# DIP ASSIGNMENT 3-5 SUMMARY

In Digital Image Processing (DIP), homomorphic filtering is a technique used to enhance images by simultaneously normalizing the brightness across an image and increasing contrast. It's particularly useful for images with non-uniform lighting, such as images taken under poor or uneven illumination.

What is Homomorphic Filtering?

Homomorphic filtering is a frequency domain filtering technique that aims to separate the illumination and reflectance components of an image and process them individually.

In an image:

- Illumination: slow varying (low frequency)
- Reflectance: fast varying (high frequency)

Goal: Suppress illumination component (low frequencies) and enhance reflectance component (high frequencies).

Image restoration (Models) in Digital Image Processing aims to recover a sharp image from one blurred by uniform linear motion using deconvolution techniques.

1. Blur Modeling:

Motion blur is modeled using a blurring kernel (PSF) — typically a 1D filter in the direction of motion.

2. Deconvolution Methods:

- Inverse Filtering: Simple but sensitive to noise.
- Wiener Filtering: Minimizes error by considering noise and blur, more robust.
- Lucy-Richardson: Iterative, handles noise well, based on Bayesian theory.
- Blind Deconvolution: Estimates both the blur kernel and image when the kernel is unknown.

3. Spatial Domain Methods:

Use operations like traveling wave equations or Hough transforms to estimate and correct blur directly in the spatial domain.

4. Frequency Domain Methods:

Blurring appears as low-pass filtering; deconvolution in this domain targets and reverses specific frequency losses.

| Feature | Image Enhancement | Image Restoration |
|---|---|---|
| **Purpose** | Improves **visual appearance** for human viewing | Recovers the **original image** from a degraded one |
| **Approach** | **Subjective** – depends on what looks better | **Objective** – aims to restore original scene |
| **Input Image** | May be clean or noisy | Always a **degraded** or noisy image |
| **Techniques Used** | Contrast stretching, histogram equalization, filtering | Deconvolution, inverse filtering, Wiener filtering |
| **Focus** | Highlighting features like edges, brightness, etc. | Removing known degradations (blur, noise, etc.) |
| **User Role** | User-defined preference plays a big role | Based on **mathematical models** and system knowledge |
| **Example** | Increasing contrast to make features stand out | Removing motion blur from a shaky photo |

image degradation refers to the process by which an image loses quality due to various factors during acquisition, processing, or transmission. Image restoration is the process of attempting to recover the original, high-quality image from the degraded version. This involves modeling the degradation and applying corrective techniques.

Causes of Degradation:

Degradation can occur due to various factors, including:

- Noise: Random variations in brightness or color, often introduced during image acquisition or transmission.
- Blur: Loss of sharpness, caused by motion, out-of-focus lenses, atmospheric conditions, or other factors.
- Sampling and Quantization: Loss of detail due to the discrete nature of digital image representation.
- Geometric Transformations: Distortion or misalignment of the image.

- Degradation Model:
  The degradation process can be mathematically modeled, often using convolution with a degradation function (H) and the addition of noise ($\eta$). The degraded image (g) can be represented as:

  $$g(x, y) = H * f(x, y) + \eta(x, y)$$

  , where f(x, y) is the original image.

Reverse filtering, also known as inverse filtering, aims to recover an original image from a degraded or blurred version, often using a known degradation function. It's essentially applying a filter that undoes the effect of a previous filter, often modeled as a blur.

## Concept of Reverse Filtering :

If a blurred image $g(x, y)$ is formed by convolving the original image $f(x, y)$ with a blur function $h(x, y)$, then in the frequency domain:

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

To recover the original image, we simply divide:

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

Then take the inverse Fourier Transform of $F(u, v)$ to get the restored image $f(x, y)$.

## Noise Models in DIP

1. Gaussian Noise
2. Rayleigh Noise
3. Gamma Noise ( Erlang )
4. Exponential Noise
5. Uniform Noise
6. Impulse Noise ( Salt and Pepper )

## Examples of Filters

1. Linear Filter – Mean (Averaging) Filter

- How it works: Replaces each pixel with the average of its neighbors.
- Best for: Gaussian or uniform noise

2. Non-Linear Filter – Median Filter

- How it works: Replaces each pixel with the median value of the neighborhood.
- Best for: Salt and pepper noise

- Preserves edges: Better than mean filter

## What is Noise PDF (Probability Density Function)?

A Noise PDF describes the statistical distribution of noise pixel values in an image. It tells us how likely each noise value is to occur and helps in selecting the right restoration or filtering method.

