Department of Electronics and Electrical Communication Engineering

Indian Institute of Technology Kharagpur

Digital Image and Video Processing Lab (EC69211)

Experiment No: 4

Title: Frequency Domain Filtering

Date of submission: 04.09.2024

Name: Unnati Singh

Roll No.: 21EC39027

Instructors:

Prof. Saumik Bhattacharya

Prof. Prabir Kumar Biswas

Introduction:

In the field of image processing, frequency domain filtering plays a crucial role in enhancing and transforming images by manipulating their frequency components. Filtering in the frequency domain consists of modifying the Fourier transform of a signal and then taking the inverse transform to obtained the filtered result. This report delves into three significant problems that exemplify the application of frequency domain filtering techniques.

Thus, given a digital signal $f(x)$ of length M, the basic filtering equation is:

$$g(x) = \mathcal{F}^{-1}[H(u)F(u)]$$

Frequency-domain filters work by following a straightforward sequence of steps:

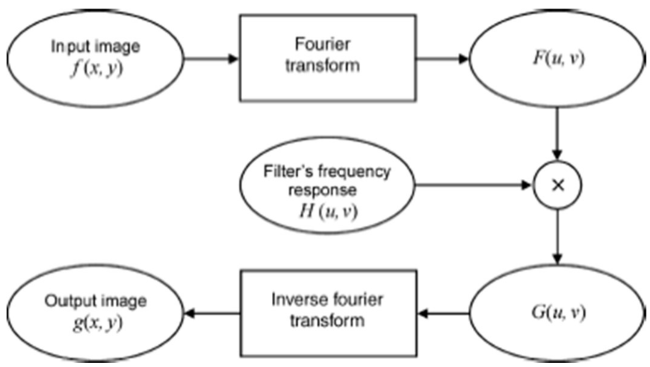


Fig. Frequency Domain Filtering Pipeline

Frequency Domain Filters:

- Ideal LPF: Passes all frequencies below a cutoff frequency D_0 and blocks higher frequencies.
- Ideal HPF: Blocks frequencies below D_0 and passes higher frequencies.

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

$$H(u,v) = 1 - H_{\text{LPF}}(u,v) \text{ (for HPF)}$$

where $D(u,v)$ is the distance from the origin in the frequency domain.

Gaussian LPF & HPF:

- Gaussian LPF: Applies a smooth transition where lower frequencies pass, and higher ones gradually attenuate.
- Gaussian HPF: Attenuates lower frequencies and passes higher frequencies smoothly.

Equations:

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

$$H(u,v) = 1 - H_{\text{LPF}}(u,v) \text{ (for HPF)}$$

Butterworth LPF & HPF:

- **Butterworth LPF:** Provides a smoother cutoff than Ideal, with order \(n \) determining the steepness.
- **Butterworth HPF:** Similar to LPF but passes higher frequencies.

Equations:

$$H(u,v) = \frac{1}{1 + \left(\frac{D(u,v)}{D_0}\right)^{2n}} \text{ (for LPF)}$$

$$H(u,v) = \frac{1}{1 + \left(\frac{D_0}{D(u,v)}\right)^{2n}} \text{ (for HPF)}$$

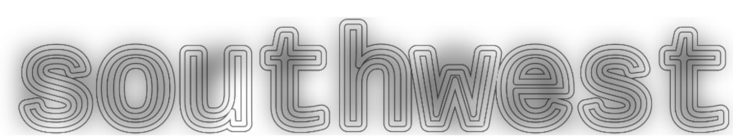
These filters are pivotal in controlling image details by selectively passing or blocking certain frequency components.

Hybrid Images:

A hybrid image is an image that is perceived in one of two different ways, depending on viewing distance, based on the way humans process visual input. Hybrid images combine the low spatial frequencies of one picture with the high spatial frequencies of another picture, producing an image with an interpretation that changes with viewing distance.

