LAB3 REPORT

Image Processing and Application - 9804

JEWEL MAHMUD NIMUL SHAMIM

201890846

Part 1: Low pass and High pass filter design

1. Image (puppy.jpg) and file *lp_hp_filters.m* is downloaded from Brightspace under Lab3 and save them in C:\Users\Wohhie\Dropbox\University\Semester 6\Labs\lab3\ working directory.

2. Read the image and convert the image to a grayscale image. Obtain the padding parameters and FT the image.

   ```
   clc;
   clear all;
   % image to a grayscale image. Obtain the padding parameters
   F_rgb=imread('puppy.jpg'); % read the gs image
   F=rgb2gray(F_rgb);
   im_size=size(F); % Obtain the size of the image
   P=2*im_size(1);Q=2*im_size(2); % Optaining padding parameters
   FTIm=fft2(double(F),P,Q); % FT with padded size
   ```

3. Design the ILPF

   ```
   D0 = 0.1*im_size(1); %Cutoff freqency radius is 0.1 times the the hight of
   the image
   n=0; %For use only in Butterworth filters. For BTW filters, Order(n)>0
   %Filter_type=('ideal' or 'btw' or 'gaussian')
   %lp_or_hp=('lp' or 'hp' for low pass or high pass),
   Filter = lp_hp_filters('ideal','lp', P, Q, D0,n); % Calculate the LPF
   ```

4. Implement the filter by multiplying the FT of the image with filter. Undo padding.

   ```
   % multiply the FT of image by the filter and apply the IDFT
   Filtered_image=real(ifft2(Filter.*FTIm));

   % Resize the image ( undo padding)
   Filtered_image=Filtered_image(1:im_size(1), 1:im_size(2));
   ```

5. Move the origin of frequency spectrum to the center and display the results

   ```
   Fim=fftshift(FTIm); % move the origin of the FT to the center
   FTI=log(1+abs(Fim)); % compute the magnitude (log to brighten display)
   Ff=fftshift(Filter); % move the origin of the FT to the center
   FTF=log(1+abs(Ff)); % compute the magnitude (log to brighten display)


   subplot(2,2,1), imshow(F,[]), title('Original Image'); % show the image
   subplot(2,2,2), imshow(FTI,[]), title('FT of Original'); % show the image
   subplot(2,2,3), imshow(FTF,[]), title('Filter in frequency domain'); % filter
   % show the image
   subplot(2,2,4), imshow(Filtered_image,[]), title('Filtered Image');
   ```

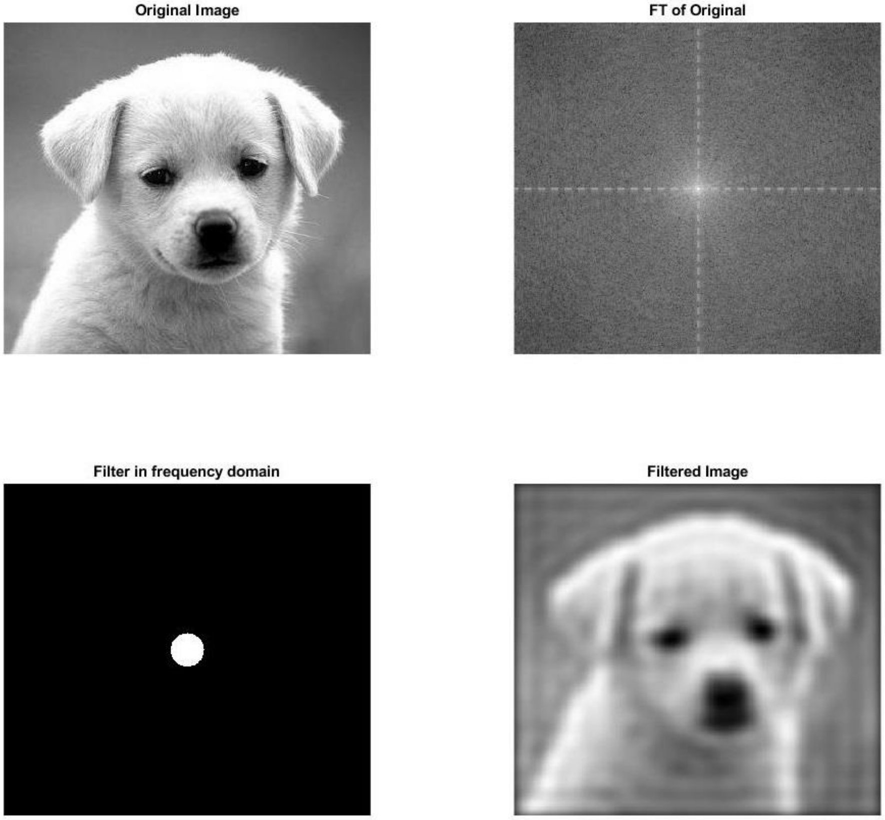


Figure 1: Result of ILPF the image

Part 2:

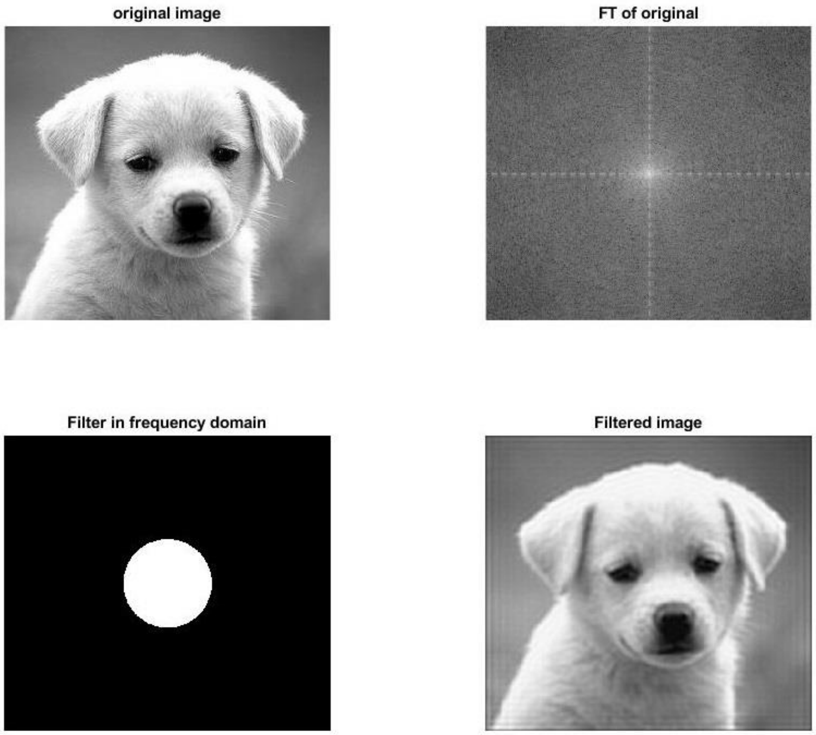
1. Implement an *Ideal* low pass filter for cutoff frequencies (0.3 and 0.7 of image height).

Ideal low pass filter for cutoff frequencies (0.3)

MATLAB Code:

```
clc;
F=imread('puppy.jpg');
F=rgb2gray(F);
im_size=size(F);
p=2*im_size(1);Q=2*im_size(2);
FTIm=fft2(double(F),p,Q);
D0= 0.3*im_size(1)
n=0
Filter=lp_hp_filters('ideal','lp',p,Q,D0,n);
Filtered_image=real(ifft2(Filter.*FTIm));
Filtered_image=Filtered_image(1:im_size(1),1:im_size(2));
Fim=fftshift(FTIm);
FTI=log(1+abs(Fim));
Ff=fftshift(Filter);
FTF=log(1+abs(Ff));
subplot(2,2,1),imshow(F,[]),title('original image');
subplot(2,2,2),imshow(FTI,[]),title('FT of original');
subplot(2,2,3),imshow(FTF,[]),title('Filter in frequency domain');
subplot(2,2,4),imshow(Filtered_image,[]),title('Filtered image');
```

Output:



original image

FT of original

Filter in frequency domain

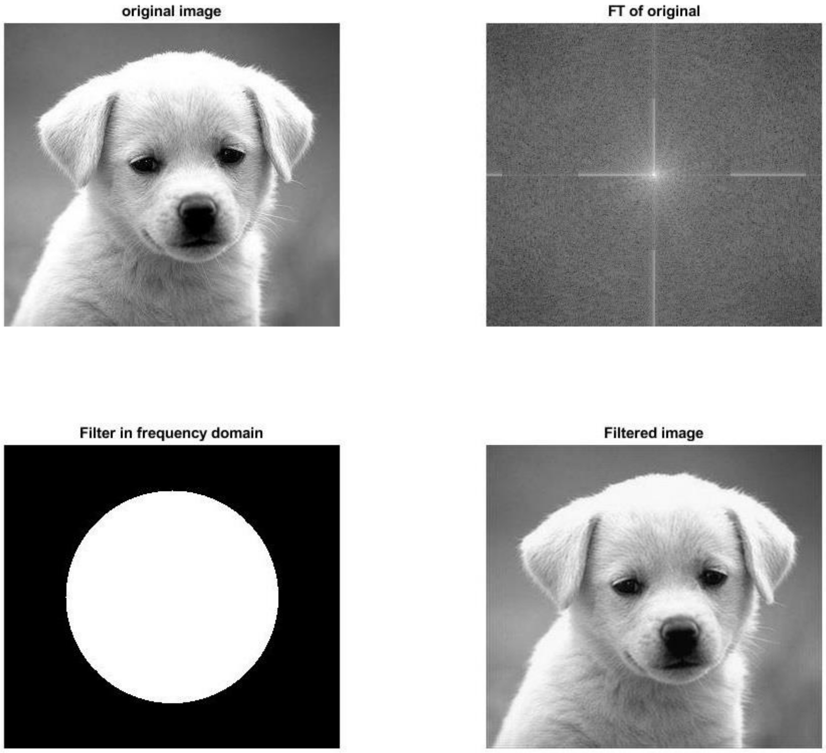
Filtered image

Ideal low pass filter for cutoff frequencies (0.7)