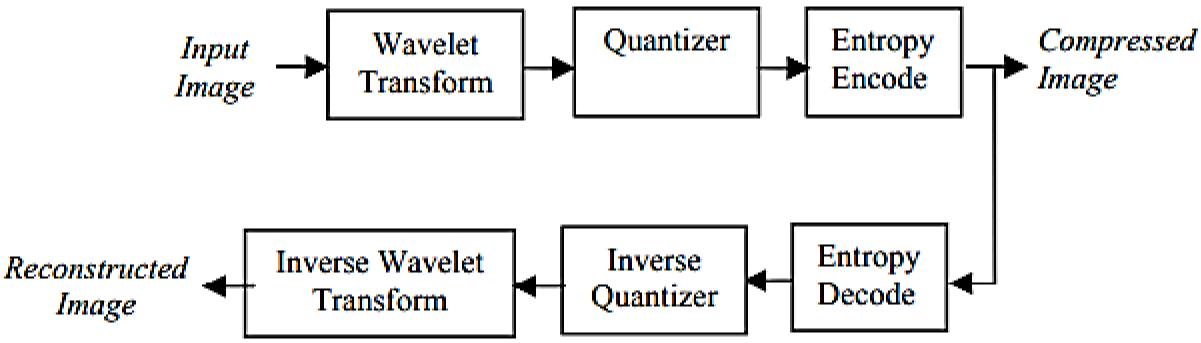
| Feature | Gaussian Noise | Impulse Noise (Salt & Pepper) |
|---|---|---|
| **Definition** | Continuous noise with values from a normal distribution | Random occurrence of extreme black and white pixels |
| **Appearance** | Grainy, smooth noise spread over the image | Random black and white dots |
| **PDF Shape** | Bell-shaped Gaussian curve | Two sharp spikes (one at black, one at white) |
| **Cause** | Sensor noise, poor lighting, thermal noise | Faulty transmission, dead pixels, bit errors |
| **Value Range** | Around a mean ($\mu$) with variance ($\sigma^2$) | Only max (255) and min (0) values |
| **Best Filter** | Mean filter, Gaussian filter, Wiener filter | **Median filter** |

# *ASSIGNMENT 4*

Lossy compression in Digital Image Processing (DIP) is a technique that reduces file size by permanently discarding data that is deemed less crucial to the image's perceived quality. This means the compressed image will not be an exact replica of the original, but the difference might not be noticeable to the human eye, especially at a glance. Popular lossy compression algorithms include Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT), and Transform Coding.

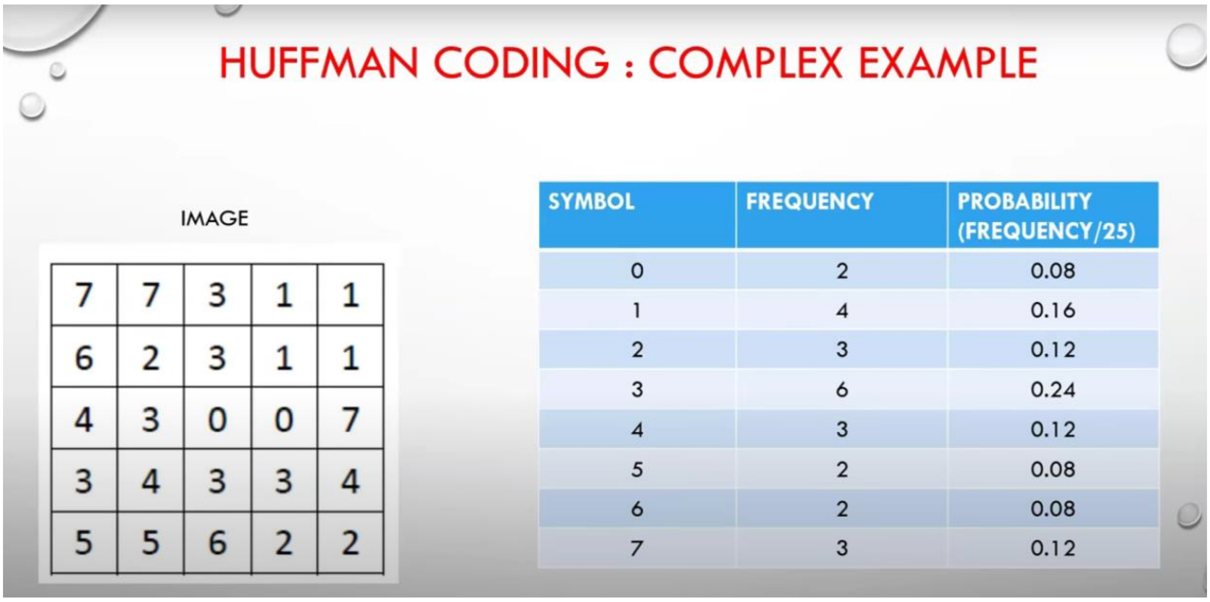
- Algorithms:
  - Discrete Cosine Transform (DCT): DCT is a fundamental technique in JPEG compression, transforming the image data into frequency components, where high-frequency details are more easily discarded without significant visual impact.
  - Discrete Wavelet Transform (DWT): DWT analyzes images at different scales, allowing for selective removal of less important details while preserving essential information.
  - Transform Coding: A general category of techniques that involve transforming data, quantizing it (reducing precision), and then reconstructing the image.

