Image Processing and Visual Communications

Nonlinear Image Filtering

Zhou Wang

Dept. of Electrical and Computer Engineering
University of Waterloo

Previous Lectures

- **Spatial Domain Linear Filters**
  - Smoothing: Averaging, Gaussian
  - Sharpening
  - ……
- **Frequency (2D-DFT) Domain Filters**
  - Lowpass, highpass, bandpass
  - Orientation selective
  - Orientation + radial selective
  - …..
- **Linear Image Restoration Filters**
  - Inverse, pseudo-inverse, radially-limited inverse
  - Wiener, Wiener denoising
  - ……

All Linear !

Nonlinear Filtering

- **Motivation: Limitation of Linear Filters**
  - Frequency shaping

    enhance some frequency components and suppress the others
  - For individual frequency component, cannot differentiate its "desirable" and "undesirable" parts

- **Nonlinear Filters**
  - Cannot be expressed as convolution
  - Cannot be expressed as frequency shaping

- **"Nonlinear" Means Everything** (other than linear)
  - Need to be more specific
  - Often heuristic
  - We will study some "nice" ones

Impulsive (Salt & Pepper) Noise

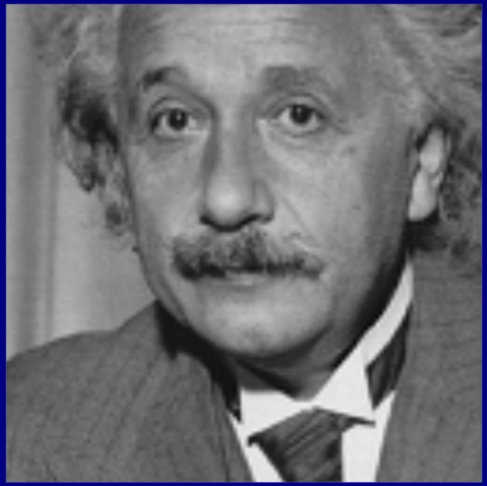
- **Definition**
  - Each pixel in an image has a probability p_a or p_b of being contaminated by a white dot (salt) or a black dot (pepper)

X: noise-free image, *Y*: noisy image

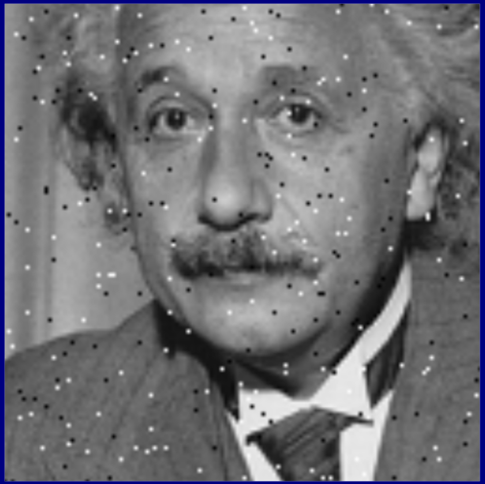
$$Y(m,n) = \begin{cases} 255 & \text{with probability } p_a \\ 0 & \text{with probability } p_b \\ X(m,n) & \text{with probability } 1 - p_a - p_b \end{cases}$$

noisy pixels

clean pixels



add salt & pepper noise



Median Filters

- **Order Statistics (OS)**
  - Given a set of numbers $\boldsymbol{x} = \{x_1, x_2, \cdots, x_{2M+1}\}$

    max value

    Denote the OS as $\boldsymbol{x}_{OS} = \{x_{(1)}, x_{(2)}, \cdots, x_{(2M+1)}\}$

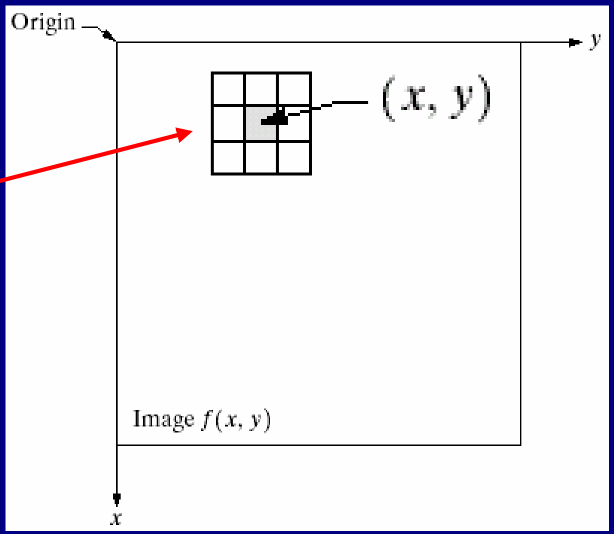
    such that $x_{(1)} \le x_{(2)} \le \cdots \le x_{(M+1)} \le \cdots \le x_{(2M+1)}$

