Chapter : 3

Fourier Transform:

The fourier transform is an important image processing tool which is used to decompose an image into it's sin & cosin components.

The output of transformation represents an image in the fourier of frequency domain while the input image is the spatial domain equivalent.

In the fourier domain image each image represent a particular frequency contained in the spatial domain image. The fourier transform is used in a wide range of application such as image analysis, image filtering, image reconstruction & image compression,

Fourier series:

A function that periodically repeats itself can be expressed as a sum of sines and cosines of different frequencies each multiplied by a different cofficient

Fourier transform:

let $f(x)$ be a continuous function of a real variable x. The fourier transform of $f(x)$ denoted by ~~$f(x)$~~ is $\mathcal{F}\{f(x)\} = F(u)$

$$= \int_{-\infty}^{\infty} f(x) \exp[-j 2\pi u x]\, dx$$

where, $j = \sqrt{-1}$

Scanned by CamScanner

Given, $F(u)$, $f(x)$ can be obtained by using the inverse fourier transform.

$$\mathcal{F}^{-1}\{F(u)\} = f(x)$$

$$= \int_{-\infty}^{\infty} F(u) \exp(j 2\pi ux) \, dx.$$

These two eqn are called the fourier transform pair. These equations exist if $f(x)$ is continuous and integrable and $F(u)$ is integrable.

We are consider with functions $f(x)$ which are real, however fourier transform of a real function is generally complex.

$$\text{So } F(u) = R(u) + jI(u)$$

where, $R(u)$ and $I(u)$ denote the real and imaginary component of $F(u)$ resp.

In extended form, $F(u)$ is expressed as,

$$F(u) = |F(u)| \, e^{j\phi(u)}$$

$$\text{where } |F(u)| = \sqrt{R^2(u) + I^2(u)}$$

$$\text{and } \phi(u) = \tan^{-1}\left[\frac{I(u)}{R(u)}\right]$$

The magnitude function $|F(u)|$ is called the Fourier spectrum of $f(x)$ and $\phi(u)$ is the phase angle.

The ~~spect~~ square of the spectrum

$$P(u) = |F(u)|^2 = R^2(u) + I^2(u)$$

is commonly called the power spectrum (or the spectral density of $f(x)$)

2D Fourier transform:

The Fourier transform can be extended to 2 dimension,

$$\mathcal{F}\{f(x,y)\} = F(u,v)$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y) \exp[-j2\pi(ux+vy)]\,dx\,dy$$

and the inverse transform

$$\mathcal{F}^{-1}\{F(u,v)\} = f(x,y)$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(u,v) \exp[j2\pi(ux+vy)]\,du\,dv.$$

The 2D Fourier's spectrum is

$$|F(u,v)| = \sqrt{R^2(u,v) + I^2(u,v)}$$

The phase angle is $\phi(u,v) = \tan^{-1}$

The power spectrum is,

$$P(u,v) = |F(u,v)|^2$$

$$= R^2(u,v) + I^2(u,v)$$

Discrete Fourier transform

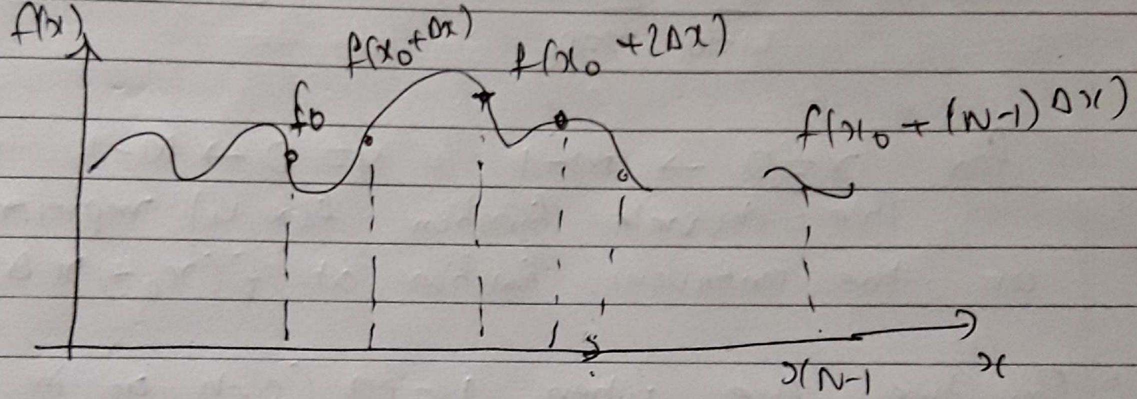
The DFT is the sampled FT and therefore doesn't contain all frequencies forming an image, but only a set of samples which is large enough to fully describe the spatial domain image. The no. of pixels in the spatial domain image.

