BLG453E COMPUTER VISION

Fall 2018 Term

Week 5

Istanbul Technical University
Computer Engineering Department

Instructor: Prof. Gözde ÜNAL

Teaching Assistant: Enes ALBAY

Learning Outcomes of the Course

Students will be able to:

1. Discuss the main problems of computer (artificial) vision, its uses and applications
2. Design and implement various image transforms: point-wise transforms, neighborhood operation-based spatial filters, and geometric transforms over images
3. Define and construct segmentation, feature extraction, and visual motion estimation algorithms to extract relevant information from images
4. Construct least squares solutions to problems in computer vision
5. Describe the idea behind dimensionality reduction and how it is used in data processing
6. Apply object and shape recognition approaches to problems in computer vision

Week 4: LOs: Spatial Image Filtering: Neighborhood Operations

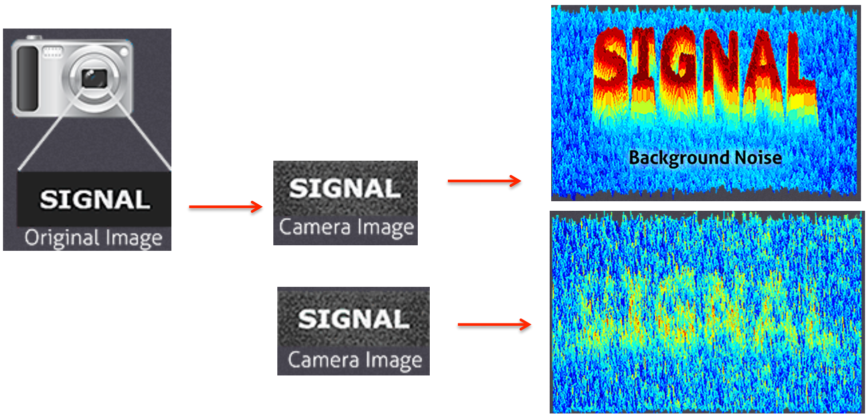
At the end of Week 4: Students will be able to:

2. Design and implement various image transforms: neighborhood operation-based spatial filters

http://en.wikipedia.org/wiki/Image_noise

NOISE: any undesired information that contaminates an image

- Digital image acquisition process, which converts a light signal into a continuous electrical signal that is then sampled, is the primary process by which noise appears in digital images.
- Noise increases with the sensitivity setting in the camera, length of the exposure, temperature, and even varies among different camera models due to different electronics.

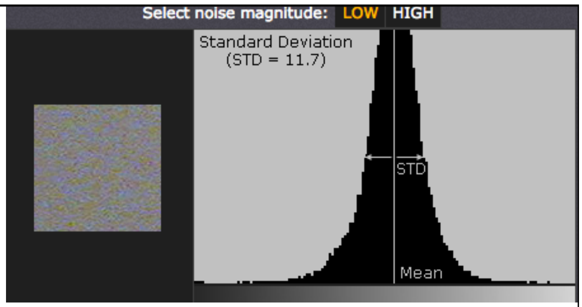


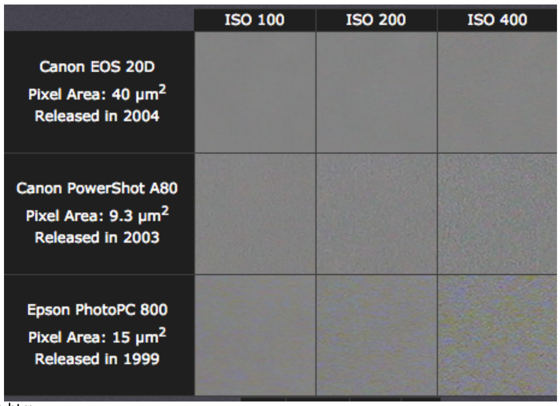
http://www.cambridgeincolour.com/tutorials/image-noise.htm

3D representation of the 2D image

Types of NOISE

- Digital cameras: Most typical: Random noise



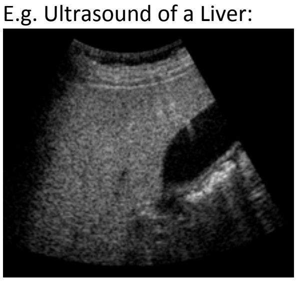


If each of the patches had zero noise, histogram would be a delta function peak located at the mean. As noise levels increase, so does the width of this histogram.

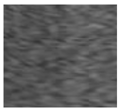
http://www.cambridgeincolour.com/tutorials/image-noise.htm

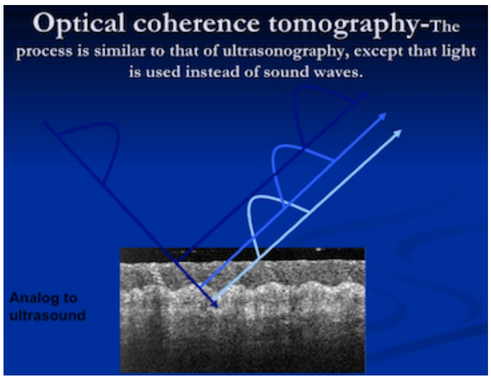
Do we only have digital camera (optical) images ?