Transform-based compression works by converting image data into another domain (usually frequency), reducing redundancies, and then encoding the compressed data efficiently.



The JPEG standard uses the Discrete Cosine Transform (DCT) primarily due to its energy compaction property and its computational efficiency.

**Huffman Coding Grid Type Example :**



HUFFMAN CODING : COMPLEX EXAMPLE

IMAGE

| | | | | |
|---|---|---|---|---|
| 7 | 7 | 3 | 1 | 1 |
| 6 | 2 | 3 | 1 | 1 |
| 4 | 3 | 0 | 0 | 7 |
| 3 | 4 | 3 | 3 | 4 |
| 5 | 5 | 6 | 2 | 2 |

| SYMBOL | FREQUENCY | PROBABILITY (FREQUENCY/25) |
|---|---|---|
| 0 | 2 | 0.08 |
| 1 | 4 | 0.16 |
| 2 | 3 | 0.12 |
| 3 | 6 | 0.24 |
| 4 | 3 | 0.12 |
| 5 | 2 | 0.08 |
| 6 | 2 | 0.08 |
| 7 | 3 | 0.12 |

*( Then follow the same Standard Steps… )*

**Short Notes :**

a) Interpixel Redundancy

- Definition: Interpixel redundancy occurs when neighboring pixels in an image are similar or correlated.
- Explanation: Since adjacent pixels often have nearly the same value (especially in smooth areas), storing all of them individually is redundant.
- Example: In a sky image, most pixels are light blue and very similar—so we don't need to store each value separately.

- Compression: Techniques like predictive coding or differential encoding are used to remove interpixel redundancy.

---

b) Psychovisual Redundancy

- Definition: This refers to image details that are not easily perceived by the human eye, and thus can be safely removed.
- Explanation: The human eye is less sensitive to high-frequency components and color differences than brightness changes.
- Example: Removing slight texture or color variations may not affect perceived quality.
- Compression: Used in lossy compression like JPEG, where visually unimportant data is discarded to reduce size.

---

c) JPEG Compression

- Definition: JPEG is a lossy image compression standard that uses the Discrete Cosine Transform (DCT).
- Steps:
    1. Image is divided into 8x8 blocks.
    2. DCT is applied to each block to convert spatial values to frequency.
    3. Quantization removes less important frequencies.

4. Entropy encoding (like Huffman) compresses the quantized values.

- Use: Widely used in digital cameras, web images, and photo storage due to high compression ratio with good quality.

---

## d) Coding Redundancy

- Definition: Coding redundancy happens when symbols are represented with more bits than necessary.
- Explanation: For example, assigning a fixed-length code to all characters even when some occur more frequently.
- Example: Using 8 bits to encode both 'E' (very frequent) and 'Z' (rare) is inefficient.
- Compression: Huffman Coding and Arithmetic Coding remove coding redundancy by assigning shorter codes to frequent symbols.

| Feature | Lossy Compression | Lossless Compression |
|---|---|---|
| **Definition** | Removes **some data** permanently to reduce size | Compresses data **without losing any information** |
| **Data Recovery** | **Original data can't be perfectly restored** | **Original data can be perfectly recovered** |
| **File Size Reduction** | Higher compression ratio, **smaller file sizes** | Lower compression ratio, **larger file sizes** |
| **Used For** | Images, audio, video (e.g., JPEG, MP3, MP4) | Text, documents, medical images (e.g., PNG, ZIP) |
| **Quality** | Quality **may degrade** after compression | **No loss** in quality |
| **Speed** | Usually faster due to aggressive reduction | May be slower due to exact data preservation |

# ASSIGNMENT 5

Short Notes :

## a) Image Segmentation

- Definition: Image segmentation is the process of dividing an image into meaningful regions or objects.
- Purpose: To simplify or change the representation of an image for easier analysis.
- Techniques: Thresholding, edge-based segmentation, region-based segmentation, clustering (like K-means).
- Example: Separating the foreground object from the background in an image.

## b) Edge Linking

- Definition: Edge linking connects discontinuous edge pixels to form complete and meaningful boundaries.
- Methods:
  - Local: Connect neighboring pixels with similar gradient direction and strength.
  - Global: Use entire image information (e.g., Hough Transform, graph-based methods).
- Use: Important in object detection and shape recognition.

## c) Thresholding

Thresholding is a simple yet powerful image segmentation technique used to separate objects from their background based on pixel intensity values. A threshold value is selected, and pixels with intensity values above (or below) this threshold are assigned to one category (e.g., object), while the remaining pixels are assigned to another category (e.g., background). The result is a binary image.