MATLAB Code:

```
clc;
F=imread('puppy.jpg');
F=rgb2gray(F);
im_size=size(F);
p=2*im_size(1);Q=2*im_size(2);
FTIm=fft2(double(F),p,Q);
D0= 0.7*im_size(1)
n=0
Filter=lp_hp_filters('ideal','lp',p,Q,D0,n);
Filtered_image=real(ifft2(Filter.*FTIm));
Filtered_image=Filtered_image(1:im_size(1),1:im_size(2));
Fim=fftshift(FTIm);
FTI=log(1+abs(Fim));
Ff=fftshift(Filter);
FTF=log(1+abs(Ff));
subplot(2,2,1),imshow(F,[]),title('original image');
subplot(2,2,2),imshow(FTI,[]),title('FT of original');
subplot(2,2,3),imshow(FTF,[]),title('Filter in frequency domain');
subplot(2,2,4),imshow(Filtered_image,[]),title('Filtered image');
```

Output:



original image

FT of original

Filter in frequency domain

Filtered image

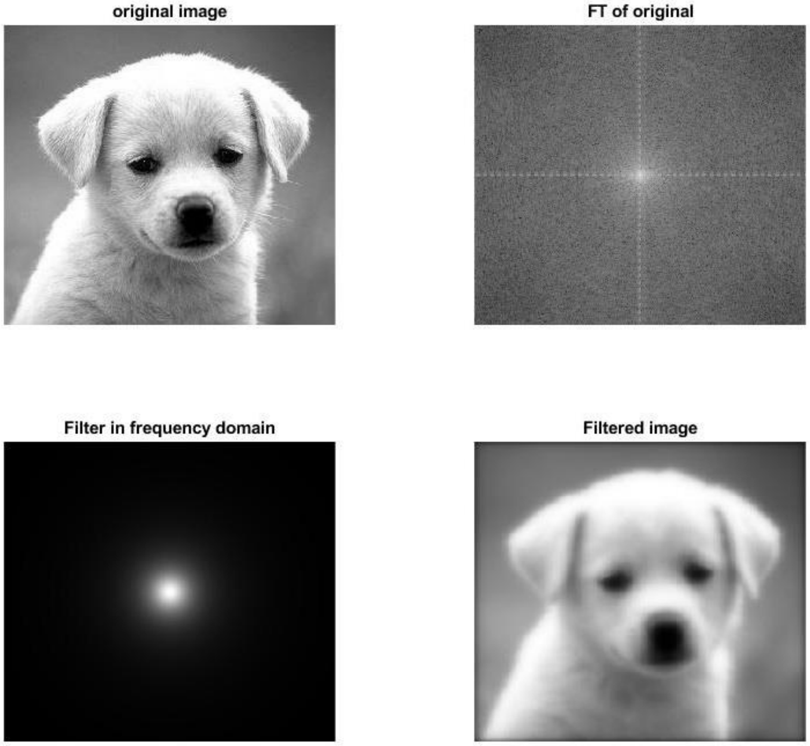
2. Implement a *Butterworth* low pass filter for cutoff frequencies (0.1 and 0.5 of image height) and order *n* = 1; 5; 20

Butterworth low pass filter for cutoff frequencies (0.1) image height and order (n = 1)

MATLAB Code:

```
clc;
F=imread('puppy.jpg');
F=rgb2gray(F);
im_size=size(F);
p=2*im_size(1);Q=2*im_size(2);
FTIm=fft2(double(F),p,Q);
D0= 0.1*im_size(1)
n=1
Filter=lp_hp_filters('btw','lp',p,Q,D0,n);
Filtered_image=real(ifft2(Filter.*FTIm));
Filtered_image=Filtered_image(1:im_size(1),1:im_size(2));
Fim=fftshift(FTIm);
FTI=log(1+abs(Fim));
Ff=fftshift(Filter);
FTF=log(1+abs(Ff));
subplot(2,2,1),imshow(F,[]),title('original image');
subplot(2,2,2),imshow(FTI,[]),title('FT of original');
subplot(2,2,3),imshow(FTF,[]),title('Filter in frequency domain');
subplot(2,2,4),imshow(Filtered_image,[]),title('Filtered image');
```

Output:

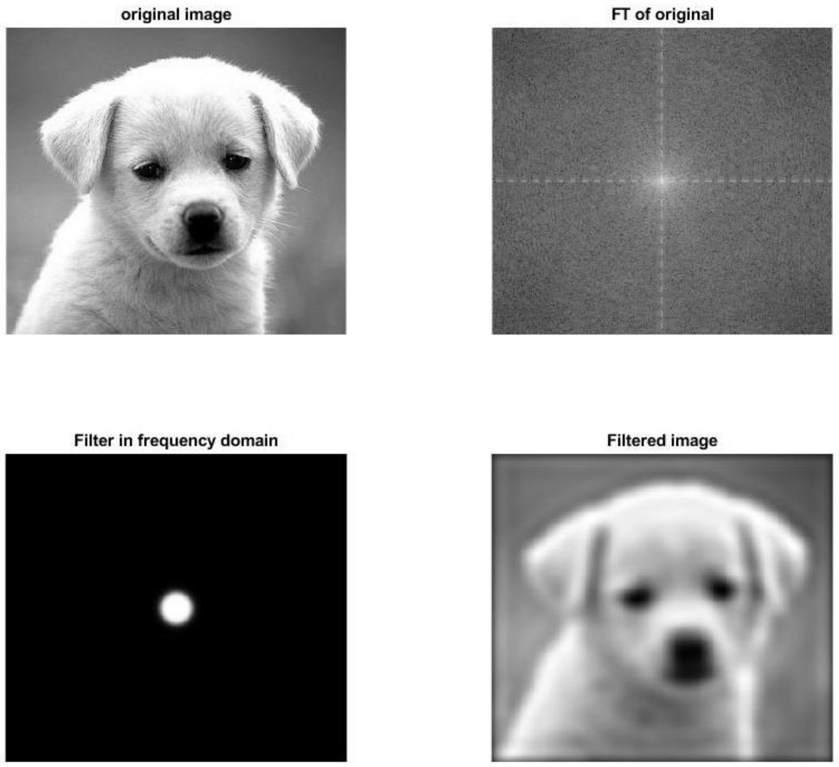


Butterworth low pass filter for cutoff frequencies (0.1) image height and order (n = 5)

MATLAB Code:

```
clc;
F=imread('puppy.jpg');
F=rgb2gray(F);
im_size=size(F);
p=2*im_size(1);Q=2*im_size(2);
FTIm=fft2(double(F),p,Q);
D0= 0.1*im_size(1)
n=5
Filter=lp_hp_filters('btw','lp',p,Q,D0,n);
Filtered_image=real(ifft2(Filter.*FTIm));
Filtered_image=Filtered_image(1:im_size(1),1:im_size(2));
Fim=fftshift(FTIm);
FTI=log(1+abs(Fim));
Ff=fftshift(Filter);
FTF=log(1+abs(Ff));
subplot(2,2,1),imshow(F,[]),title('original image');
subplot(2,2,2),imshow(FTI,[]),title('FT of original');
subplot(2,2,3),imshow(FTF,[]),title('Filter in frequency domain');
subplot(2,2,4),imshow(Filtered_image,[]),title('Filtered image');
```

Output:

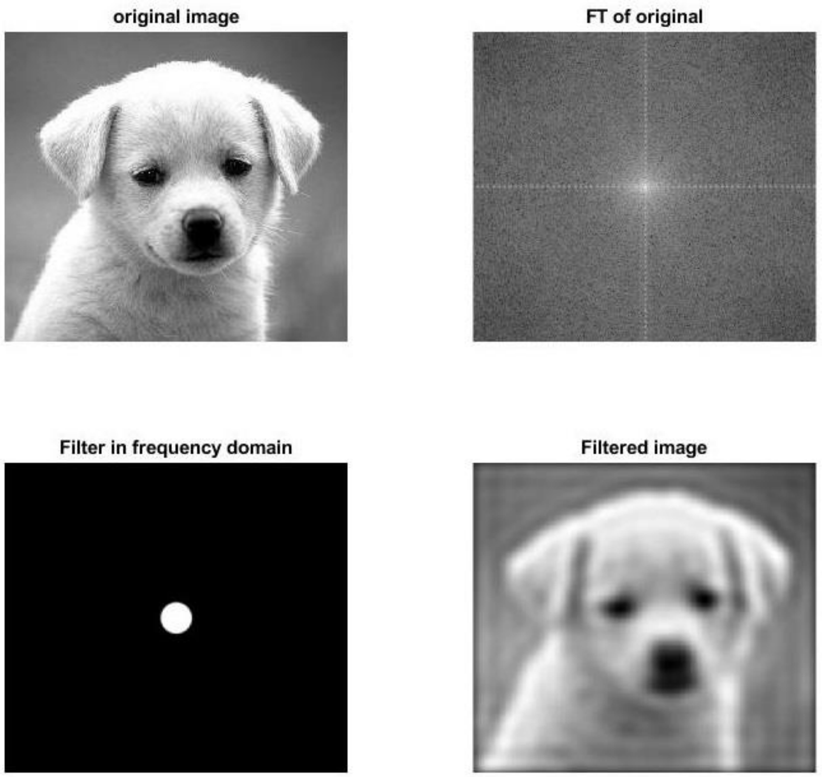


Butterworth low pass filter for cutoff frequencies (0.1) image height and order (n = 20)

