3.3.2 Histogram Matching (specification)

We may want to transform a image's gray scale histogram to a particular shape, i.e., specify it.

This is a generalization of histogram equalization

	output image	input image
these are the probability distribution functions, i.e., histograms.	$p_z(z)$	$p_r(r)$
	↑ This is the desired distribution	↑ This is the original distribution

Consider a new random variable s given by

$$s = T(r) = \int_0^r p_r(w)\,dw \qquad (1)$$

just like we did for histogram equalization.

Now define another random variable z

$$G(z) = \int_0^z p_z(t)\,dt = s \qquad (2)$$

$\therefore \quad G(z) = T(r)$

or $\quad z = G^{-1}(s) = G^{-1}[T(r)]$

This can be computed using (1) from the input image

G can be similarily computed from (2) for the desired image distribution.

The real issue is how to invert G to get G^{-1}

This can be done readily for digital (discrete functions) k-th value of transformation.

k-th value of r_k (gray level)

$$s_k = T(r_k) = \sum_{j=0}^{k} P_r(r_j) = \sum_{j=0}^{k} \frac{n_j}{n}$$

This is the cumulative distributive function for each gray level k

probability of each gray level k

$$v_k = G(z_k) = \sum_{i=0}^{k} P_z(z_i) = s_k$$

This is the cumulative distribution function for the desired distribution $P_z(z_i)$

