DIP IA3

M4Q1) Explain the basics of Filtering in the Frequency Domain.

Filtering in the Frequency Domain involves processing an image by modifying its Fourier transform, rather than manipulating pixel values directly in the spatial domain. This method is particularly useful for implementing convolution operations efficiently and for designing filters that act selectively on specific frequency components of an image.

Key Concepts

1. **Fourier Transform in Image Processing:**
   - Any image can be transformed from the spatial domain to the frequency domain using the **2D Discrete Fourier Transform (DFT)**.
   - The DFT decomposes an image into its **sinusoidal frequency components**, allowing manipulation of these components before transforming back to the spatial domain.
2. **Why Frequency Domain Filtering?**
   - **Convolution in spatial domain = multiplication in frequency domain**.
   - Certain types of filters (like low-pass and high-pass) are easier to implement and design in the frequency domain.
3. **Steps in Frequency Domain Filtering:**
   a. **Compute the DFT** of the input image, $f(x, y)$, to get $F(u, v)$.

   b. **Multiply** the DFT by a filter function $H(u, v)$, the frequency response of the desired filter.

   c. **Compute the inverse DFT** of the result to get the filtered image back in the spatial domain.

   $$g(x, y) = \mathcal{F}^{-1}\{H(u, v) \cdot F(u, v)\}$$
4. **Types of Frequency Domain Filters:**
   - **Low-Pass Filters (LPF):** Attenuate high frequencies (remove noise, smooth images).

- **High-Pass Filters (HPF):** Attenuate low frequencies (enhance edges, sharpen images).
- **Band-Pass / Band-Reject Filters:** Focus on specific frequency ranges.

5. **Practical Considerations:**
   - **Centering the Fourier Spectrum** (multiply by$(-1)^{x+y}$) before applying DFT to bring the low-frequency components to the center.
   - **Padding the image** to avoid circular convolution effects.
   - **Filtering may introduce ringing artifacts** especially with ideal filters.

M4Q2) Explain with block diagram the basic steps for image filtering in frequency domain.

Filtering in the frequency domain involves transforming an image into its frequency representation, modifying it using a filter, and then transforming it back to the spatial domain. The process is systematic and can be illustrated using a **block diagram**.

Basic Steps in Frequency Domain Filtering:

1. **Obtain the image** $f(x, y)$
   - This is the original input image in the spatial domain.
2. **Multiply by** $(-1)^{x+y}$ **to center the transform**
   - This step centers the low-frequency components in the Fourier spectrum (shifts zero frequency to the center of the spectrum).
3. **Compute the 2D DFT (Fourier Transform)**
   - Converts the spatial image to the frequency domain:

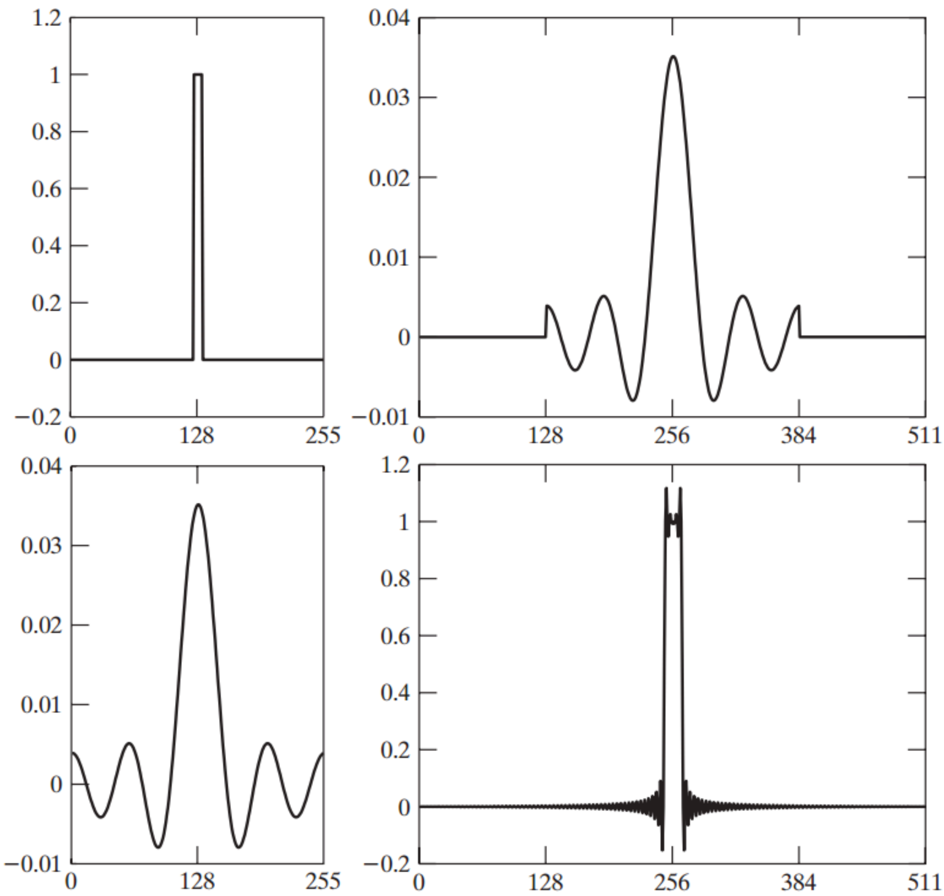
     F$(u, v) = \mathcal{F}\{f(x, y)\}$
4. **Multiply by the filter function** $H(u, v)$
   - This represents the actual filtering operation in the frequency domain:

     G$(u, v) = H(u, v) \cdot F(u, v)$
5. **Compute the inverse DFT**
   - Converts the filtered result back to the spatial domain:

$g'(x, y) = \mathcal{F}^{-1}\{G(u, v)\}$

6. **Multiply by $(-1)^{x+y}$ again**
   - Reverses the centering done initially and yields the final filtered image:

     $g(x, y) = (-1)^{x+y} \cdot g'(x, y)$



(a) Original filter specified in the (centered) frequency domain. (b) Spatial representation obtained by computing the IDFT of (a). (c) Result of padding (b) to twice its length (note the discontinuities). (d) Corresponding filter in the frequency domain obtained by computing the DFT of (c). Note the ringing caused by the discontinuities in (c). (The curves appear continuous because the points were joined to simplify visual analysis.)

M4Q3) What are the advantages and applications of filtering in frequency domain?

Filtering in the frequency domain is a powerful technique in digital image processing, particularly useful for tasks like image enhancement, noise reduction, and edge detection. It involves applying filters to the Fourier-transformed image rather than the original spatial image.

Advantages of Frequency Domain Filtering:

1. **Efficient Convolution:**
    - Convolution in the spatial domain corresponds to multiplication in the frequency domain, which is computationally more efficient, especially for large filters.
2. **Design Simplicity:**
    - Filters like low-pass, high-pass, and band-pass can be easily designed and analyzed in the frequency domain using mathematical functions.
3. **Better Control Over Frequency Components:**
    - Specific frequency components (like noise or unwanted detail) can be directly targeted and attenuated or amplified.
4. **Useful for Periodic Noise Removal:**
    - Frequency domain filters are particularly effective in removing periodic noise using notch filters.
5. **Global Perspective:**
    - Frequency domain operations consider the entire image, allowing for globally consistent filtering effects.
6. **Flexible Filtering Operations:**
    - Frequency domain filtering can implement filters that are difficult to express in spatial terms (e.g., homomorphic filters, high-frequency emphasis).

Applications of Frequency Domain Filtering:

1. **Image Smoothing and Denoising:**
    - Using low-pass filters to remove high-frequency noise (Section 4.8).
2. **Image Sharpening and Enhancement:**
    - High-pass filters enhance edges and fine details in images (Section 4.9).
3. **Medical Imaging:**
    - Enhancing structures in CT, MRI, and X-ray images.
4. **Satellite and Remote Sensing:**

- Enhancing satellite imagery and removing periodic noise patterns.