Thus, the image looks something from a distance and something else from up close. Some examples include:

Hybrid image example. From close it reads "Love", from far it reads "WAR"

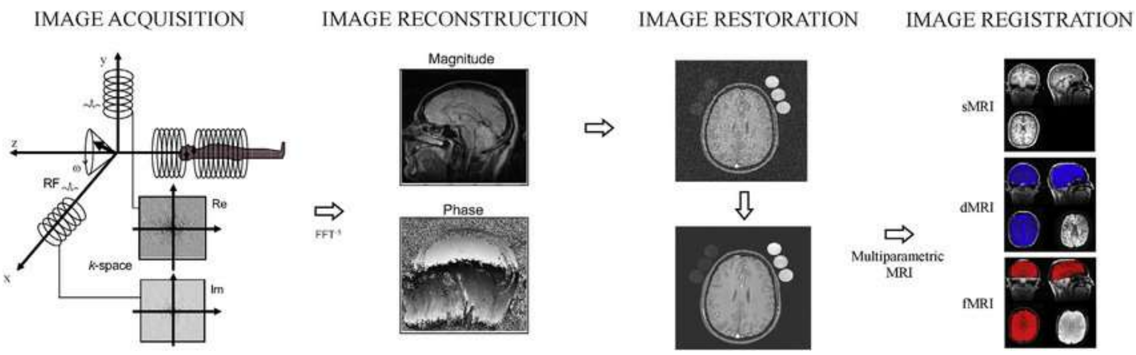


A textual hybrid image reading "southwest" up close and "northeast" from afar

De-noising:

The denoising process can be described as to cancel the noise while retaining and not distorting the quality of processed signal or image. All digital images are degraded by different types of noise during their acquisition and transmission.

The image de-noising pathway can be shown as:



There are different techniques of performing the de-noising in both spatial and frequency domain depending on the type of noise present.

Key Functions in code:

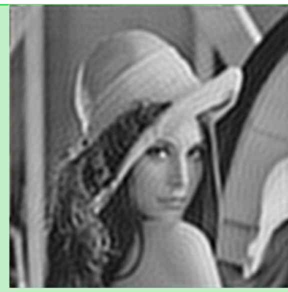
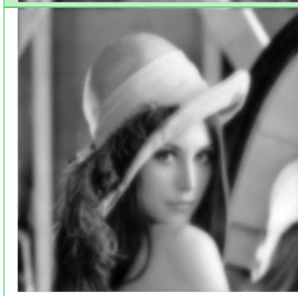
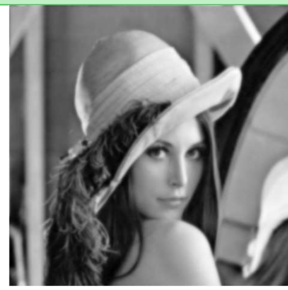
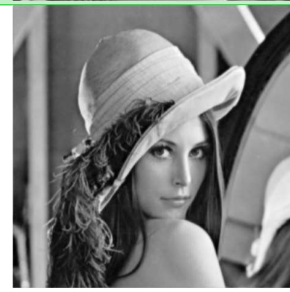
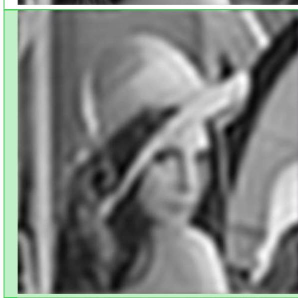
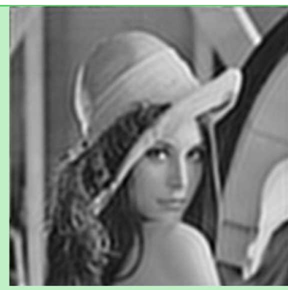
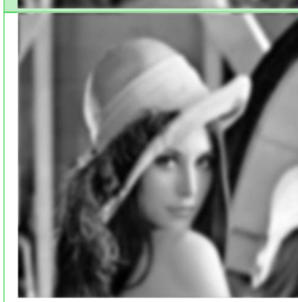
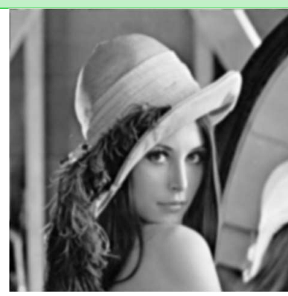
1. `lpf(lpf_type, image_path, cutoff, order=1):`
    - Purpose: Applies a low-pass filter to an image, allowing only low-frequency components to pass through.
    - How It works:
        - The function generates a mask based on the specified filter type (Butterworth, Gaussian, or Ideal) and the cutoff frequency.
        - This mask is then applied to the frequency domain representation of the image to filter out high-frequency components.