MATLAB Code:

```
clc;
F=imread('puppy.jpg');
F=rgb2gray(F);
im_size=size(F);
p=2*im_size(1);Q=2*im_size(2);
FTIm=fft2(double(F),p,Q);
D0= 0.1*im_size(1)
n=20
Filter=lp_hp_filters('btw','lp',p,Q,D0,n);
Filtered_image=real(ifft2(Filter.*FTIm));
Filtered_image=Filtered_image(1:im_size(1),1:im_size(2));
Fim=fftshift(FTIm);
FTI=log(1+abs(Fim));
Ff=fftshift(Filter);
FTF=log(1+abs(Ff));
subplot(2,2,1),imshow(F,[]),title('original image');
subplot(2,2,2),imshow(FTI,[]),title('FT of original');
subplot(2,2,3),imshow(FTF,[]),title('Filter in frequency domain');
subplot(2,2,4),imshow(Filtered_image,[]),title('Filtered image');
```

Output:



original image

FT of original

Filter in frequency domain

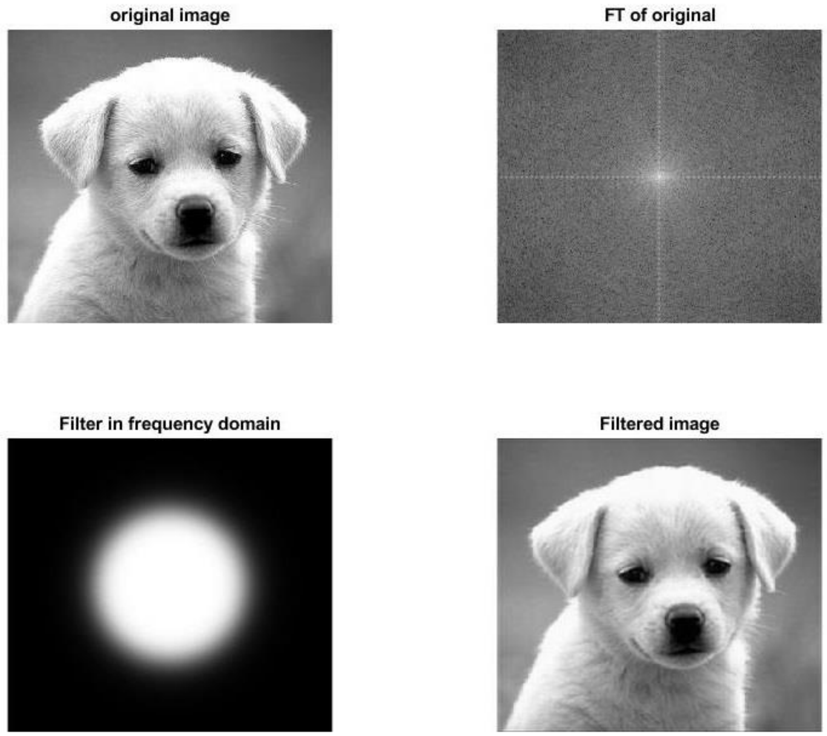
Filtered image

Butterworth low pass filter for cutoff frequencies (0.5) image height and order (n = 1)

MATLAB Code:

```
clc;
F=imread('puppy.jpg');
F=rgb2gray(F);
im_size=size(F);
p=2*im_size(1);Q=2*im_size(2);
FTIm=fft2(double(F),p,Q);
D0= 0.5*im_size(1)
n=1
Filter=lp_hp_filters('btw','lp',p,Q,D0,n);
Filtered_image=real(ifft2(Filter.*FTIm));
Filtered_image=Filtered_image(1:im_size(1),1:im_size(2));
Fim=fftshift(FTIm);
FTI=log(1+abs(Fim));
Ff=fftshift(Filter);
FTF=log(1+abs(Ff));
subplot(2,2,1),imshow(F,[]),title('original image');
subplot(2,2,2),imshow(FTI,[]),title('FT of original');
subplot(2,2,3),imshow(FTF,[]),title('Filter in frequency domain');
subplot(2,2,4),imshow(Filtered_image,[]),title('Filtered image');
```

Output:

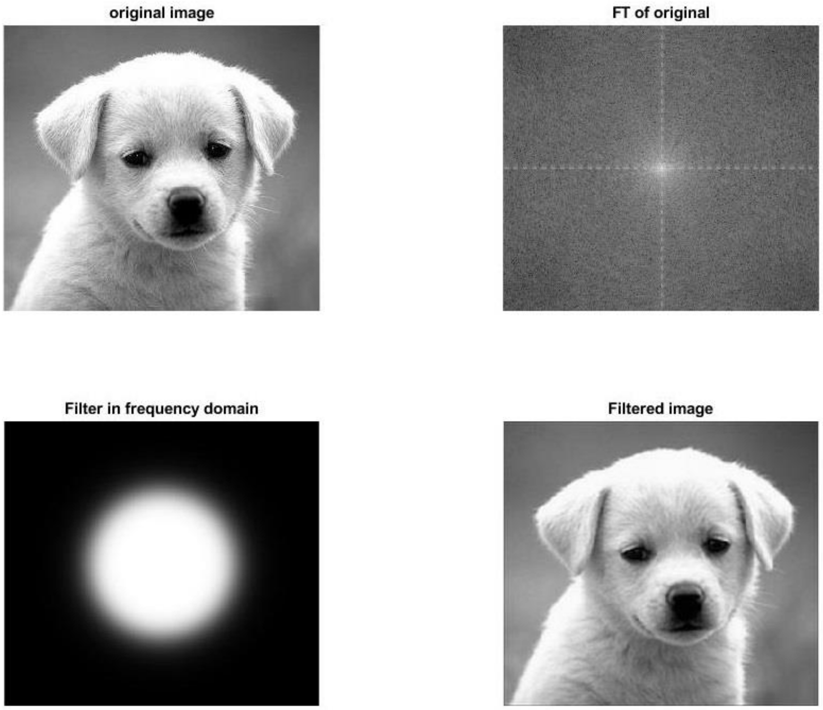


Butterworth low pass filter for cutoff frequencies (0.5) image height and order (n = 5)

MATLAB Code:

```
clc;
F=imread('puppy.jpg');
F=rgb2gray(F);
im_size=size(F);
p=2*im_size(1);Q=2*im_size(2);
FTIm=fft2(double(F),p,Q);
D0= 0.5*im_size(1)
n=5
Filter=lp_hp_filters('btw','lp',p,Q,D0,n);
Filtered_image=real(ifft2(Filter.*FTIm));
Filtered_image=Filtered_image(1:im_size(1),1:im_size(2));
Fim=fftshift(FTIm);
FTI=log(1+abs(Fim));
Ff=fftshift(Filter);
FTF=log(1+abs(Ff));
subplot(2,2,1),imshow(F,[]),title('original image');
subplot(2,2,2),imshow(FTI,[]),title('FT of original');
subplot(2,2,3),imshow(FTF,[]),title('Filter in frequency domain');
subplot(2,2,4),imshow(Filtered_image,[]),title('Filtered image');
```

Output:

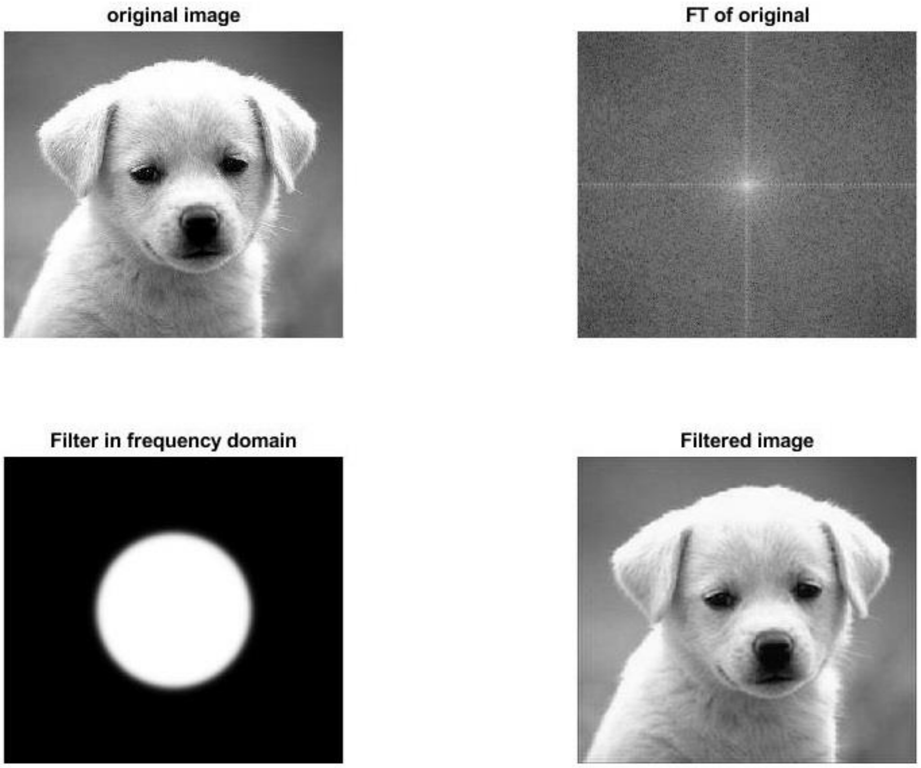


Butterworth low pass filter for cutoff frequencies (0.5) image height and order (n = 20)