5. **Pattern Recognition and Feature Extraction:**
   - Frequency analysis helps extract relevant features for classification tasks.
6. **Compression and Data Reduction:**
   - Frequency-domain transformations (e.g., DFT, DCT) are used in JPEG and MPEG compression.

M4Q4) Describe the Laplacian in the frequency domain for image sharpening. Derive the relevant equations and explain its implementation.

The **Laplacian operator** is a second-order derivative operator used to highlight regions of rapid intensity change, making it useful for **image sharpening**. In the **frequency domain**, the Laplacian has a specific form that simplifies its implementation via the Fourier transform.

The Laplacian in the Spatial Domain

The 2D Laplacian operator in the spatial domain is defined as:

$$\boxed{\nabla^2 f(x,y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}}$$

Laplacian in the Frequency Domain – Derivation

By applying the Fourier Transform properties of derivatives:

$\mathcal{F}\left\{\frac{\partial^2 f(x,y)}{\partial x^2}\right\} = -(2\pi u)^2 F(u,v)$

$\mathcal{F}\left\{\frac{\partial^2 f(x,y)}{\partial y^2}\right\} = -(2\pi v)^2 F(u,v)$

So the Fourier transform of the Laplacian is:

$\mathcal{F}\left\{\nabla^2 f(x,y)\right\} = -4\pi^2(u^2+v^2)F(u,v)$

Thus, applying the Laplacian in the frequency domain is equivalent to:

$\mathcal{F}\left\{\nabla^2 f(x,y)\right\} = H(u,v) \cdot F(u,v)$

Where:

$$H(u,v) = -4\pi^2(u^2 + v^2)$$

Image Sharpening Using the Laplacian (Implementation)

To sharpen an image $f(x,y)$, we **add** the Laplacian to the original image:

$$g(x,y) = f(x,y) + c \cdot \nabla^2 f(x,y)$$

Where $c = -1$ usually (so sharpening adds the negative Laplacian, enhancing edges).

In the frequency domain:

$$G(u,v) = F(u,v) + c \cdot H(u,v) \cdot F(u,v)$$

$$G(u,v) = [1 + c \cdot H(u,v)]\, F(u,v)$$

So the effective filter becomes:

$$H_{\text{sharp}}(u,v) = 1 + c \cdot H(u,v)$$

M4Q5) Explain Homomorphic filtering for image enhancement. Describe the process and derive the necessary expressions.

Homomorphic filtering is a frequency domain technique used for **image enhancement**, particularly to correct **non-uniform illumination** and to **enhance contrast and details**. It is based on separating the **illumination** and **reflectance** components of an image and selectively filtering them.

Image Formation Model

An image can be modeled as the product of illumination $i(x,y)$ and reflectance $r(x,y)$:

$$f(x,y) = i(x,y) \cdot r(x,y)$$

- **Illumination $i(x,y)$:** Generally varies slowly ⇒ low-frequency component.
- **Reflectance $r(x,y)$:** Changes abruptly ⇒ high-frequency component.

Transforming the Model

Because Fourier transform works on additive components, we **convert multiplication to addition** using the logarithm:

$$\ln f(x,y) = \ln[i(x,y) \cdot r(x,y)] = \ln i(x,y) + \ln r(x,y)$$

Homomorphic Filtering Process (Steps)

1. **Take the natural log** of the image:

   $$z(x, y) = \ln f(x, y)$$

2. **Compute the Fourier Transform** of $z(x, y)$:

   $$Z(u, v) = \mathcal{F}\{z(x, y)\}$$

3. **Apply a highpass filter** H(u, v) to suppress low-frequency (illumination) and boost high-frequency (reflectance):

   $$S(u, v) = H(u, v) \cdot Z(u, v)$$

4. **Compute the inverse Fourier Transform**:

   $$s(x, y) = \mathcal{F}^{-1}\{S(u, v)\}$$

5. **Exponentiate to recover the enhanced image**:

   $$g(x, y) = \exp[s(x, y)]$$

M4Q6) Given HSL = (356.99, 0.534, 0.3453), convert this into RGB model.