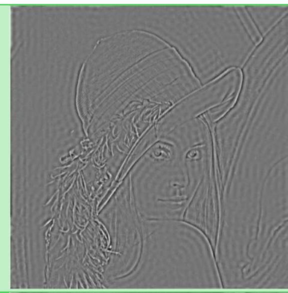
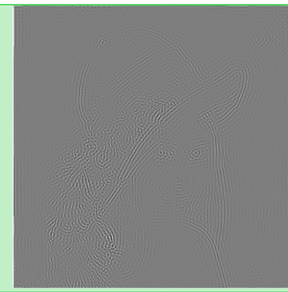
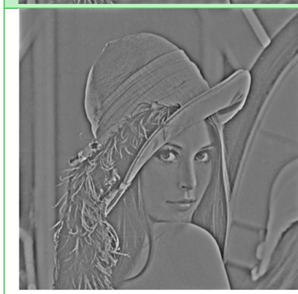
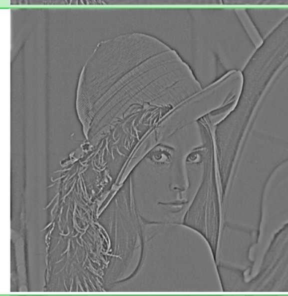
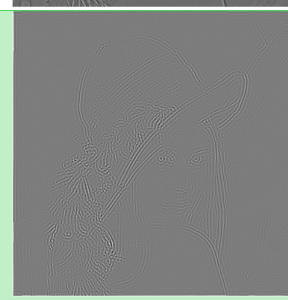
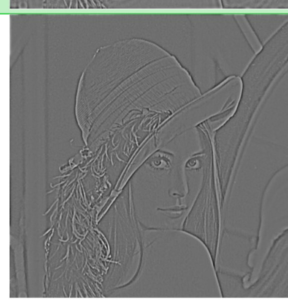
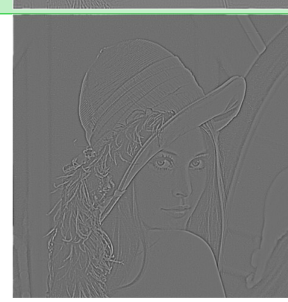
2. `hpf(hpf_type, image_path, cutoff, order=1):`
    - Purpose: Applies a high-pass filter to an image, allowing only high-frequency components to pass through.
    - How It works: Similar to the low-pass filter, this function creates a mask to filter out low-frequency components based on the selected filter type and cutoff frequency.

3. `hybrid(image1='input/einstein.png', image2='input/marilyn.png'):`
    - Purpose: Creates a hybrid image by combining the high-frequency components of one image with the low-frequency components of another.
    - How It works:
        - The function applies a low-pass filter to one image and a high-pass filter to the other.
        - These filtered images are then combined to create the hybrid image, which shows different features depending on the viewing distance.

4. `denoising(img_path):`
    - Purpose: Attempts to remove noise from an image by masking specific frequency components.
    - How it works:
        - The function identifies and suppresses the brightest spots in the magnitude spectrum that are likely to correspond to noise.
        - It then reconstructs the image from the filtered frequency domain representation.

Output Images:

