Lab 05 and 06 Manual

1. Frequency Transformation using Fourier Transform

Objective:
To convert an image from the spatial domain to the frequency domain using the Fourier Transform, and to reconstruct it using the Inverse Fourier Transform.

Theory:

- Fourier Transform represents the image in terms of its frequency components, making it easier to apply frequency-based operations.
- The transformed image highlights edges and fine textures (high frequencies) and smooth regions (low frequencies).
- The Inverse Fourier Transform is used to revert the image back to its spatial form after processing.

Functions Used:

- fft2() – Compute the 2D Fourier Transform
- fftshift() – Shift the zero-frequency component to the center
- abs() – Calculate the magnitude of complex values
- log() – Enhance visibility of the magnitude spectrum
- ifftshift() – Inverse of fftshift
- ifft2() – Compute the 2D Inverse Fourier Transform
- imshow() – Display images
- subplot() – Show images side by side

2. High Pass Filtering in Frequency Domain

Objective:
To sharpen an image by filtering high frequency components using Ideal, Butterworth, and Gaussian High Pass Filters.

Theory:

- High Pass Filters allow high-frequency content (edges, fine details) to pass while suppressing low-frequency components (smooth areas).
- These filters emphasize image details and are used for image enhancement.

Filter Types:

- **Ideal High Pass Filter (IHPF):** Sharp cutoff; allows all frequencies above a certain threshold.
- **Butterworth High Pass Filter (BHPF):** Smooth transition between passband and stopband; controlled sharpness.
- **Gaussian High Pass Filter (GHPF):** Smoothest transition; no ringing artifacts.

Functions Used:

- meshgrid() – Generate frequency grid
- sqrt() – Compute Euclidean distance in frequency space
- exp() – Used in Gaussian filter design
- ifft2() – Reconstruct image from frequency domain
- fftshift() / ifftshift() – Shift frequency components
- subplot() – Compare filtered results
- imshow() – Visualize outputs

3. Low Pass Filtering in Frequency Domain

Objective:
To smooth an image by filtering low frequency components using Ideal, Butterworth, and Gaussian Low Pass Filters.

Theory:

- Low Pass Filters preserve low-frequency content (smooth regions) while suppressing high-frequency content (edges, noise).
- Used for denoising or smoothing the image.

Filter Types:

- **Ideal Low Pass Filter (ILPF):** Allows all frequencies below a threshold; may cause ringing.
- **Butterworth Low Pass Filter (BLPF):** Gentle transition; parameterized by filter order.
- **Gaussian Low Pass Filter (GLPF):** Smooth transition and no artifacts; mimics natural blurring.

Functions Used:

- meshgrid() – Create frequency coordinate matrix
- exp() – Construct Gaussian smoothing profile
- fft2() / ifft2() – Transform to and from frequency domain
- abs() / log() – Process magnitude spectrum
- subplot() – Compare smoothing effects visually
- imshow() – Show the smoothed images

4. Performance Comparison of Frequency Filters

Objective:
To visually compare the effects of different frequency filters for both sharpening and smoothing on a sample image.

Theory:
This step evaluates the effectiveness of frequency domain filtering techniques by examining how well

each high pass filter sharpens and each low pass filter smooths the image. The comparison helps determine the most appropriate filter for enhancement or noise reduction.

Functions Used:

- All relevant high and low pass filtering functions
- subplot() – Display all results side by side
- imshow() – Present outputs for visual assessment

5. Image Compression using Transform Techniques

Objective:
To compress grayscale medical images using different transform techniques and evaluate their performance based on compression ratio and PSNR (Peak Signal-to-Noise Ratio).

Theory:

- **Discrete Cosine Transform (DCT):** Transforms image data into frequency components. High energy compaction makes it effective for compression.
- **Haar Transform:** A type of wavelet transform, fast and simple, suited for hierarchical image representation.
- **DCT-Haar Hybrid:** Combines the energy compaction of DCT with the multi-resolution property of Haar transform.

Functions Used:

- dct2() – Compute 2D DCT of an image
- idct2() – Compute 2D inverse DCT
- wavedec2() / haart2() – Perform Haar transform
- psnr() – Calculate Peak Signal-to-Noise Ratio
- Custom calculations – Compute Compression Ratio
- imshow() – Show compressed and reconstructed images

- subplot() – Display results side by side

6. Restoration of Noisy Image using Mean Filters

Objective:
To restore an image corrupted with Gaussian noise using different averaging techniques: Geometric Mean, Harmonic Mean, and Contra-harmonic Mean filters.

Theory:

- **Geometric Mean Filter:** Multiplies pixel values and takes the nth root, preserving detail while reducing noise.
- **Harmonic Mean Filter:** Effective against Gaussian and salt noise; uses reciprocal averaging.
- **Contra-harmonic Mean Filter:** Designed to reduce either salt or pepper noise depending on the parameter value QQQ.

Functions Used:

- imnoise() – Add Gaussian noise to image
- Custom implementations – Apply geometric, harmonic, and contra-harmonic mean filters
- subplot() – Compare original, noisy, and restored images
- imshow() – Display visual results

7. Restoration using Order Statistic Filters

Objective:
To apply different order-statistic filters to a noisy image and analyze their restoration quality.

Theory:
Order-statistic filters are non-linear filters that operate based on the intensity order of neighboring pixels.

Filter Types:

- **Median Filter:** Removes salt-and-pepper noise effectively.
- **Maximum Filter:** Highlights bright details, can reduce pepper noise.
- **Minimum Filter:** Highlights dark details, can reduce salt noise.
- **Midpoint Filter:** Averages the max and min; balances both noise types.
- **Alpha-trimmed Mean Filter:** Removes a portion of extreme values and averages the rest.
- **Trimmed Mean Filter:** Similar to alpha-trimmed; averages data after trimming ends.

Functions Used:

- medfilt2() – Apply median filter
- ordfilt2() – Implement min, max, and midpoint filters
- Custom function – Apply alpha-trimmed and trimmed filters
- imshow() – Show filtered images
- subplot() – Compare outputs

8. Comparative Evaluation of Filters

Objective:
To determine which filter restores the noisy image closest to its original state based on visual assessment and quantitative metrics.

Theory:
Visual inspection and metrics such as PSNR are used to determine the filter's effectiveness. Filters are evaluated based on their ability to reduce noise while preserving image detail.

Functions Used:

- psnr() – Evaluate similarity with the original image
- subplot() – Display side-by-side results

- imshow() – Visual comparison of restored outputs