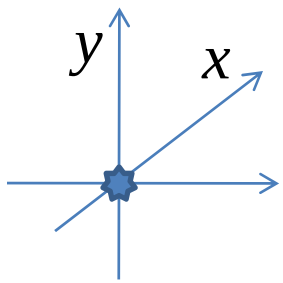
Lecture 5 Mask/Filter Transformation

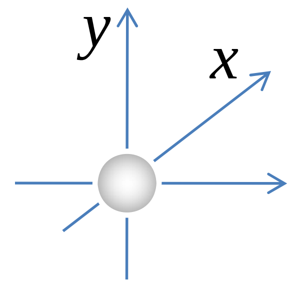
1. Math foundation of filtering
2. Smoother filters
3. Filter transformation with MatLab

Foundation of image acquisition and filtering

- Point light source => image acquisition (lens) => Point spread (sensor)



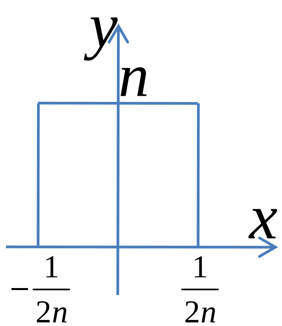
Lens (passive)



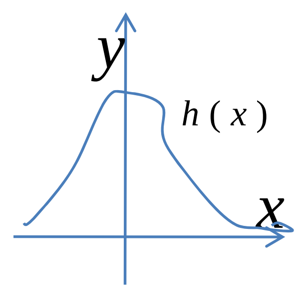
Point light source unit energy

Point spread

- Intensity distribution of the unit light energy



L



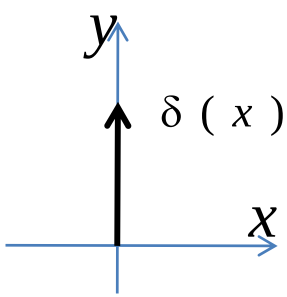
$$P_n(x) = \begin{cases} n, & |x| \le \frac{1}{2n} \\ 0, & |x| > \frac{1}{2n} \end{cases}, E = \int_{-\infty}^{\infty} P_n(x)dx = 1$$

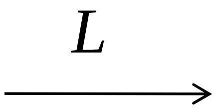
$$h(x) = L[P_n(x)]$$

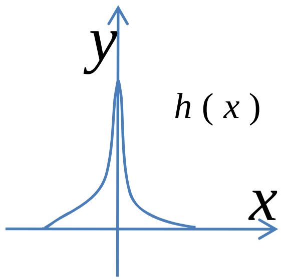
$$\int_{-\infty}^{\infty} h(x)dx = 1$$

Foundation of Image Acquisition

- Ideal unit energy point intensity function







$$\delta(x)=\begin{cases}\infty, & x=0\\ 0, & x\neq 0\end{cases},\int_{-\infty}^{\infty}\delta(x)dx=1$$

