CSI 4133 Computer Methods in Picture Processing and Analysis

Fall 2024

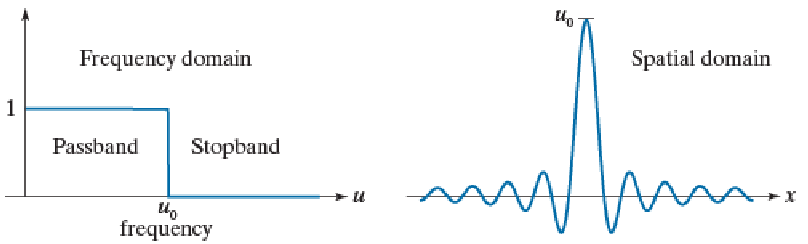
Pengcheng Xi, Ph.D.

Spatial and frequency domains

- Connection: use **Fourier** transform to go from the spatial to the frequency domain; use inverse Fourier transform to return to the spatial domain
- **Convolution**, which is the basis for filtering in the spatial domain, is equivalent to *multiplication* in the frequency domain, and vice versa
- An impulse of strength A in the spatial domain is a constant of value A in the frequency domain, and vice versa

a b

FIGURE 3.38
(a) Ideal 1-D low-pass filter transfer function in the frequency domain. (b) Corresponding filter kernel in the spatial domain.



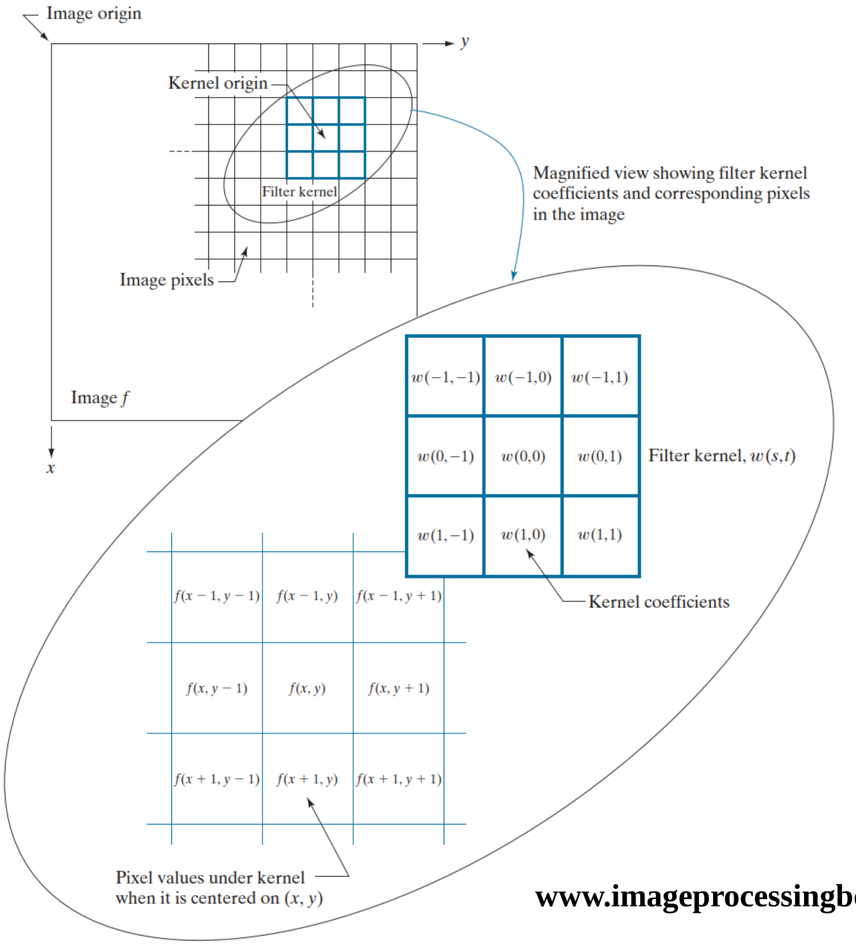
Spatial and frequency domains

- A function (e.g., an image) satisfying some mild conditions, can be expressed as the sum of sinusoids of different frequencies and amplitudes.
- The appearance of an image depends on the **frequencies** of its *sinusoidal* components
  - Change the frequencies of those components will change the *appearance* of the image
- Possible to associate certain frequency bands with image characteristics
  - Regions of an image with intensities that vary slowly are characterized by sinusoids of low frequencies
  - Edges and other sharp intensity transitions are characterized by high frequencies
- Reducing the high-frequency components of an image will tend to blur it

Basics of spatial filtering

- Neighborhood operations

FIGURE 3.28 The mechanics of linear spatial filtering using a 3×3 kernel. The pixels are shown as squares to simplify the graphics. Note that the origin of the image is at the top left, but the origin of the kernel is at its center. Placing the origin at the center of spatially symmetric kernels simplifies writing expressions for linear filtering.



