Toc138876574)

[Ringing Effect 22](#_Toc138876575)

[Butterworth Filter 23](#_Toc138876576)

[Gaussian Filter 26](#_Toc138876577)

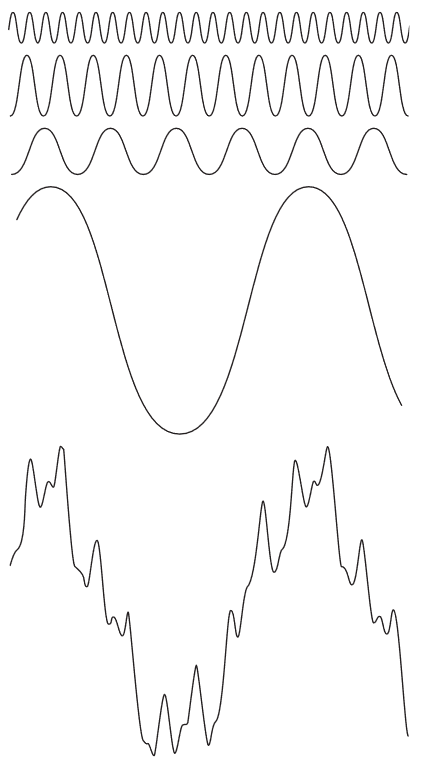
[Padding 27](#_Toc138876578)

[Highpass Filters 28](#_Toc138876579)

All the processing techniques we have seen so far have been in the **spatial domain**. It is, however, possible to transform images to move them to the frequency domain, called the **transform domain**. Certain processing techniques become easier if we make this shift.

## Fourier Transform

The specific type of transformation we will be studying is called the **Fourier Transform**. The Fourier theorem says that a periodic function can be represented as a weighted sum of sines and cosines of different frequencies. This is called the Fourier Series.



**Non-periodic functions** can also be represented as the integral of sines and cosines multiplied by a weighing function. The only requirement is that the area beneath the curve for the function be finite. This is called the Fourier Transform. Images are basically 2D non-periodic functions.

If is the non-periodic function we are dealing with, then the Fourier Transform is given by

where .

From the Euler formula, . Here, . is called the **frequency variable**.

To go back to the spatial domain, we can invert the equation above.

The pair of equations above is called the Fourier transform pair.

For two variables, as in images, the pair looks like this:

The values of or are **complex numbers**, meaning they have the form . However, it is difficult to graphically represent a complex number, so we display the **magnitude** or **Fourier Spectrum** instead.

We can also use **polar coordinates** to represent the function.

where . is called the **phase angle**.

The square of the spectrum is called the **power spectrum** of .

Suppose we have a 1D signal and its corresponding Fourier Spectrum.



In the diagram above, refers to the image size. From the pixels, of them have an amplitude of . This graph is given by the equation .

If the value of is doubled, the Fourier Spectrum changes.



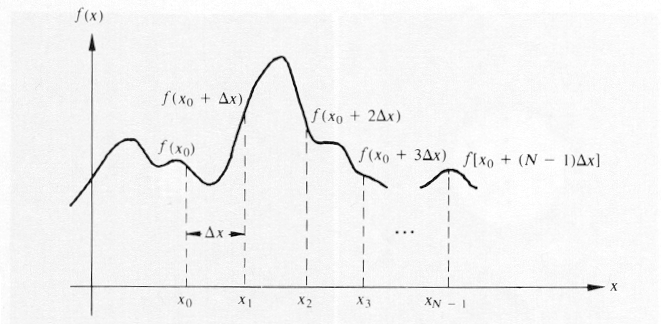
The centre of the Fourier Transform shows the highest value, which is equal to the **average intensity** of the image. The fact that this value occurs at means that this value has a frequency. Thus, this is the **DC component** of the image.

In the graph, we also see some smaller non-zero values occurring at regular intervals. If there are points in the image which have non-zero values, the graph will have a non-zero value at every interval.

## Discrete Fourier Transform

Digital images have **discrete signals**, so instead of integration, we should be using a summation. The continuous function can be discretized into the following sequence:

Thus, we have samples, each units apart.



Thus, .

From this, we can re-create the Fourier transform function pair:

For two dimensions, this becomes:

Both and (and their one-dimensional counterparts) are **complex values**, meaning they have a real and an imaginary part. If we want to display any image, we have to compute the **magnitude** of the values and display that instead.

Notice that the value of works out to , meaning the value at the is the **average intensity**.

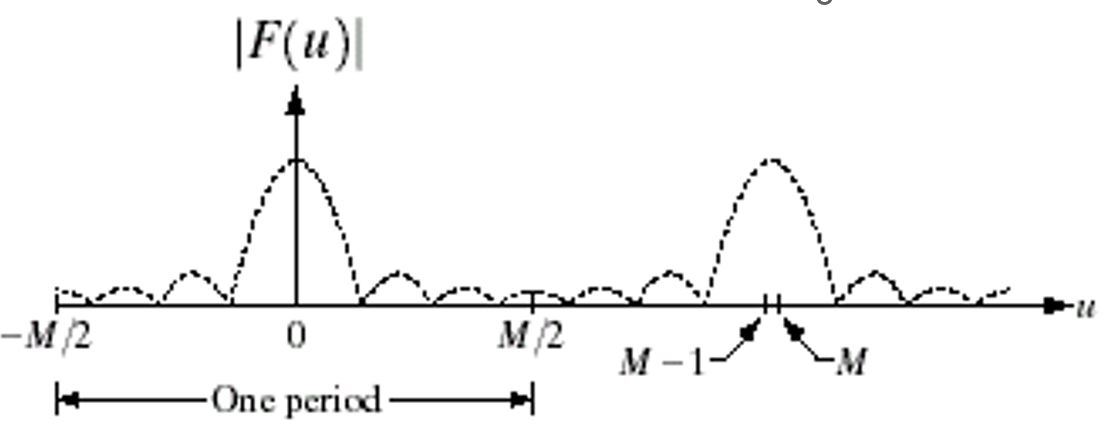
## Periodicity

The values of range from to . Since is the measure of frequency, the range is from low-frequency pixels to high-frequency pixels. is said to have a **period** of .

