CMP362/CMPN446: Image Processing and Computer Vision

Lecture 03: Image Preprocessing
Local Preprocessing- Smoothing

Mayada Hadhoud

Computer Engineering Department

Cairo University

Agenda

- Introduction
- Image Noise
- Local Preprocessing Techniques-Smoothing
  - Using Linear Filters (Averaging, Gaussian)
  - Using Non- Linear Filters (Min, Max , Median)
  - Using Limited Data Validity
  - Using Rotating Masks

Introduction

- Pre-processing methods (also called **filters**) use a **small neighborhood** of a pixel in an input image to get a new brightness value in the output image.
- Local pre-processing methods can be divided into **two groups according to the goal** of the processing:
  - **Smoothing: suppresses high frequencies**
  - **Gradient operators: suppresses low frequencies**

Introduction

- Clearly, **smoothing** and **gradient operators** have **conflicting aims**.
- As **Smoothing** suppresses noise or other small fluctuations in the image, unfortunately, smoothing also **blurs all sharp edges** that bear important information about the image.
- As **Gradient** operators suppress low frequencies in the frequency domain.

Image Noise

- What kind of noise?
  - Additive Noise (Ex. Gaussian)

    Random noise $n(i, j)$ *added to pixel value* $I(i, j)$

    $$\hat{I}(i, j) = I(i, j) + n(i, j)$$

  - Impulsive (Salt and Pepper)

Principal sources of Gaussian noise in digital images arise during
acquisition e.g. sensor noise caused by poor illumination and/or high
temperature, and/or transmission
An image containing salt-and-pepper noise will have dark pixels in
bright regions and bright pixels in dark regions. [8] This type of noise
can be caused by analog-to-digital converter errors, bit errors in
transmission

Figure: Random noise and Impulse noise.

Image Smoothing

- Calculation of the new value is **based on averaging** of brightness values.
- problem of **blurring** sharp edges
- **Edge preserving** techniques are based on the general idea that the average is computed only from those points in the neighborhood which have similar properties to the processed point

Image Smoothing Techniques

- **Averaging to eliminate noise**
  - For n images of the same scene, **smoothing** can be accomplished **without blurring** the image by:

$$f(i,j) = \frac{1}{n}\sum_{k=1}^{n} g_k(i,j)$$

Image Smoothing Techniques

- **Averaging to eliminate zero mean noise**
  - images g1,…,g{n} contain noise values
    V{1},……, V{n}
  - hence

$$\frac{g_1 + \ldots + g_n}{n} + \frac{v_1 + \ldots + v_n}{n}$$

=0

Image Smoothing Techniques

Original image

Corrupted image

8 images

16 images

64 images

128 images

Image Smoothing Techniques

- **Averaging using averaging filters (Linear Filters)**
  - In many cases only one image with noise is available, and averaging is then realized in a local neighborhood by using the convolution mask(**Mean Filter**):

$$h = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

  - **blurring** of edges is a serious disadvantage.

Linear Filters use convolution to apply the filter

Original Image

3x3 Filter

Original Image

5x5 Filter

Original Image

11x11 Filter

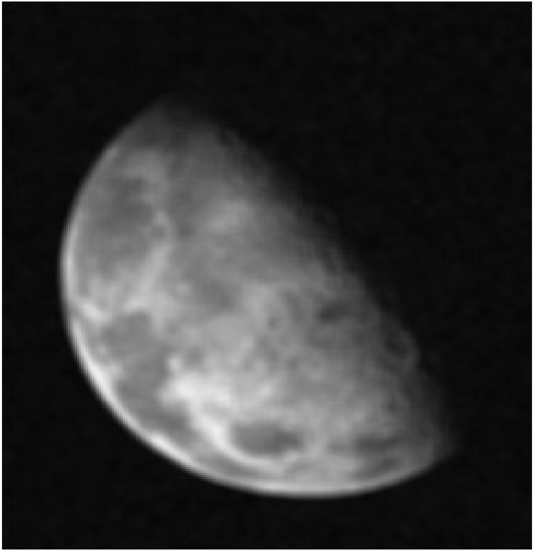


Image Smoothing Techniques

- **Averaging using averaging filters (Linear Filters)**
  - **Gaussian Filter**

- Gaussian distribution $G(x,y) = \frac{1}{2\pi} e^{-\frac{1}{2} \cdot \frac{x^2+y^2}{\sigma^2}}$

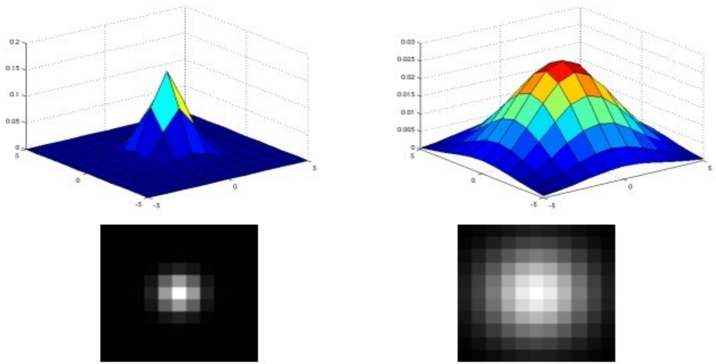


Figure: Gaussian distribution for $\sigma = 1$ and $\sigma = 2.5$ and corresponded Gaussian kernels of the size 11×11.