Suppose a continuous function $f(x)$ is digitized into a sequence,

$$\{f(x_0), f(x_0 + \Delta x), f(x_0 + 2\Delta x) \dots f(x_0 + [N-1]\Delta x)\}$$

where we take N samples Δx units apart let x refer to either a continuous or discrete value, where $f(x) = f(x_0 + x\Delta x)$

Here x assumes the descrete vlue 0, 1 ... N-1 and $\{f(0), f(1) \dots f(N-1)\}$ denotes any N uniformly spaced samples from a corresponding continuous function.



discrete FT is given by

$$F(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) \exp[-j\, 2\pi\, ux/N]$$

$$u = 0, 1, \dots N-1$$

The fourier image can be re transformed to the spatial domain by taking inverse FT. The discrete inverse FT is given by,

$$f(x) = \sum_{u=0} F(u) \exp[j\, 2\pi u x/N]$$

$$x = 0, 1, \dots N-1$$

classmate

Date
Page

In the 2D case,

$$F(u,v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \exp\left[-j 2\pi \left(ux/M + vy/N\right)\right]$$

for $u = 0 \to M-1$ & $v = 0 \to N-1$

$$f(x,y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u,v) \exp\left[j 2\pi \left(ux/M + vy/N\right)\right]$$

for $x = 0 \to M-1$ & $y = 0 \to N-1$

The discrete function $f(x,y)$ represents samples of the continuous function at $f(x_0 + x\Delta x, y_0 + y\Delta y)$

For the case when $N = M$ (such as in a sequence image)

$$F(u,v) = \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) \exp\left[-j 2\pi (ux + vy)/N\right]$$

and

$$f(x,y) = \frac{1}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} F(u,v) \exp\left(j 2\pi (ux + vy)/N\right]$$

* The basis function are sines & cosines waves with increasing frequencies.

$F(0,0)$ represent the DC component of image which correspond to average brightness and $F(N-1, N-1)$ represent the highest frequency. In most implementation $F(0,0)$ is displayed at the center of image. The further away from the center with higher frequency components.

The result show that the image contains components of all frequencies, but that their magnitude gets smaller for higher frequencies. Hence, low frequency components contain more image information than higher ones.

- As we move away from origin of transforms the low frequency correspond to slowly varying correspond to slowly varying components of an image and the high frequencies being to correspond to faster gray level changes in the image. These are the edges of object components characterized by abrupt change in gray level.

~~For 2D case~~

Properties of 2D Fourier Transformation:

→ The dynamic range of Fourier spectrum is generally higher than can be displayed.

→ A common technique is to display the function.

$$D(u,v) = c \log [1 + |F(u,v)|]$$

~~which is chosen to scale the dat~~

where c is the scaling factor which is chosen to scale the data into the range of display device and the logarithm function perform a "compression" of the data.

1). Translation:

The fourier transform pair has the following translation properties.