    min value

    middle value

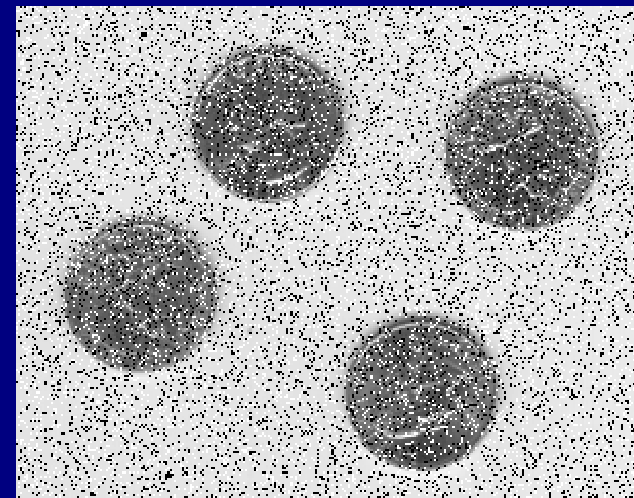
- **Median**
  - Define $Median\{x_1, x_2, \cdots, x_{2M+1}\} = x_{(M+1)}$

- **Applying Median Filters to Images**
  - Use sliding windows (similar to spatial linear filters)
  - Typical windows: 3x3, 5x5, 7x7, other shapes ……



Median Filters

original



noisy ($p_a = p_b = 0.1$)

median filtered 3x3 window

median filtered 5x5 window

From MATLAB sample images

Iterative Median Filters

- **Idea:** repeatedly apply median filters

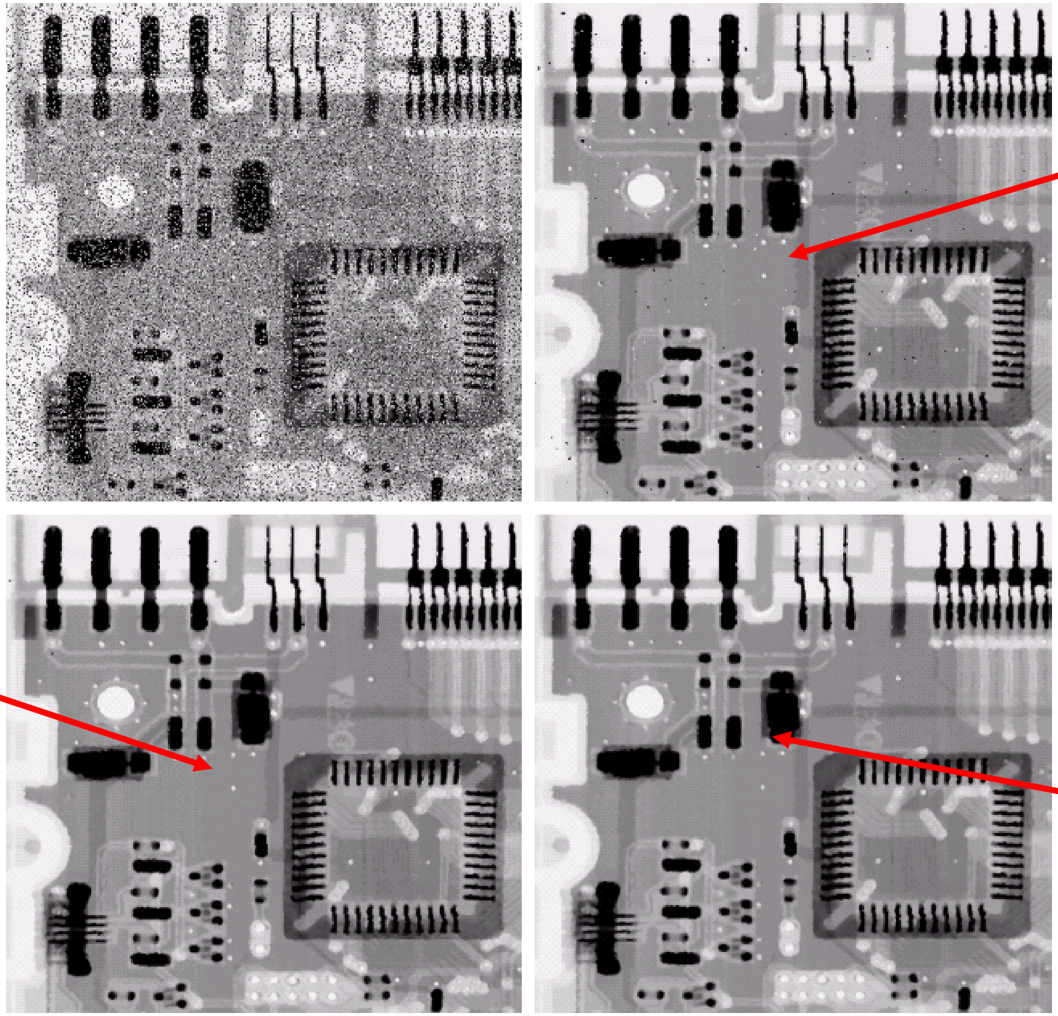
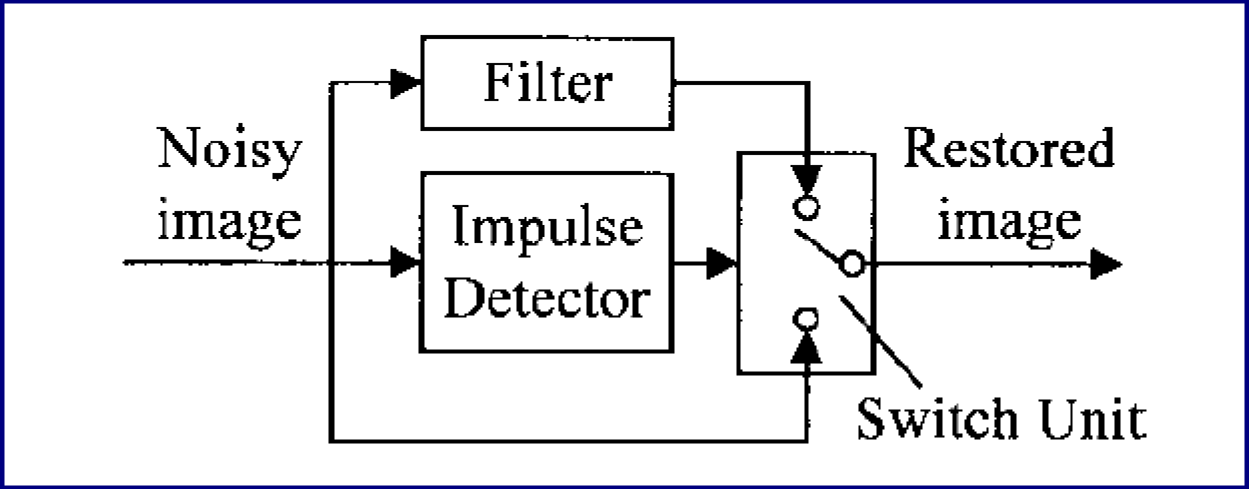