$$h(x)=L[\delta(x)]$$

$$\int_{-\infty}^{\infty}h(x)dx=1$$

Properties of Lens Transformation

- Lens transformation system

$$input \xrightarrow{L} output$$

$$\delta(x) \xrightarrow{L} h(x)$$

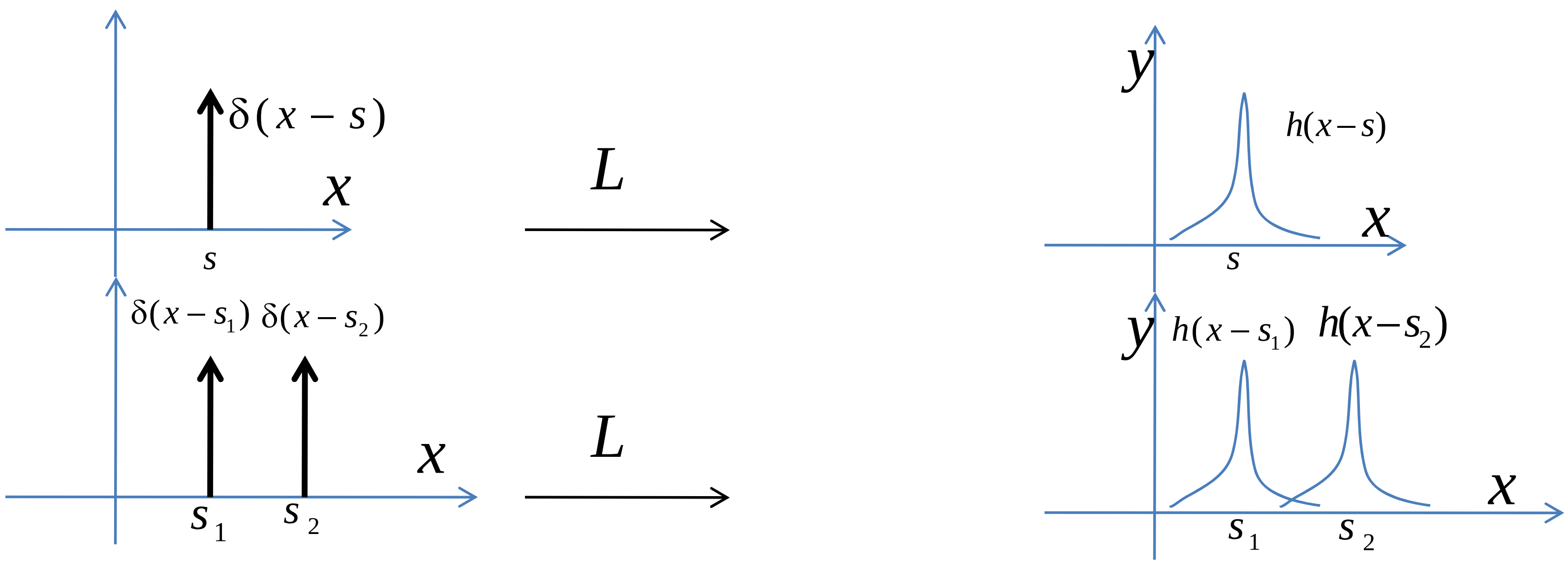
$$f(x) \xrightarrow{L} g(x)$$

- Linear Shift-Invariant System (LSI): If the system L has the properties

  (a) Homogeneity: $Af(x) \xrightarrow{L} Ag(x)$

  (b) Shift-invariant: $\delta(x-s) \xrightarrow{L} h(x-s)$

  (c) Superposition: $f_1(x) + f_2(x) \xrightarrow{L} g_1(x) + g_2(x)$



Signals in LSI

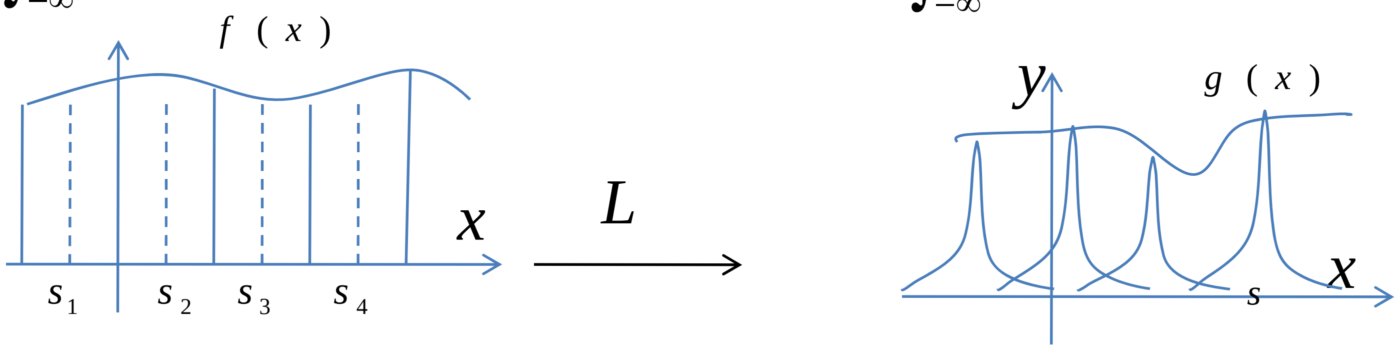
- By homogeneity and shift-invariant:

$$f(s)\delta(x-s) \xrightarrow{L} f(s)h(x-s)$$

- Then by superposition

$$f(x) = \lim_{n\to\infty} \frac{1}{n}\sum_{i=0}^{K} f(s_i)P_n(x-s_i)$$

$$= \int_{-\infty}^{\infty} f(s)\delta(x-s)ds \qquad \xrightarrow{L} g(x) = \int_{-\infty}^{\infty} f(s)h(x-s)ds$$



$\delta(x - s)$

Convolution

- Definition:

$$f(x) * h(x) = \int_{-\infty}^{\infty} f(s)h(x-s)ds$$

f(x) convoluted with h(x), or convolution of f(x) with h(x)

- In a LSI system, signal received g(x) the convolution of input f(x) with impulse response h(x). As long as we know f(x) and h(x), we can compute the output signal by their convolution

Example of Convolution

Given

$$f(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}, h(x) = \begin{cases} 1, & |x| \leq \frac{1}{2} \\ 0, & else \end{cases}$$

Compute f(x)*h(x)

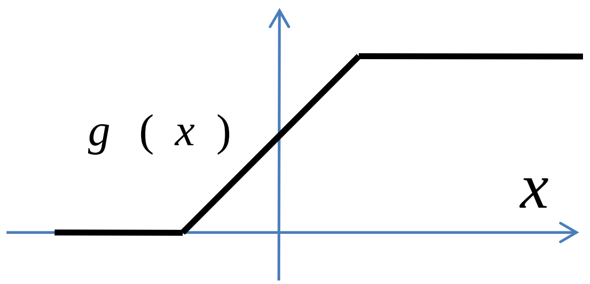
Sol: See the next slide for the case analysis