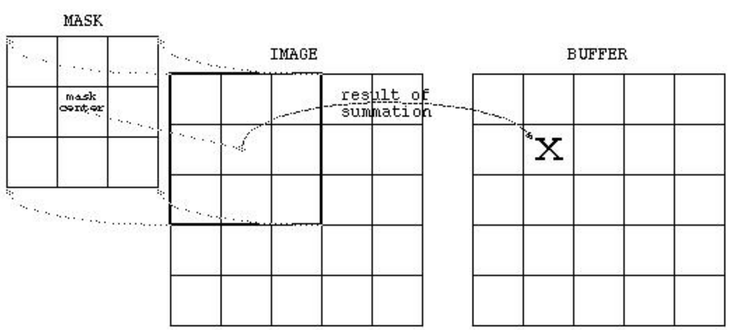
$$g(x,y) = w(-1,-1)f(x-1,y-1) + w(-1,0)f(x-1,y) + \ldots + w(0,0)f(x,y) + \ldots + w(1,1)f(x+1,y+1)$$

$$g(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t)$$

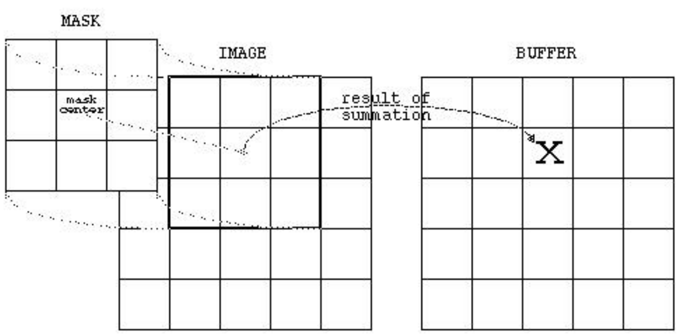
www.imageprocessingbook.com

Convolution process

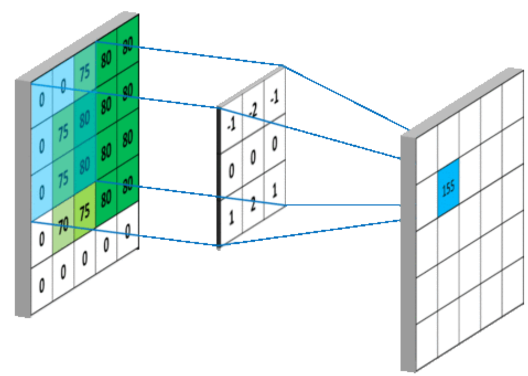
Figure 3.2-3: THE CONVOLUTION PROCESS



a) Overlay the convolution mask in upper left corner of the image. Multiply coincident terms, sum, put result into the image buffer at the location that corresponds to the mask's current center, which is (r,c) = (1,1).



b) Move the mask one pixel to the right, multiply coincident terms, sum, and place the new result into the buffer at the location that corresponds to the new center location of the convolution mask, now at (r,c) = (1,2). Continue to the end of the row.



Convolution

35	40	41	45	50
40	40	42	46	52
42	46	50	55	55
48	52	56	58	60
56	60	65	70	75

×

	0	1	0	
	0	0	0	
	0	0	0	

=

		42		

Computing an average is analogous to integration;
A filter that computes the local derivative of an image sharpens the image

- Sharpening

0	0	0	0	0
0	0	-1	0	0
0	-1	5	-1	0
0	0	-1	0	0
0	0	0	0	0

- Blur

0	0	0	0	0
0	1	1	1	0
0	1	1	1	0
0	1	1	1	0
0	0	0	0	0

- Edge enhance

	0	1	0	
	1	-4	1	
	0	1	0	

edge detect

	0	1	0	
	1	-4	1	
	0	1	0	

Emboss

	-2	-1	0	
	-1	1	1	
	0	1	2	

Smoothing spatial filters

- Linear filters
  - 3x3 smoothing (averaging) filter masks

?