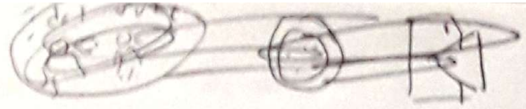
$$f(x,y)\, e^{j2\pi(u_0 x/N + v_0 y/N)} \Leftrightarrow F(u-u_0, v-v_0)$$

and,

$$f(x-x_0, y-y_0) \Leftrightarrow F(u,v)\, e^{-j2\pi(u x_0/N + v y_0/N)}$$

where the double arrow indicates a correspondance between a function and it's FT

Multiplying $f(x,y)$ by the exponential and taking the transform results in a shift of the origin of the frequency plane to the point (u_0, v_0).



For our purpose, $u_0 = v_0 = N/2$

Thus,

$$\exp[j2\pi(u_0 x + v_0 y)/N] = e^{j\pi(x+y)} = (-1)^{x+y}$$

and $f(x,y)(-1)^{x+y} \Longleftrightarrow F(u - N/2, v - N/2)$

So the origin of FT of $f(x,y)$ can be moved to the center of the corresponding $N \times N$ simply by multiplying $f(x,y)$ by $(-1)^{x+y}$ before taking transform.

(2) Distributivity and Scaling:

The fourier tranform (and it's inverse) are distributive over addition but not over multiplication.

So,

$$\mathcal{F}\{f_1(x,y) + f_2(x,y)\} = \mathcal{F}\{f_1(x,y)\} + \mathcal{F}\{f_2(x,y)\}$$

$$\mathcal{F}\{f_1(x,y) * f_2(x,y)\} \neq \mathcal{F}\{f_1(x,y)\} * \mathcal{F}\{f_2(x,y)\}$$

For two scalars a and b,

$$a f(x,y) \longleftrightarrow a F(u,v)$$

$$f(ax, by) \longleftrightarrow \frac{1}{|ab|} F(u/a, v/b)$$

(3) Separability:

we have the discrete FT of a function/image $f(x,y)$ of size $M \times N$ is given by,

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

It can be expressed in separable form as,

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

$$= \frac{1}{M}\sum_{x=0}^{M-1} F(x,v)\, e^{-j2\pi ux/M}$$

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

It tells us that we can compute the 2D transform by first computing a 1D transform along each row of the input image and then compute a 1D transform along each column of this intermediate result.

$$\boxed{f(x,y)} \xrightarrow[\text{1D row transforms}]{} \boxed{F(x,v)} \xrightarrow[\text{1D column transform}]{} \boxed{F(u,v)}$$

Fig:- computation of the 2D Fourier transform as a series of 1D transform.

4. Conjugate Symmetry of the fourier transform.
If $f(x,y)$ is real, the FT exhibits conjugate symmetry.

$$F(u,v) = F^*(-u,-v)$$
or, $$|F(u,v)| = |F(-u,-v)|$$

where $F^*(u,v)$ is the complex conjugate of $F(u,v)$.

(5) Periodicity of FT

The discrete Fourier transform (and it's inverse) are periodic with period N.

$$F(u,v) = F(u+N,v) = F(u,v+N) = F(u+N,v+N)$$

Altough $F(u,v)$ repeats itself infinitely for many values of u and v, only N values of each variable are require to obtain $f(x,y)$ from $F(u,v)$

→ only one period of transform is necessary to specify $F(u,v)$ in the frequency domain.

Filtering in frequency domain:

Frequency domain is the ~~specific~~ space defined by FT and it's frequency variables, (u,v).

→ The discrete FT of a function $f(x,y)$ of size $M \times N$ is given by the equation.

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

Here, each term of $F(u,v)$ contains values of $f(x,y)$ modified by the values of exponential terms.

Steps:-

1). Multiply the input image by $(-1)^{x+y}$ to center the transform.

$$\mathcal{F}\left[f(x,y)(-1)^{x+y}\right] = F(u - M/2, v - N/2)$$

In other words, multiplying $f(x,y)$ by $(-1)^{x+y}$ shifts the origin of $F(u,v)$ to frequency coordinates $(M/2, N/2)$ which is the center of $M \times N$ area occupied by 2D DFT

(2) Compute $F(u,v)$, the DFT of the image from 1.