Case 1: x < -1/2, $g(x) = f(x) * h(x) = \int_{-\infty}^{\infty} f(s)h(x-s)ds = \int_{-\infty}^{\infty} 0ds = 0$

Case 2: |x|<1/2, $g(x) = f(x) * h(x) = \int_{-\infty}^{\infty} f(s)h(x-s)ds = \int_{0}^{x+\frac{1}{2}} 1ds = x + \frac{1}{2}$

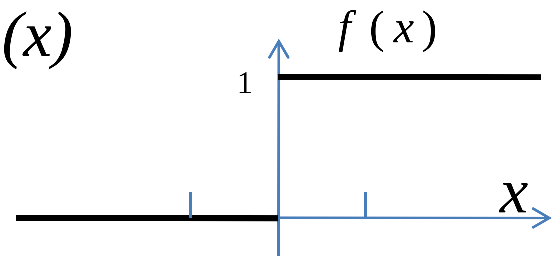
Case 3. x> ½, $g(x) = f(x) * h(x) = \int_{-\infty}^{\infty} f(s)h(x-s)ds = \int_{x-\frac{1}{2}}^{x+\frac{1}{2}} 1ds = 1$

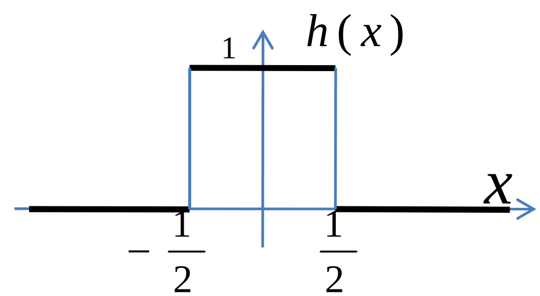
$$g(x) = \begin{cases} 0, & x <\leq -1/2 \\ x+1/2, & |x| < 1/2 \\ 1, & x \geq 1/2 \end{cases}$$



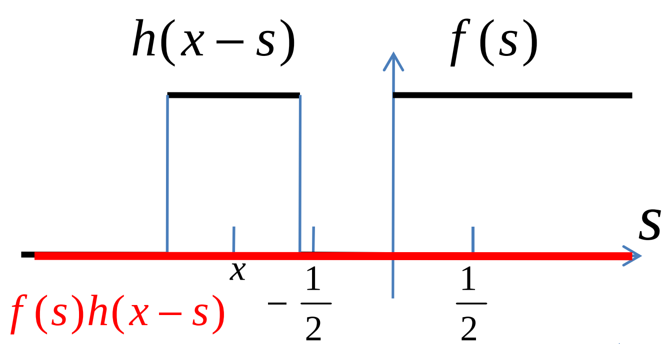
Case analysis for convolution computing

- *f(x)* and *h(x)*

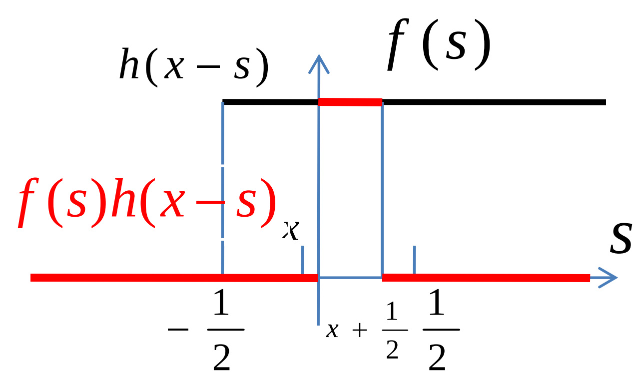




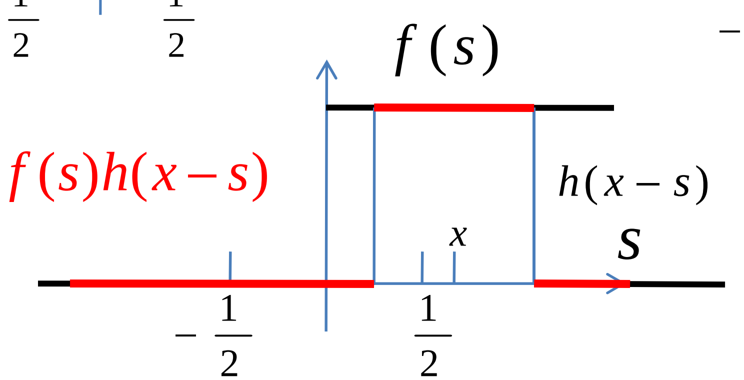
- *Case 1: x<-1/2*



Case 2: |x|<1/2



Case 3: x>1/2



Categories of Spatial filters

- Linear filters
  - Correlation
  - Convolution
- Nonlinear filters

- Smooth filters

- Sharpening filters

Mechanics of linear spatial filter

- Move the mask over the image, calculate the pixel value using correlation or convolution