To convert from **HSL (Hue, Saturation, Lightness)** to **RGB**, we follow a standard algorithm. Here's how to convert:

Given:

- **H = 356.99°**
- **S = 0.534**
- **L = 0.3453**

Steps to Convert HSL → RGB

Step 1: Calculate C, X, and m

$$C = (1 - |2L - 1|) \cdot S = (1 - |2 \cdot 0.3453 - 1|) \cdot 0.534$$

$$= (1 - |0.6906 - 1|) \cdot 0.534 = (1 - 0.3094) \cdot 0.534 = 0.6906 \cdot 0.534 = 0.3687$$

Now compute:

$$H' = \frac{H}{60} = \frac{356.99}{60} \approx 5.9498$$

$X = C \cdot (1 - |H' \mod 2 - 1|) = 0.3687 \cdot (1 - |5.9498 \mod 2 - 1|)$

$H' \mod 2 = 1.9498 \Rightarrow |1.9498 - 1| = 0.9498$

$X = 0.3687 \cdot (1 - 0.9498) = 0.3687 \cdot 0.0502 \approx 0.0185$

$m = L - \frac{C}{2} = 0.3453 - \frac{0.3687}{2} = 0.3453 - 0.18435 = 0.16095$

Step 2: Determine RGB (before adding m)

Since $H' \approx 5.95$, it lies in **[5, 6)**, so:

- $R' = C$
- $G' = 0$
- $B' = X$

$R' = 0.3687, \quad G' = 0, \quad B' = 0.0185$

Now add m to get final RGB values:

$R = R' + m = 0.3687 + 0.16095 = 0.5297$

$G = G' + m = 0 + 0.16095 = 0.16095$

$B = B' + m = 0.0185 + 0.16095 = 0.17945$

Final RGB (Normalized [0–1] Scale):

$$\boxed{R = 0.5297, \quad G = 0.16095, \quad B = 0.17945}$$

Final RGB (0–255 Scale):

$$\boxed{R \approx 135, \quad G \approx 41, \quad B \approx 46}$$

M4Q7) Define the following terms: a) Hue b) Saturation c) Intensity d) Chromaticity.

a) Hue

- **Definition:** Hue refers to the **dominant color perceived** by the human eye and is associated with the **wavelength** of light.
- **Explanation:** It distinguishes colors such as **red, green, blue, or yellow**.
- **Measurement:** Represented as an **angle** (typically 0°–360°) on the color wheel.

- 0° → Red
- 120° → Green
- 240° → Blue

b) Saturation

- **Definition:** Saturation refers to the **purity or vividness of a color**, i.e., how much white light is mixed with the hue.
- **Explanation:**
  - **High saturation** → pure, vivid color
  - **Low saturation** → washed out, pale color (closer to gray)
- **Range:** From 0 (gray) to 1 (pure color)

c) Intensity

- **Definition:** Intensity is the **brightness or lightness** of a color, representing the **amount of light** emitted or reflected by an object.
- **Explanation:** Often computed as the **average** of the RGB components:
  $\text{Intensity} = \frac{R+G+B}{3}$
- **Important in:** Grayscale conversion and light perception

d) Chromaticity

- **Definition:** Chromaticity defines the **quality of a color regardless of brightness**. It is specified by two parameters: **hue** and **saturation**.
- **Explanation:** Two colors with different brightness but the same hue and saturation have the **same chromaticity**.
- **Chromaticity diagram** is used in color science to plot these values.

M4Q8) What is pseudocolor image processing, and how is it different from true colour image processing?