1	1	1
1	1	1
1	1	1

?

1	2	1
2	4	2
1	2	1

Smoothing spatial filters

- Box kernels

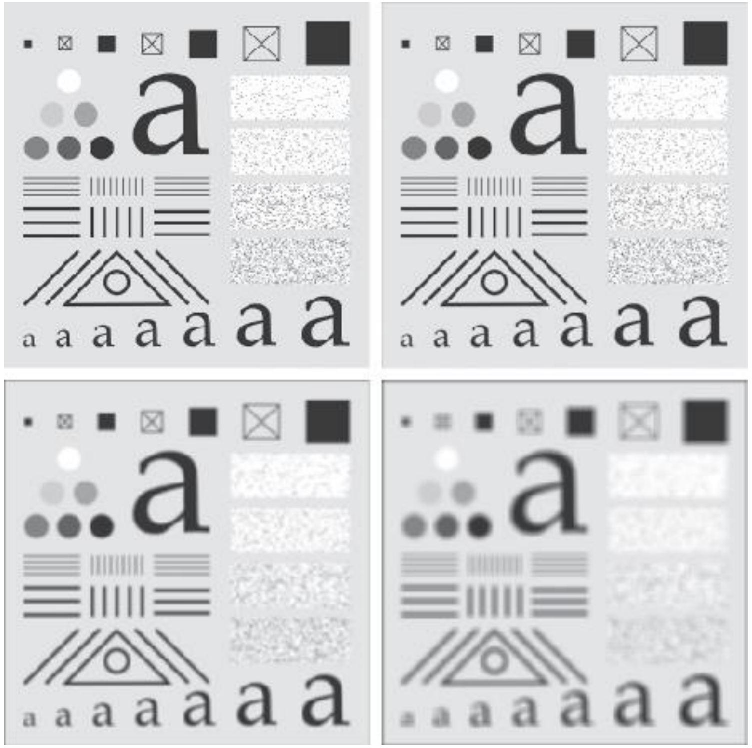
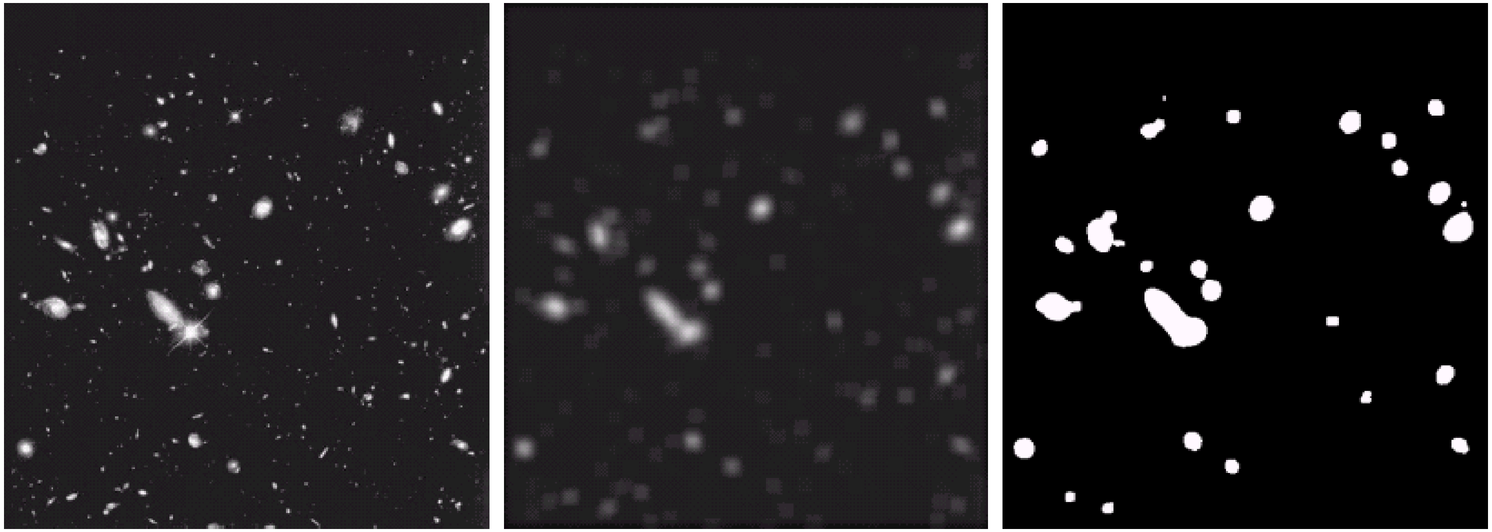