MATLAB Code:

```
clc;
F=imread('puppy.jpg');
F=rgb2gray(F);
im_size=size(F);
p=2*im_size(1);Q=2*im_size(2);
FTIm=fft2(double(F),p,Q);
D0= 0.5*im_size(1)
n=20
Filter=lp_hp_filters('btw','lp',p,Q,D0,n);
Filtered_image=real(ifft2(Filter.*FTIm));
Filtered_image=Filtered_image(1:im_size(1),1:im_size(2));
Fim=fftshift(FTIm);
FTI=log(1+abs(Fim));
Ff=fftshift(Filter);
FTF=log(1+abs(Ff));
subplot(2,2,1),imshow(F,[]),title('original image');
subplot(2,2,2),imshow(FTI,[]),title('FT of original');
subplot(2,2,3),imshow(FTF,[]),title('Filter in frequency domain');
subplot(2,2,4),imshow(Filtered_image,[]),title('Filtered image');
```

Output:



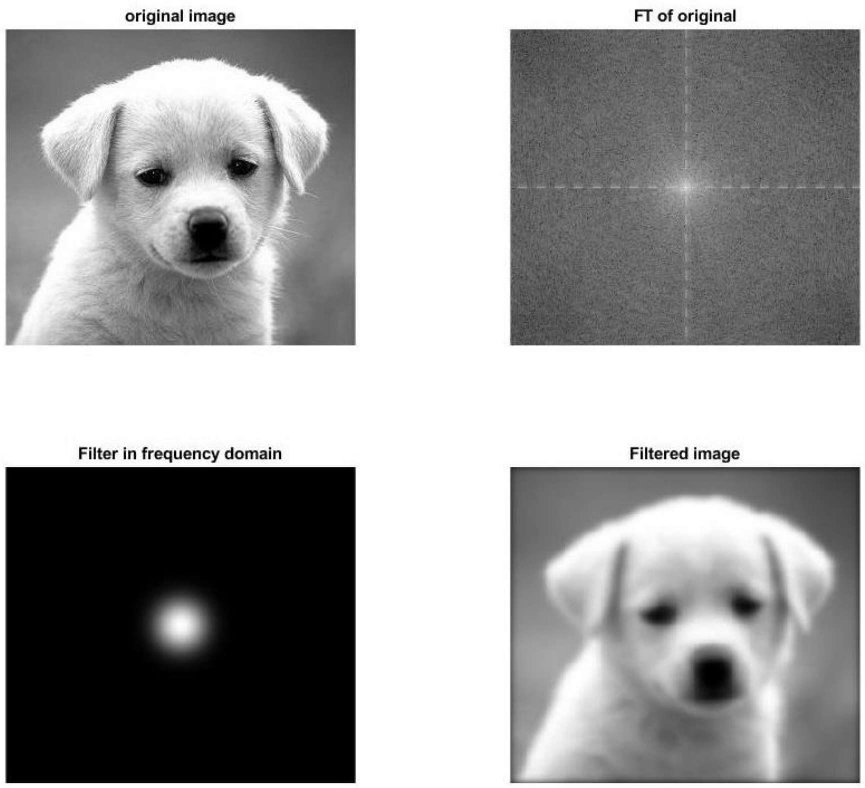
3. Implement a *Gaussian* low pass filter for cutoff frequencies (0.1, 0.3 and 0.7 of image height) .

Gaussian low pass filter for cutoff frequencies (0.1) image height

MATLAB Code:

```
clc;
F=imread('puppy.jpg');
F=rgb2gray(F);
im_size=size(F);
p=2*im_size(1);Q=2*im_size(2);
FTIm=fft2(double(F),p,Q);
D0= 0.1*im_size(1)
n=0
Filter=lp_hp_filters('gaussian','lp',p,Q,D0,n);
Filtered_image=real(ifft2(Filter.*FTIm));
Filtered_image=Filtered_image(1:im_size(1),1:im_size(2));
Fim=fftshift(FTIm);
FTI=log(1+abs(Fim));
Ff=fftshift(Filter);
FTF=log(1+abs(Ff));
subplot(2,2,1),imshow(F,[]),title('original image');
subplot(2,2,2),imshow(FTI,[]),title('FT of original');
subplot(2,2,3),imshow(FTF,[]),title('Filter in frequency domain');
subplot(2,2,4),imshow(Filtered_image,[]),title('Filtered image');
```

Output:

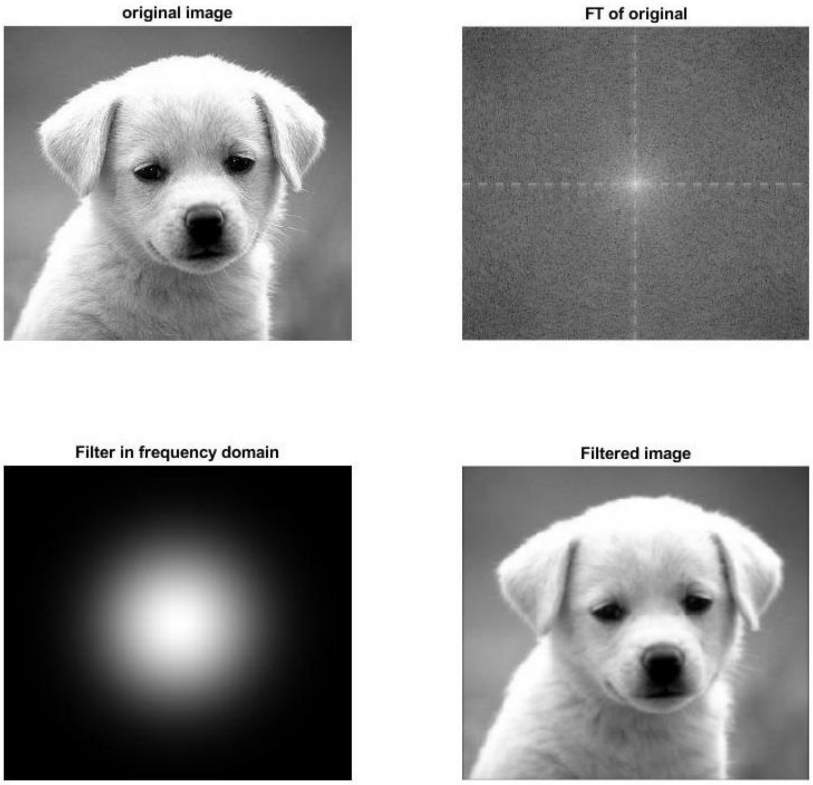


Gaussian low pass filter for cutoff frequencies (0.3) image height

MATLAB Code:

```
clc;
F=imread('puppy.jpg');
F=rgb2gray(F);
im_size=size(F);
p=2*im_size(1);Q=2*im_size(2);
FTIm=fft2(double(F),p,Q);
D0= 0.3*im_size(1)
n=0
Filter=lp_hp_filters('gaussian','lp',p,Q,D0,n);
Filtered_image=real(ifft2(Filter.*FTIm));
Filtered_image=Filtered_image(1:im_size(1),1:im_size(2));
Fim=fftshift(FTIm);
FTI=log(1+abs(Fim));
Ff=fftshift(Filter);
FTF=log(1+abs(Ff));
subplot(2,2,1),imshow(F,[]),title('original image');
subplot(2,2,2),imshow(FTI,[]),title('FT of original');
subplot(2,2,3),imshow(FTF,[]),title('Filter in frequency domain');
subplot(2,2,4),imshow(Filtered_image,[]),title('Filtered image');
```

Output:

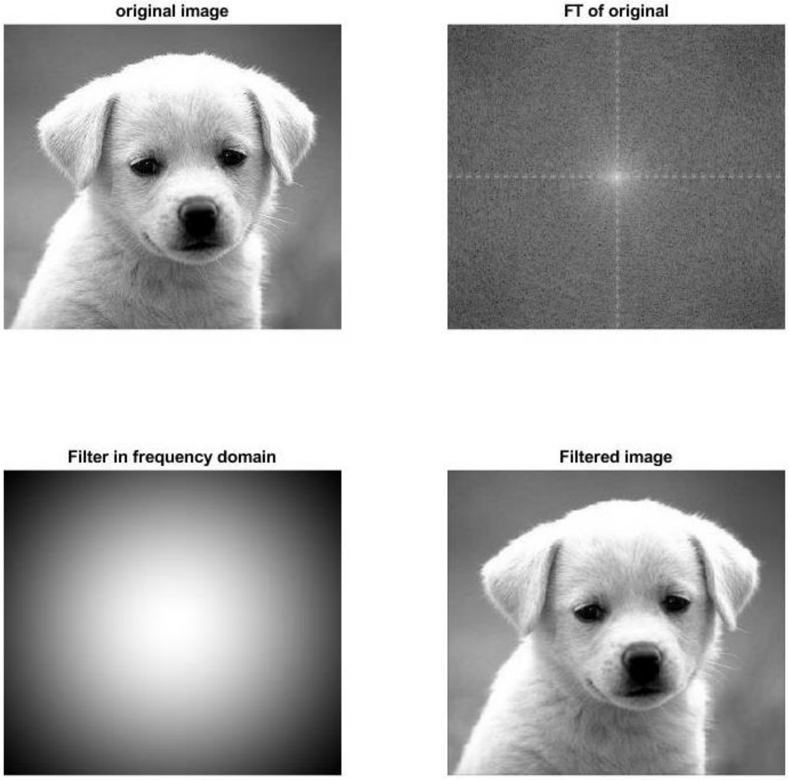


Gaussian low pass filter for cutoff frequencies (0.7) image height

MATLAB Code:

```
clc;
F=imread('puppy.jpg');
F=rgb2gray(F);
im_size=size(F);
p=2*im_size(1);Q=2*im_size(2);
FTIm=fft2(double(F),p,Q);
D0= 0.7*im_size(1)
n=0
Filter=lp_hp_filters('gaussian','lp',p,Q,D0,n);
Filtered_image=real(ifft2(Filter.*FTIm));
Filtered_image=Filtered_image(1:im_size(1),1:im_size(2));
Fim=fftshift(FTIm);
FTI=log(1+abs(Fim));
Ff=fftshift(Filter);
FTF=log(1+abs(Ff));
subplot(2,2,1),imshow(F,[]),title('original image');
subplot(2,2,2),imshow(FTI,[]),title('FT of original');
subplot(2,2,3),imshow(FTF,[]),title('Filter in frequency domain');
subplot(2,2,4),imshow(Filtered_image,[]),title('Filtered image');
```