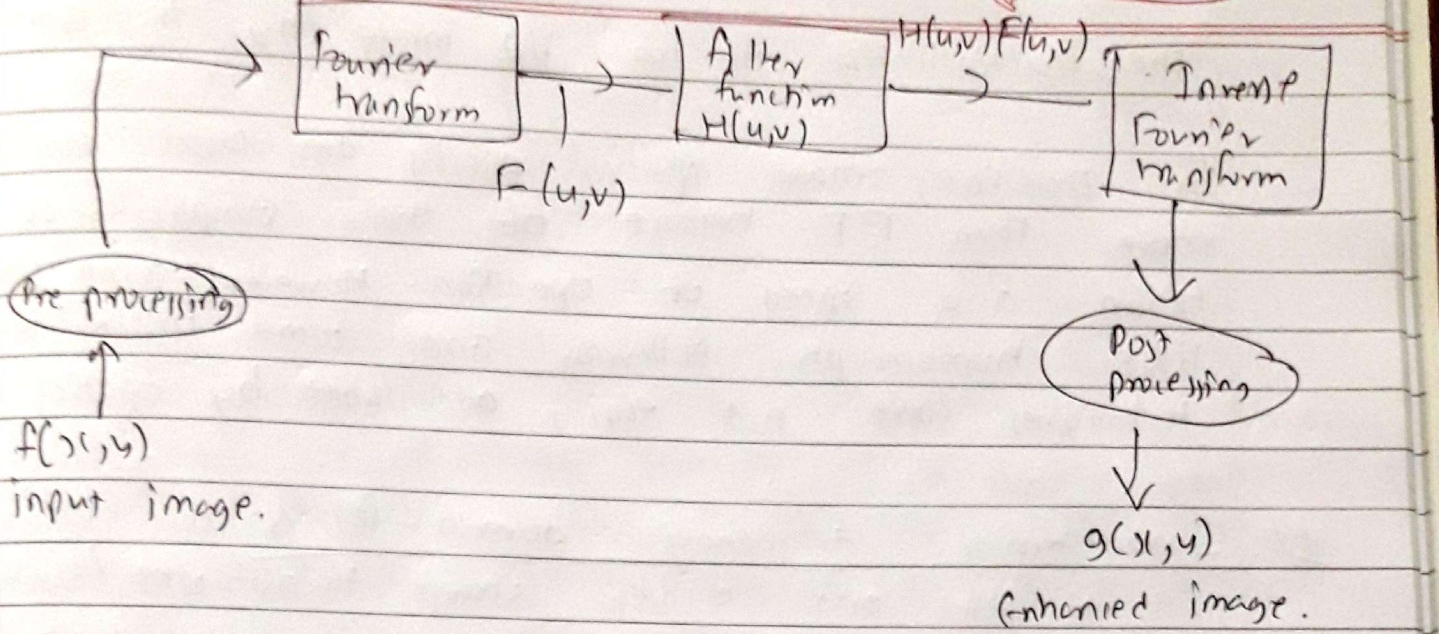
(3) Multiply $F(u,v)$ by a filter function $H(u,v)$.

Here, $H(u,v)$ is called filter as it suppresses certain frequencies in the transform while leaving others unchanged.

4) Compute the inverse DFT of the result in (3)

(5) Obtain the real part of the image in 4.

(6) Multiply the result in 5 by $(-)^{x+y}$



Let $f(x,y)$ represent the i/p image in step 1 and $F(u,v)$ it's Fourier transform of an image to be smoothed.

Then the FT of o/p image is given by,

$$G(u,v) = H(u,v)\,F(u,v)$$

where each component of H multiplies both the real and imaginary parts of the corresponding components in F. Such filters are called zero phase shift filters. These filters do not change the phase of the transform.