Combining these

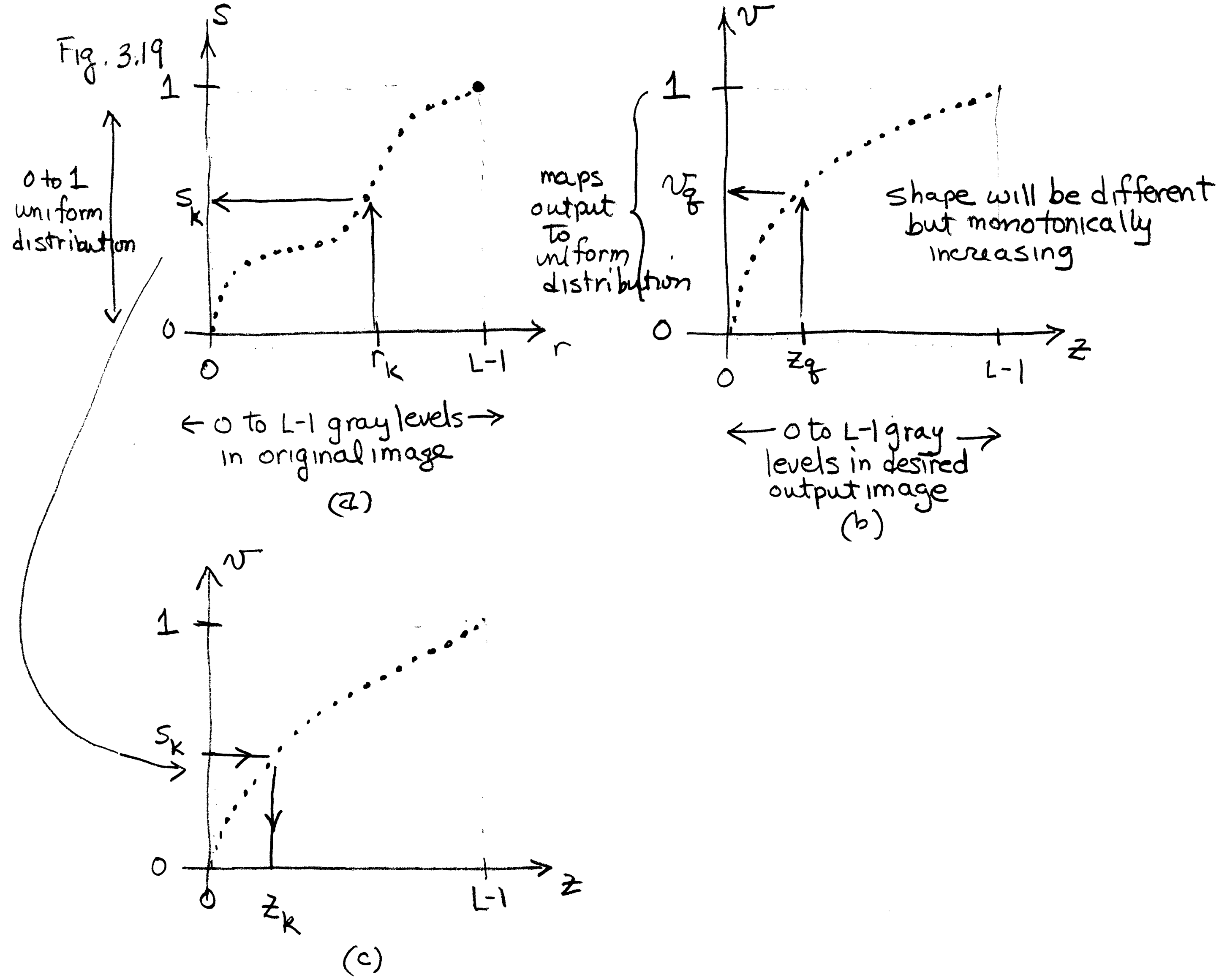
output (desired) gray scales

$$G(z_k) = s_k = T(r_k)$$

Invert to get the transform

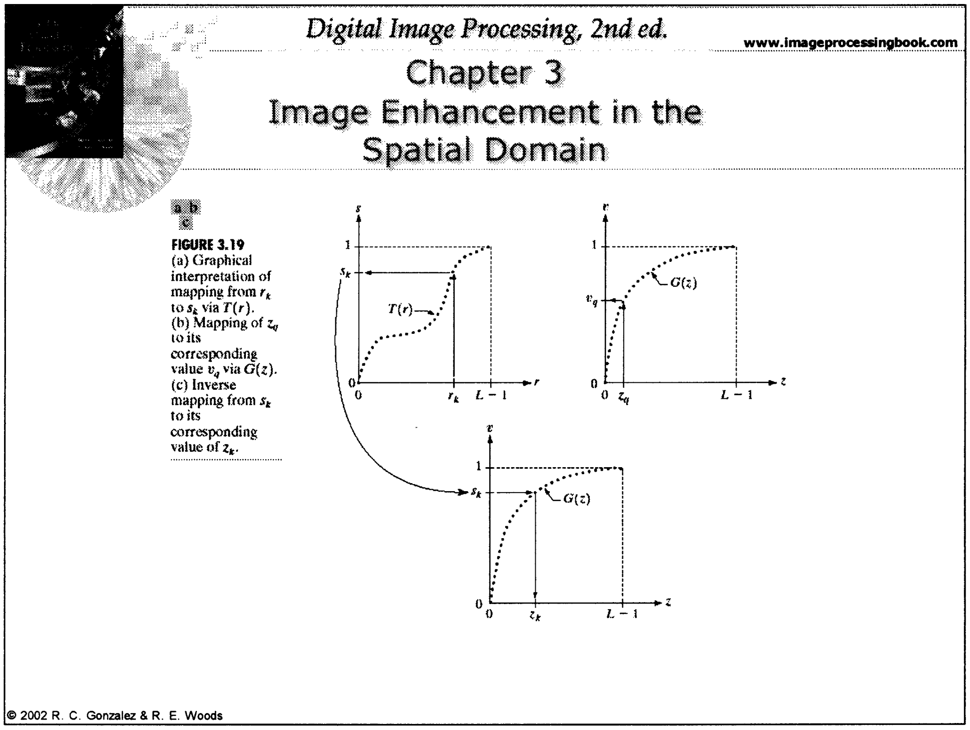
$$z_k = G^{-1}\left[T(r_k)\right]$$

Fig. 3.19



Algorithm

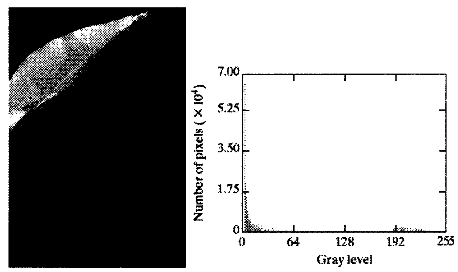
1. compute histogram of original image
2. compute the uniform mapping s_k by computing C.D.F.
3. compute the uniform transformation for the desired output transformation
4. starting at $s_k = 0$ iterate (increase) to find smallest value $\hat{z}_k$ which approximately satisifies (c) ; repeat for all s_k up to 1
5. combine $r_k \rightarrow s_k$ with results of ④ to get $r_k \rightarrow \hat{z}_k$ transformation and transform table.



