INDIAN INSTITUTE OF TECHNOLOGY

KHARAGPUR

DEPARTMENT OF ELECTRONICS AND ELECTRICAL COMMUNICATION

ASSIGNMENT NUMBER 4

ARNAB BISWAS

21EC65R01

M.Tech

2021 – 2022

Introduction:

The objective of this experiment is to apply following Frequency Filtering operations on an image:
1. Ideal Low-pass Filter
2. Ideal High-pass Filter
3. Butterworth Low-pass Filter
4. Butterworth High-pass Filter
5. Gaussian Low-pass Filter
6. Gaussian High-pass Filter.

Each of the above filter helps in modification or extracting features by manipulating frequency spectrum of the image. The image is Fourier transformed, multiplied with the filter function and then re-transformed into the spatial domain. Supressing low frequencies result in an enhancement of edges, while supressing high frequencies results in a smoother image in the spatial domain. All frequency filters can also be implemented in the spatial domain and, if there exists a simple kernel for the desired filter effect, it is computationally less expensive to perform the filtering in the spatial domain. Frequency filtering is more appropriate if no straightforward kernel can be found in the spatial domain, and may also be more efficient. Frequency filters process an image by attenuating certain frequencies. G(u,v) =H(u,v)F(u,v).Functions for various filters are as follows:

Ideal Low-pass filter:

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

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

Gaussian Low-Pass Filter:

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

Butter-worth Low-pass Filter:

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

High-pass filter:

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

Algorithm:

2-D FFT requires 1-D FFT on rows first and then on columns. To do this, we first perform FFT on each row using Divide and Conquer strategy. The we transpose this coefficient matrix, and perform FFT on rows again. Same strategy is used for Inverse FFT. We calculate filter coefficients by using the functions as mentioned above.

```
complex **FFT2( string filename ){
// temporary variable

complex temp;
// Opening image

Mat img = imread(filename, -1);
// Variable for storing input and output

complex **inpData = new complex* [ img.rows ];
complex **fftData = new complex* [ img.rows ];

if (!img.data){
cout << "Error:Image not found" <<endl;
return inpData;
}
// Initializaing arrays
for(int i = 0; i < img.rows; i++){
inpData[i] = new complex [ img.cols ];
fftData[i] = new complex [ img.cols ];
}

// Loading image data

for(int i = 0; i < img.rows; i++)
for(int j = 0; j < img.cols; j++)
inpData[i][j] = complex( (double)img.at(i,j), 0.0);

// Performing FFT on each row

for( int i = 0; i < img.rows; i++)
FFT( img.cols, inpData[i], fftData[i]);

// Transpose the image
```

// Perform FFt on each row (column previously)

```
for(int i = 0; i < img.rows; i++)
FFT(img.cols, fftData[i], inpData[i]);
```