E.g. Ultrasound of a Liver:



Ultrasound Images exhibit Speckle Noise





E.g. Optical Coherence Tomography image of retina

http://www.slideshare.net/tapan_jakkal/optical-coherence-tomography

NOISE PROBABILITY DENSITY FUNCTIONS (PDFs)

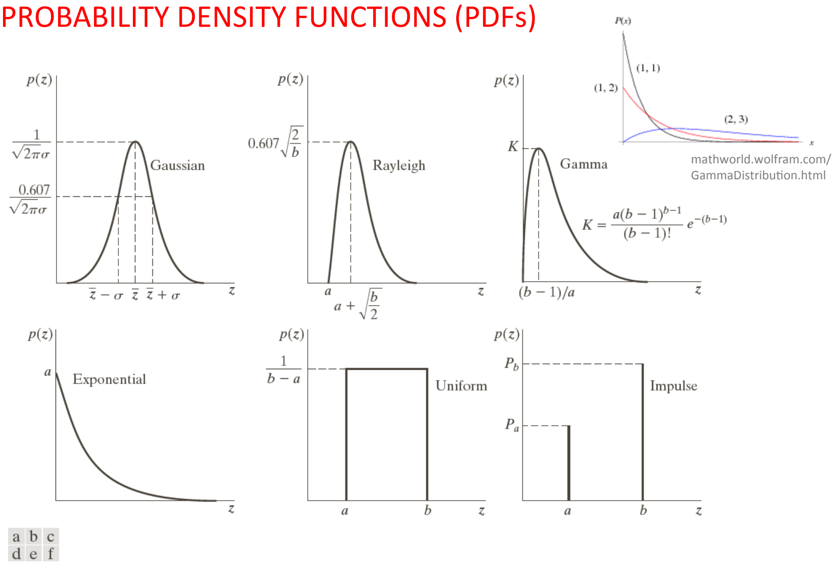


FIGURE 5.2 Some important probability density functions.

Digital Image Processing: Gonzalez and Woods Book: Filtering Chapter

Mathematical Expressions for the PDFs

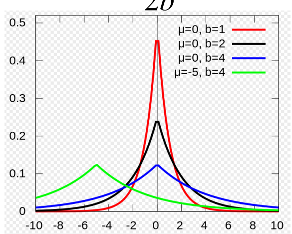
$$H_{Gaussian} = \frac{1}{\sqrt{2\pi\sigma}} e^{-(g-m)^2/2\sigma^2}$$

$$H_{Rayleigh} = \frac{2g}{\alpha} e^{-g^2/\alpha}$$

$$H_{Exponential} = \frac{e^{-g/\alpha}}{\alpha}$$

$$H_{Uniform} = \begin{cases} \dfrac{1}{b-a} & \text{for a} \le \text{g} \le \text{b} \\ 0 & \text{elsewhere} \end{cases}$$

$$H_{Laplace} = \frac{1}{2b} e^{-|g-m|/b}$$



$$H_{Salt\&Pepper} = \begin{cases} A & \text{for g = a(pepper)} \\ B & \text{for g = b(salt)} \end{cases}$$

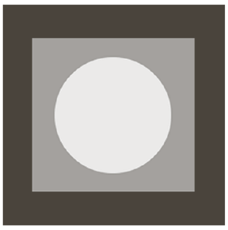


FIGURE 5.3 Test pattern used to illustrate the characteristics of the noise PDFs shown in Fig. 5.2.

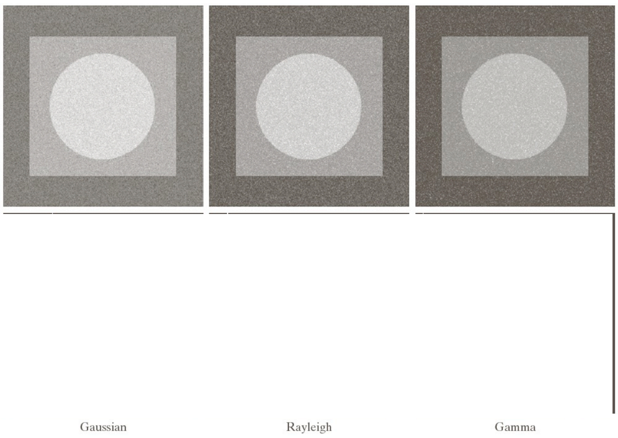


FIGURE 5.4 Images and histograms resulting from adding Gaussian, Rayleigh, and gamma noise to the image in Fig. 5.3.

Digital Image Processing: Gonzalez and Woods Book: Filtering Chapter

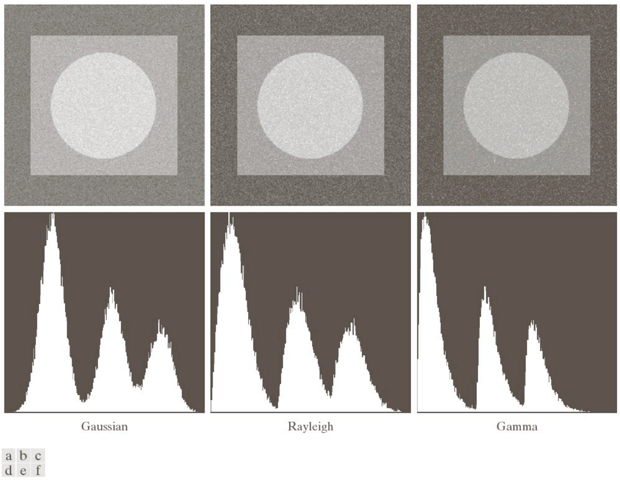


FIGURE 5.4 Images and histograms resulting from adding Gaussian, Rayleigh, and gamma noise to the image in Fig. 5.3.