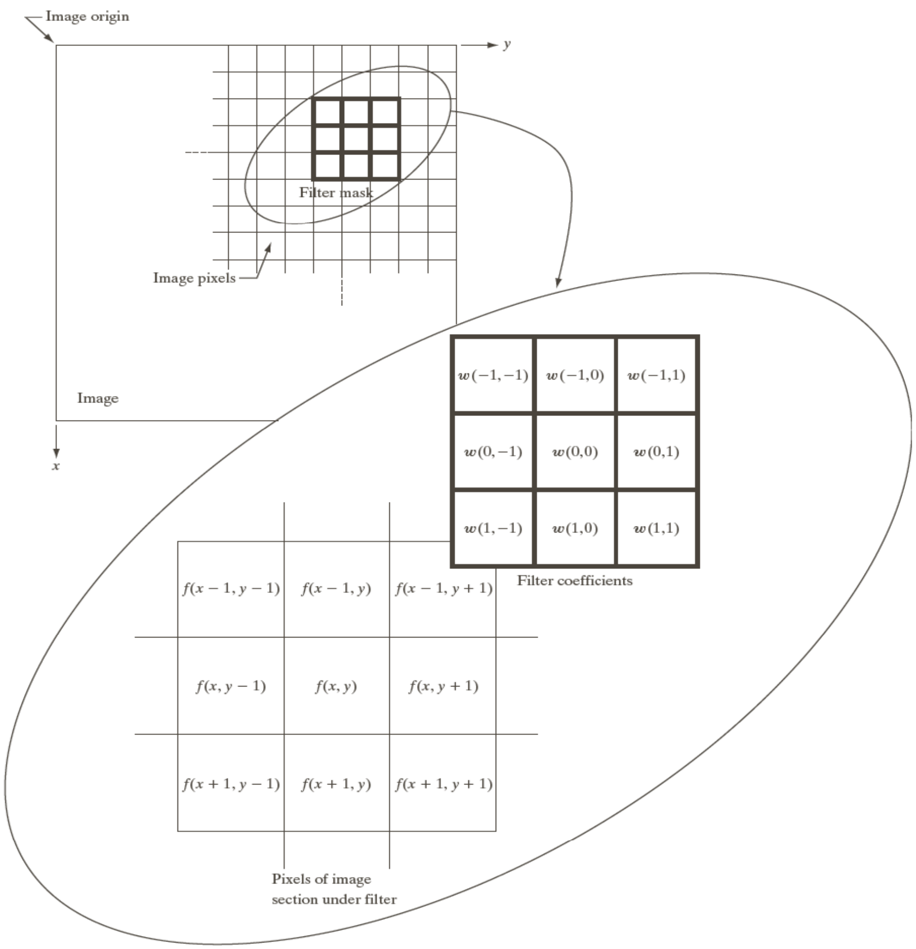


FIGURE 3.28 The mechanics of linear spatial filtering using a 3 × 3 filter mask. The form chosen to denote the coordinates of the filter mask coefficients simplifies writing expressions for linear filtering.

1-D Correlation and Convolution

- Correlation of two functions

$$g(x) = w(x) \circ f(x) = \sum_{s=-a}^{a} w(s) f(x+s)$$

- Convolution of two functions (like rotate w for 180 degree)