FIGURE 5.10 (a) Image corrupted by salt-and-pepper noise with probabilities $P_a = P_b = 0.1$. (b) Result of one pass with a median filter of size 3×3. (c) Result of processing (b) with this filter. (d) Result of processing (c) with the same filter.

From [Gonzalez & Woods]

Switching Median Filters

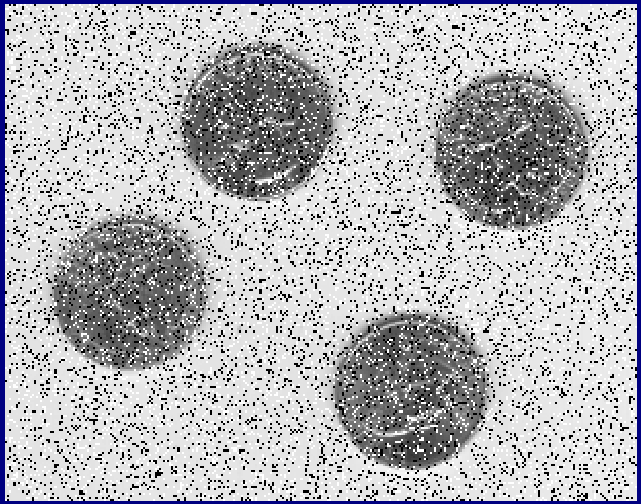
- **Motivation**
  - Regular median filters change both “bad” and “good” pixels
- **Idea**
  - Detect/classify “bad” and “good” pixels
  - Filter “bad” pixels only



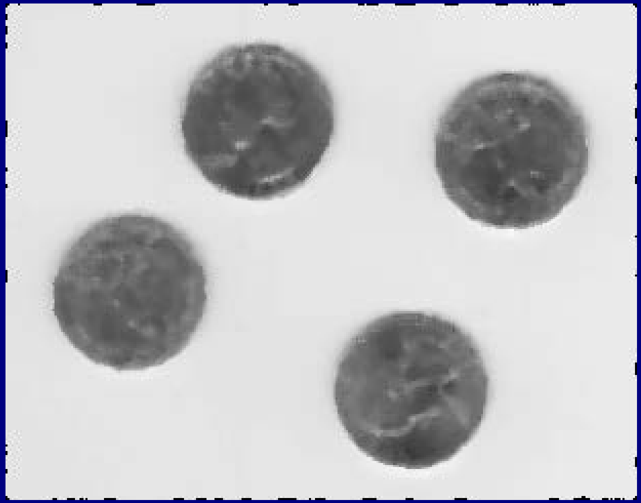
From [Wang & Zhang]