Algorithm

1. compute histogram of original image
2. compute uniform mapping S_K onto $[0,1]$ by computing cumulative distribution function (CDF)
3. similarily compute the CDF for the desired output transformation
4. starting at $S_K = 0$ iterate (increase) to find value $\hat{Z}_K$ which approximately satisfies the function in (c). Repeat for all S_K up to 1.
5. combine the $r_K \rightarrow S_K$ mapping with the results of ④ to get the functional mapping $r_K \rightarrow \hat{Z}_K$ and construct a transformation table

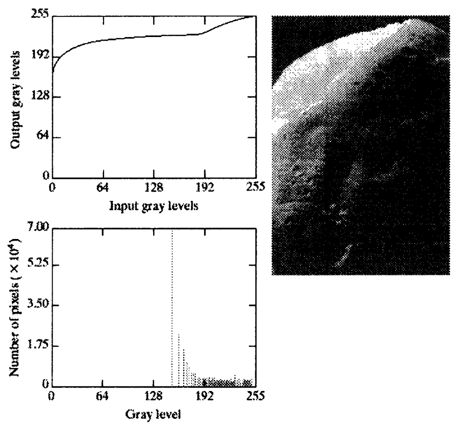
Chapter 3
Image Enhancement in the Spatial Domain



a b

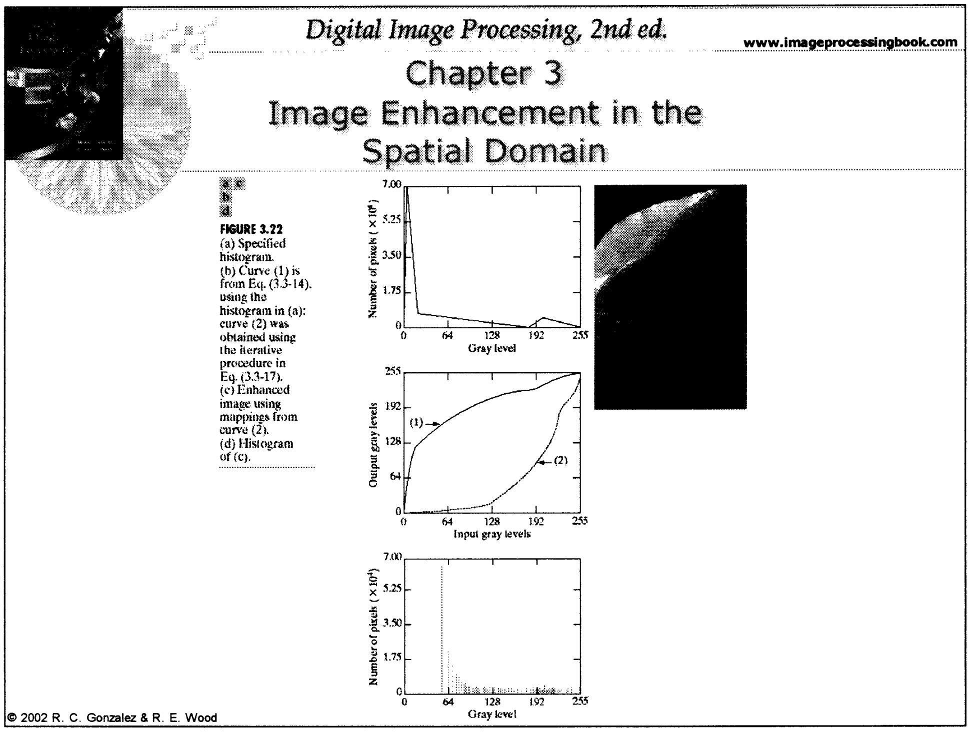
FIGURE 3.20 (a) Image of the Mars moon Photos taken by NASA's ***Mars Global Surveyor.*** (b) Histogram. (Original image courtesy of NASA.)

Chapter 3
Image Enhancement in the Spatial Domain



a b
c

FIGURE 3.21 (a) Transformation function for histogram equalization. (b) Histogram-equalized image (note the washed-out appearance). (c) Histogram of (b).