$$g(x) = w(x) * f(x) = \sum_{s=-a}^{a} w(s) f(x-s)$$

with image of size M and mask size m, a = (m-1)/2, x = 0, …, M-1

1-D Correlation and Convolution

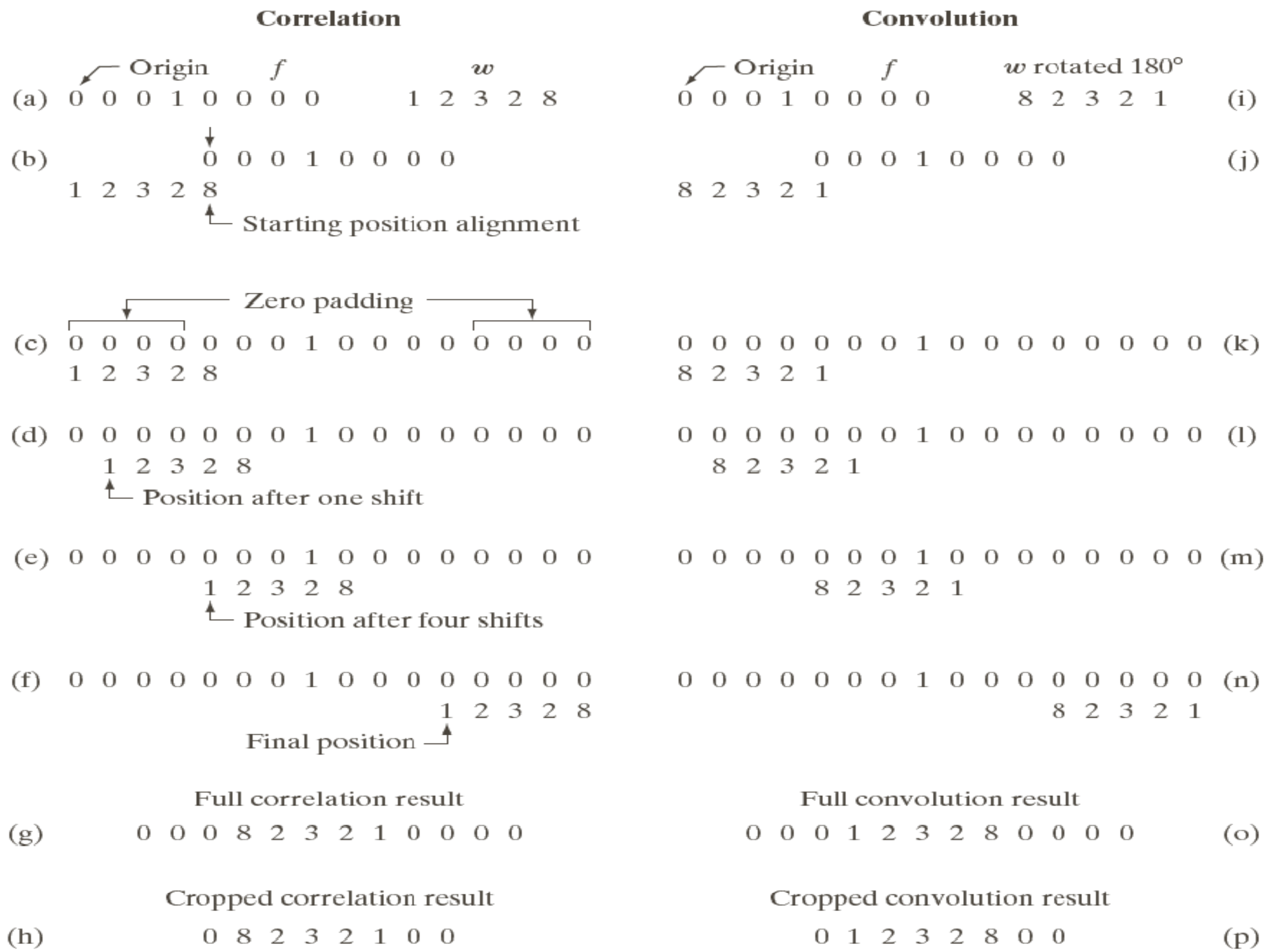


FIGURE 3.29 Illustration of 1-D correlation and convolution of a filter with a discrete unit impulse. Note that correlation and convolution are functions of *displacement*.

Spatial Filtering

- Spatial filter: also called mask/kernel/template or window)
  - Consist of a neighbourhood with coefficients on pixels
  - Example: masks of odd sizes, e.g. 3x3, 5x5,…
- Apply a filter to an image: simply move the filter mask from point to point in an image. At each point (x, y), the response of the filter at that point is calculated using a predefined relationship.
- Example

Linear filter

$$R = w_1 z_1 + w_2 z_2 + \ldots + w_{mn} z_{mn}$$
$$= \sum_{i=i}^{mn} w_i z_i$$

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

2-D Correlation and Convolution

- Correlation of two functions

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

- Convolution of two functions (like rotate w for 180 degree)

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

- Linear Filtering of an image f of size MxN filter mask of size mxn is given by the expression, a = (m-1)/2, b = (n-1)/2,
  x = 0, 1, 2, … , M-1 and y = 0, 1, 2, … , N-1