Switching Median Filters

original



noisy ($p_a = p_b = 0.1$)



regular 5x5 median filtered

switching 5x5 median filtered

From MATLAB sample images

Order Statistics (OS) Filters

- **Recall Order Statistics:**

For $\boldsymbol{x} = \{x_1, x_2, \cdots, x_{2M+1}\}$

OS $\boldsymbol{x}_{OS} = \{x_{(1)}, x_{(2)}, \cdots, x_{(2M+1)}\}$

such that $x_{(1)} \le x_{(2)} \le \cdots \le x_{(M+1)} \le \cdots \le x_{(2M+1)}$

- **OS filter: General Form**

$$OS\{x_1, x_2, \cdots, x_{2M+1}\} = \sum_{i=1}^{2M+1} w_i x_{(i)} \quad \text{where} \quad \sum_{i=1}^{2M+1} w_i = 1$$

- **Special Cases**

$Min\{x_1, x_2, \cdots, x_{2M+1}\} = x_{(1)}$ $\{w_i\} = \{1,0,\cdots 0,\cdots,0\}$

$Max\{x_1, x_2, \cdots, x_{2M+1}\} = x_{(2M+1)}$ $\{w_i\} = \{0,0,\cdots 0,\cdots,1\}$

(M+1)-th

$Median\{x_1, x_2, \cdots, x_{2M+1}\} = x_{(M+1)}$ $\{w_i\} = \{0,0,\cdots 1,\cdots,0\}$

Order Statistics (OS) Filters

- **Note: An OS Filter is Uniquely Defined by** $\{w_i\}$

- **Example 1:** $\{w_i\} = \{0, \cdots, 1/4, 1/2, 1/4, \cdots, 0\}$

M-th $\quad$ $(M+1)$-th $\quad$ $(M+2)$-th

then $\quad OS\{x_1, x_2, \cdots, x_{2M+1}\} = 0.25x_{(M)} + 0.5x_{(M+1)} + 0.25x_{(M+2)}$

- **Example 2:** $\{w_i\} = \{1, 1, \cdots, 1\}/(2M+1)$

then $\quad OS\{x_1, x_2, \cdots, x_{2M+1}\} = \sum_{i=1}^{2M+1} \frac{1}{2M+1} x_{(i)}$

$$= Mean\{x_{(i)}, i = 1, \cdots, 2M+1\} = Mean\{x_i,, i = 1, \cdots, 2M+1\}$$

Examples

- A 4x4 grayscale image is given by

5	6	7	8
0	6	7	8
5	6	15	8
5	6	7	8

impulse?

impulse?

1) Filter the image with a 3x3 median filter, after **zero-padding** at the image borders

5	6	7	8
0	6	7	8
5	6	15	8
5	6	7	8

zero-padding →

0	0	0	0	0	0
0	5	6	7	8	0
0	0	6	7	8	0
0	5	6	15	8	0
0	5	6	7	8	0
0	0	0	0	0	0

median filtering →

0	5	6	0
5	6	7	7
5	6	7	7
0	5	6	0

Examples

2) Filter the image with a 3x3 median filter, after **replicate-padding** at the image borders

5	6	7	8
0	6	7	8
5	6	15	8
5	6	7	8

replicate
-padding

5	5	6	7	8	8
5	5	6	7	8	8
0	0	6	7	8	8
5	5	6	15	8	8
5	5	6	7	8	8
5	5	6	7	8	8

median
filtering

5	6	7	8
5	6	7	8
5	6	7	8
5	6	7	8

impulse cleaned!

Examples

3) Filter the image with a 3x3 OS filter, after **replicate-padding** at the image borders. The weighting factors of the OS filter are given by

$$\{w_i \mid i = 1, \ldots, 9\} = \{0, 0, 0, ¼, ½, ¼, 0, 0, 0\}$$

5	6	7	8
0	6	7	8
5	6	15	8
5	6	7	8

replicate -padding →

5	5	6	7	8	8
5	5	6	7	8	8
0	0	6	7	8	8
5	5	6	15	8	8
5	5	6	7	8	8
5	5	6	7	8	8

OS filtering →

5	6	7	8
5	6	7.25	8
5	6	7.25	8
5	6	7.25	8