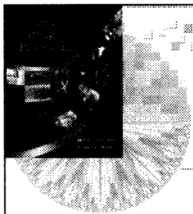
(a) This is the desired manually generated histogram. Note that this is a continuous function.

(b) (1) is the computed C.D.F. of the desired histogram given in (a).
(2) is the transformation resulting from applying the algorithm of Figure 3.19 to this image.

(c) is the result of using (b)(2) to transform the original image.

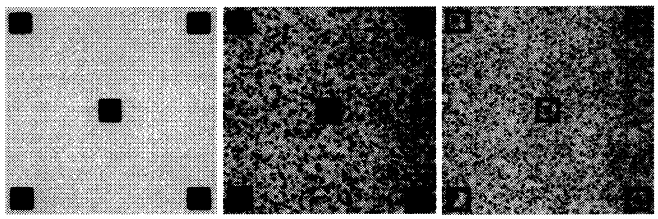
(d) is the histogram of the image produced in (c).

Note that the histogram does not appear uniform because our derivation was for continuous variables whereas our images and solution are discrete.



Digital Image Processing, 2nd ed. www.imageprocessingbook.com

Chapter 3
Image Enhancement in the Spatial Domain

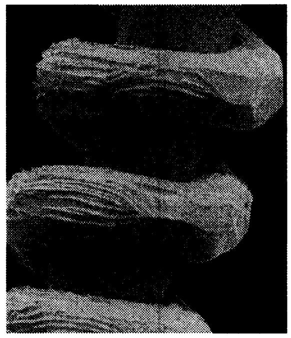


a b c

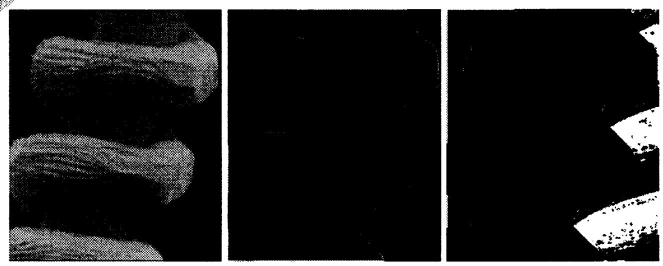
FIGURE 3.23 (a) Original image. (b) Result of global histogram equalization. (c) Result of local histogram equalization using a 7×7 neighborhood about each pixel.

Chapter 3
Image Enhancement in the Spatial Domain

FIGURE 3.24 SEM image of a tungsten filament and support, magnified approximately 130×. (Original image courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene).



Chapter 3
Image Enhancement in the Spatial Domain



a b c

FIGURE 3.25 (a) Image formed from all local means obtained from Fig. 3.24 using Eq. (3.3-21). (b) Image formed from all local standard deviations obtained from Fig. 3.24 using Eq. (3.3-22). (c) Image formed from all multiplication constants used to produce the enhanced image shown in Fig. 3.26.

Chapter 3
Image Enhancement in the Spatial Domain

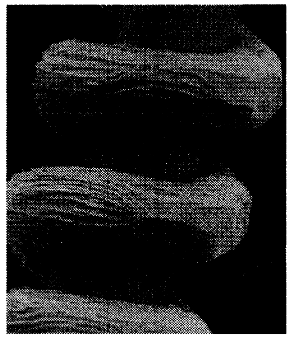
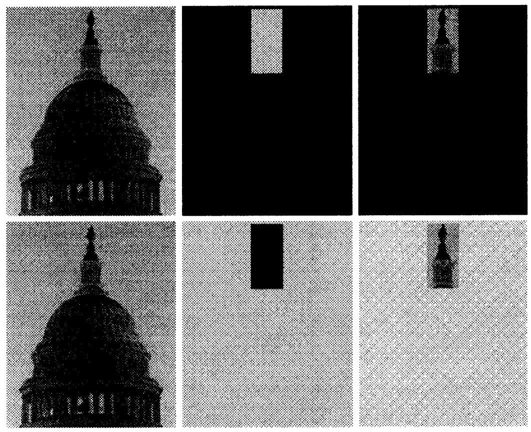


FIGURE 3.26 Enhanced SEM image. Compare with Fig. 3.24. Note in particular the enhanced area on the right side of the image.