Output:



original image

FT of original

Filter in frequency domain

Filtered image

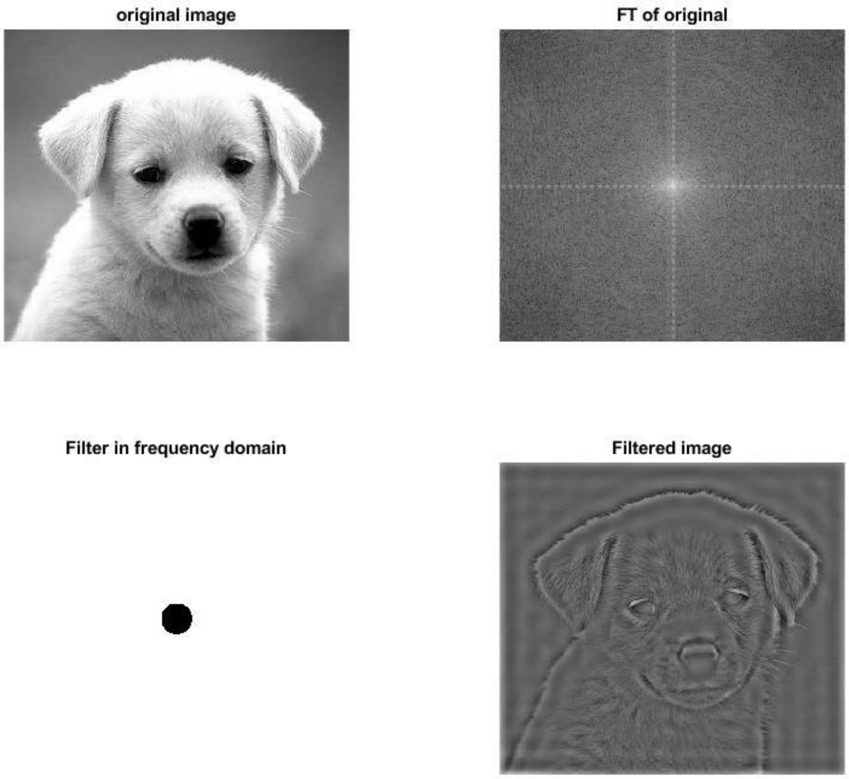
4. Implement an *Ideal* high-pass filter for cutoff frequencies (0.1, 0.3 and 0.7 of image height).

Ideal high-pass filter for cutoff frequencies (0.1) image height

MATLAB Code:

```
clc;
F=imread('puppy.jpg');
F=rgb2gray(F);
im_size=size(F);
p=2*im_size(1);Q=2*im_size(2);
FTIm=fft2(double(F),p,Q);
D0= 0.1*im_size(1)
n=0
Filter=lp_hp_filters('ideal','hp',p,Q,D0,n);
Filtered_image=real(ifft2(Filter.*FTIm));
Filtered_image=Filtered_image(1:im_size(1),1:im_size(2));
Fim=fftshift(FTIm);
FTI=log(1+abs(Fim));
Ff=fftshift(Filter);
FTF=log(1+abs(Ff));
subplot(2,2,1),imshow(F,[]),title('original image');
subplot(2,2,2),imshow(FTI,[]),title('FT of original');
subplot(2,2,3),imshow(FTF,[]),title('Filter in frequency domain');
subplot(2,2,4),imshow(Filtered_image,[]),title('Filtered image');
```

Output:



original image

FT of original

Filter in frequency domain

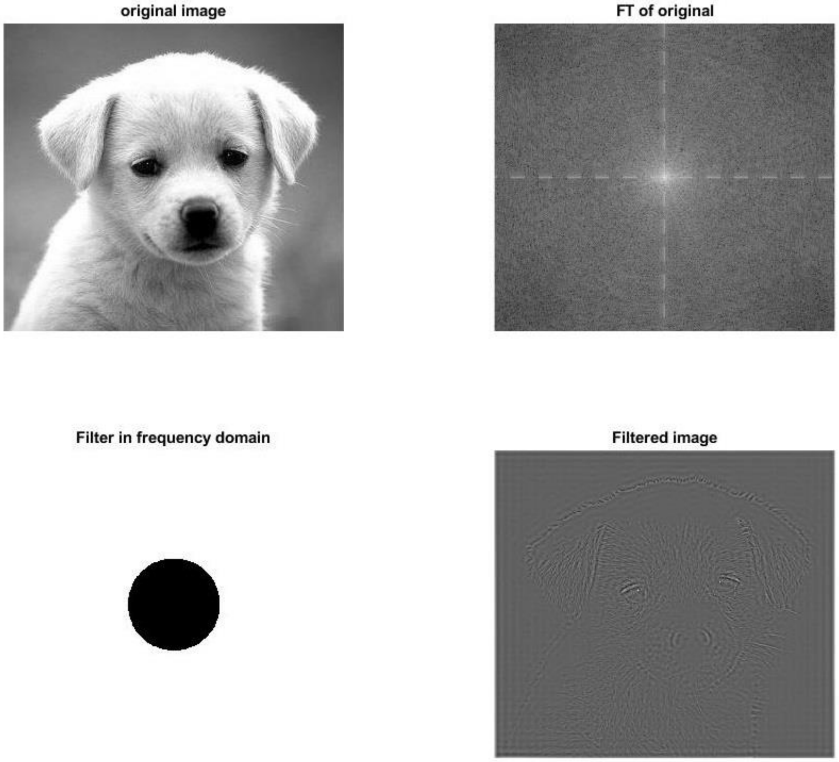
Filtered image

Ideal high-pass filter for cutoff frequencies (0.3) image height

MATLAB Code:

```
clc;
F=imread('puppy.jpg');
F=rgb2gray(F);
im_size=size(F);
p=2*im_size(1);Q=2*im_size(2);
FTIm=fft2(double(F),p,Q);
D0= 0.3*im_size(1)
n=0
Filter=lp_hp_filters('ideal','hp',p,Q,D0,n);
Filtered_image=real(ifft2(Filter.*FTIm));
Filtered_image=Filtered_image(1:im_size(1),1:im_size(2));
Fim=fftshift(FTIm);
FTI=log(1+abs(Fim));
Ff=fftshift(Filter);
FTF=log(1+abs(Ff));
subplot(2,2,1),imshow(F,[]),title('original image');
subplot(2,2,2),imshow(FTI,[]),title('FT of original');
subplot(2,2,3),imshow(FTF,[]),title('Filter in frequency domain');
subplot(2,2,4),imshow(Filtered_image,[]),title('Filtered image');
```

Output:

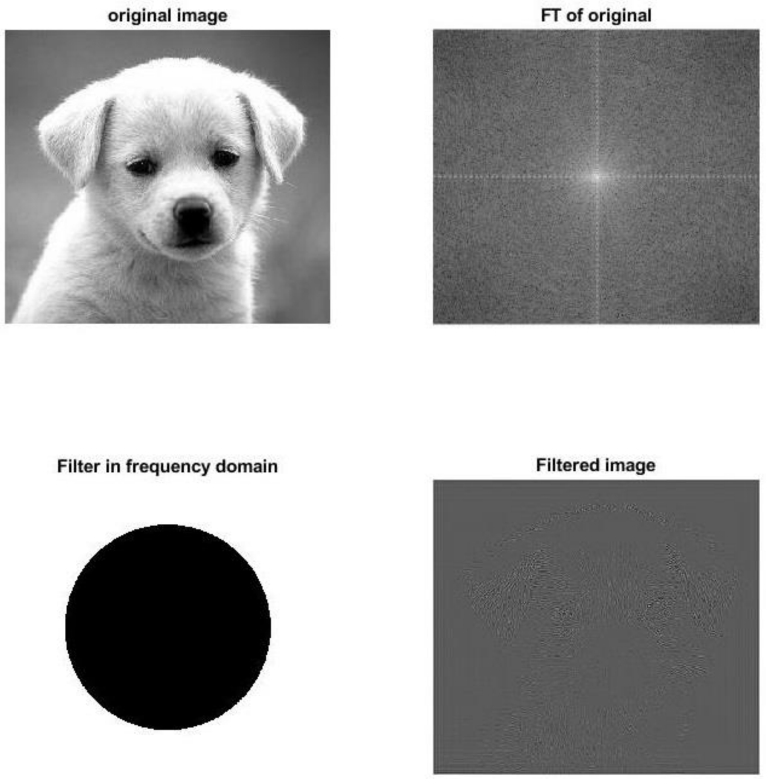


Ideal high-pass filter for cutoff frequencies (0.7) image height

MATLAB Code:

```
clc;
F=imread('puppy.jpg');
F=rgb2gray(F);
im_size=size(F);
p=2*im_size(1);Q=2*im_size(2);
FTIm=fft2(double(F),p,Q);
D0= 0.7*im_size(1)
n=0
Filter=lp_hp_filters('ideal','hp',p,Q,D0,n);
Filtered_image=real(ifft2(Filter.*FTIm));
Filtered_image=Filtered_image(1:im_size(1),1:im_size(2));
Fim=fftshift(FTIm);
FTI=log(1+abs(Fim));
Ff=fftshift(Filter);
FTF=log(1+abs(Ff));
subplot(2,2,1),imshow(F,[]),title('original image');
subplot(2,2,2),imshow(FTI,[]),title('FT of original');
subplot(2,2,3),imshow(FTF,[]),title('Filter in frequency domain');
subplot(2,2,4),imshow(Filtered_image,[]),title('Filtered image');
```