Gaussian Filter Examples

$$h = \frac{1}{10}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

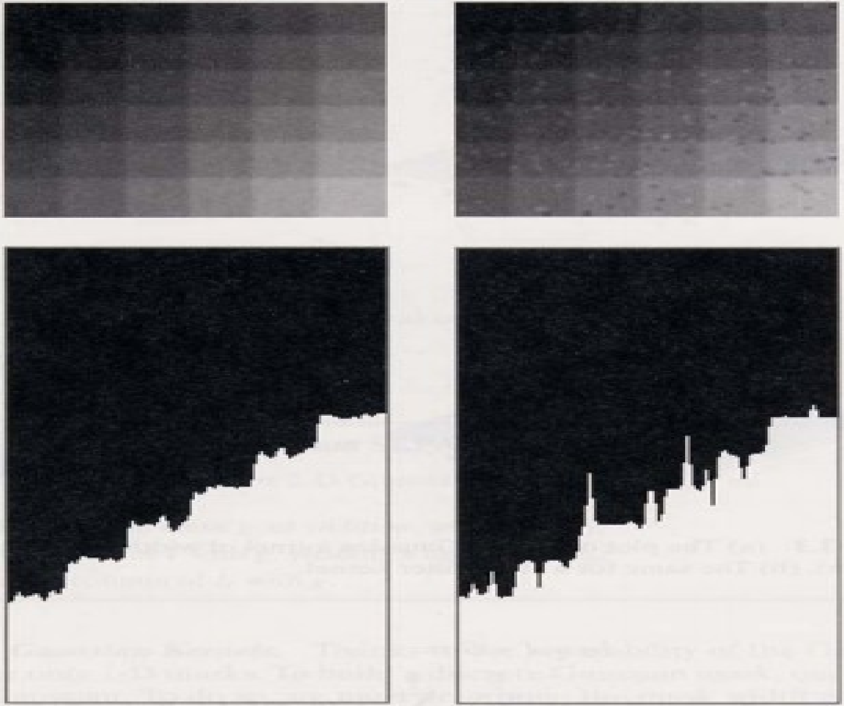
$$h = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

11×11 with $\sigma = 1$

Original Image

Original Image

11×11 with $\sigma = 2.5$.



(a) (b)

Figure 3.2 (a) Results of applying Gaussian filtering (kernel width 5 pixel, $\sigma = 1$) to the "checkerboard" image corrupted by Gaussian noise, and grey-level profile along a row. (b) Same for the "checkerboard" image corrupted by salt and pepper noise.

Image Smoothing Techniques

- **Averaging with NonLinear Filters**
  - Min, Max Filters

- Max filter - The maximum value replaces the current pixel
  $\Rightarrow$ brighter image
- Min filter - The minimum value replaces the current pixel
  $\Rightarrow$ darker image

Figure: Original image; Max filter with 3×3 kernel and Min filter with 3×3 kernel.

Nonlinear filters don’t use convolution
It performs an operation to the pixels in the mask like Max, Min, Median

Image Smoothing Techniques

- **Averaging with NonLinear Filters**
  - **Median Filter**
    - Good for **impulsive noise** (Salt and Pepper Noise)
    - Sharp edges are kept as no new values are created
    - **How it works**:
      - The median filter works by moving through the image pixel by pixel, replacing each value with the median value of neighboring pixels

    - **Disadvantages:** **fine details are erased as very thin lines and sharp corners are damaged**
    - Time-consuming to sort values compared to average values
    - **By choosing other shapes than rectangular neighborhoods (thin lines may be preserved)**

2D Median filtering example using a 3 x 3 sampling window:

Keeping border values unchanged

Sorted: 0,0,1,1,1,2,2,4,4

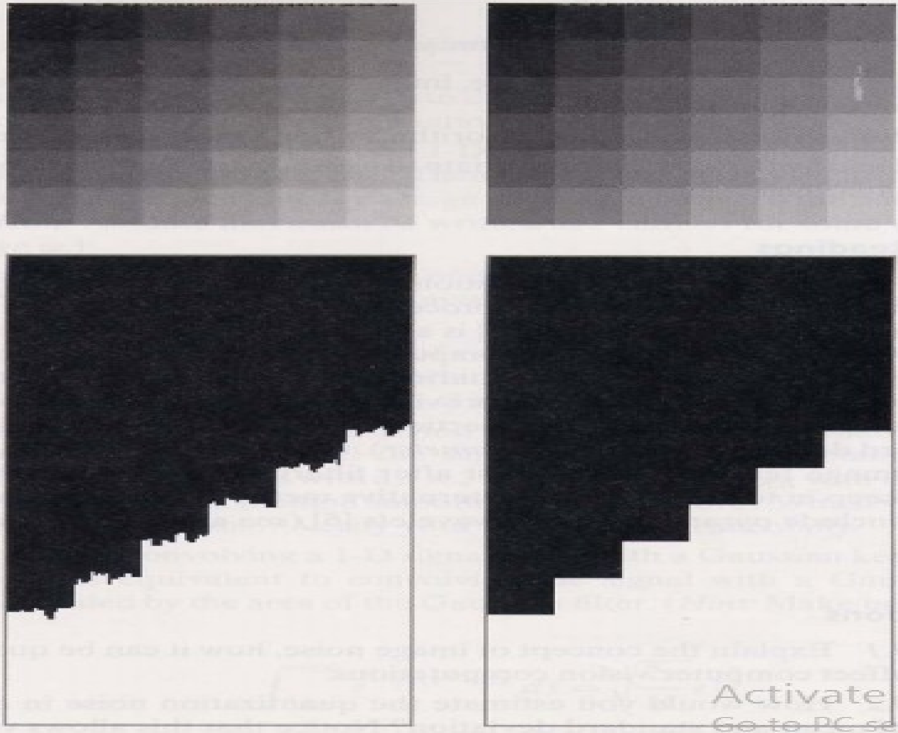
Input

1	4	0	1	3	1
2	2	4	2	2	3
1	0	1	0	1	0
1	2	1	0	2	2
2	5	3	1	2	5
1	1	4	2	3	0

Output

1	4	0	1	3	1
2	1	1	1	1	3
1	1	1	1	2	0
1	1	1	1	1	2
2	2	2	2	2	5
1	1	4	2	3	0

Figure: Mean and Median filter with respect to: Random noise (top row) and Impulse noise (bottom row).



(a) (b)

Image Smoothing Techniques

- **Averaging with limited data validity**
  - Avoids blurring by averaging only those **pixels which satisfy some criterion** and prevent involving pixels that are part of a separate feature.
  - A very simple criterion is to use only pixels in the original image with brightness in a **predefined interval** [min,max].

$$h(i,j) = \begin{cases} 1 & \text{for } g(m+i, n+j) \in [min, max] \\ 0 & \text{otherwise} \end{cases}$$

Image Smoothing Techniques

- **Averaging with limited data validity**
  - **Using edge strength**, e.g. only pixels with gradient below a certain value are averaged. The magnitude of some gradient operator is first computed for the entire image.

Image Smoothing Techniques

- Averaging using a rotating mask
  - Avoids edge blurring by searching for the homogeneous part around the current pixel
  - average is calculated only within the homogeneous region
  - a brightness dispersion σ^2 is used as the region homogeneity measure

Image Smoothing Techniques

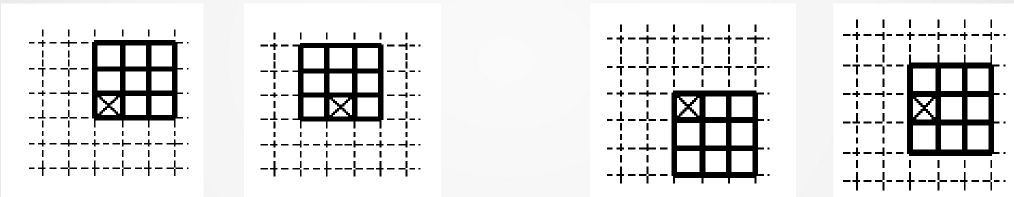
- **Averaging using a rotating mask**
  - The dispersion **σ^2 is given by:**

$$\sigma^2 = \frac{1}{n}\left(\sum_{(i,j)\in R}\left(g(i,j) - \frac{1}{n}\sum_{(i,j)\in R} g(i,j)\right)^2\right)$$

$$\frac{1}{n}\left(\sum_{(i,j)\in R}\left(g(i,j)\right)^2 - \frac{\left(\sum_{(i,j)\in R} g(i,j)\right)^2}{n}\right)$$

Image Smoothing Techniques

- **Averaging using a rotating mask**
  - **Rotated Masks:**



1 2 7
8

Image Smoothing Techniques

- **Averaging using a rotating mask**
  - **Algorithm:**

1. Consider each image pixel (i, j).
2. Calculate dispersion in the mask for all possible mask rotations about pixel (i, j) according to equation
3. Choose the mask with minimum dispersion.
4. Assign to the pixel $g(i, j)$ in the output image the average brightness in the chosen mask.