Fourier Transform for Edge Detection

Task Objective: This task explores how the Fourier Transform can be used to detect edges in digital images.

Materials:

- Programming environment with image processing libraries (e.g., Python with OpenCV)
- A test image

Steps:

1. Load the Image: Begin by loading your chosen test image into your programming environment. If it's colored, convert it to grayscale.
2. Implement the Fourier Transform: Apply the Fast Fourier Transform (FFT) to the grayscale image. This transforms the image from the spatial domain (pixel intensities) to the frequency domain (spectrum of frequencies).
3. Design a High-Pass Filter: In the frequency domain, create a high-pass filter. This filter will allow high-frequency components (edges) to pass through while attenuating low-frequency components (smooth areas). There are various filter design techniques you can explore (e.g., ideal filter, Butterworth filter).
4. Apply the Filter: Multiply the FFT of the image with the designed high-pass filter. This emphasizes the high-frequency components corresponding to edges.
5. Inverse Fourier Transform: Apply the Inverse Fast Fourier Transform (IFFT) to the filtered frequency domain image. This transforms the data back into the spatial domain, resulting in an image where edges are enhanced.
6. Comparison: Compare the original image with the edge-detected image obtained through the Fourier Transform method. Analyze how well the edges are identified.

Bonus:

- Experiment with different high-pass filter designs and observe their impact on the edge detection results.

Learning Outcomes:

By completing this task, you will gain hands-on experience with:

- Applying the Fourier Transform for image processing.
- Understanding the relationship between frequency components and edges in images.
- Designing filters in the frequency domain for specific tasks.
- Evaluating the effectiveness of different edge detection techniques