Example of 2-D correlation and convolution

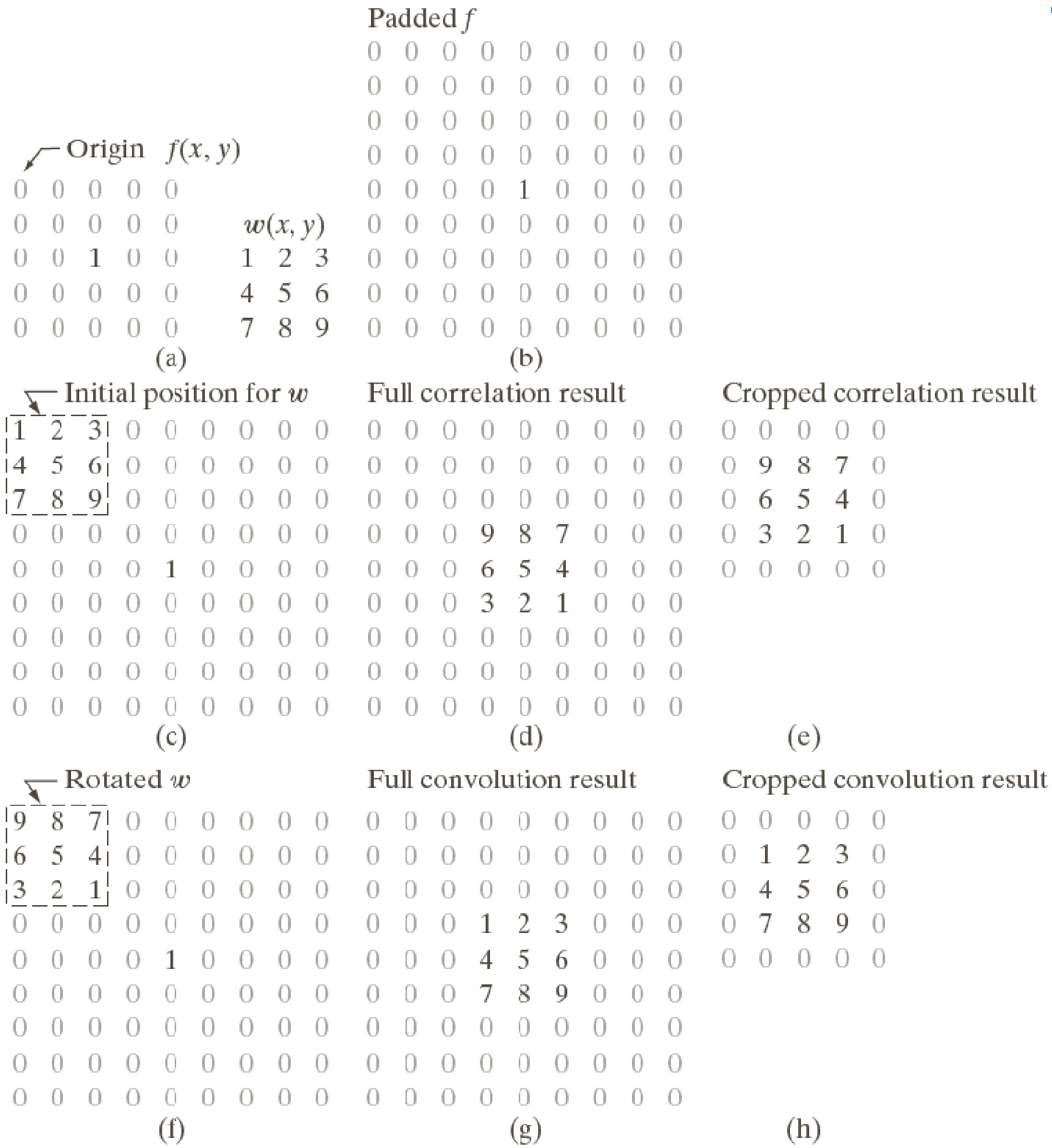


FIGURE 3.30 Correlation (middle row) and convolution (last row) of a 2-D filter with a 2-D discrete, unit impulse. The 0s are shown in gray to simplify visual analysis.

Smoothing Spatial Filters

- Used for blurring and for noise reduction
- Blurring is used in preprocessing steps, such as
  - removal of small details from an image prior to object extraction
  - bridging of small gaps in lines or curves

- Noise reduction can be accomplished by blurring with a linear filter and also by a nonlinear filter
- General form of smooth filter (size mxn, m and n are odd)

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

Smoothing Linear Filters

- Output is simply the average of the pixels contained in the neighborhood of the filter mask. Also called averaging filters or low pass filters

- Replacing the value of every pixel in an image by the average of the gray levels in the neighborhood will reduce the “sharp” transitions in gray levels.

- Sharp transitions
– random noise in the image
– edges of objects in the image
– Smoothing can reduce noises (desirable) and blur edges (undesirable)