Digital Image Processing: Gonzalez and Woods Book: Filtering Chapter

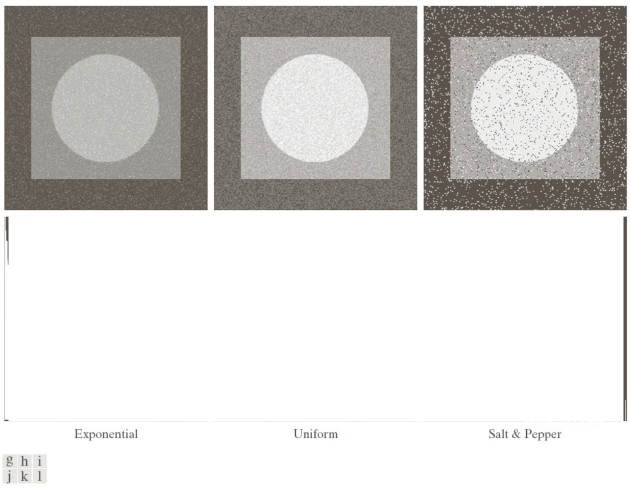


FIGURE 5.4 *(Continued)* Images and histograms resulting from adding exponential, uniform, and salt and pepper noise to the image in Fig. 5.3.

Digital Image Processing: Gonzalez and Woods Book: Filtering Chapter

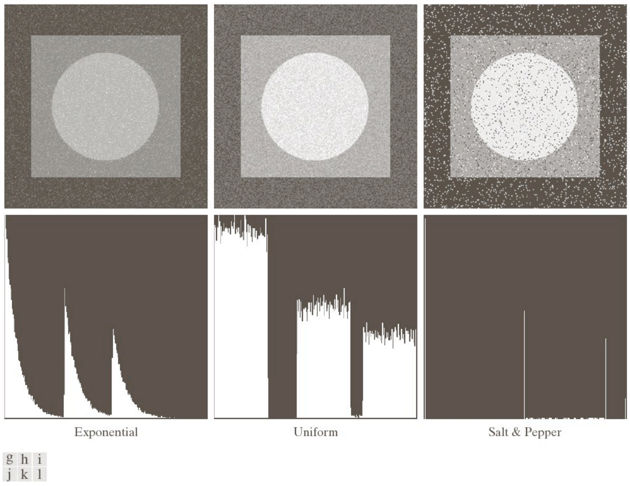


FIGURE 5.4 *(Continued)* Images and histograms resulting from adding exponential, uniform, and salt and pepper noise to the image in Fig. 5.3.

Digital Image Processing: Gonzalez and Woods Book: Filtering Chapter

Original. Gaussian noise. Impulse noise.

Impulse type noise in an image

- Random appearance of black and white pixel intensity levels throughout the image

* Can be caused by analog-to-digital converter errors, bit errors in transmission, …

Impulse Noise

- Impulse noise may corrupt any signal including digital images just due to occasional inversion of a single bit representing the intensity value in some pixel
- The general model of impulse noise is

$$g(x,y) = \begin{cases} p_n, \eta(x,y) \\ 1 - p_n, f(x,y) \end{cases}$$

where p_n is the probability of distortion (p_n in percents $p_n \cdot 100\%$ is called the corruption rate)

η A certain intensity value to replace the image intensity f(x,y)

18

Impulse Noise

- Unlike additive noise, which just distorts intensity values, impulse noise completely replaces the intensity values in those pixels that are corrupted.
- The higher is corruption rate, the more pixels are affected by noise and the more difficult is filtering

19

Salt-and-Pepper Impulse Noise

- Salt-and-Pepper impulse noise replaces the intensity values in the image $f(x,y)$ by 0s and 255s with some certain probabilities

$$g(x,y) = \begin{cases} p_0, & 0 \\ p_{255}, & 255 \\ 1-(p_0+p_{255}), & f(x,y) \end{cases}$$

- Since 0 is black and 255 is white, a corrupted image is covered by white and black impulses ("salt-and-pepper")
- The corruption rate is

$$(p_0 + p_{255}) \cdot 100\%$$

21

FILTERING

Spatial Filtering or Frequency Filtering ?

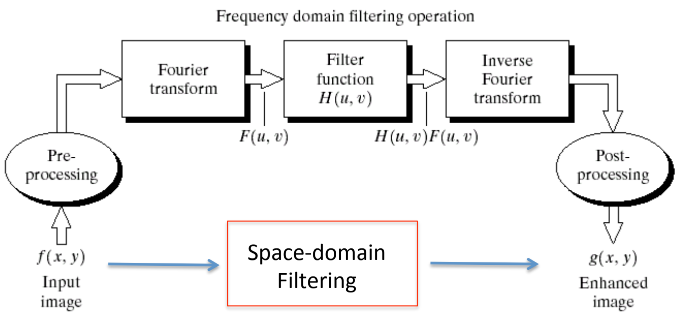


FIGURE 4.5 Basic steps for filtering in the frequency domain.

In this course, we will work with spatial filtering only.