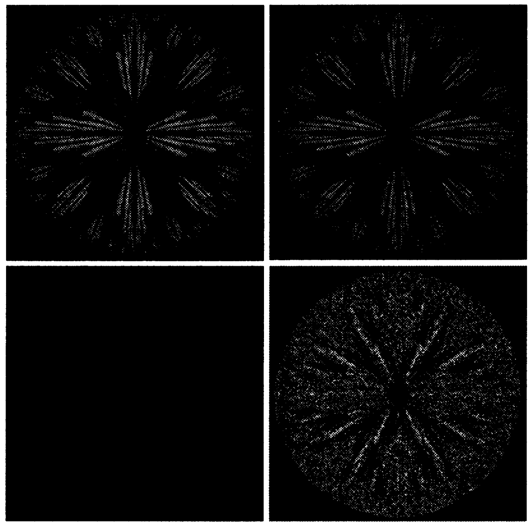
Chapter 3
Image Enhancement in the Spatial Domain



a b c
d e f

FIGURE 3.27
(a) Original image. (b) AND image mask. (c) Result of the AND operation on images (a) and (b). (d) Original image. (e) OR image mask. (f) Result of operation OR on images (d) and (e).

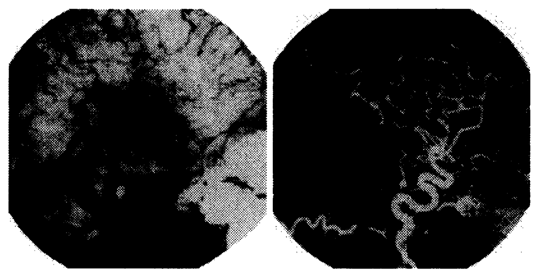
Chapter 3
Image Enhancement in the Spatial Domain



a b
c d

FIGURE 3.28 (a) Original fractal image. (b) Result of setting the four lower-order bit planes to zero. (c) Difference between (a) and (b). (d) Histogram-equalized difference image. (Original image courtesy of Ms. Melissa D. Binde, Swarthmore College, Swarthmore, PA).

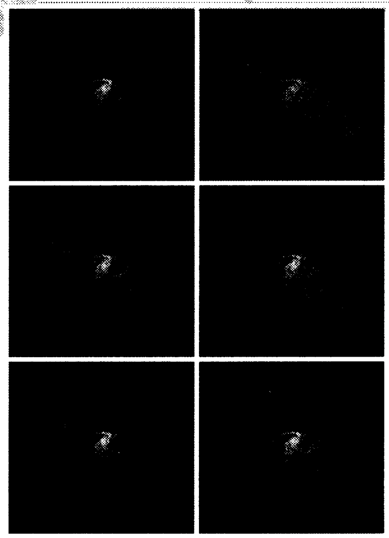
Chapter 3
Image Enhancement in the Spatial Domain



a b

FIGURE 3.29 Enhancement by image subtraction. (a) Mask image. (b) An image (taken after injection of a contrast medium into the bloodstream) with mask subtracted out.

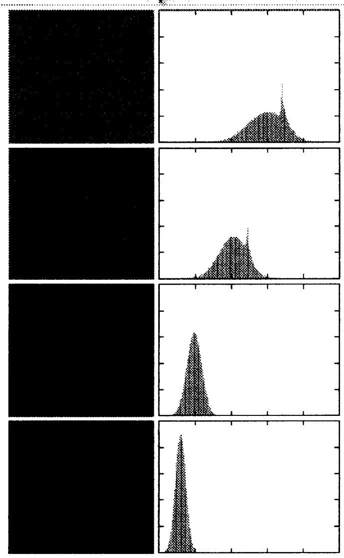
Chapter 3 Image Enhancement in the Spatial Domain