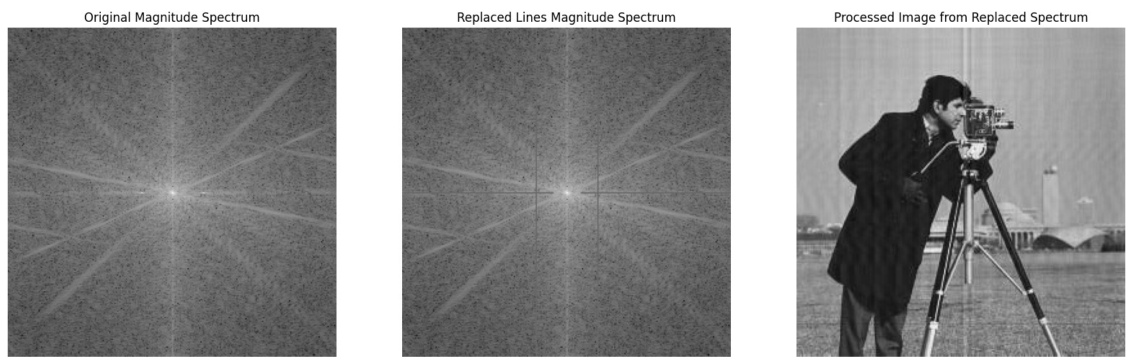
Low-pass Filter Type	Cutoff = 20	Cutoff = 40	Cutoff = 80
Ideal			
Butterworth (Order = 1)			
Butterworth (Order = 10)			
Gaussian			

High-pass Filter Type	Cutoff = 20	Cutoff = 40	Cutoff = 80
Ideal			
Butterworth (Order = 1)			
Butterworth (Order = 10)			
Gaussian			

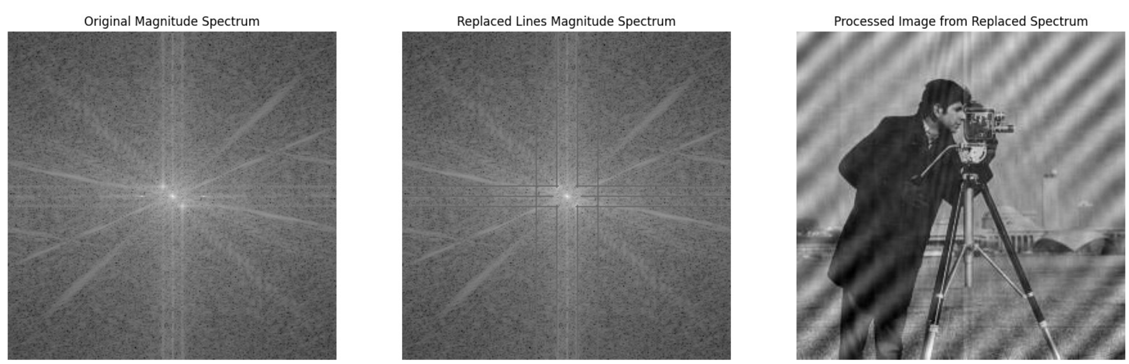
The hybrid image created can be seen as:

The image de-noising results can be seen as:

1. `cameraman_noisy1.jpg`



2. `cameraman_noisy2.jpg`



Discussion:

The primary goal of this experiment was to understand the application of low-pass and high-pass filters in the frequency domain using FFT. By manipulating the frequency components, we can emphasize or suppress certain features in an image, which is crucial for various image processing tasks such as edge detection, blurring, and noise reduction. The creation of hybrid images and the denoising of images further demonstrate the power of frequency domain processing.

We created different low-pass and high-pass filters and observed their properties. During filtering, the ideal and higher order Butterworth filters showed rippling effect. The rippling effect, or "ringing artifacts," is a common issue in image processing that arises when applying filters, particularly those with sharp frequency cutoffs like Ideal Low-Pass Filters. This effect occurs due to abrupt changes in the frequency spectrum, leading to oscillations in the spatial domain that manifest as visible ripples or halos around edges and high-contrast areas in the image. Closely related to the Gibbs Phenomenon, the rippling effect degrades image quality by introducing unwanted artifacts and can obscure fine details, making it problematic in applications where precision and clarity are essential. A hybrid image was also created by the help of these filters which gave us a rudimentary optical illusion.

A basic attempt at denoising was also done in this experiment. We identified the differences in the frequency domains of the original and noisy images and attempted to mask them. While vertical banding was taken care of in a fairly decent manner, the diagonal banding proved to be a challenge and we were not able to denoise that image effectively.

The experiment provided practical experience with frequency domain filters, highlighting the importance of selecting appropriate filter parameters (e.g., cutoff frequency and filter type) for different image processing objectives.