MECHANICS OF SPATIAL FILTERING

https://towardsdatascience.com/intuitively-understanding-convolutions-for-deep-learning-1f6f42faee1

MECHANICS OF SPATIAL FILTERING

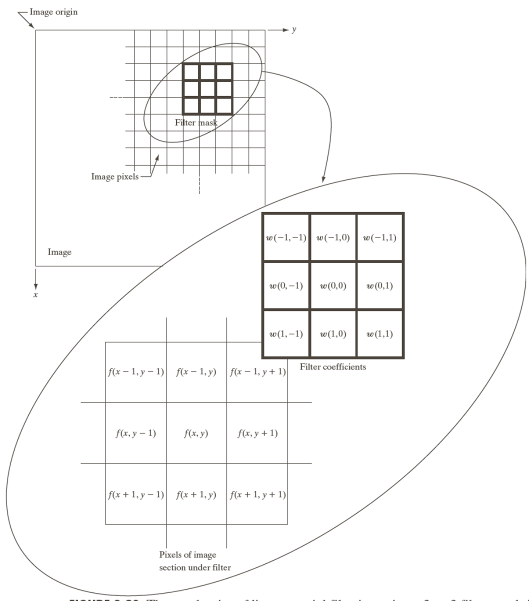


FIGURE 3.28 The mechanics of linear spatial filtering using a 3 × 3 filter mask. The form chosen to denote the coordinates of the filter mask coefficients simplifies writing expressions for linear filtering.

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

f(x,y): image
w(s,t): filter mask
g(x,y): filter response: Convolution

Each pixel in *w* visits every pixel in *f*.

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

Figure 4.11 Local average mask.

101 * 1/9	100 * 1/9	103 * 1/9	105	107	105	103	110
110 * 1/9	140 * 1/9	120 * 1/9	122	130	130	121	120
134 * 1/9	134 * 1/9	135 * 1/9	131	137	138	120	121
132	132	132	133	133	150	160	155
134	140	140	135	140	156	160	174
130	138	139	150	169	175	170	165
126	133	138	149	163	169	180	185
130	140	150	169	178	185	190	200

Figure 4.12 Image smoothing using local average mask.

1/9 * (101 + …+ 135) = 119.67
1/9 * (100 + …+ 131) = 121.11
….

Correlation

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

Convolution

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

Correlation yields a copy of the function, but rotated by 180°.

Spatial Correlation And Convolution

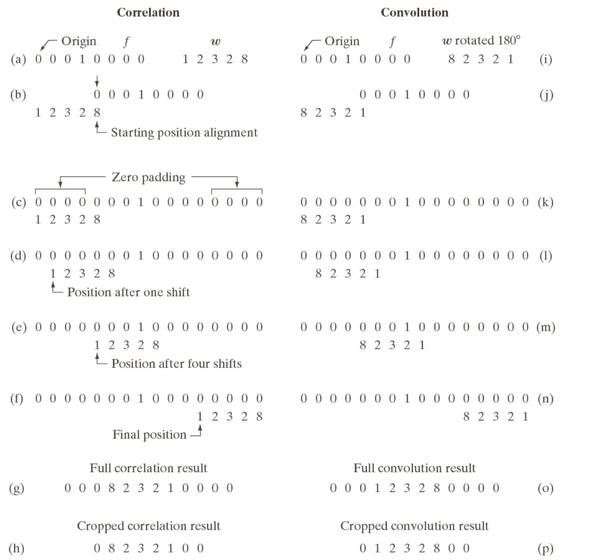


FIGURE 3.29 Illustration of 1-D correlation and convolution of a filter with a discrete unit impulse. Note that correlation and convolution are functions of *displacement*.

Digital Image Processing: Gonzalez and Woods Book: Filtering Chapter

Example in 2D

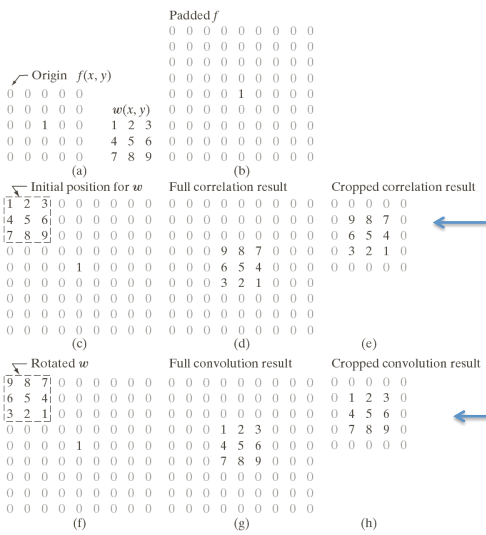


FIGURE 3.30 Correlation (middle row) and convolution (last row) of a 2-D filter with a 2-D discrete, unit impulse. The 0s are shown in gray to simplify visual analysis.

Result is flipped in both axes (horizontal and vertical x and y) compared to convolution result

Convolution result

Digital Image Processing: Gonzalez and Woods Book: Filtering Chapter

NOISE REMOVAL USING SPATIAL FILTERS

- Smoothing Spatial Filters
  - Averaging (linear) filters
  - Order-statistic (nonlinear) filters
  - Adaptive filters
- Sharpening Spatial Filters
  - Unsharp Masking and Highboost filtering
- Morphological Image Filters: If time permits, but you should take a look yourself. Used widely in image filtering
- Typically these filters operate on small subimages, *windows*.

SMOOTHING SPATIAL FILTERS

Why smooth?

To reduce noise!

To increase signal to noise ratio!

SMOOTHING SPATIAL FILTERS

Why smooth?