a b
c d
e f

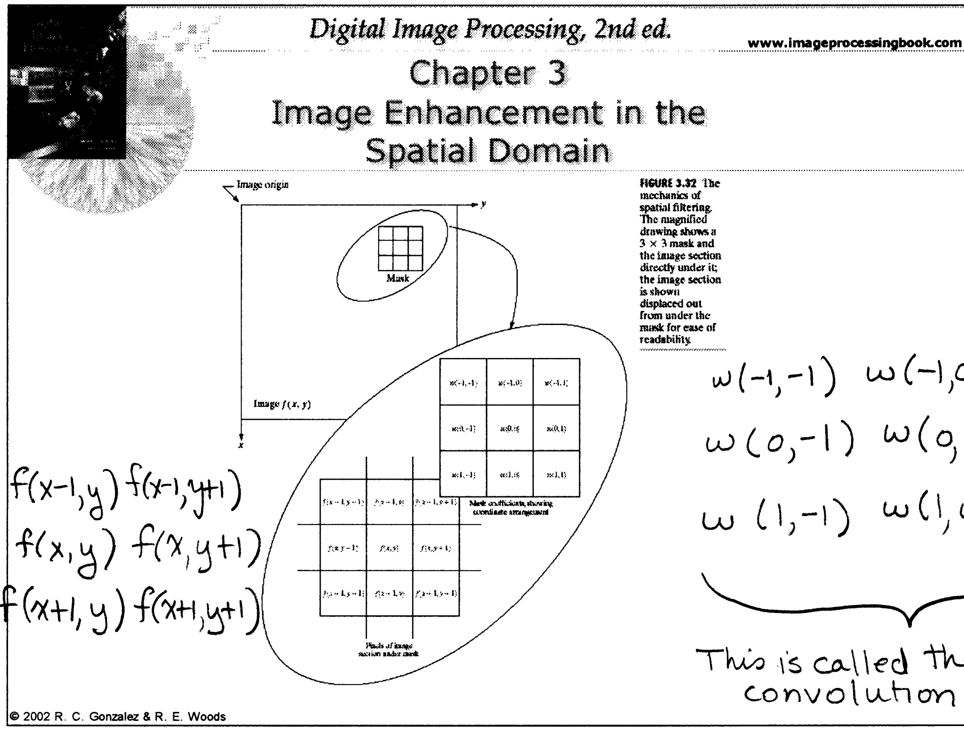
FIGURE 3.30 (a) Image of Galaxy Pair NGC 3314. (b) Image corrupted by additive Gaussian noise with zero mean and a standard deviation of 64 gray levels. (c)–(f) Results of averaging $K = 8, 16, 64,$ and 128 noisy images. (Original image courtesy of NASA.)

Chapter 3
Image Enhancement in the Spatial Domain



a b

FIGURE 3.31
(a) From top to bottom: Difference images between Fig. 3.30(a) and the four images in Figs. 3.30(c) through (f), respectively. (b) Corresponding histograms.



$f(x-1,y-1)$ $f(x-1,y)$ $f(x-1,y+1)$
$f(x,y-1)$ $f(x,y)$ $f(x,y+1)$
$f(x+1,y-1)$ $f(x+1,y)$ $f(x+1,y+1)$

$w(-1,-1)$ $w(-1,0)$ $w(-1,1)$
$w(0,-1)$ $w(0,0)$ $w(0,1)$
$w(1,-1)$ $w(1,0)$ $w(1,1)$

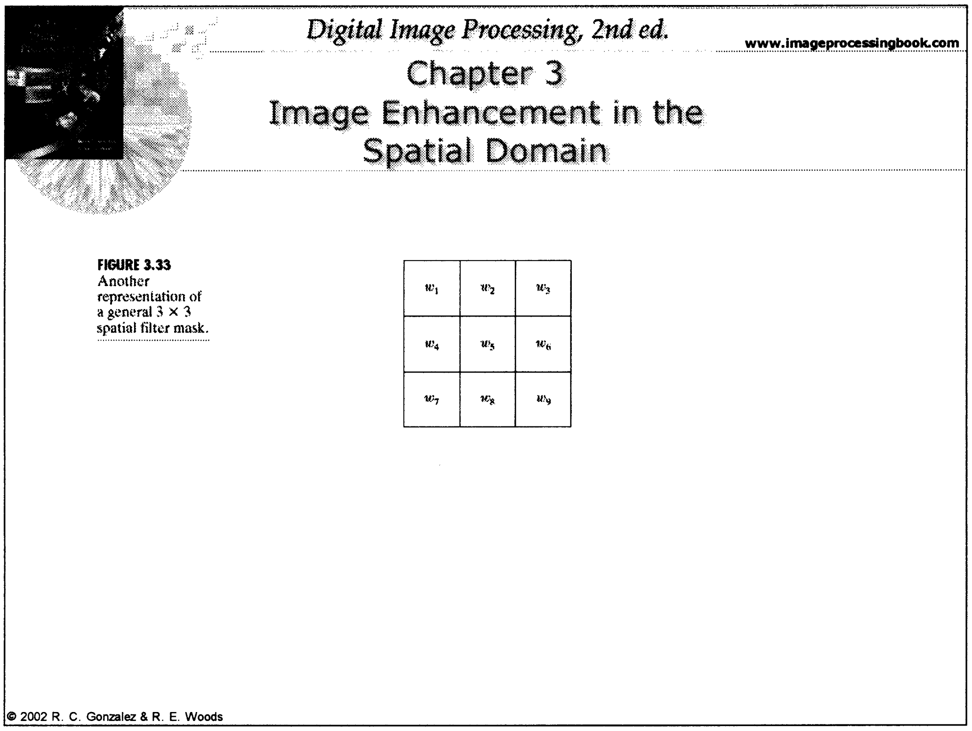
This is called the convolution mask.