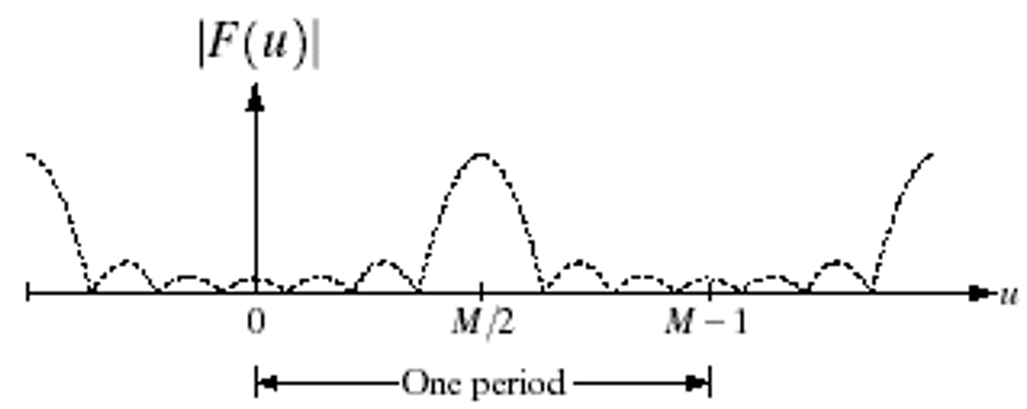
If we try to go beyond this range, then the values will loop around, meaning , and so on. Thus, , which is why is the period.

The same applies for the negative direction, i.e., .

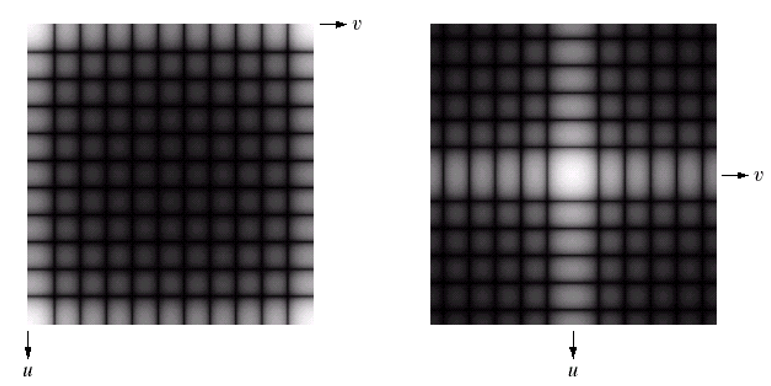
Normally, with values ranging from to , we would have a Fourier transform graph like this:



Because of the periodicity property, we have a **repeating pattern**. This allows us to **shift** the signal and make it range from to . This does not cause us to lose any information.



In two-dimensions, this is seen as moving the parts with the lowest frequencies from the edges to the centre.



Visualizing this in four sections makes it clear that there is simply an exchange between sections and and sections and .



The reason we make this shift is to bring the **low-frequencies** closer to the origin. These values fluctuate less, which makes them easier to deal with. Having them close to the origin is useful to work with.

The reason we can make this shift at all without causing ourselves any headaches later is because **this is not an image**. This is just a set of spectrum values. As long as we are not losing any information, we will be able to get back the exact same image as we had before.

Since we are making these shifts, it is important to remember the difference between an **index** and a **frequency index**. The index values still range from to , but the frequency index values range from to . Thus, the value at index is the value of frequency index .

A common way to perform the shift is to multiply the image in the spatial domain with , i.e.,

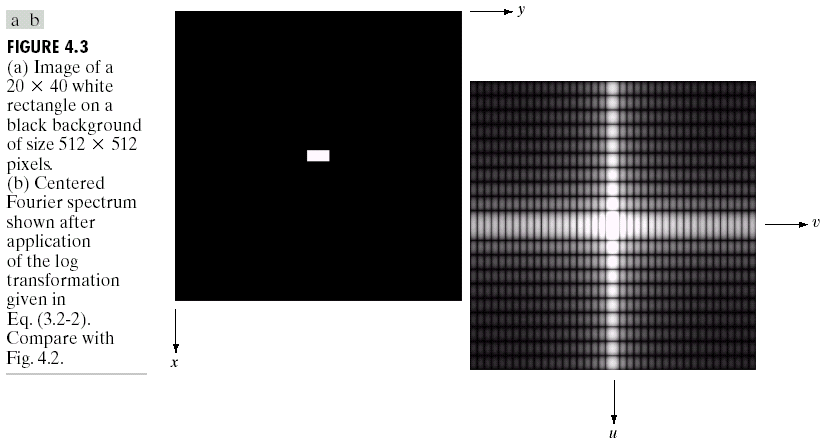
We will not be looking into the mathematics behind why this works.

## Conjugate Symmetry

The **complex conjugate** of a complex number is . In terms of Fourier transform values, . Practically, this means .

Example

The images below show a 2D image and the corresponding Fourier transform (shifted).