→ The filtered image is obtained simply by taking the inverse FT of $G(u,v)$. Filtered image

$$= \mathcal{F}^{-1}[G(u,v)].$$

→ The final image is obtained by taking the real part of the result and multiplying by $(-1)^{x+y}$ to cancel the multiplication of the i/p image by this quantity.

In practice, small spatial masks are used considerably more than FT because of their simplicity of implementation and speed of operation. However, some problems like homomorphic filtering and some image restoration techniques are not easily addressable by spatial techniques.

II. Smoothing frequency domain filters:

Edges and other sharp transitions (such as noise) in the gray levels of an image contribute significantly to the high frequency components of it's fourier transform.

→ Hence, smoothing is acheived in the frequency domain by attenuating a specified range of high frequency components in the transform of a given image.

we have,

$$G(u,v) = H(u,v)\, F(u,v)$$

where, $F(u,v)$ is the FT of image to be smoothed.

The objective is to select a filter transfer function $H(u,v)$ that yields $G(u,v)$ by attenuating the high frequency components of $F(u,v)$

Scanned by CamScanner

classmate

Date

Page

Types of filters:

→ ideal
→ Butterworth
→ Gaussian

#. Ideal low pass filters:

The ideal low pass filter rejects all high frequency components of the FT that are at a distance greater than a specified distance D_0 from the origin of the (centered) transform. It has the transfer function,

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

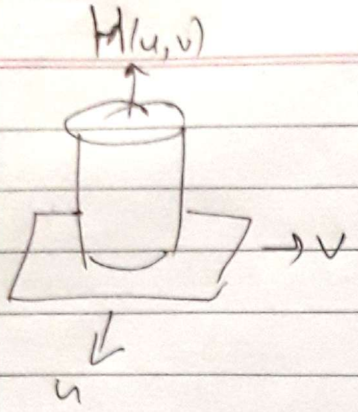
where D_0 is the specified non negative quantity and $D(u,v)$ is the distance from point (u,v) to the origin.

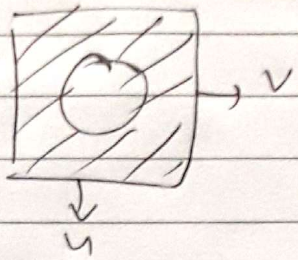
If the image is of size $M \times N$, then it's transform is also of this size, so the center of frequency rectangle is at $(u,v) = (M/2, N/2)$.

In this case, the distance of any point (u,v) to the center (origin) of the FT is given by,

$$D(u,v) = \left[\left(u - \frac{M}{2}\right)^2 + \left(v - \frac{N}{2}\right)^2 \right]^{1/2}$$

$$\left(\frac{1}{1+(\omega_N)^{2N}}\right)$$





radial cross section.

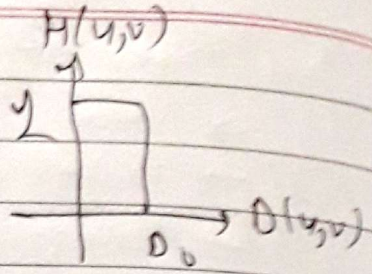


fig - ideal LPF

Here all frequencies inside a circle of radius D_0 are passed with no attenuation whereas all frequencies outside this circle are completely attenuated.

#. Butterworth lowpass filter:

The transfer function of a Butterworth LPF of order n and with cutoff frequency at a distance D_0 from the origin is difined as,