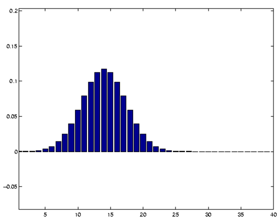
To increase signal to noise ratio!

$$SNR = \frac{\mu_{signal}}{\sigma_{noise}}$$

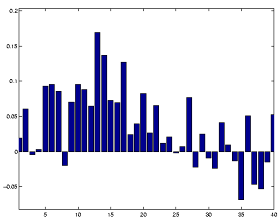
Signal average value

Noise (or background) standard deviation

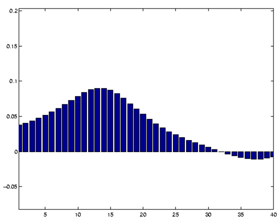
Histograms:



Original signal



Original w. random noise



Recovered signal by Gaussian smoothing

Averaging (smoothing) filter masks

Don't forget: You have to normalize the filter coefficients to sum to 1 for averaging filters

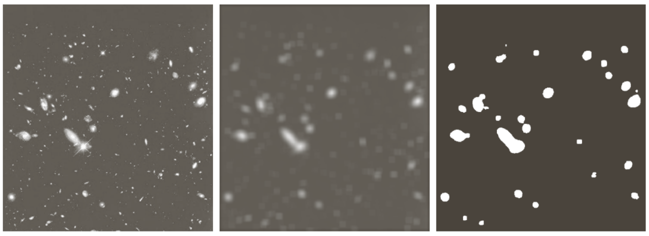
$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad \frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

a b

FIGURE 3.32 Two 3×3 smoothing (averaging) filter masks. The constant multiplier in front of each mask is equal to 1 divided by the sum of the values of its coefficients, as is required to compute an average.

$$g(x,y) = \sum_{s=-1}^{1}\sum_{t=-1}^{1} w(s,t) f(x+s, y+t) \qquad g(x,y) = \frac{\sum_{s=-1}^{1}\sum_{t=-1}^{1} w(s,t) f(x+s, y+t)}{\sum_{s=-1}^{1}\sum_{t=-1}^{1} w(s,t)}$$

Note that: Spatial filtering is not only for noise reduction! You can use it as pre-processing for object/blob detection, etc.



a b c

FIGURE 3.34 (a) Image of size 528 × 485 pixels from the Hubble Space Telescope. (b) Image filtered with a 15 × 15 averaging mask. (c) Result of thresholding (b). (Original image courtesy of NASA.)

Digital Image Processing: Gonzalez and Woods Book: Filtering Chapter

Some advanced filtering examples from medical imaging:

Fig. 4 **a** X-ray image of the coronary arteries. Iodine contrast medium is injected in the vessels, which causes them to absorb more X-ray radiation than the surrounding tissues. **b** The vesselness transform of the X-ray image enhances the tubular structures, and suppresses the other image features

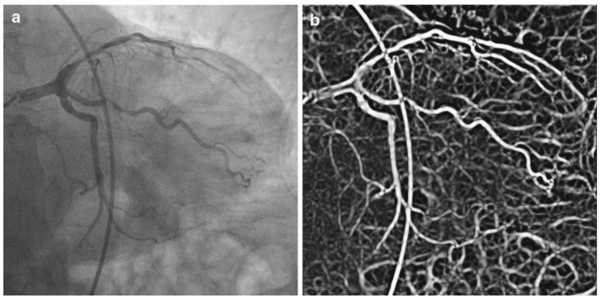
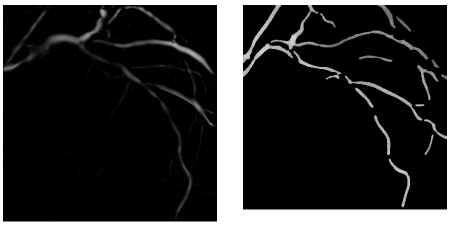


Figure from: D. Ruijters et al. Vesselness-based 2D–3D registration of the coronary arteries, Int J CARS, 2009.

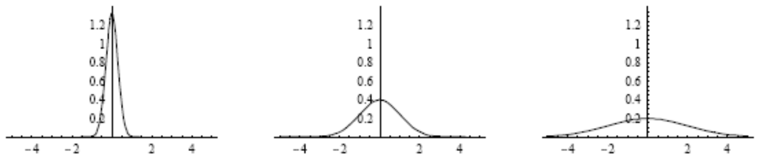
Original Image

Vesselness filtering



Gaussian Kernel

$$G_{1D}(x;\sigma)=\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{x^2}{2\sigma^2}}$$



$$G_{2D}(x,y;\sigma)=\frac{1}{2\pi\sigma}e^{-\frac{x^2+y^2}{2\sigma^2}}$$

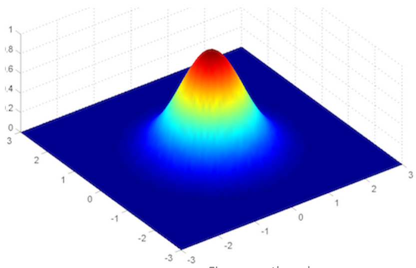


Figure: mathworks.com

$$G_{ND}(\vec{x};\sigma)=\frac{1}{\left(\sqrt{2\pi}\sigma\right)^N}e^{-\frac{|\vec{x}|^2}{2\sigma^2}}$$

Gaussian kernels

σ=0.391 pixels (3x3)

1	4	1
4	12	4
1	4	1

σ=0.625 pixels (5x5)

1	2	3	2	1
2	7	11	7	2
3	11	17	11	3
2	7	11	7	2
1	2	3	2	1

σ=1.0 pixels (9x9)

0	0	1	1	1	1	1	0	0
0	1	2	3	3	3	2	1	0
1	2	3	6	7	6	3	2	1
1	3	6	9	11	9	6	3	1
1	3	7	11	12	11	7	3	1
1	3	6	9	11	9	6	3	1
1	2	3	6	7	6	3	2	1
0	1	2	3	3	3	2	1	0
0	0	1	1	1	1	1	0	0

σ=1.6 pixels (11x11)