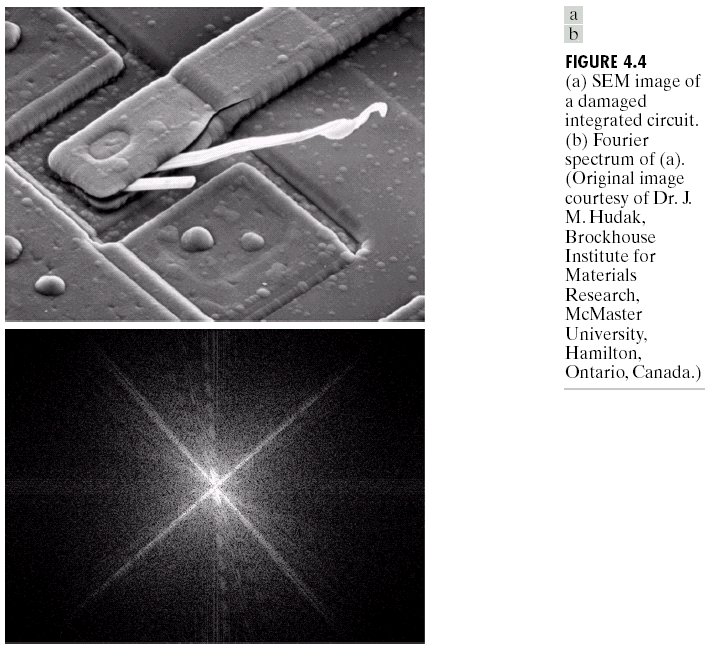
Notice the black **horizontal lines** along the direction. These correspond to the zero-values we saw in the Fourier transform graph. This means that if we have a larger number of non-zero values in our image, the **spacing** between the lines will decrease. Since the image above is longer in the direction, the **vertical lines** along the direction have a smaller amount of space between them.





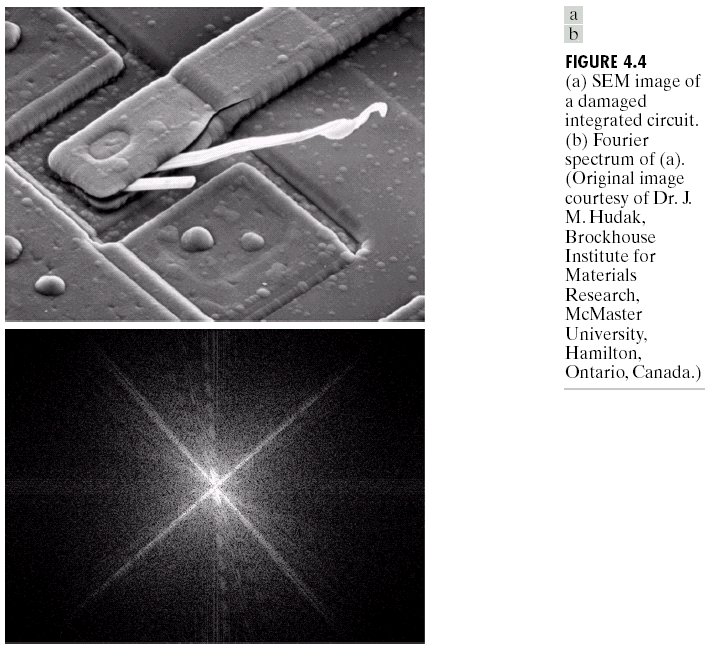
Moreover, we can also see that the central region is brighter than the other regions. This also corresponds to the graph.

In the actual image, we have edges along the vertical and horizontal directions. This causes the strong responses along those directions in the Fourier transform. Thus, we are also able to obtain some **edge information**. If the edges were at a angle, we would see a strong response along the angle. Thus, the direction of the response is rotated by . A more complicated image should make this more obvious.



We can also analyse some information about the **smooth regions**. The intensities are changing slowly in the smooth regions, meaning they have a **low frequency**. The edges by contrast have a fast change in intensity, so they have a **high frequency**.

We will now start examining different ways in which we can manipulate images in the frequency domain.



In the image above, we can see that there are some white defects in roughly the horizontal direction. The corresponding response in the Fourier transform can be seen in roughly the vertical direction.

## Filtering

The basic steps to perform filtering in the frequency domain are as follows:

1. Multiply the image by to centre the transform to and (assuming and are even numbers).
2. Compute .
3. Multiply by a **filter function** .
4. Compute .
5. Obtain the real part of the results.
6. Multiply the results by to cancel out the effects of step 1.

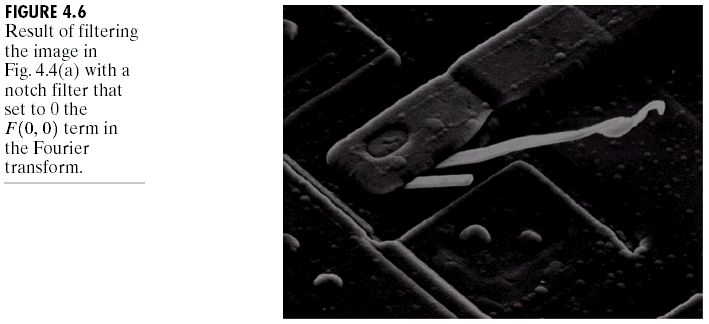
Notice that, unlike in the spatial domain, the filter is not performing a convolution operation. Instead, we do simple **element-wise multiplication**. This requires the filter to be of size .

Since we are doing element-wise multiplication, every pixel in the input affects every pixel in the output, unlike in the spatial domain.

## Notch Filter

The **notch filter** has a 1 everywhere except at the centre (index ). Thus, a 3x3 filter would look like this:

The result of will thus remove the value at the centre. This removes most of the smooth areas from the image but **retains edge information**. All of the low frequencies are not removed.

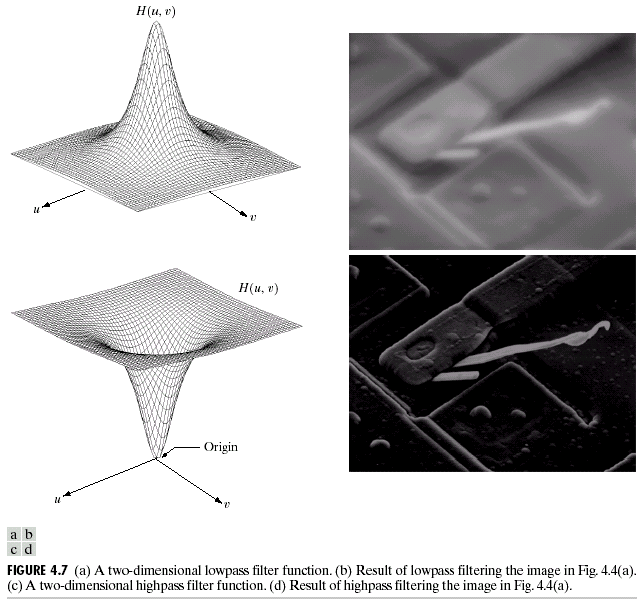


If we make the central value , theoretically, the average intensity of the image should be . However, this cannot be displayed, since it would require **negative grey levels**. Instead, we just display the magnitude.