Linear filtering of an image f of size $M \times N$ with a filter mask w of size $m \times n$ is given by

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

This is essentially a digital convolution.

In practice, we multiply each element of the mask by the corresponding image element (underneath it). We then slide the mask one to the right and repeat. At the end of each row we move the template to the beginning of the next row and repeat.

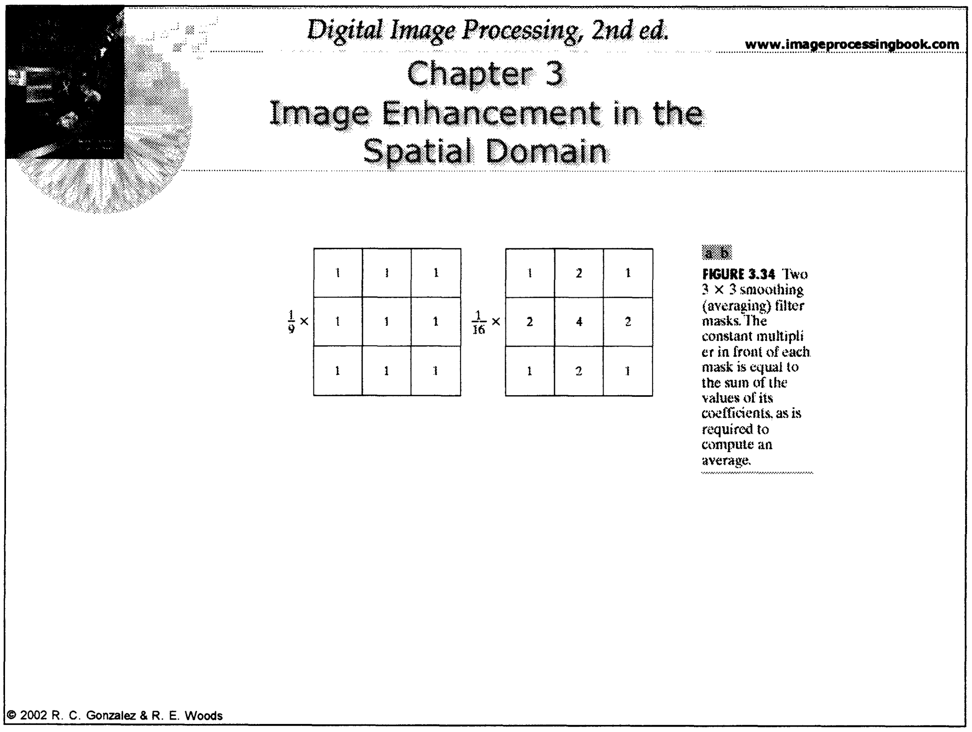


In essence, the response is a sum of products given by

$$R = \sum_{i=1}^{9} w_i z_i \qquad \text{for a 3x3 mask}$$

But what happens when the filter runs out of pixels at the edge of the image.

- simply limit the mask so it can't go outside the image (this reduces the image size!)
- filter all pixels only with the part of the mask fully contained in the image, i.e., a partial mask
- padding – either replicate rows & columns at the edges, or extend the image by adding rows & columns of some constant gray level such as "1"



One of the simplest things you can do is average the pixels contained in the mask neighborhood. This has the effect of smoothing and will be shown to be a low-pass filter.