FIGURE 3.39
(a) Test pattern of size 1024×1024 pixels.
(b)-(d) Results of lowpass filtering with box kernels of sizes 3×3, 11×11, and 21×21, respectively.

www.imageprocessingbook.com

Smoothing spatial filters

- Application of an averaging mask



a b c

FIGURE 3.36 (a) Image from the Hubble Space Telescope. (b) Image processed by a 15 × 15 averaging mask. (c) Result of thresholding (b). (Original image courtesy of NASA.)

www.imageprocessingbook.com

Smoothing kernels

a b

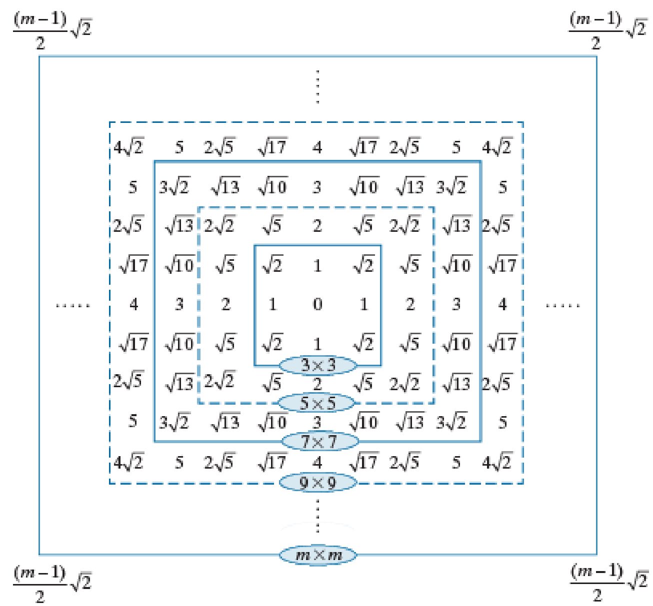
FIGURE 3.37 Examples of smoothing kernels: (a) is a *box* kernel; (b) is a *Gaussian* kernel.

$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad \frac{1}{4.8976} \times \begin{bmatrix} 0.3679 & 0.6065 & 0.3679 \\ 0.6065 & 1.0000 & 0.6065 \\ 0.3679 & 0.6065 & 0.3679 \end{bmatrix}$$

- **Box filters** are suitable for quick experimentation
  - Yield smoothing results that are visually acceptable
  - Limitations:
    - Poor approximations to the blurring characteristics of lenses
    - Favors blurring along perpendicular directions
- **Gaussian kernels**
  - $w(s,t) = G(s,t) = Ke^{-\frac{s^2+t^2}{2\sigma^2}}$, or
  - $G(r) = K\,e^{-\frac{r^2}{2\sigma^2}}$, where $r = \sqrt{s^2+t^2}$

Gaussian kernels - distances

FIGURE 3.40 Distances from the center for various sizes of square kernels.



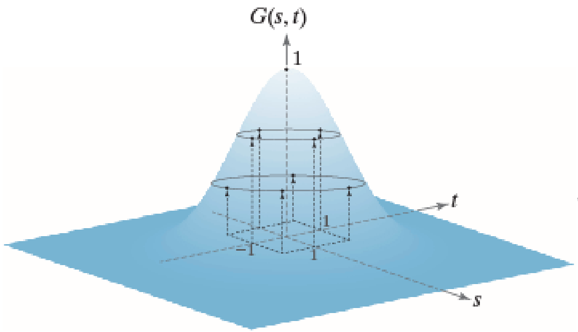
- Samples from a Gaussian function can be used to construct a weighted-average (lowpass) filter
- These 2D spatial functions sometimes are generated as the inverse Fourier transform of 2D filters specified in the frequency domain

www.imageprocessingbook.com

Gaussian kernels

a b

FIGURE 3.41
(a) Sampling a Gaussian function to obtain a discrete Gaussian kernel. The values shown are for $K = 1$ and $\sigma = 1$. (b) Resulting 3×3 kernel [this is the same as Fig. 3.37(b)].