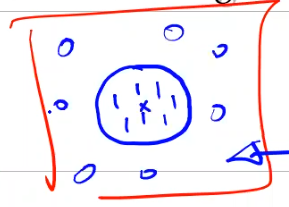
## Lowpass and Highpass Filtering

A **lowpass filter** allows only lower frequencies to pass. Thus, smooth regions of the image remain while edge regions are removed. This causes a **blurring effect**. Thus, the lowpass filter is comparable to the average filter from the spatial domain.

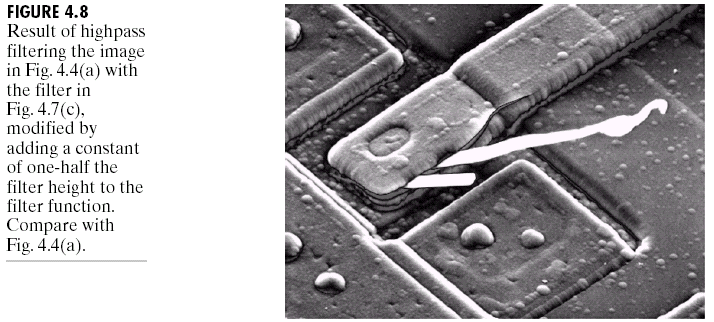
A **highpass filter** does the opposite, allowing only high frequencies to pass. Thus, edge information remains while smooth regions are removed. Note that this in itself does not cause sharpening. It only gives us the edge information. Adding this to the original image is what will give us a sharpened image. Thus, the highpass filter is comparable to the Laplacian filter from the spatial domain.



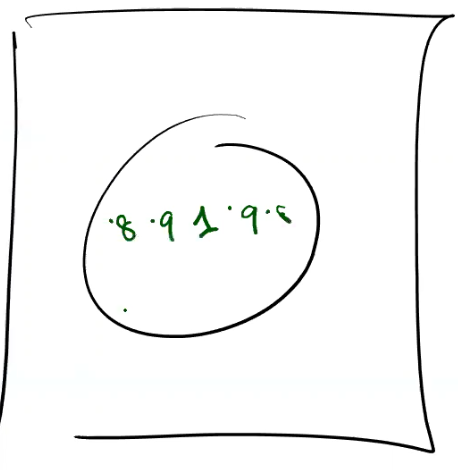
To design a filter that only allows lower frequencies to pass, we need to remember that the lower frequencies are around the centre of the Fourier transform. Thus, a filter like this would work:



For a highpass filter, the values would just be flipped. Additionally, since we need to add the edge information to the original image to obtain sharpening when using the highpass filter, it is practically the same if we start the filter values from suppose 1.5 instead of 1 in this case. On the flip side, we cannot multiply the lower frequencies with 0. Instead, we use some low value, suppose 0.5. Thus, we are adding a **constant value**, 0.5 in this case.



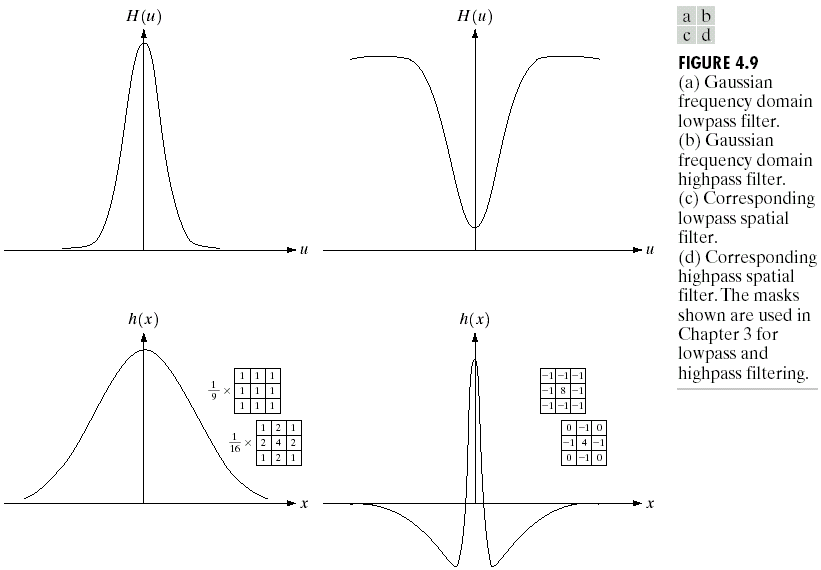
Instead of directly assigning a specific portion to 1, we can also change the values **gradually**, somewhat like a gaussian filter.



We can also have a corresponding version for a highpass filter. This is the manner in which lowpass and highpass filters are practically applied.

## Converting Between Spatial and Frequency Domain Filters

Interestingly, filters in the spatial domain and the frequency domain form pairs. This means that given a filter in the frequency domain, we can get the corresponding filters in the spatial domain by performing an inverse Fourier transform operation on it and vice versa.



Notice how the upwards translation applied to the highpass filter to achieve sharpening results in the modified form of the Laplacian filter we studied earlier that was created to achieve the same effect.

However, there is one issue. We saw that the filters in the spatial domain were all **smaller** than the image and that we had to apply convolution on the images. Converting a filter from the spatial domain the frequency domain should thus also give a filter that is smaller than the image. In the frequency domain however, we need the filter to be of the same size as the image. We have the opposite problem in the reverse direction.

These problems are fortunately automatically solved for us by the process. Converting from to results in a filter is **padded** with 0s, which we can **discard**. When converting from to , we must pad the filter with 0s.