1	1	1	2	2	2	2	2	1	1	1
1	2	2	3	4	4	4	3	2	2	1
1	2	4	5	6	7	6	5	4	2	1
2	3	5	7	8	9	8	7	5	3	2
2	4	6	8	10	11	10	8	6	4	2
2	4	7	9	11	12	11	9	7	4	2
2	4	6	8	10	11	10	8	6	4	2
2	3	5	7	8	9	8	7	5	3	2
1	2	4	5	6	7	6	5	4	2	1
1	2	2	3	4	4	4	3	2	2	1
1	1	1	2	2	2	2	2	1	1	1

σ=2.56 pixels (15x15)

2	2	3	4	5	5	6	6	6	5	5	4	3	2	2
2	3	4	5	7	7	8	8	8	7	7	5	4	3	2
3	4	6	7	9	10	10	11	10	10	9	7	6	4	3
4	5	7	9	10	12	13	13	13	12	10	9	7	5	4
5	7	9	11	13	14	15	16	15	14	13	11	9	7	5
5	7	10	12	14	16	17	18	17	16	14	12	10	7	5
6	8	10	13	15	17	19	19	19	17	15	13	10	8	6
6	8	11	13	16	18	19	20	19	18	16	13	11	8	6
6	8	10	13	15	17	19	19	19	17	15	13	10	8	6
5	7	10	12	14	16	17	18	17	16	14	12	10	7	5
5	7	9	11	13	14	15	16	15	14	13	11	9	7	5
4	5	7	9	10	12	13	13	13	12	10	9	7	5	4
3	4	6	7	9	10	10	11	10	10	9	7	6	4	3
2	3	4	5	7	7	8	8	8	7	7	5	4	3	2
2	2	3	4	5	5	6	6	6	5	5	4	3	2	2

Original.

Gaussian noise.

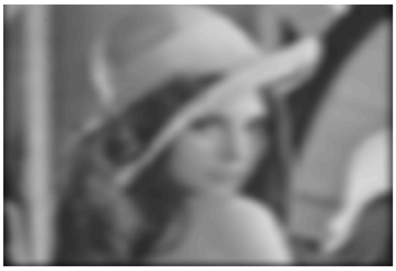
Impulse noise.

Gaussian Smoothing

Noisy.

Smoothed.

Smoothed a bit more.



… and even more.

Additive Noise model and Multiplicative Noise model

* Additive noise corrupts the data by an addition process

added noise $\eta(x,y)$

original image $f(x,y) \longrightarrow \oplus \longrightarrow g(x,y)$ Noise corrupted image

$$g(x,y) = f(x,y) + \eta(x,y)$$

e.g. Thermal noise

To remove additive random noise such as Gaussian noise, linear filters such as averaging filters can be used

* Multiplicative noise corrupts the data through a multiplicative process

$\eta(x,y)$

original image $f(x,y) \longrightarrow \otimes \longrightarrow g(x,y)$ Noise corrupted image

$$g(x,y) = f(x,y) \times \eta(x,y)$$

e.g. Salt and Pepper noise is multiplicative
Speckle noise in sound/ultrasound, radar, global illumination variations, ...

e.g. global illumination in an image is like a noise mask multiplying pixels

Averaging (mean) filters

Response based on averaging pixel intensities in a neighborhood/window around the current pixel

$$g(x,y) \longrightarrow \boxed{\text{Filtering}} \longrightarrow \hat{f}(x,y)$$

Noise corrupted image — Filtered image

Arithmetic mean

- Noise reduced by blurring
- Works well for random noise, Gaussian noise...

$$\hat{f}(x,y) = \frac{1}{mn} \sum_{(s,t)\in S_{xy}} g(s,t)$$

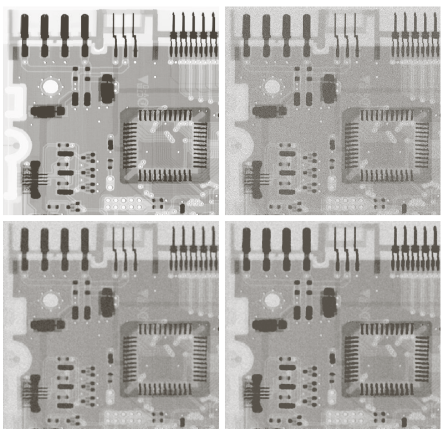
S_{xy} : set of pixels (window) around pixel (x,y)

Geometric mean

- Works well for Gaussian noise
- Loses less image detail than arithmetic mean filter

$$\hat{f}(x,y) = \left[\prod_{(s,t)\in S_{xy}} g(s,t) \right]^{\frac{1}{mn}}$$

original — w/ Gaussian noise



Arithmetic mean filtered — Geometric mean filtered

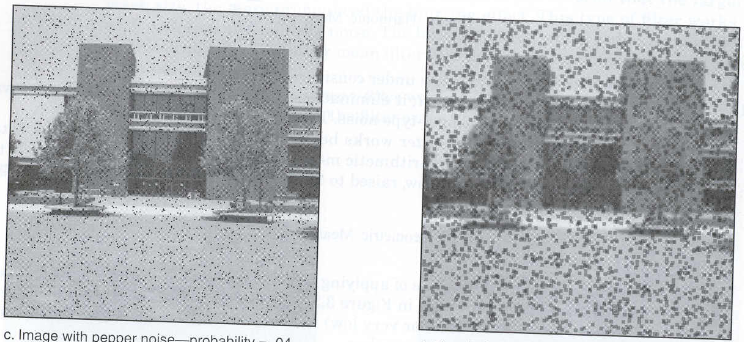
a b
c d

FIGURE 5.7 (a) X-ray image. (b) Image corrupted by additive Gaussian noise. (c) Result of filtering with an arithmetic mean filter of size 3×3. (d) Result of filtering with a geometric mean filter of the same size. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