Output:


original image

FT of original

Filter in frequency domain

Filtered image

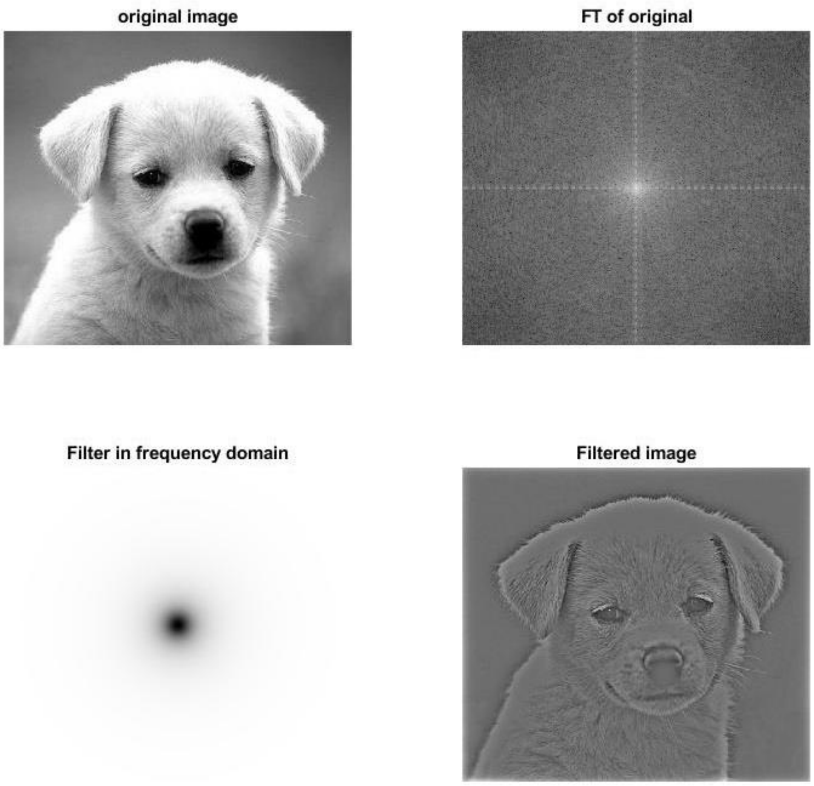
5. Implement a *Butterworth* high pass filter for cutoff frequencies (0.1 and 0.5 of image height) and order *n* = 1; 5; 20

Butterworth high pass filter for cutoff frequencies (0.1) image height and order n = 1

MATLAB Code:

```
clc;
F=imread('puppy.jpg');
F=rgb2gray(F);
im_size=size(F);
p=2*im_size(1);Q=2*im_size(2);
FTIm=fft2(double(F),p,Q);
D0= 0.1*im_size(1)
n=1
Filter=lp_hp_filters('btw','hp',p,Q,D0,n);
Filtered_image=real(ifft2(Filter.*FTIm));
Filtered_image=Filtered_image(1:im_size(1),1:im_size(2));
Fim=fftshift(FTIm);
FTI=log(1+abs(Fim));
Ff=fftshift(Filter);
FTF=log(1+abs(Ff));
subplot(2,2,1),imshow(F,[]),title('original image');
subplot(2,2,2),imshow(FTI,[]),title('FT of original');
subplot(2,2,3),imshow(FTF,[]),title('Filter in frequency domain');
subplot(2,2,4),imshow(Filtered_image,[]),title('Filtered image');
```

Output

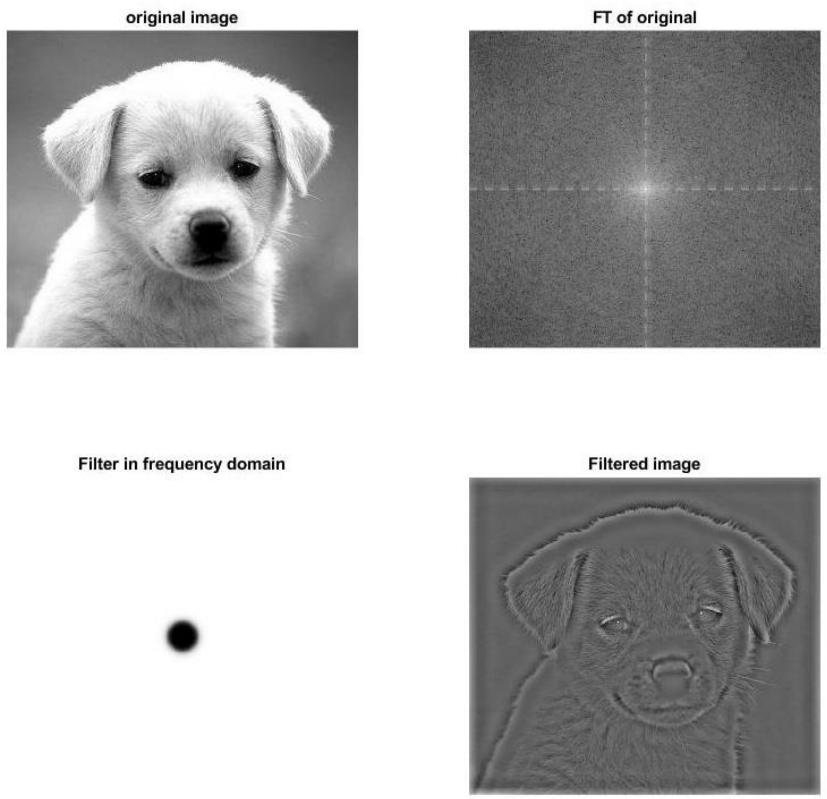


Butterworth high pass filter for cutoff frequencies (0.1) image height and order n = 5

MATLAB Code:

```
clc;
F=imread('puppy.jpg');
F=rgb2gray(F);
im_size=size(F);
p=2*im_size(1);Q=2*im_size(2);
FTIm=fft2(double(F),p,Q);
D0= 0.1*im_size(1)
n=5
Filter=lp_hp_filters('btw','hp',p,Q,D0,n);
Filtered_image=real(ifft2(Filter.*FTIm));
Filtered_image=Filtered_image(1:im_size(1),1:im_size(2));
Fim=fftshift(FTIm);
FTI=log(1+abs(Fim));
Ff=fftshift(Filter);
FTF=log(1+abs(Ff));
subplot(2,2,1),imshow(F,[]),title('original image');
subplot(2,2,2),imshow(FTI,[]),title('FT of original');
subplot(2,2,3),imshow(FTF,[]),title('Filter in frequency domain');
subplot(2,2,4),imshow(Filtered_image,[]),title('Filtered image');
```

Output

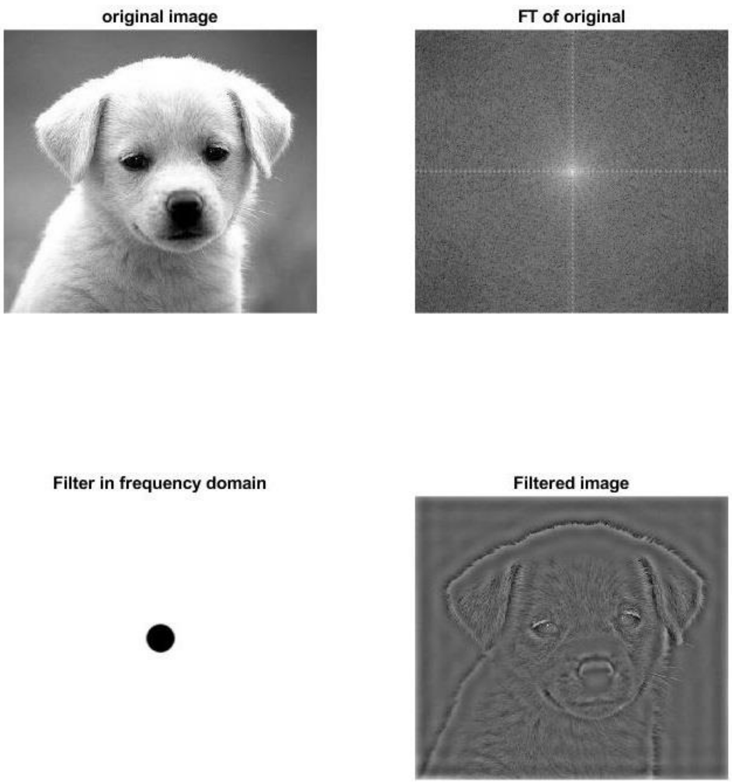


Butterworth high pass filter for cutoff frequencies (0.1) image height and order n = 20

MATLAB Code:

```
clc;
F=imread('puppy.jpg');
F=rgb2gray(F);
im_size=size(F);
p=2*im_size(1);Q=2*im_size(2);
FTIm=fft2(double(F),p,Q);
D0= 0.1*im_size(1)
n=20
Filter=lp_hp_filters('btw','hp',p,Q,D0,n);
Filtered_image=real(ifft2(Filter.*FTIm));
Filtered_image=Filtered_image(1:im_size(1),1:im_size(2));
Fim=fftshift(FTIm);
FTI=log(1+abs(Fim));
Ff=fftshift(Filter);
FTF=log(1+abs(Ff));
subplot(2,2,1),imshow(F,[]),title('original image');
subplot(2,2,2),imshow(FTI,[]),title('FT of original');
subplot(2,2,3),imshow(FTF,[]),title('Filter in frequency domain');
subplot(2,2,4),imshow(Filtered_image,[]),title('Filtered image');
```