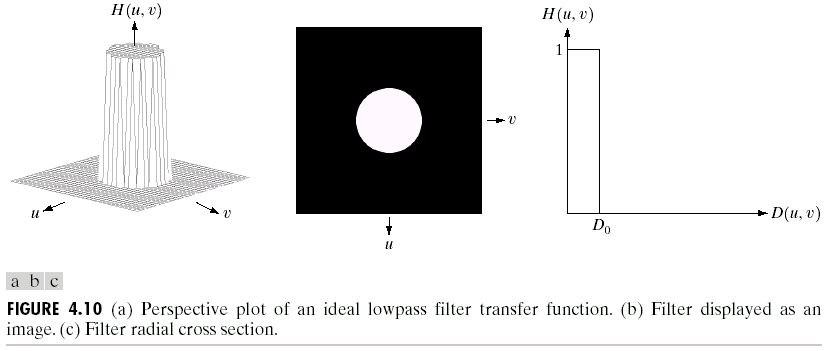
## Low Pass Filters

We will be studying three different types of low-pass filters, Ideal filters, Butterworth filters, and Gaussian filters.

### Ideal Filters

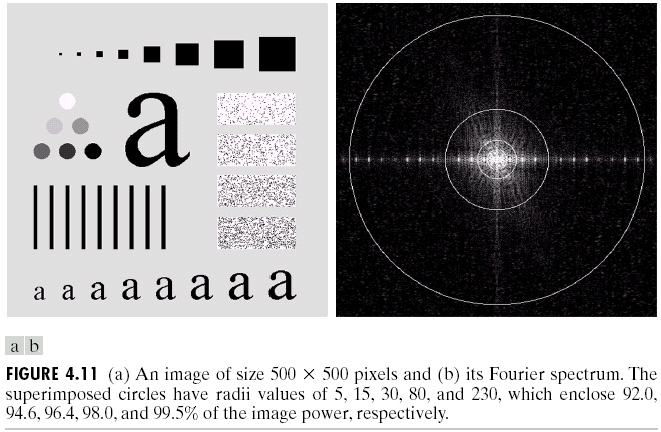
An **ideal filter** is the first filter that we saw. We pick some distance and set all pixels that are less than that distance from the centre to and all pixels that are more than that distance from the centre to .

The distance is calculated as .

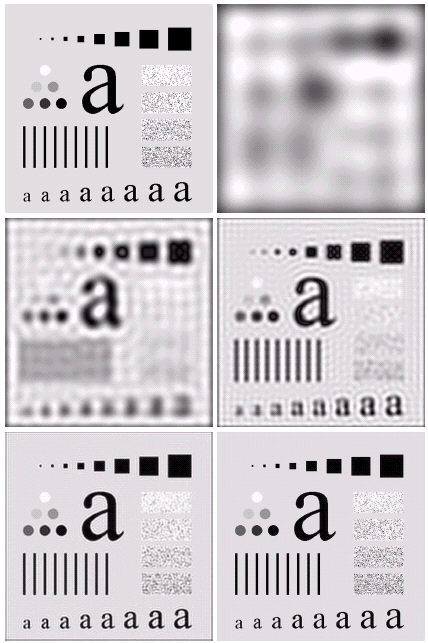


Inside the circle, there is said to be **no attenuation**, while outside it, all the pixels are said to be completely **attenuated**. The point at which this transition occurs is called the **cutoff frequency**.

The square of a spectrum value is said to be the **power** of that value. The power values are actually decreasing as we go further away from the centre. If the entire domain contains 100% of the power in the image below, the smallest circle with a radius of 5 has 92% of the power.



This means that most of the power lies in the low frequency components. The larger the cutoff frequency of our ideal filter, the more power we retain. The image is less likely to change much if we have larger cutoff frequencies.



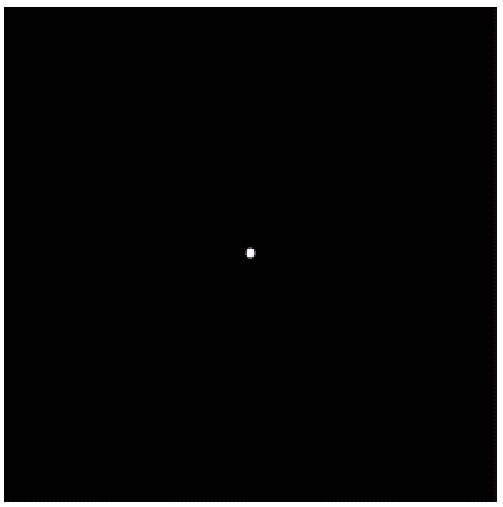
For lower cutoff frequencies, the filter in the spatial domain will be larger. For higher cutoff frequencies, the filter in the spatial domain will be smaller.

The opposite effect will be seen for highpass filters. With a larger cutoff frequency, we will lose more power while with a smaller cutoff frequency, we will lose less power.

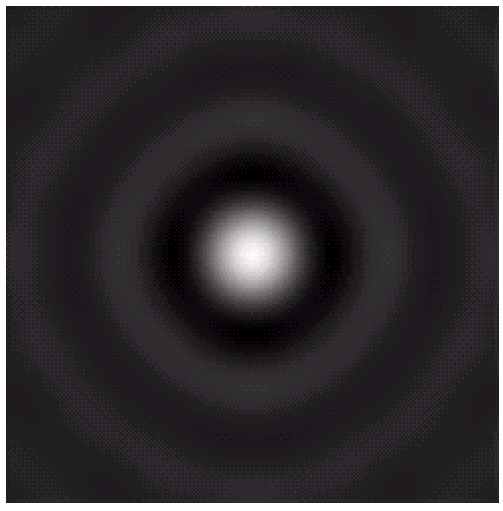
### Ringing Effect

For the smaller cutoff frequencies in the ideal lowpass filter, notice that we start to see some ripples. This effect is called the **ringing effect**.