Digital Image Processing: Gonzalez and Woods Book: Filtering Chapter

If you apply the wrong type of filter…

Figure 3.3-8 (Continued)

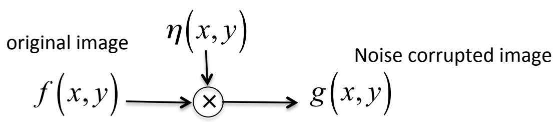


c. Image with pepper noise—probability = .04.

d. Result of geometric mean filter on image with pepper noise; mask size = 3.

Multiplicative Noise model

* Recall Multiplicative noise model:



$$g(x,y) = f(x,y) \times \eta(x,y)$$

* An idea: linearize the model by a nonlinear operation such as taking logarithm:

$$\log g(x,y) = \log f(x,y) + \log \eta(x,y)$$

Then use a linear filter to remove noise, and transform back by $\log^{-1}$

* Typically what we do to remove speckle noise: a nonlinear filters such as order statistics filters are used

see next slide

Order-statistic filters

$$g(x,y) \longrightarrow \boxed{\text{Nonlinear Filtering}} \longrightarrow \hat{f}(x,y)$$

Response based on ordering (ranking) pixel intensity values in a window (neighborhood) around the current pixel location

Median Filter

$$\hat{f}(x,y) = \underset{(s,t)\in S_{xy}}{\text{median}}\{g(s,t)\}$$

- Good for impulse noise
- Less smoothing than averaging filters

Max Filter & Min Filter

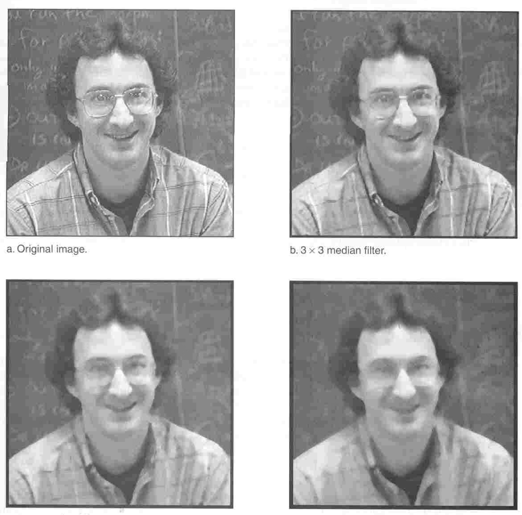
$$\hat{f}(x,y) = \max_{(s,t)\in S_{xy}}\{g(s,t)\}$$

$$\hat{f}(x,y) = \min_{(s,t)\in S_{xy}}\{g(s,t)\}$$

- Max finds brightest points (reduces pepper noise –dark dots)
- Min finds darkest points (reduces salt noise – bright dots)

Image smoothing with median filtering…

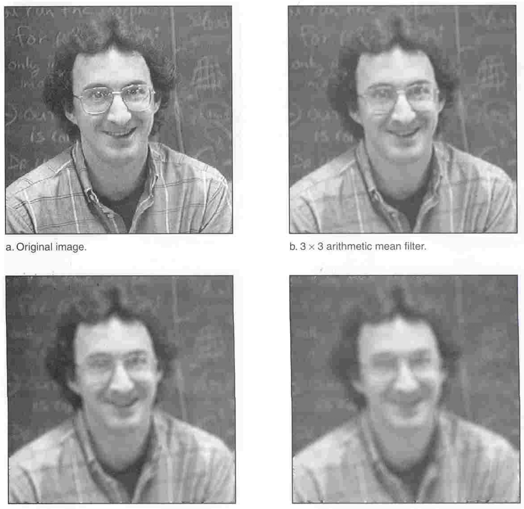
Figure 4.4-3 Image Smoothing with a Median Filter



a. Original image. b. 3×3 median filter. c. 5×5 median filter. d. 7×7 median filter.

Compare to arithmetic mean filter

Figure 4.4-2 Image Smoothing with an Arithmetic Mean Filter



a. Original image.
b. 3 × 3 arithmetic mean filter.
c. 5 × 5 arithmetic mean filter.
d. 7 × 7 arithmetic mean filter.

Median filtering…

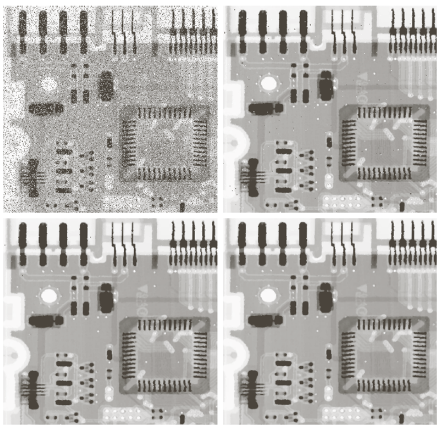


FIGURE 5.10 (a) Image corrupted by salt-and-pepper noise with probabilities $P_a = P_b = 0.1$. (b) Result of one pass with a median filter of size 3 × 3. (c) Result of processing (b) with this filter. (d) Result of processing (c) with the same filter.

Digital Image Processing: Gonzalez and Woods Book: Filtering Chapter

Gaussian noise.

Median filtered.

Impulse noise.

Median filtered.