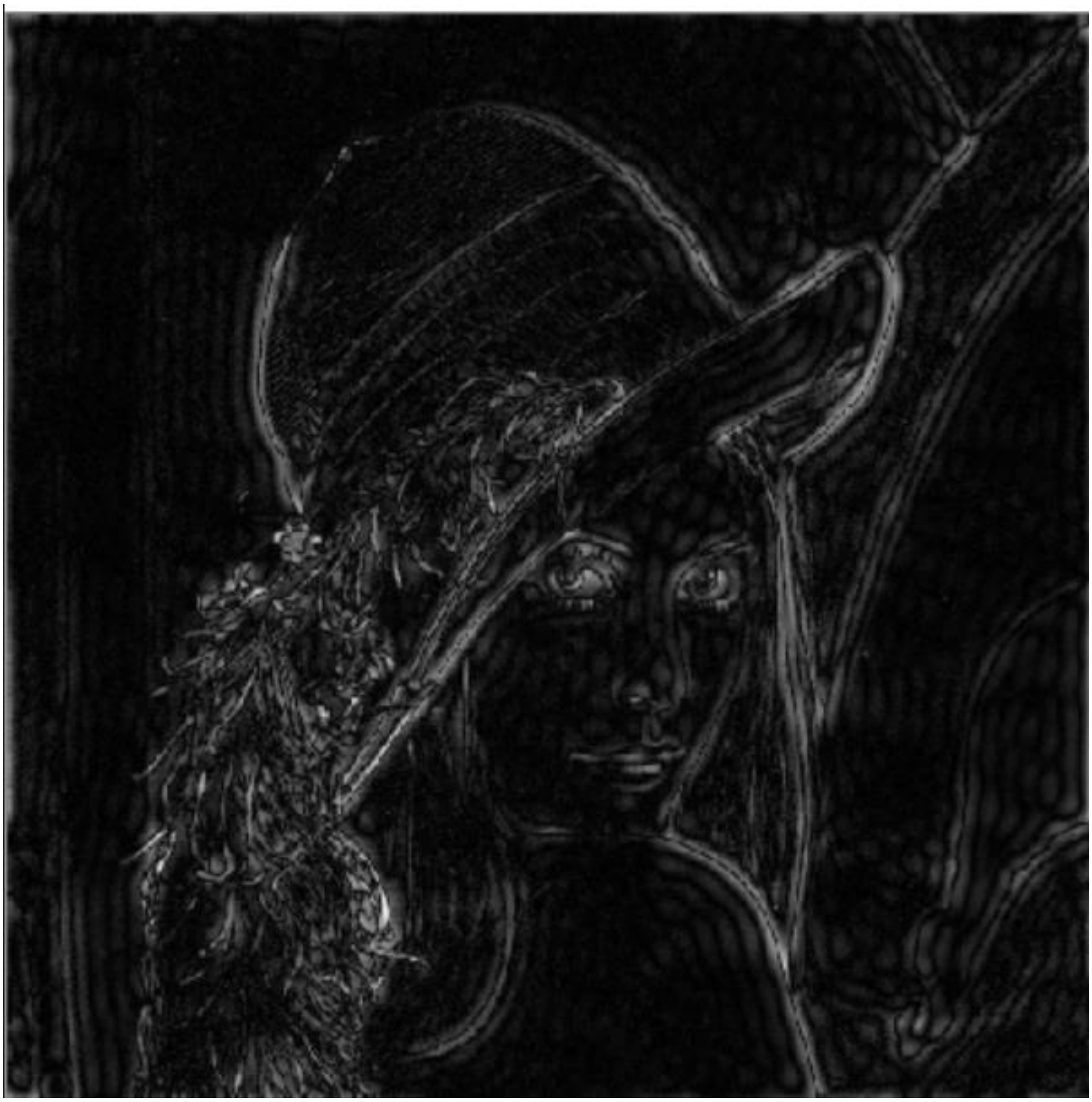
// Return the transformed data

```
return inpData;
}
```

Output:

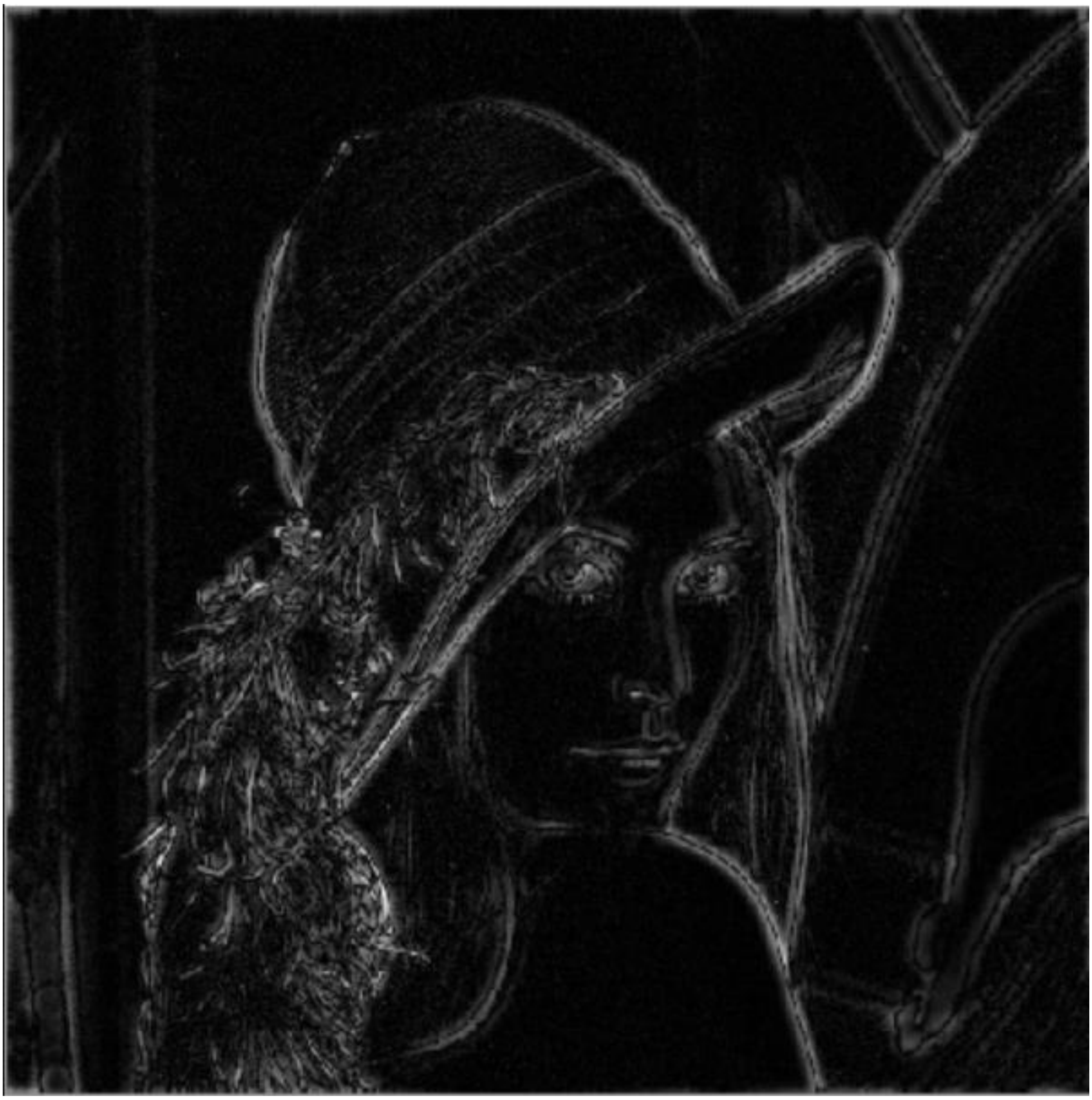
Ideal Low-pass filter:

Ideal High-pass Filter:



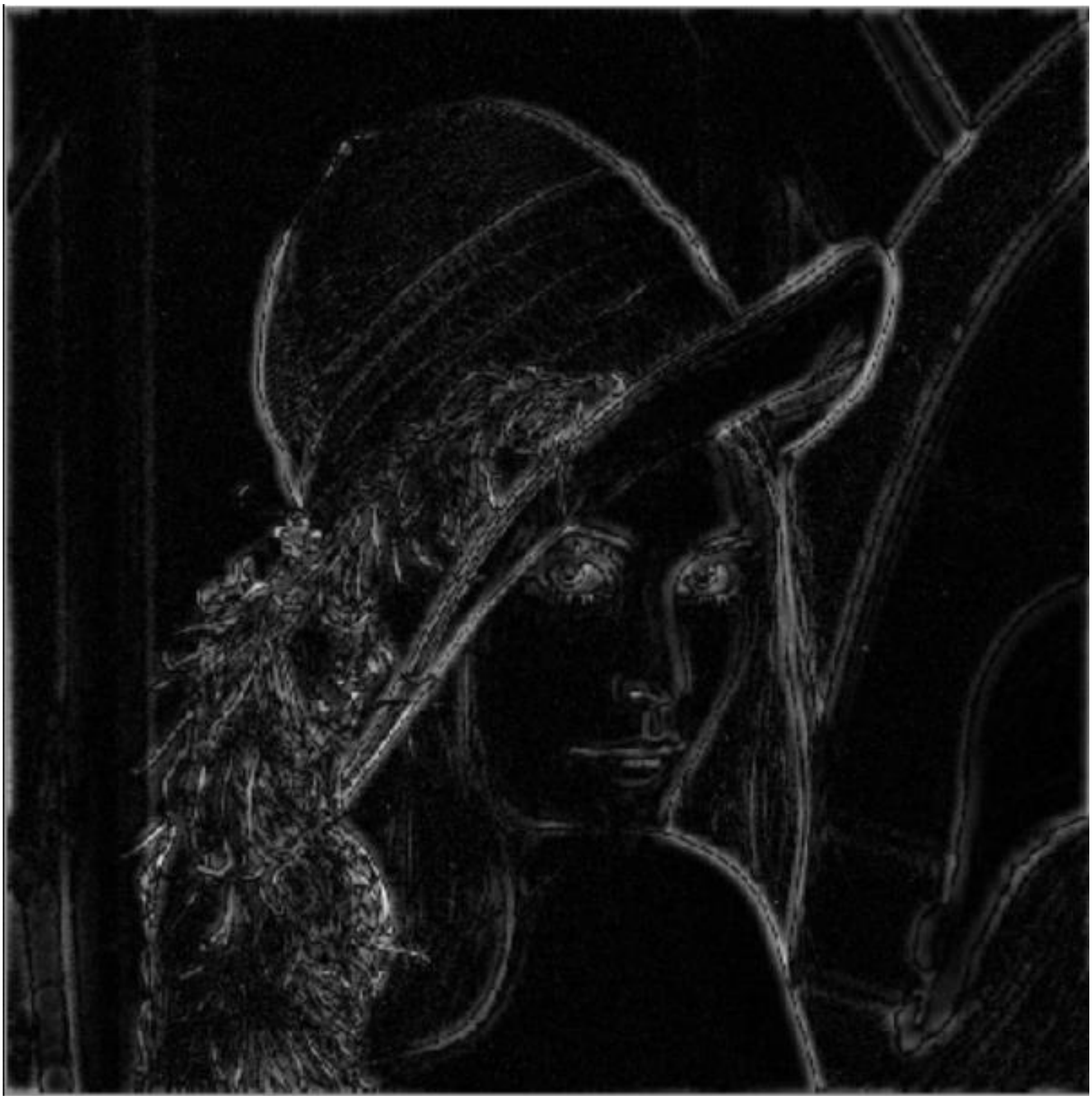
Gaussian Low-Pass Filter:

Gaussian high-pass filter:



Butterfly Low Pass Filter:

Butterfly High Pass Filter:



Conclusion:

Frequency domain filtering provides more control over frequency. For example, in an ideal low-pass filter we can remove high frequency components easily. Thus, they are easier for filtering operations. Frequency filtering are more computationally expensive than the spatial filters because of the requirement of fourier transform and inverse fourier transform. Thus, they are used only in case we can't create a mask in spatial domain for an operation.Ideal filters suffer from ringing (Gibbs) phenomenon. It means some similar to oscillation response are observed in the output. It is due to non-continues nature of the filter.To avoid ringing effects in ideal filters which occurs due to discontinuity in filter response, Butterworth and Gaussian filters are used which are continuous. The parameters (like σ or order) decide how close they are to the ideal filters by affecting the slope in transition band.

Sources:

https://en.wikipedia.org/wiki/Butterworth_filter
https://en.wikipedia.org/wiki/Gaussian_filter
https://classes.soe.ucsc.edu/ee264/Fall11/LecturePDF/8-SpectralFiltering.pdf