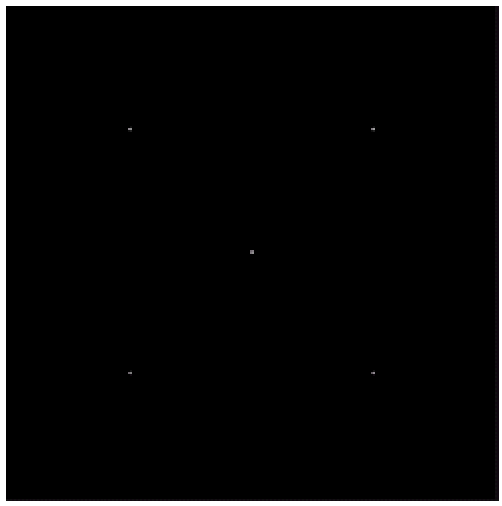
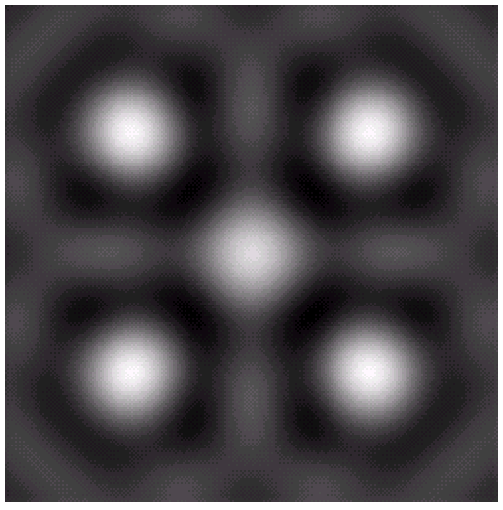
Suppose we have a filter in the frequency domain that is just a point at the centre of the image.



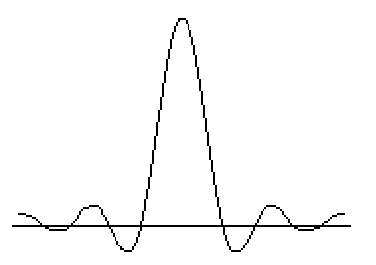
Such a filter, when converted to the spatial domain, will create an image with a ripple effect centred at that point.



If we have multiple 1-bit points on an image in the space domain and we apply the above filter on that image, each point will cause a separate ripple.

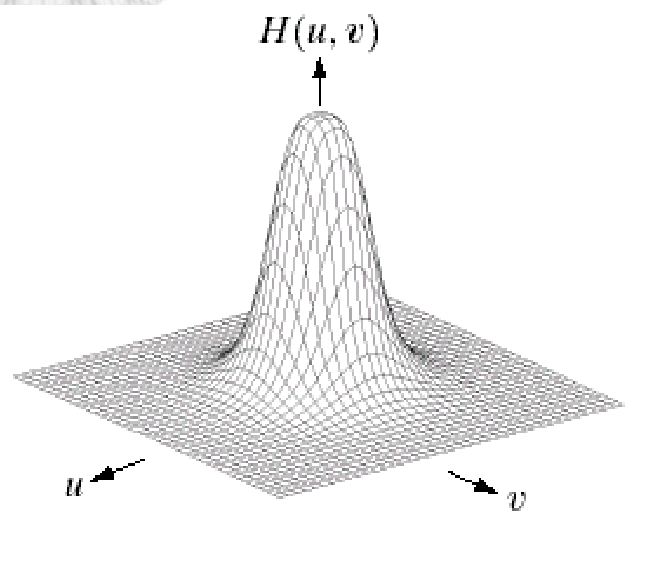
 

Taking a look at the graph for the filter converted to the spatial domain should make it clear why we have repeating patterns of high and low values (thus causing ripples).

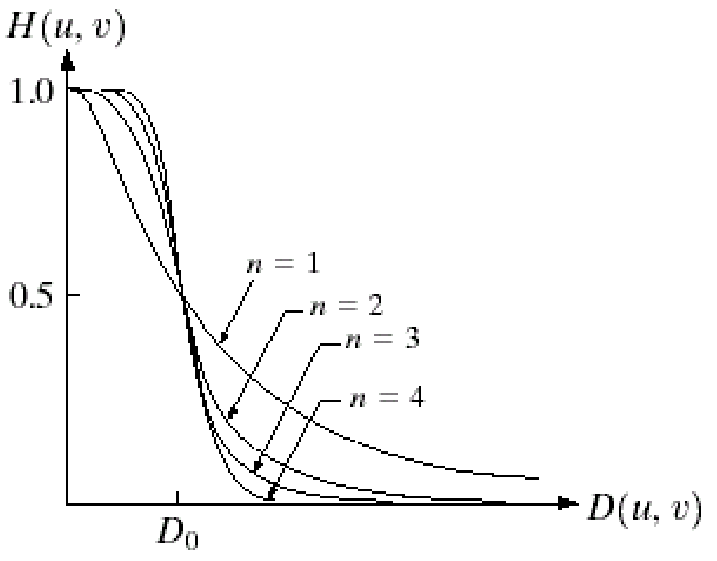


### Butterworth Filter

The **Butterworth Filter** looks quite similar to the Gaussian filter.



The difference is that this filter has an additional parameter, , which can be tuned. The cross-sectional view shows the variation of the curve as changes.

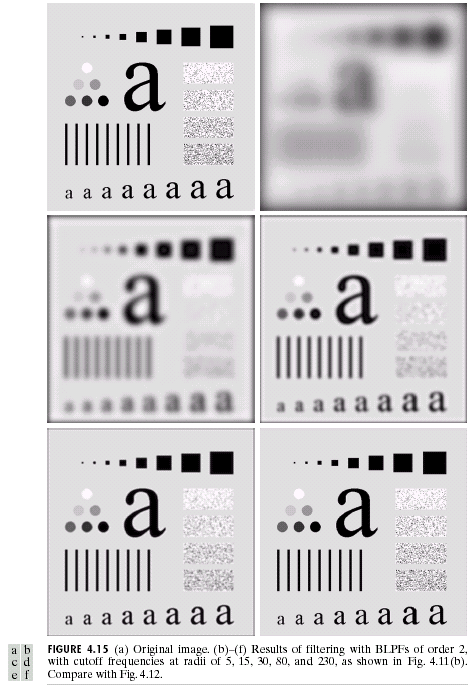


A unique trend for the Butterworth filter is that at the point where , the value of is exactly .