Output

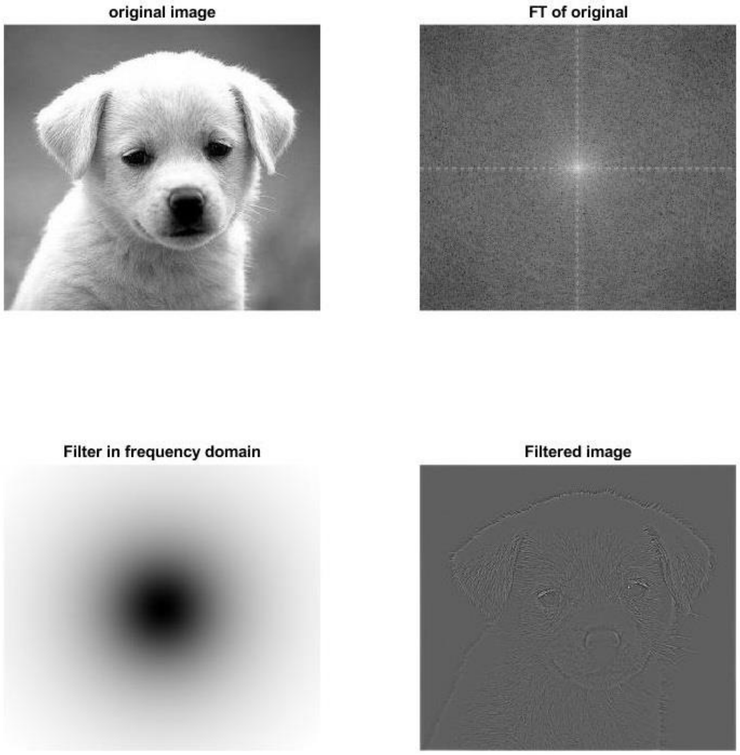


Butterworth high pass filter for cutoff frequencies (0.5) image height and order n = 1

MATLAB Code:

```
clc;
F=imread('puppy.jpg');
F=rgb2gray(F);
im_size=size(F);
p=2*im_size(1);Q=2*im_size(2);
FTIm=fft2(double(F),p,Q);
D0= 0.5*im_size(1)
n=1
Filter=lp_hp_filters('btw','hp',p,Q,D0,n);
Filtered_image=real(ifft2(Filter.*FTIm));
Filtered_image=Filtered_image(1:im_size(1),1:im_size(2));
Fim=fftshift(FTIm);
FTI=log(1+abs(Fim));
Ff=fftshift(Filter);
FTF=log(1+abs(Ff));
subplot(2,2,1),imshow(F,[]),title('original image');
subplot(2,2,2),imshow(FTI,[]),title('FT of original');
subplot(2,2,3),imshow(FTF,[]),title('Filter in frequency domain');
subplot(2,2,4),imshow(Filtered_image,[]),title('Filtered image');
```

Output

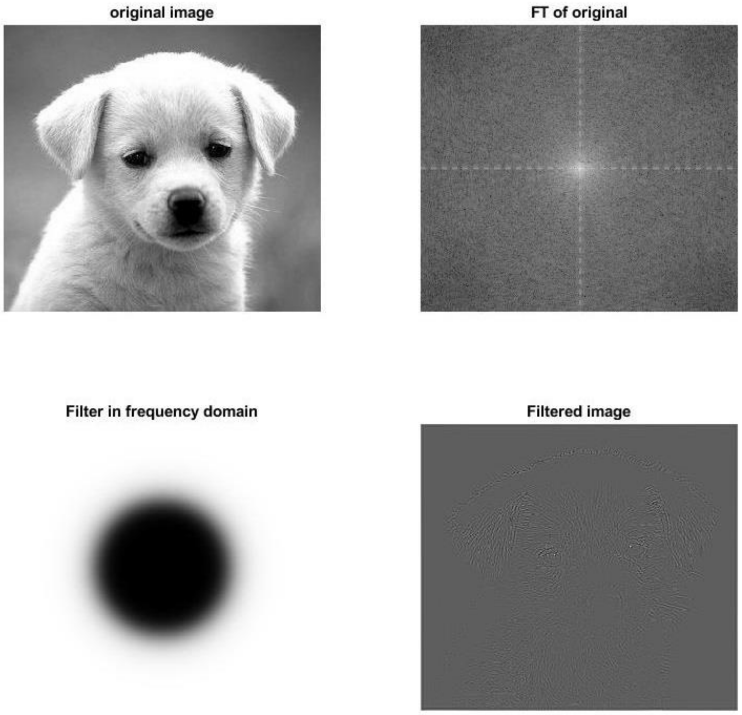


Butterworth high pass filter for cutoff frequencies (0.5) image height and order n = 5

MATLAB Code:

```
clc;
F=imread('puppy.jpg');
F=rgb2gray(F);
im_size=size(F);
p=2*im_size(1);Q=2*im_size(2);
FTIm=fft2(double(F),p,Q);
D0= 0.5*im_size(1)
n=5
Filter=lp_hp_filters('btw','hp',p,Q,D0,n);
Filtered_image=real(ifft2(Filter.*FTIm));
Filtered_image=Filtered_image(1:im_size(1),1:im_size(2));
Fim=fftshift(FTIm);
FTI=log(1+abs(Fim));
Ff=fftshift(Filter);
FTF=log(1+abs(Ff));
subplot(2,2,1),imshow(F,[]),title('original image');
subplot(2,2,2),imshow(FTI,[]),title('FT of original');
subplot(2,2,3),imshow(FTF,[]),title('Filter in frequency domain');
subplot(2,2,4),imshow(Filtered_image,[]),title('Filtered image');
```

Output



original image

FT of original

Filter in frequency domain

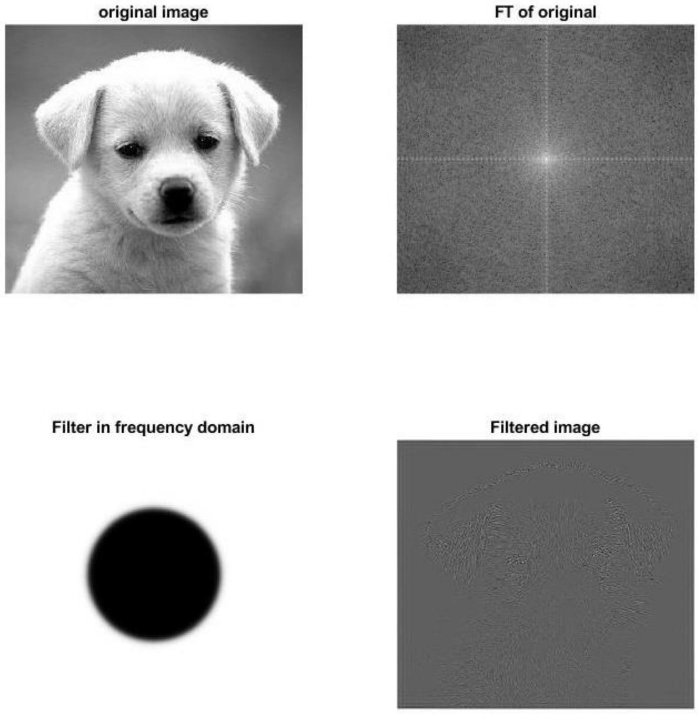
Filtered image

Butterworth high pass filter for cutoff frequencies (0.5) image height and order n = 20

MATLAB Code:

```
clc;
F=imread('puppy.jpg');
F=rgb2gray(F);
im_size=size(F);
p=2*im_size(1);Q=2*im_size(2);
FTIm=fft2(double(F),p,Q);
D0= 0.5*im_size(1)
n=20
Filter=lp_hp_filters('btw','hp',p,Q,D0,n);
Filtered_image=real(ifft2(Filter.*FTIm));
Filtered_image=Filtered_image(1:im_size(1),1:im_size(2));
Fim=fftshift(FTIm);
FTI=log(1+abs(Fim));
Ff=fftshift(Filter);
FTF=log(1+abs(Ff));
subplot(2,2,1),imshow(F,[]),title('original image');
subplot(2,2,2),imshow(FTI,[]),title('FT of original');
subplot(2,2,3),imshow(FTF,[]),title('Filter in frequency domain');
subplot(2,2,4),imshow(Filtered_image,[]),title('Filtered image');
```

Output



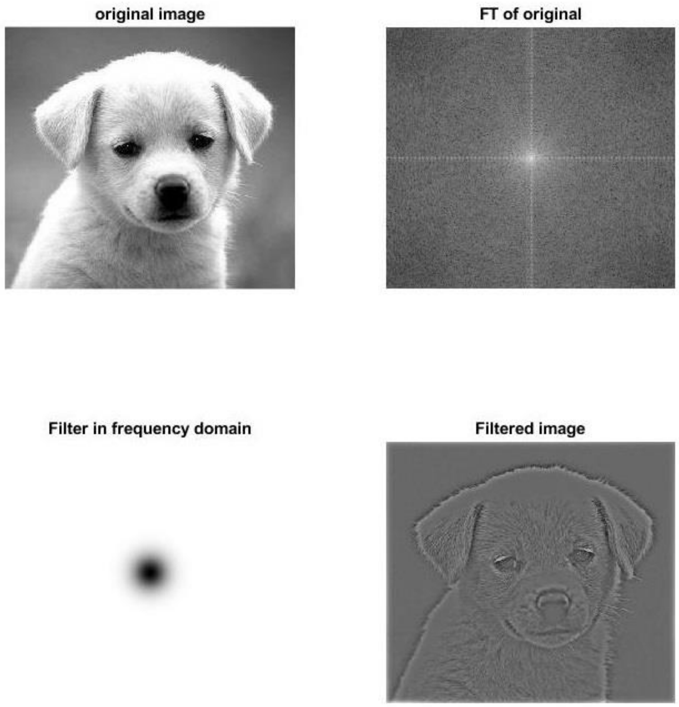
6. Perform a *Gaussian* high pass filtering for cutoff frequencies (0.1, 0.3 and 0.7 of image height)