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

where, $D(u,v) = \left[\left(u-M/2\right)^2 + \left(v-N/2\right)^2\right]^{1/2}$

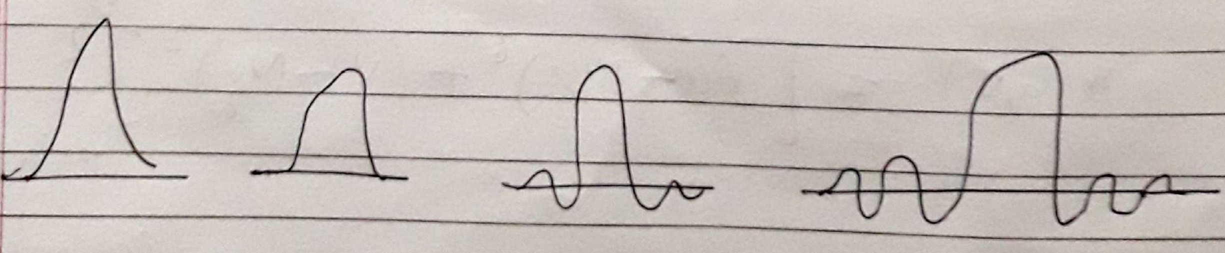


fig: Spatial representation of BLPF of order 1, 2, 5 & 20.

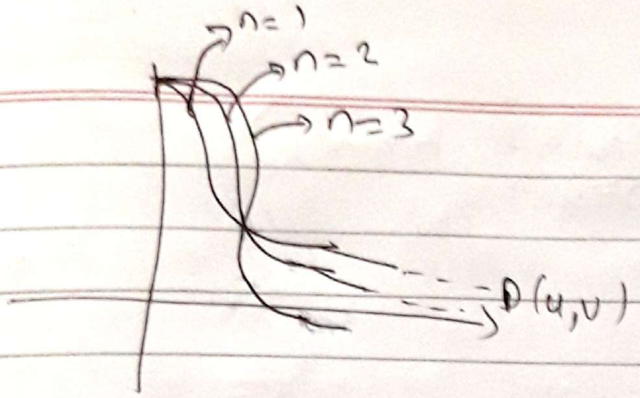


fig: butterworth filter response.

- The BLPF of order 1 has neither ringing not negative values.
- The filter of order 2 show mild ringing and small negative values but they certainly are less pronounced than in the ILPF.
- Ringing in the BLPF becomes significant for higher order filters.
- Butterworth filter of order 20 nearly exhibits the characteristics of ILPF.

Gaussian low pass filters:

The transfer function of a Gaussian low pass filter is defined as,

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

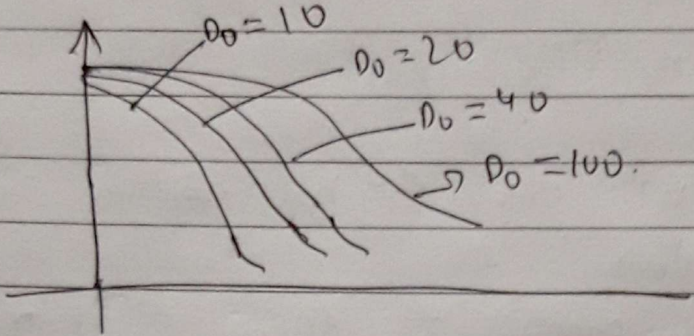
where,

$D(u,v)$ is the distance from origin of fourier transform.

σ is a measure of the spread of Gaussian curve.

By letting $\sigma = D_0$ (cutoff frequency)

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



A spatial Gaussian filter obtained by computing the inverse FT of these equations.

→ Gaussian filters have no ringing effect.

→ The three filters cover the range from very sharp (ideal) to very smooth (Gaussian) filter functions. The Butterworth filter has a parameter called the filter order.

→ For high values of this parameter the Butterworth filter approaches the form of the ideal filter. For low order values, the Butterworth filter has a smooth form similar to the Gaussian filter.

Scanned by CamScanner

Examples:

→ Text of poor resolution - when one encounters the characters having distorted shapes due to lack of resolution and if many characters are broken (which is very difficult for machine recognition system to read) then we bridge small gaps in the input image by blurring it using low pass filter which attenuate high frequency components of it's Fourier transform.

→ low pass filtering is a staple in the printing and publishing industry, where it is used for numerous preprocessing functions.

→ we can produce smoother softer looking result from a sharp original image.