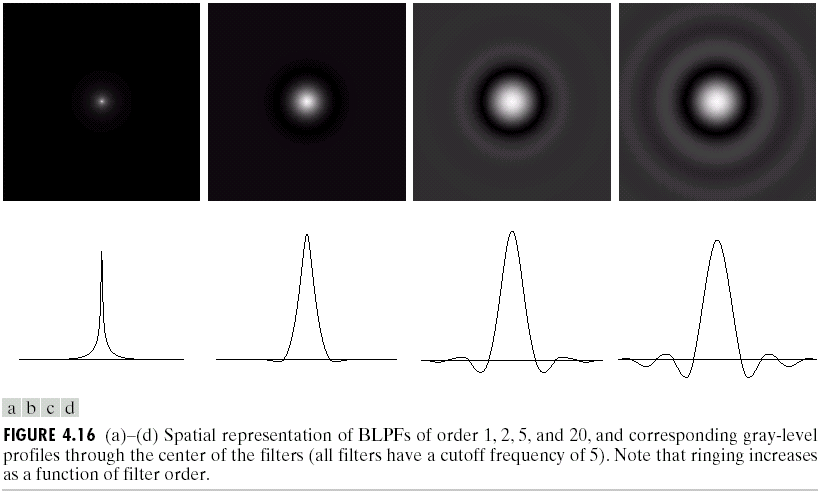
As the value of increases, the curve becomes sharper and will eventually become the same as the ideal filter.

The Butterworth filter can be represented mathematically as:

Like the Ideal filter, the Butterworth filter also shows increased blurring with a lower cutoff frequency.

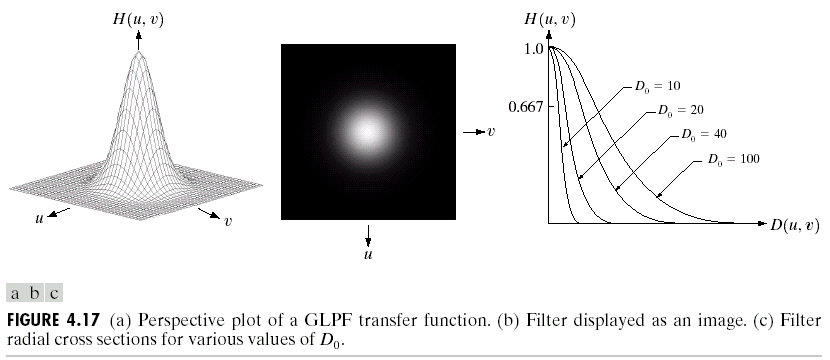


However, notice that there is no ringing effect. This is because of the gradual transition. However, if we increase the order, the transition will become sharper. Eventually, we will start seeing the ringing effect.



### Gaussian Filter

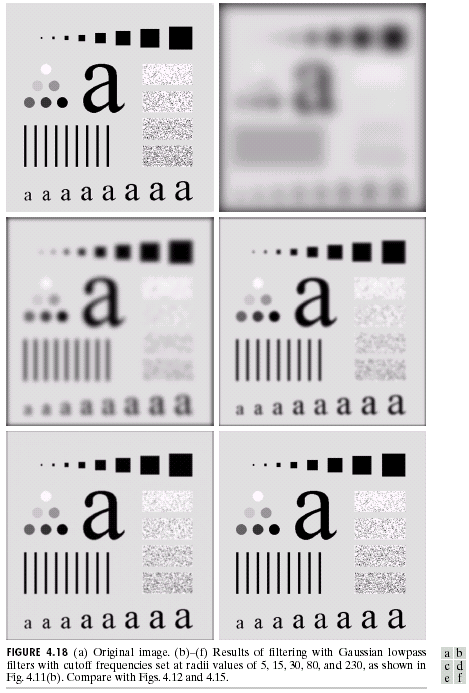
The **Gaussian Filter** varies its steepness based on where we define the value of . Similar to the Butterworth filter, the Gaussian filter also shows a unique trend. Here, at the point where , .



The mathematical representation for this filter is:

Here, is the cutoff frequency.

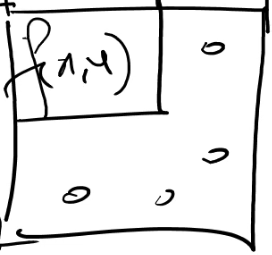
Lower values of means we have lower cutoff frequencies, which in turn makes the image blurrier.



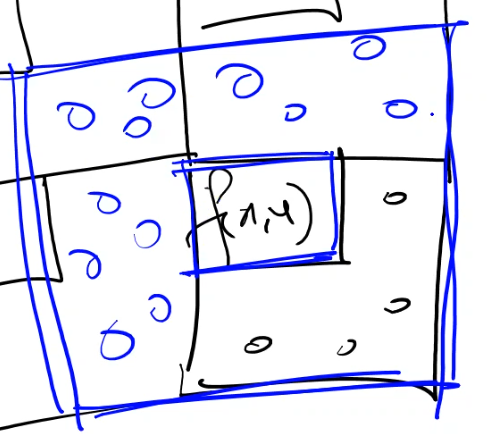
Since we cannot modify the value of , we cannot make the Gaussian filter behave like an Ideal filter. Consequently, there is no ringing effect.

## Padding

If the image we are working with requires **padding**, we cannot add 0s all around the image anymore. This is because we will be shifting the image. Instead, we add 0s to the right and the bottom.