Example: 3x3 Smoothing Linear Filters

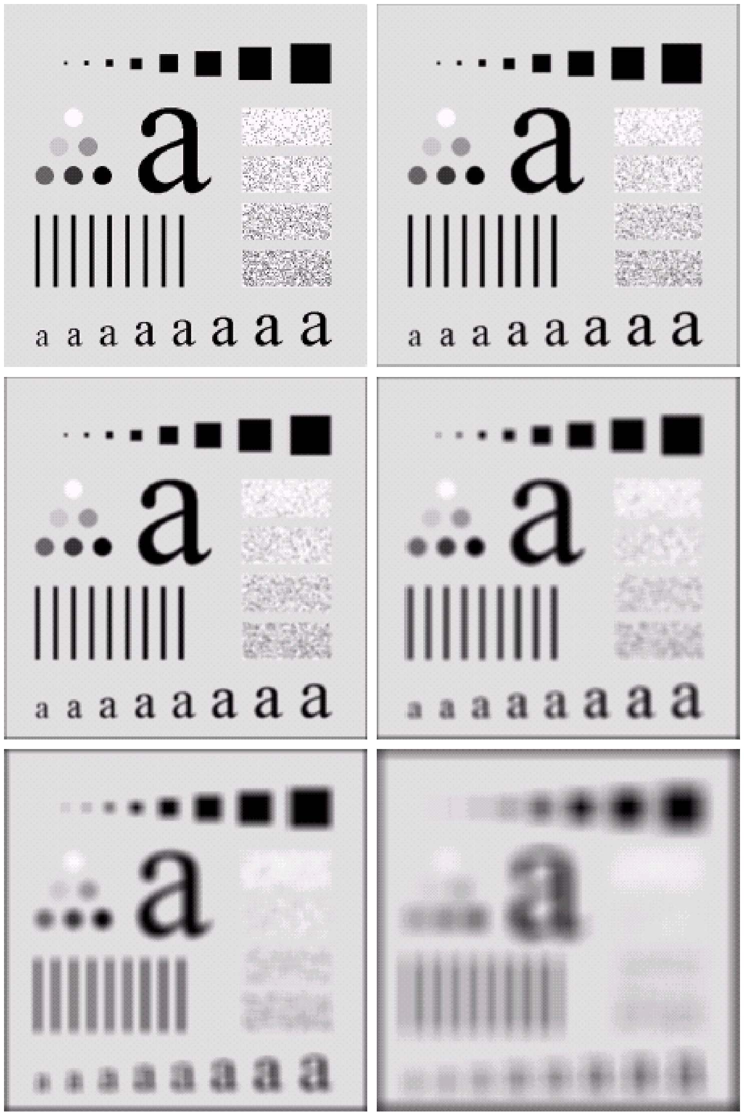
$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

box filter

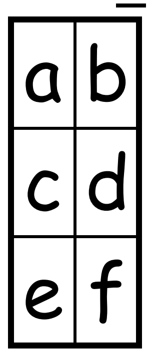
$$\frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

weighted average

Example



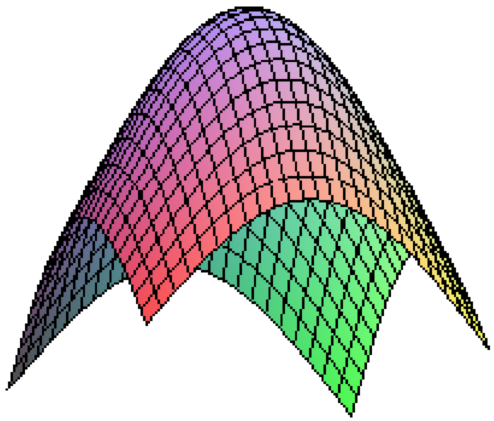
- a). original image 500x500 pixel
- b). - f). results of smoothing with square averaging filter masks of size n = 3, 5, 9, 15 and 35, respectively.
- Note:
  - big mask is used to eliminate small objects from an image.
  - the size of the mask establishes the relative size of the objects that will be blended with the background.



Gauss function filter

- Use Gauss function to generate a mask of m x m matrix w

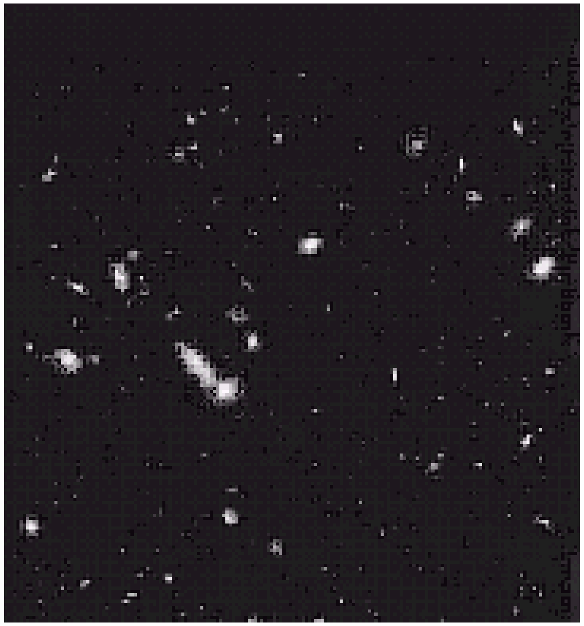
$$h(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}}$$