Sharpening frequency domain filters:

Edges and other abrupt changes in gray levels are associated with high frequency components, hence image sharpening can be achieved in the frequency domain by a high pass filtering process which attenuates the low frequency components without disturbing high frequency information in the Fourier transform.

→ we consider only zero phase shift filters that are radially symmetric,

→ The function of high pass filter is to perform the reverse operation of the ideal low pass filters. The transfer function of the high pass filter is given by,

$$H_{hp}(u, v) = 1 - H_{lp}(u, v)$$

~~the~~ ideal high pass filter:

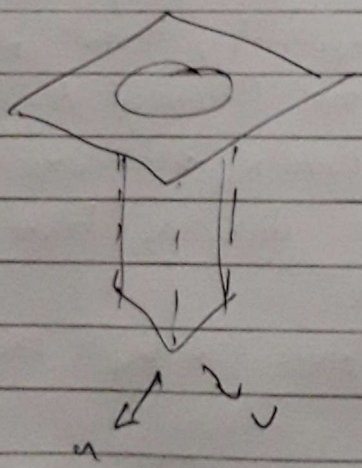
The ideal high pass filter is given as:

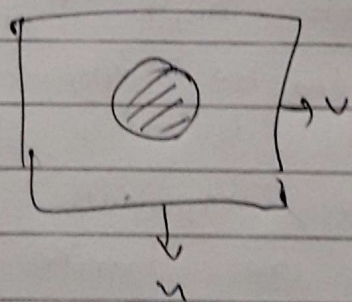
$$H(u,v) = \begin{cases} 0 & \text{if } D(u,v) \leq D_0 \\ 1 & \text{if } D(u,v) > D_0 \end{cases}$$

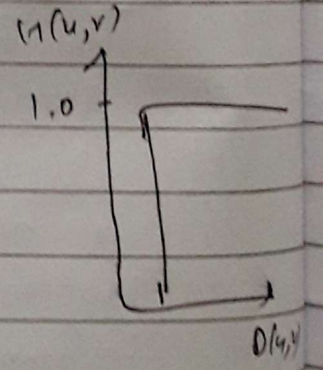
where, D_0 is the cut off distance as before.

- It sets to zero for all frequencies inside a circle of radius D_0 while passes all frequencies outside the circle without attenuation.

As in case of ideal low pass filter, the IHPF is not physically realizable with electronic components.



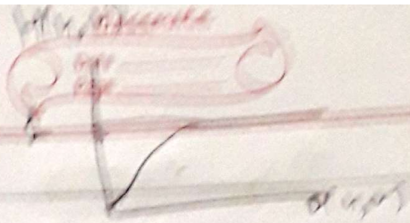




Butterworth High Pass filters:

The Butterworth high pass filter is given as:

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



where n is the order and D_0 is the cut off distance as before.

→ As in case of low pass filter, we can expect BHPF to behave smoother than IHPFs.

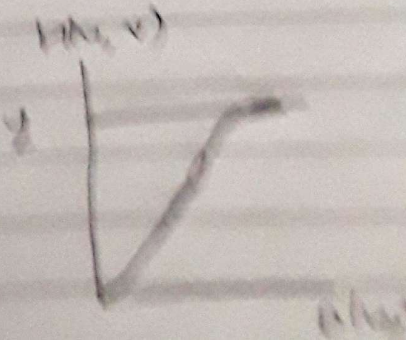
Gaussian high pass filter:

The Gaussian high pass filter is given as:

$$H(u,v) = 1 - e^{-D^2(u,v)\ 2D_0^2}$$

where D_0 is the cut off distance as before.

The results obtained are smoother than the previous two filters. Even the filtering of the smaller objects and thin bars is cleaner with Gaussian filter.



4. Frequency Domain Filtering & Spatial Domain Filtering:

Similar jobs can be done in the spatial and frequency domains.

Filtering in the spatial domain can be easier to understand.

Filtering in the frequency domain can be much faster - especially for large images.