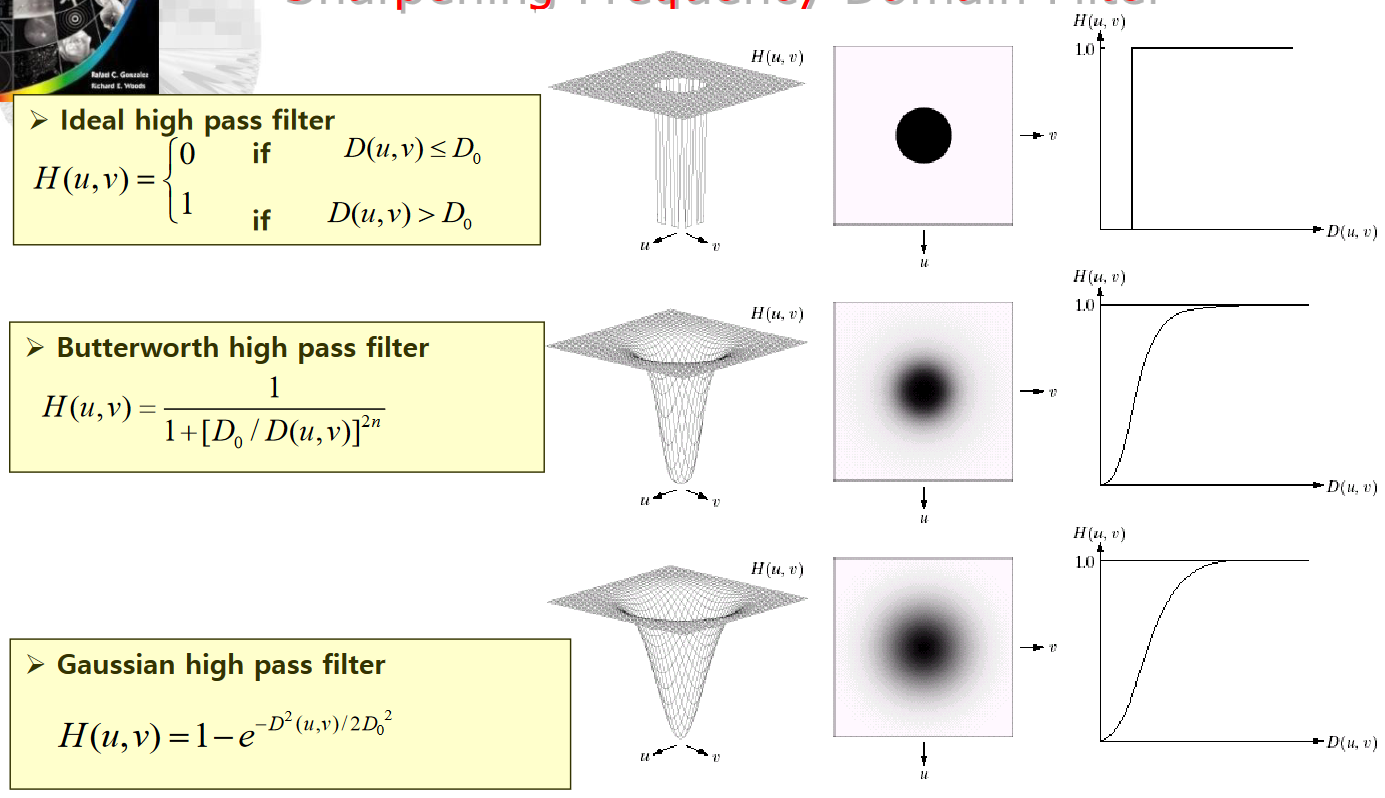


After shifting (multiplying with ), the image becomes like this:



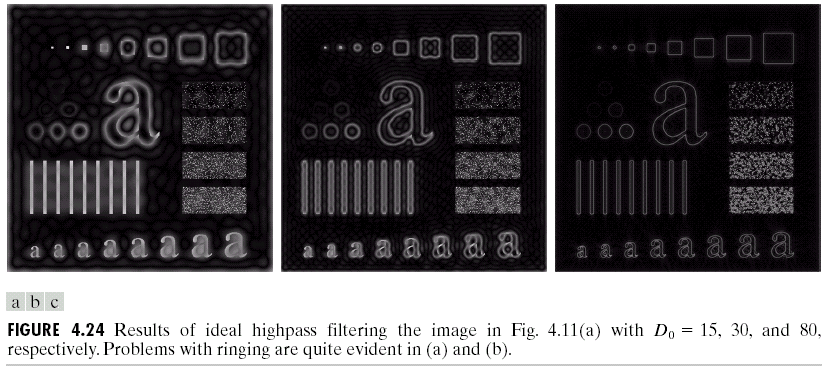
## Highpass Filters

Each of the filters that we have seen as lowpass filters can also be defined for **highpass filters**.

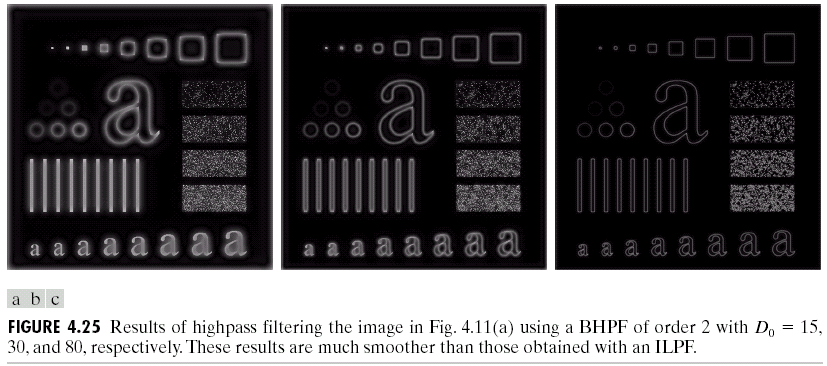


The properties for each of the filters still hold.

For the ideal filter, the ringing effect and the amount of smooth region information retained both decrease as we increase the cutoff frequency.



A similar situation occurs for the Butterworth filter. The ringing effect will be comparatively lower than that of an Ideal filter, given that we use a low value of .



The Gaussian filter achieves the same results without the risk of any ringing effect.