$$\frac{1}{4.8976} \times \begin{array}{|c|c|c|} \hline 0.3679 & 0.6065 & 0.3679 \\ \hline 0.6065 & 1.0000 & 0.6065 \\ \hline 0.3679 & 0.6065 & 0.3679 \\ \hline \end{array}$$

www.imageprocessingbook.com

Smoothing spatial filters

a b c

FIGURE 3.42 (a) A test pattern of size 1024×1024. (b) Result of lowpass filtering the pattern with a Gaussian kernel of size 21×21, with standard deviations $\sigma = 3.5$. (c) Result of using a kernel of size 43×43, with $\sigma = 7$. This result is comparable to Fig. 3.39(d). We used $K = 1$ in all cases.

- Gaussian kernels need to be larger than box filters to achieve the same degree of blurring
- The values of Gaussian kernel coefficients decreases as a function of distance from the kernel centre
- We use a filter size $\lceil 6\sigma \rceil \times \lceil 6\sigma \rceil$
  - The percentage of area under ± 3σ in Gaussian frequency distribution is 99.7%

www.imageprocessingbook.com

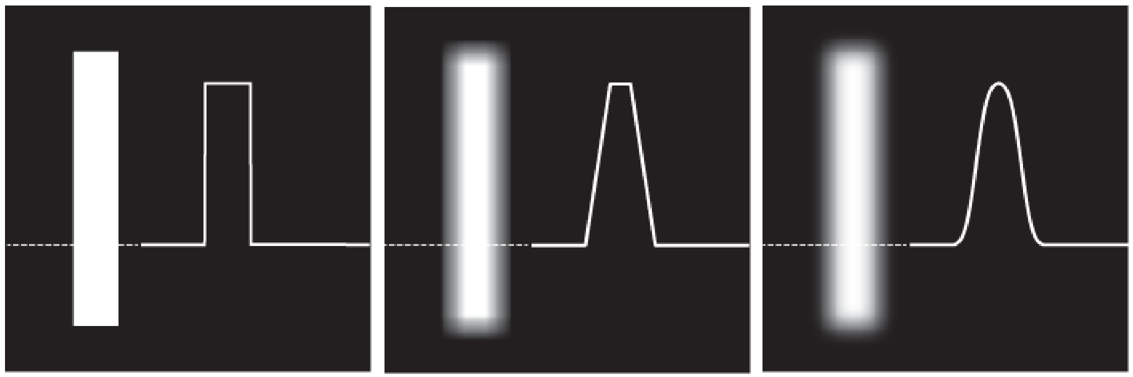
Effect of increasing kernel size

a b c

FIGURE 3.43 (a) Result of filtering Fig. 3.42(a) using a Gaussian kernels of size 43×43, with $\sigma = 7$. (b) Result of using a kernel of 85×85, with the same value of σ. (c) Difference image.

www.imageprocessingbook.com

Comparison of Gaussian and box filter smoothing characteristics



a b c

FIGURE 3.44 (a) Image of a white rectangle on a black background, and a horizontal intensity profile along the scan line shown dotted. (b) Result of smoothing this image with a box kernel of size 71×71, and corresponding intensity profile. (c) Result of smoothing the image using a Gaussian kernel of size 151×151, with $K = 1$ and $\sigma = 25$. Note the smoothness of the profile in (c) compared to (b). The image and rectangle are of sizes 1024×1024 and 768×128 pixels, respectively.

www.imageprocessingbook.com

The padding effect

a b c

FIGURE 3.45 Result of filtering the test pattern in Fig. 3.42(a) using (a) zero padding, (b) mirror padding, and (c) replicate padding. A Gaussian kernel of size 187×187, with $K = 1$ and $\sigma = 31$ was used in all three cases.

