Video Signals

4. Image Spatial Processing

Unat Teksen

1 Fundamentals of Spatial Processing

1.1 Definition

Spatial processing methods work directly in the spatial domain, which is the plane of image pixels. This is distinct from frequency domain processing (Fourier Transform).

The general mathematical model for spatial processing is:

$$g(x, y) = T[f(x, y)] \tag{1}$$

Where:

- $f(x, y)$ is the input image.
- $g(x, y)$ is the processed output image.
- T is an operator defined over some neighborhood of (x, y).

1.2 The Concept of a "Mask"

The operator T is typically implemented using a square sub-image, often referred to as:

- **Mask**
- **Kernel**
- **Template** or **Window**

This mask moves across the image (convolution or correlation). At each pixel location, the response is calculated using the pixels strictly within the mask's neighborhood.

2 Linear Spatial Filters

Linear filtering involves the convolution of the image with a constant matrix (kernel).

2.1 Smoothing Filters (Low-Pass)

These filters attenuate high-frequency components (edges/noise) and pass low-frequency components.

1. Averaging (Box) Filter

- **Operation:** Replaces the value of every pixel with the average value of its neighbors.
- **Effect:** Blurs the image and reduces sharp transitions.
- **Kernel Example (3 × 3):**

$$H_{avg} = \frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix} \tag{2}$$

2. Gaussian Filter

- **Operation:** A weighted average where pixels closer to the center contribute more.
- **Shape:** Based on the 2D Gaussian distribution function (Bell curve).
- **Advantage:** Provides gentler smoothing and is naturally **isotropic** (rotationally symmetric).

2.2 The Separable Property

This is a critical property for optimizing the performance of linear filters.

Definition: A 2D filter kernel $H(x, y)$ is *separable* if it can be expressed as the outer product of two 1D vectors (a column vector v and a row vector h).

$$H = v \times h \tag{3}$$

Mathematical Example: Consider a generic matrix being decomposed:

$$\begin{bmatrix}1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1\end{bmatrix} = \begin{bmatrix}1\\ 2\\ 1\end{bmatrix} \times \begin{bmatrix}1 & 2 & 1\end{bmatrix} \tag{4}$$

Computational Complexity Analysis: Why is separability important?

1. **Standard 2D Convolution:** For an image of size $M \times N$ and a kernel of size $K \times K$:

   - Operations $\approx M \times N \times K^2$ multiplications.

2. **Separable 1D Convolution:** We first convolve columns with the column vector $(K \times 1)$, then convolve rows with the row vector $(1 \times K)$.

   - Operations $\approx M \times N \times 2K$ multiplications.

Conclusion: For a standard kernel (e.g., $K = 3$), the speedup is moderate (9 ops vs 6 ops). However, for large smoothing kernels (e.g., $K = 15$), the difference is massive (225 ops vs 30 ops). Gaussian and Box filters are separable.

3 Sharpening Filters (High-Pass)

Sharpening focuses on emphasizing intensity transitions. Since averaging (integration) blurs images, differentiation (derivatives) sharpens them.

3.1 Derivative Properties

- **First Derivative:** Detects the presence of an edge (non-zero at ramp).
- **Second Derivative:** Detects the onset and end of an edge (zero crossing). It is very sensitive to fine details and noise.

3.2 The Laplacian Operator

The Laplacian is the most common isotropic operator for sharpening.

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \tag{5}$$

Discrete Approximation Masks: Common 3×3 masks used to implement the Laplacian:

$$H_1 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad H_2 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix} \tag{6}$$

Note: The sum of coefficients in these high-pass filters is always 0, meaning constant intensity areas result in 0 output (black).

3.3 Image Sharpening Implementation

The Laplacian image contains only the edges. To sharpen the original image, we add or subtract this edge map:

$$g(x, y) = f(x, y) + c \cdot [\nabla^2 f(x, y)] \tag{7}$$

Where c controls the strength of the sharpening.

- If the center coefficient of the mask is negative, we **subtract**.
- If the center coefficient is positive, we **add**.

4 Non-Linear Spatial Filters

Non-linear filters do not rely on linear convolution. Instead, they use the ranking of pixel values within the neighborhood. These are collectively called **Order-Statistic Filters**.

4.1 The Median Filter

This is the most famous non-linear filter used in image processing.

Algorithm: For every pixel (x, y) in the image:

1. Extract the values of all pixels in the $n \times n$ neighborhood.
2. **Sort** these values in ascending order.
3. Select the **Median** (the middle value) from the sorted list.
4. Replace the center pixel with this median value.

4.2 Application: Salt and Pepper Noise

Median filters are specifically designed to handle "Impulse Noise," often called **Salt and Pepper Noise**.

- **Salt Noise:** Random white pixels (value 255).
- **Pepper Noise:** Random black pixels (value 0).

Why Median Works: An averaging filter would include the 0 or 255 values in the calculation, smearing the noise out. A median filter forces the value to be one of the existing neighbors. Since noise is typically an outlier (extreme value at the start or end of the sorted list), the median (middle value) is almost always a valid, clean pixel.

4.3 Comparison: Linear vs Non-Linear

Linear (Smoothing)	**Non-Linear (Median)**
Calculates average	Calculates median/rank
Blurs edges	**Preserves edges**
Smears impulse noise	Eliminates impulse noise
Efficient (Separable)	Computationally intensive (Sorting)

Table 1: Comparison of Smoothing Techniques