- Types:
  - Global Thresholding: One threshold for the whole image.
  - Adaptive Thresholding: Different thresholds for different regions.
- Example: Converting a grayscale image to binary by setting all pixels > 128 to 1 and the rest to 0.

d) Boundary Linking

Boundary linking is closely related to edge linking but often focuses specifically on connecting the boundaries of segmented regions. After an image has been segmented into distinct regions, the boundaries of these regions might have gaps or discontinuities. Boundary linking algorithms aim to close these gaps and create continuous closed contours that accurately represent the shapes of the segmented objects.

Derivative operators, particularly first and second derivatives, are Valuable tools in image segmentation for edge detection, which is a crucial step in separating image regions. Image segmentation aims to divide an image into distinct regions or objects, based on properties like intensity, color, or texture.

How Derivative Operators Help:

- Edge Detection:

    - First Derivative (Gradient): Measures the rate of change in intensity between neighboring pixels. High gradient values indicate abrupt changes, suggesting edges where different objects or regions meet.
    - Second Derivative (Laplacian): Identifies points where the intensity changes most rapidly, which can be used to detect edges and even locate the center of thick edges.
- Region Boundary Identification:
    - Derivative operators help in identifying boundaries between regions by highlighting areas where there are significant changes in intensity or other image properties.
- Point Detection:
    - The second derivative, specifically the Laplacian operator, is used to detect isolated points in an image.

The Role of Segmentation:

Image segmentation is a fundamental process in computer vision with various applications:

- Object Recognition:

Segmentation helps isolate objects in an image, making it easier to identify and classify them.

- Image Analysis:

Segmented images allow for more focused analysis of specific regions or objects.

- Medical Imaging:

Segmentation is used in medical imaging to identify and analyze tissues, organs, or tumors.

- Robotics and Autonomous Systems:

Segmentation is used to understand the environment and navigate objects.

- Content-Based Image Retrieval:

Segmentation can help in organizing and searching images based on their content.

Two common boundary region schemes in Digital Image Processing (DIP) are boundary tracing and chain coding.

1. Boundary Tracing:

- Concept:

Boundary tracing algorithms identify the boundary pixels of a region by systematically following the edge of the region. This involves moving from one boundary pixel to its neighbor, typically in a clockwise or counter-clockwise direction, until the entire boundary is traversed.

2. Chain Coding:

- Concept:

Chain coding represents the boundary as a sequence of line segments, where each segment is assigned a direction (eg, North, Northeast, East, etc.). This compact representation allows for efficient storage and manipulation of the boundary.

## Region-Based Segmentation

In this segmentation, we grow regions by recursively including the neighboring pixels that are similar and connected to the seed pixel. We use similarity measures such as differences in gray levels for regions with homogeneous gray levels. We use connectivity to prevent connecting different parts of the image.

### ✅ Example of Region Growing:

Let's say you have the following grayscale image:

```
100 101 102  50  52
 98 100  99  48  49
101  99  97  47  46
 10  12  11 200 199
```

If you pick a **seed pixel at (0,0)** with value `100` and set the threshold to `±5`, region growing will group all connected pixels between `95–105`. Resulting region:

```nginx
R  R  R  .  .
R  R  R  .  .
R  R  R  .  .
.  .  .  .  .
```

(Here `R` means included in the region.)

## Need for Image Segmentation:

- Object Recognition:

Segmentation helps isolate objects of interest, making them easier to identify and classify.

- Scene Understanding:

By partitioning the image into regions, segmentation can help understand the context and relationships between objects in a scene.

- Feature Extraction:

Segmentation can extract key features like edges, lines, and points, which are essential for subsequent analysis.

- Image Compression:

Segmentation can help reduce redundancy in an image, allowing for more efficient compression without significant loss of important details.

- Medical Imaging:

Segmentation is crucial in medical imaging for tasks like organ segmentation, lesion detection, and 3D reconstruction of anatomical structures.

Edge Detection:

- Edges are the boundaries between regions of an image where there are significant changes in intensity.
- They are detected using techniques that analyze the gradient or derivatives of the image intensity.
- Examples of edge detection algorithms include Sobel, Prewitt, and Canny.

Line Detection:

- Line detection identifies straight lines in an image, often by searching for lines that connect edge points.
- The Hough transform is a popular method for line detection, which votes for potential lines based on edge points.

- Convolution-based techniques, using masks that are sensitive to different line orientations, can also be used.

Point Detection:

- Point detection aims to identify specific points, often corners, which are important features for image analysis.
- Corner detection techniques look for points where the image intensity changes significantly in multiple directions, as opposed to a single edge.
- Point detection can be used for tasks like object recognition, matching features between images, and determining object pose.