www.imageprocessingbook.com

Smoothing performance as a function of kernel and image size

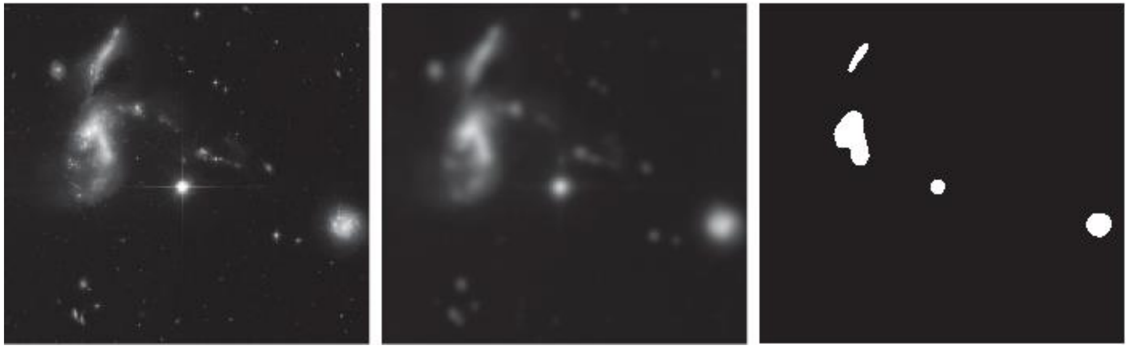
a b c

FIGURE 3.46 (a) Test pattern of size 4096 × 4096 pixels. (b) Result of filtering the test pattern with the same Gaussian kernel used in Fig. 3.45. (c) Result of filtering the pattern using a Gaussian kernel of size 745 × 745 elements, with $K = 1$ and $\sigma = 124$. Mirror padding was used throughout.

Image size is increased to four times
Need to understand the relationship between kernel size and object size

www.imageprocessingbook.com

Using low pass filtering and thresholding for region extractions



a b c

FIGURE 3.47 (a) A 2566 × 2758 Hubble Telescope image of the *Hickson Compact Group*. (b) Result of lowpass filtering with a Gaussian kernel. (c) Result of thresholding the filtered image (intensities were scaled to the range [0, 1]). The Hickson Compact Group contains dwarf galaxies that have come together, setting off thousands of new star clusters. (Original image courtesy of NASA.)

www.imageprocessingbook.com

Shading correction using lowpass filtering

a b c

FIGURE 3.48 (a) Image shaded by a shading pattern oriented in the −45° direction. (b) Estimate of the shading patterns obtained using lowpass filtering. (c) Result of dividing (a) by (b). (See Section 9.8 for a morphological approach to shading correction).

www.imageprocessingbook.com

Spatial filters

- Non-linear filters
  - Order-statistics filters
    - The response is based on ordering (ranking) the pixels contained in the image area encompassed by the filter, and then replacing the value of the center pixel with the value determined by the ranking result
    - E.g., median filter

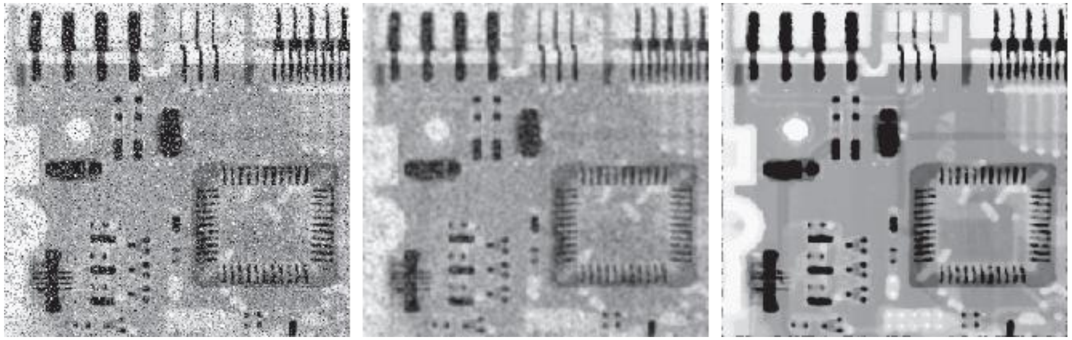
Spatial filter

- Median filter
  - The median, ξ – half the values in the set are less than or equal to ξ and half are greater than or equal to ξ.
  - Popular because, for certain types of random noise, they provide excellent noise-reduction capabilities with considerably less blurring than linear smoothing filers of similar size.

Mean vs. median

5x5 mean filter

5x5 median filter



a b c

FIGURE 3.49 (a) X-ray image of a circuit board, corrupted by salt-and-pepper noise. (b) Noise reduction using a 19×19 Gaussian lowpass filter kernel with $\sigma = 3$. (c) Noise reduction using a 7×7 median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

www.imageprocessingbook.com

Salt-and-pepper noise

- Some pixel values are lost (replaced by black or white dots)
  - Median filter: effective in the presence of impulse noise