Gaussian high pass filter for cutoff frequencies (0.1) image height

MATLAB Code:

```
clc;
F=imread('puppy.jpg');
F=rgb2gray(F);
im_size=size(F);
p=2*im_size(1);Q=2*im_size(2);
FTIm=fft2(double(F),p,Q);
D0= 0.1*im_size(1)
n=0
Filter=lp_hp_filters('gaussian','hp',p,Q,D0,n);
Filtered_image=real(ifft2(Filter.*FTIm));
Filtered_image=Filtered_image(1:im_size(1),1:im_size(2));
Fim=fftshift(FTIm);
FTI=log(1+abs(Fim));
Ff=fftshift(Filter);
FTF=log(1+abs(Ff));
subplot(2,2,1),imshow(F,[]),title('original image');
subplot(2,2,2),imshow(FTI,[]),title('FT of original');
subplot(2,2,3),imshow(FTF,[]),title('Filter in frequency domain');
subplot(2,2,4),imshow(Filtered_image,[]),title('Filtered image');
```

Output



original image

FT of original

Filter in frequency domain

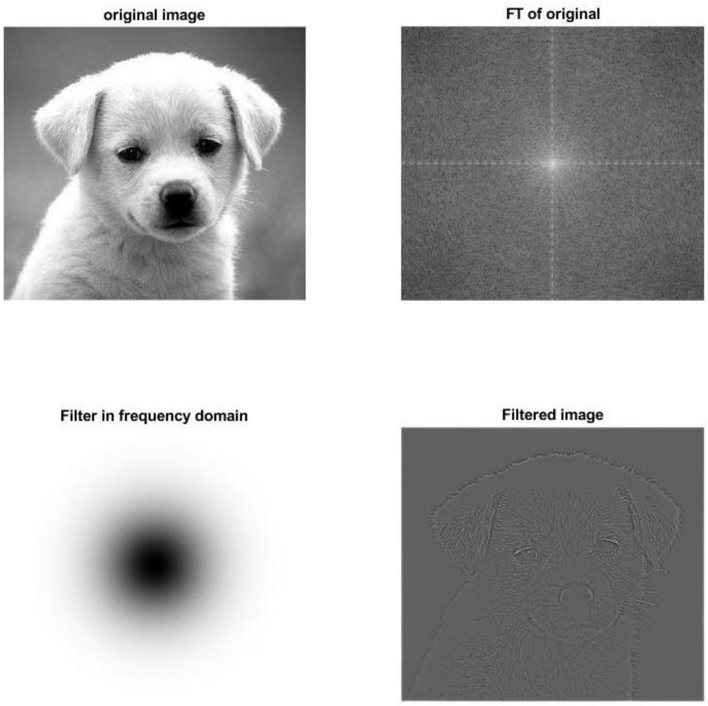
Filtered image

Gaussian high pass filter for cutoff frequencies (0.3) image height

MATLAB Code:

```
clc;
F=imread('puppy.jpg');
F=rgb2gray(F);
im_size=size(F);
p=2*im_size(1);Q=2*im_size(2);
FTIm=fft2(double(F),p,Q);
D0= 0.3*im_size(1)
n=0
Filter=lp_hp_filters('gaussian','hp',p,Q,D0,n);
Filtered_image=real(ifft2(Filter.*FTIm));
Filtered_image=Filtered_image(1:im_size(1),1:im_size(2));
Fim=fftshift(FTIm);
FTI=log(1+abs(Fim));
Ff=fftshift(Filter);
FTF=log(1+abs(Ff));
subplot(2,2,1),imshow(F,[]),title('original image');
subplot(2,2,2),imshow(FTI,[]),title('FT of original');
subplot(2,2,3),imshow(FTF,[]),title('Filter in frequency domain');
subplot(2,2,4),imshow(Filtered_image,[]),title('Filtered image');
```

Output

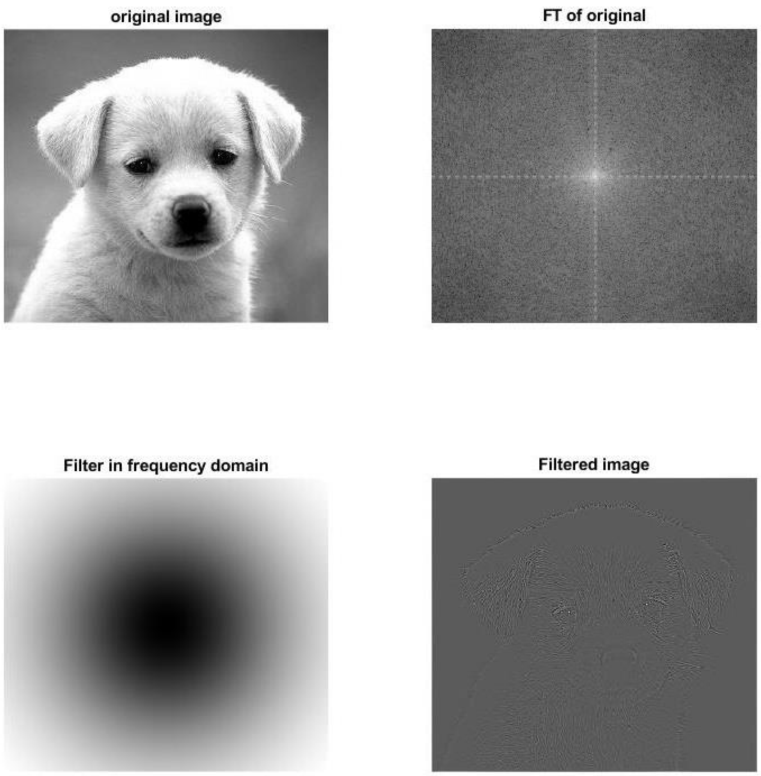


Gaussian high pass filter for cutoff frequencies (0.7) image height

MATLAB Code:

```
clc;
F=imread('puppy.jpg');
F=rgb2gray(F);
im_size=size(F);
p=2*im_size(1);Q=2*im_size(2);
FTIm=fft2(double(F),p,Q);
D0= 0.7*im_size(1)
n=0
Filter=lp_hp_filters('gaussian','hp',p,Q,D0,n);
Filtered_image=real(ifft2(Filter.*FTIm));
Filtered_image=Filtered_image(1:im_size(1),1:im_size(2));
Fim=fftshift(FTIm);
FTI=log(1+abs(Fim));
Ff=fftshift(Filter);
FTF=log(1+abs(Ff));
subplot(2,2,1),imshow(F,[]),title('original image');
subplot(2,2,2),imshow(FTI,[]),title('FT of original');
subplot(2,2,3),imshow(FTF,[]),title('Filter in frequency domain');
subplot(2,2,4),imshow(Filtered_image,[]),title('Filtered image');
```

Output



original image

FT of original

Filter in frequency domain

Filtered image

Discussion:

1. ***What can you observe when increasing the cut off frequency radius of the Ideal filter?***

   Ideal Low Pass Filter:

   The ideal low pass filter is used to reduce the high-frequency noise and pass the low-frequency components. i.e., the ideal low pass filter allows the frequencies within the circle and filtered-out all the frequencies above the circle radius.
   That is why increasing the cutoff frequencies resulting in less blurred/smooth images. If the radius or cutoff frequency increases from 0.3 to 0.7, the smoothness of the image will be better.

   Ideal High Pass Filter:

   The ideal high pass filter is used to reduce the low-frequency noise and pass the high-frequency components. i.e., the ideal high pass filter allows the frequencies above the circle radius and filtered-out all the frequencies below the circle radius. In image processing, the ideal high pass filter is used to sharpen the given image.

   That is why the Ideal high pass filter detects more edge information when it increases. We can conclude that if the radius or cutoff frequency increases from 0.1 to 0.7, the sharpness of the image will be reduced.

2. ***What can you observe when increasing the order of the Butterworth filter when the cut-off frequency remains the same?***

   The Butterworth low pass filter is used to attenuate the high-frequency noise and pass the low-frequency components. i.e., the Butterworth low pass filter allows the frequencies within the circle and filtered-out all the frequencies above the circle radius.   In image processing, the ideal Butterworth low pass filter is used to smoothen the given image. The difference between the ideal LPF and Butterworth filter is that the Butterworth LPF introduces a smooth transition from 1 to 0, whereas in ideal LPF the transition is sharp in nature.

   we can conclude that if the order of the filter increases with the constant cut-off frequency, then the smoothness of the image will be better, because of the reduction of ringing artifacts in the Butterworth filter. Also, for the order n=1, there is no ringing artifact. In Butterworth LPF, the ringing artifact increases with the increase in the order n.

3. ***Discuss the general performance of the Gaussian filters in comparison to the Ideal and Butterworth filters. You do not need to discuss each possible comparison case. Only discuss any interesting phenomenon or performance observations.***

When we compare the Gaussian and ideal LPF with the same cut-off frequency, the image quality of Gaussian is better.

Besides, when we compare the Gaussian with the cut-off frequency 0.7 and the Butterworth filter with the cut-off frequency 0.5, the ringing wave effect avoided in the Gaussian filter, the image quality of Gaussian is better.