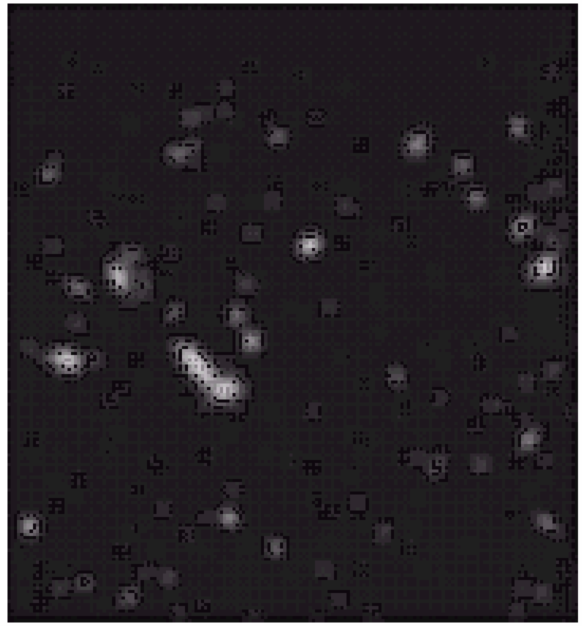
E.g. m = 3

$$w = \begin{bmatrix} h(-1,-1) & h(-1,0) & h(-1,1) \\ h(0,-1) & h(0,0) & h(0,1) \\ h(1,-1) & h(1,0) & h(1,1) \end{bmatrix}$$

Example



original image



result after smoothing
with 15x15 averaging mask

result of thresholding

we can see that the result after smoothing and thresholding, the remains are the largest and brightest objects in the image.

Nonlinear Filters

- Order-Statistics Filters
  - The response is based on ordering (ranking) the pixels contained in the image area encompassed by the filter

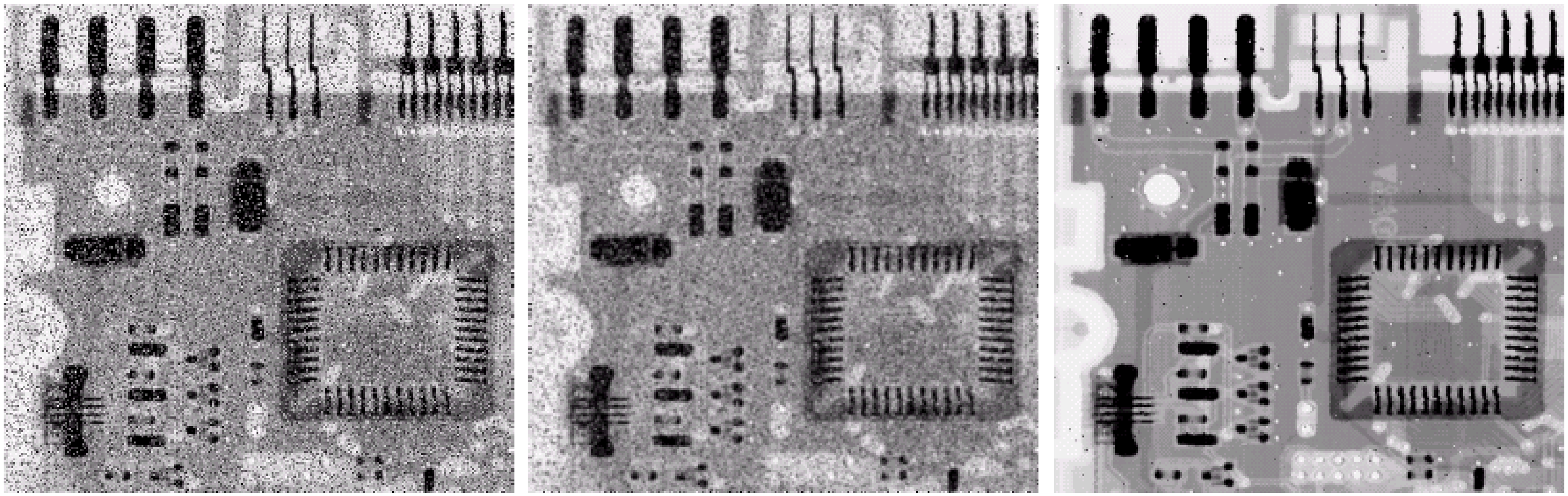
- Examples
  - median filter : $R = \text{median}\{z_k \mid k = 1,2,\ldots,n \times n\}$
  - max filter : $R = \max\{z_k \mid k = 1,2,\ldots,n \times n\}$
  - min filter : $R = \min\{z_k \mid k = 1,2,\ldots,n \times n\}$

note: n x n is the size of the mask

Median Filters

- Median filters
  - replaces the value of a pixel by the median of the gray levels in the neighborhood of that pixel (the original value of the pixel is included in the computation of the median)
  - quite popular because for certain types of random noise (impulse noise ⇨ salt and pepper noise) , they provide excellent noise-reduction capabilities, with considering less blurring than linear smoothing filters of similar size.
  - forces the points with distinct gray levels to be more like their neighbors.
  - isolated clusters of pixels that are light or dark with respect to their neighbors, and whose area is less than $n^2/2$ (one-half the filter area), are eliminated by an n x n median filter.
  - eliminated = forced to have the value equal the median intensity of the neighbors. larger clusters are affected considerably less

Example : Median Filters



a b c

FIGURE 3.37 (a) X-ray image of circuit board corrupted by salt-and-pepper noise. (b) Noise reduction with a 3 × 3 averaging mask. (c) Noise reduction with a 3 × 3 median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)