Pseudocolor image processing involves **assigning colors to grayscale images** to enhance interpretation and visual analysis. The goal is to represent different intensity levels (gray values) using **artificially assigned colors**, which makes the image more informative to human observers.

Key Techniques in Pseudocolor Processing:

1. **Intensity Slicing:**
    - The intensity range is divided into intervals, and each interval is mapped to a specific color.
    - Useful for highlighting specific ranges in an image (e.g., thermal or medical images).
2. **Intensity-to-Color Transformation:**
    - A continuous mapping from grayscale values to a color spectrum (e.g., blue → green → red).
    - Often implemented using lookup tables (LUTs).

True Color Image Processing:

In **true color processing**, images contain **actual color information** typically stored in the **RGB model**, where each pixel is defined by its **red, green, and blue** components.

- Each component usually has 8 bits ⇒ 24-bit color image (over 16 million colors).
- True color images come from sensors like digital cameras or scanners that capture real-world colors.

Key Differences:

Feature	**Pseudocolor**	**True Color**
Input Image	Grayscale	Color (RGB or other models)
Color Mapping	Artificially assigned	Captured directly from scene
Purpose	Visual enhancement	Realistic representation
Bit Depth	8-bit (input) → mapped to color	Typically 24-bit (8 bits per RGB channel)
Example Use Cases	Medical imaging, thermal imaging	Photography, video, graphics

M5Q1) Explain Inverse filter and Weiner filter with the help of equations. Explain the advantages of Weiner filter over inverse filter.

Inverse Filtering Definition:

Inverse filtering attempts to **reverse the effects of degradation**, assuming that the degradation function is known and that **no noise** is present.

Mathematical Model:

Let:

- $g(x, y)$: observed degraded image
- $h(x, y)$: degradation function (blur)
- $n(x, y)$: additive noise
- $f(x, y)$: original image

In the frequency domain:

$G(u, v) = H(u, v) \cdot F(u, v) + N(u, v)$

Inverse Filter assumes noise N(u, v) = 0:

$F(u, v) = \frac{G(u,v)}{H(u,v)}$

So, the restored image is obtained by:

$$\boxed{F_{\text{restored}}(u, v) = \frac{G(u, v)}{H(u, v)}}$$

Limitation:

- **Fails when** $H(u, v) \approx 0$ (division by small values amplifies noise)
- Sensitive to **noise** and **instability**

Wiener Filter Definition:

The Wiener filter is designed to **minimize the overall mean square error** between the estimated image and the original image. It accounts for both **degradation** and **noise**.

Mathematical Expression:

$$\boxed{F(u,v) = \left[\frac{H^*(u,v)}{|H(u,v)|^2 + \frac{S_N(u,v)}{S_F(u,v)}}\right] \cdot G(u,v)}$$

Where:

- $H^*(u,v)$: complex conjugate of degradation function
- $S_N(u,v)$: power spectrum of noise
- $S_F(u,v)$: power spectrum of original image

If $\frac{S_N(u,v)}{S_F(u,v)} = K$ (a constant), then:

$F(u,v) = \left[\frac{H^*(u,v)}{|H(u,v)|^2+K}\right] \cdot G(u,v)$

Advantages of Wiener Filter Over Inverse Filter

Feature	Inverse Filter	Wiener Filter
Noise Handling	Ignores noise	Handles noise statistically
Stability	Unstable near$H(u,v) = 0$	More stable due to regularization term KK
Error Minimization	No error control	Minimizes **mean square error**
Practical Use	Limited to ideal/noiseless cases	More effective in **real-world** scenarios

---

M5Q2) Explain how image degradation is estimated using (i) Observation (ii) Mathematical Modelling

Image degradation refers to the deterioration of image quality due to various factors such as noise, motion blur, or lens imperfections. Estimating the **degradation function** H(u,v)H(u,v) is a critical step in image restoration. There are two primary approaches:

(i) Estimation by Observation

- **Description**: This approach involves examining the degraded image and using external measurements or visual cues to deduce the degradation model.