From Fig 3.33 $\sum_{i=1}^{9} w_i z_i$

a real average would be $\frac{1}{9} \sum_{i=1}^{9} z_i$

↑ scaling constant to make sure result is in range [0,1] or [0,255] as appropriate.

Other constants can be used. Or you can use other weights as shown in (b). This is a center weighted average.

Digital Image Processing, 2nd ed. www.imageprocessingbook.com

Chapter 3
Image Enhancement in the Spatial Domain

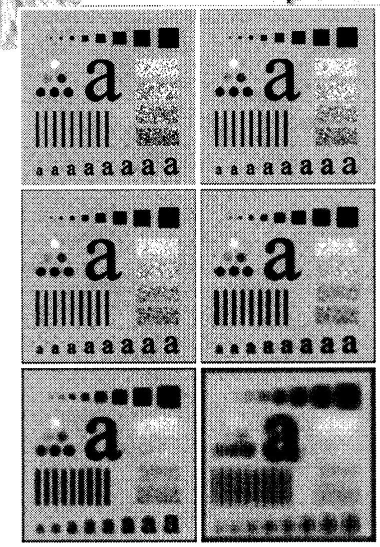


FIGURE 3.35 (a) Original image, of size 500 × 500 pixels. (b)–(f) Results of smoothing with square averaging filter masks of sizes n = 3, 5, 9, 15, and 35, respectively. The black squares at the top are of sizes 3, 5, 9, 15, 25, 35, 45, and 55 pixels, respectively; their borders are 25 pixels apart. The letters at the bottom range in size from 10 to 24 points, in increments of 2 points; the large letter at the top is 60 points. The vertical bars are 5 pixels wide and 100 pixels high; their separation is 20 pixels. The diameter of the circles is 25 pixels, and their borders are 15 pixels apart; their gray levels range from 0% to 100% black in increments of 20%. The background of the image is 10% black. The noisy rectangles are of size 50 × 120 pixels.

The size of the mask determines the amount of smoothing.

(b) 3x3 The smallest amount of smoothing

(c) 5x5

(d) 9x9

(e) 15x15

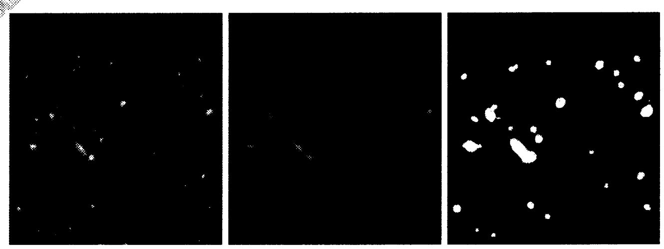
(f) 35x35 most smoothing

vertical bars are 5x 100 pixels with 20 pixel separation

Squares are 3x3 to 55x55

noisy rectangles are 50 x 120 pixels

Chapter 3
Image Enhancement in the Spatial Domain



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.)

original — 15x15 averaging — thresholding

Chapter 3
Image Enhancement in the Spatial Domain

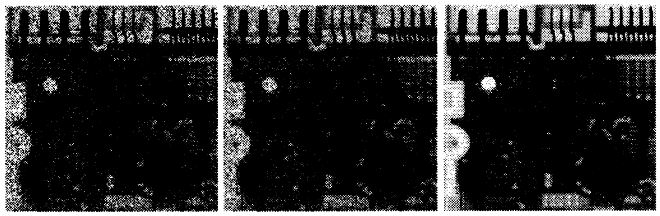


FIGURE 3.37 (a) X-ray image of circuit board corrupted by salt-and-pepper noise. (b) Noise reduction with a 3 × 3 averaging mask. (c) Noise reduction with a 3 × 3 median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

original "noisy" image | 3x3 averaging | 3x3 median

This is an example of an order-statistics filter. Produce a sorted list of the 9 pixel values in the mask and replace the input pixel by some function of the ranking.

median filter – replace the input pixel by the median (middle) of the 9 pixel values (for a 3x3 mask)

good match to eliminating certain types of noise salt & pepper (impulse noise)

x-ray detectors have a lot of impulse noise