Name : Sarvagya Singh **SAPID : 60009200030** **Batch : K1**

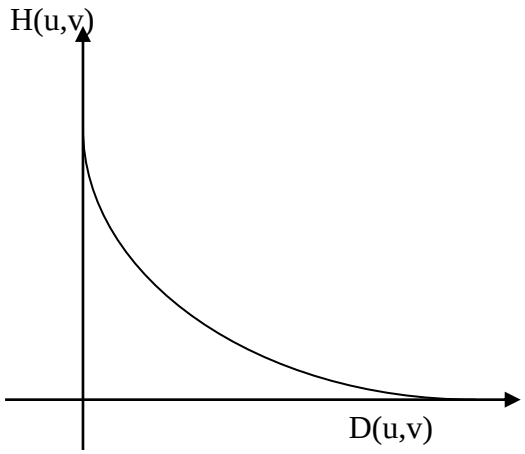
AIM: To implement Gaussian Low Pass & High Pass Filtering techniques on an image

THEORY:

1. Gaussian Low Pass Filter

Gaussian LPF is given by,

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



Where, σ is the standard deviation and is a measure of spread of the Gaussian curve. If we put $\sigma = D_0$ we get,

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

The response of the Gaussian LPF is similar to that of BLPF but there are no ringing effects.

2. Gaussian High Pass Filter

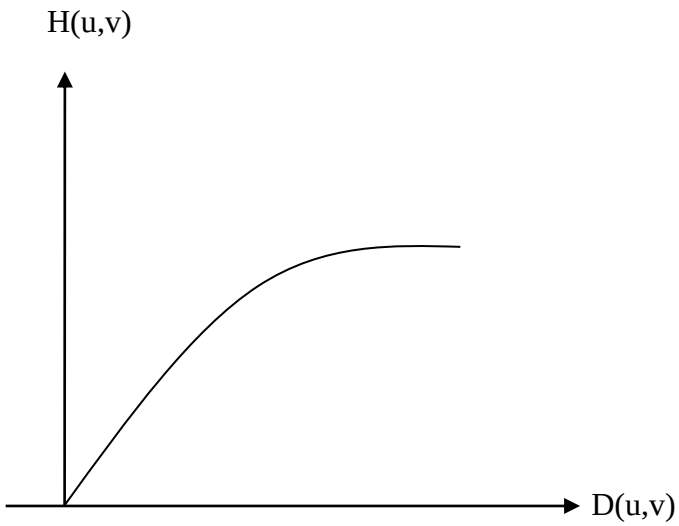
The basic formula is,

$$H\ hp\ (u,v) = 1- H\ lp\ (u,v)$$

Therefore ,

$$H\ Gaussian\ hp\ (u,v) = 1- H\ Gaussian\ lp\ (u,v)$$

$$H\ GHPF = 1- e^{\wedge} -D^2 (u,v)/2D0^2$$



The results of Gaussian high pass filter are smoother and also cleaner.

RESULT:

The Gaussian filters are generally used to reduced the size of the image thereby blurring the image. When downsampling an image, it is common to apply a low-pass filter to the image prior to resampling. This is to ensure that spurious high-frequency information does not appear in the downsampled image.

Ideal Gaussian low pass filter

Sarvayga Singh -- 60009200030 (K1)

Lab9

In [1]:
```
import cv2
import numpy as np
from google.colab.patches import cv2_imshow
import matplotlib.pyplot as plt
```

In [2]:
```
img = cv2.imread('/content/image.png',0)
# cv2_imshow(img)
plt.imshow(img, cmap='gray')
plt.axis('off')
plt.show()
```

In [6]:
```
m,n = img.shape[0],img.shape[1]
m,n
```

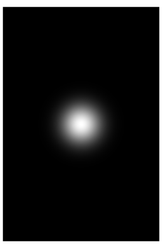
Out[6]:
```
(900, 600)
```

In [4]:
```
np.max(img),np.min(img)
```

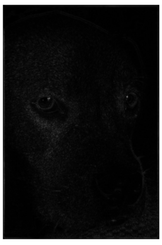
Out[4]:
```
(255, 0)
```

In [5]:
```
def dist(u,v):
  return np.sqrt((u - M/2)**2+(v-N/2)**2)
```

In [7]:
```
D0 = 50
F = np.fft.fft2(img)
Fshift = np.fft.fftshift(F)
H = np.zeros((m,n), dtype=np.float32)
for u in range(m):
    for v in range(n):
        D = np.sqrt((u-m/2)**2 + (v-n/2)**2)
        H[u,v] = np.exp(-(D**2)/(2*(D0**2)))

plt.imshow(img, cmap='gray')
plt.axis('off')
plt.show()

plt.imshow(H, cmap='gray')
plt.axis('off')
plt.show()

Gshift = Fshift * H
G = np.fft.ifftshift(Gshift)
g = np.abs(np.fft.ifft2(G))
plt.imshow(g, cmap='gray')
plt.axis('off')
plt.show()
```



In [8]:
```
# Filter: High pass filter
H = 1 - H
plt.imshow(img, cmap='gray')
plt.axis('off')
plt.show()

plt.imshow(H, cmap='gray')
plt.axis('off')
plt.show()

Gshift = Fshift * H
G = np.fft.ifftshift(Gshift)
g = np.abs(np.fft.ifft2(G))
plt.imshow(g, cmap='gray')
plt.axis('off')
plt.show()
```



In [ ]:
```
```