- **Example**: If the image is blurred due to motion, the length and direction of the blur can often be estimated from the appearance of specific features (e.g., streaks).
- **Procedure**:
  - Capture images of known reference objects under the same degradation conditions.
  - Analyze repeated patterns of degradation (e.g., blurring direction, periodic noise).
- **Limitation**: Accuracy depends on the quality of observations and may not be precise without auxiliary data.

(ii) Estimation by Mathematical Modelling

- **Description**: This method involves creating a mathematical model that describes the physical process causing the degradation.
- **Example**:
  - For **motion blur**, the degradation function $H(u, v)$ can be modeled using the linear motion of the camera or object.
  - For **atmospheric turbulence**, wave propagation models are used.
- **Procedure**:
  - Define the system response (e.g., point spread function).
  - Use physical laws (like motion equations or optics) to derive $H(u, v)$.
- **Advantage**: More accurate and generalized; useful when system characteristics are known.
- **Limitation**: May be complex to derive and may not account for unpredictable factors.

M5Q3) Discuss the importance of Adaptive filters in image restoration system, highlight its working of Adaptive filters.

Importance of Adaptive Filters

1. **Spatially Varying Noise**: In real-world images, noise often varies across regions. Adaptive filters are ideal in such scenarios as they adjust locally.
2. **Preservation of Edges and Details**: Adaptive filters analyze local variance to **preserve edges and important features**, unlike mean or median filters which may blur them.
3. **Better Performance**: They provide **improved noise reduction** while minimizing distortion to the original signal.
4. **Flexibility**: Can switch between smoothing and preserving sharp features based on the **local image context**.

Working of Adaptive Filters

The basic principle involves computing **local statistics** such as **mean** and **variance** within a window around each pixel, and adjusting the filtering accordingly.

A widely used adaptive filter is defined as:

$$\hat{f}(x, y) = g(x, y) - \frac{\sigma_\eta^2}{\sigma_L^2(x,y)} [g(x, y) - \mu_L(x, y)]$$

Where:

- $\hat{f}(x, y)$: Restored pixel value
- $g(x, y)$: Noisy image pixel
- $\mu_L(x, y)$: Local mean
- $\sigma_L^2(x, y)$: Local variance
- σ_η^2: Noise variance (assumed known or estimated)

M5Q4) Explain minimum mean square error filtering method of restoring images.

The **Minimum Mean Square Error (MMSE) filter**, also known as the **Wiener filter**, is a powerful technique used in image restoration to **minimize the overall error** between the original image and the restored image in a **statistical sense**.

It works by considering both **degradation** and **noise** and attempts to **minimize the mean square error** between the estimated and the true image.

Mathematical Formulation

In the frequency domain, the MMSE-restored image $\hat{F}(u,v)$ is given by:

$$\boxed{\hat{F}(u,v) = \left[\frac{|H(u,v)|^2}{|H(u,v)|^2 + \frac{S_\eta(u,v)}{S_f(u,v)}}\right] \cdot \frac{G(u,v)}{H(u,v)}}$$

Where:

- $G(u,v)$: Degraded image in frequency domain
- $H(u,v)$: Degradation function (e.g., blur function)
- $S_\eta(u,v)$: Power spectrum of noise
- $S_f(u,v)$: Power spectrum of the original image
- $\hat{F}(u,v)$: Restored image in frequency domain

Key Characteristics

1. **Takes Noise into Account**: Unlike inverse filtering, MMSE filtering **considers noise** in the degradation model.
2. **Reduces Amplification of Noise**: The term

   $$\frac{|H(u,v)|^2}{|H(u,v)|^2 + \frac{S_\eta(u,v)}{S_f(u,v)}}$$

   prevents division by near-zero $H(u,v)$, avoiding noise amplification at high frequencies.
3. **Optimal in MSE Sense**: It is statistically **optimal** under the assumption that the signal and noise are uncorrelated, and both are stationary random processes.

********************************** EOF **********************************