More Order-statistic filters

Response based on ordering (ranking) pixel intensities

Midpoint Filter

- Combines order-statistic and averaging
- Good for randomly distributed noise

(e.g. Gaussian, uniform)

$$\hat{f}(x,y) = \frac{1}{2}\left[\max_{(s,t)\in S_{xy}}\{g(s,t)\} + \min_{(s,t)\in S_{xy}}\{g(s,t)\}\right]$$

Alpha-trimmed mean Filter

- Remove d/2 highest & lowest intensities
- Useful in removing multiple types of noise

(e.g. S&P + Gaussian)

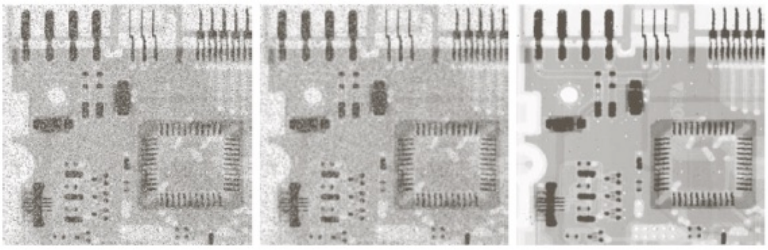
- d=0: arithmetic mean
- d=mn-1: median

$$\hat{f}(x,y) = \frac{1}{mn-d}\sum_{(s,t)\in S_{xy}} g_r(s,t)$$

w. S&P noise

Arithmetic mean filtered

Median filtered



a b c

FIGURE 3.35 (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.)

Digital Image Processing: Gonzalez and Woods Book: Filtering Chapter

w. uniform noise

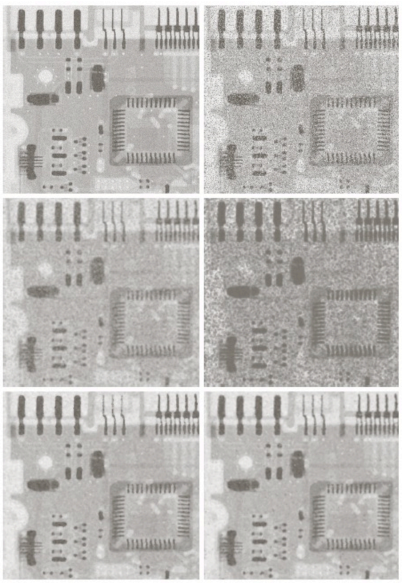
+ S&P noise

Arithmetic mean filtered

Geometric mean filtered

Median filtered

Alpha-trimmed mean filtered



a b
c d
e f

FIGURE 5.12 (a) Image corrupted by additive uniform noise. (b) Image additionally corrupted by additive salt-and-pepper noise. Image (b) filtered with a 5 × 5; (c) arithmetic mean filter; (d) geometric mean filter; (e) median filter; and (f) alpha-trimmed mean filter with $d = 5$.

Digital Image Processing: Gonzalez and Woods Book: Filtering Chapter

Mean-Median Filter

Similarly, for mixed noise (e.g. Gaussian type and impulse type), you can design a blended mean-median filter:

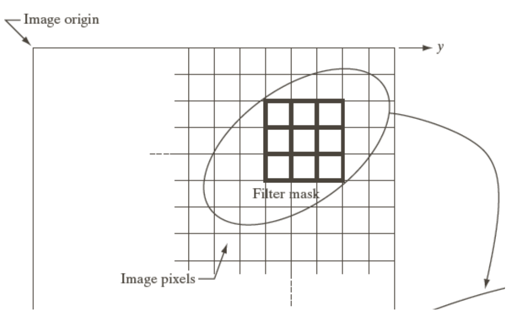
$$\hat{I} = \alpha \ I_{mean} + (1-\alpha) \ I_{median}$$

- α : the blending parameter
- I_{mean} : the mean intensity in your filter window
- $I_{median:}$ the median intensity in your filter window

Adaptive Filters

Adaptive to image characteristic in the neighborhood.

Modification of the gray-level values within an image based on some criterion that adjusts its parameters as local image characteristics change.



Adaptive local noise reduction filter

Makes use of mean (average intensity) and variance (measure of contrast) in an image window (filter size) and global intensity characteristics

$$\hat{f}(x,y) = g(x,y) - \frac{\sigma_n^2}{\sigma_L^2}\left[g(x,y) - m_L\right]$$

m_L : local mean of pixels in S_{xy}

σ_L^2 : local variance of pixels in S_{xy}

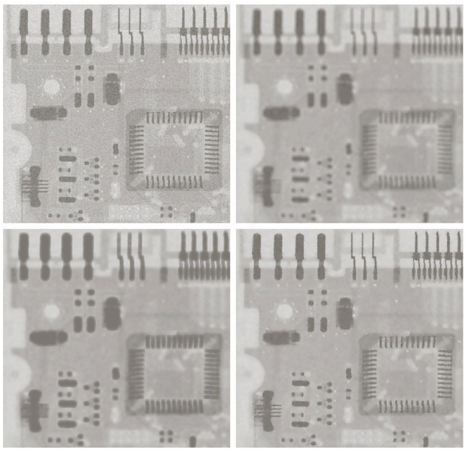
σ_n^2 : variance of overall noise (unknown)

Works best with Gaussian and uniform noise

Issues like how to estimate σ_n^2

w. Gaussian noise

Arithmetic mean filtered



a b
c d

FIGURE 5.13 (a) Image corrupted by additive Gaussian noise of zero mean and variance 1000. (b) Result of arithmetic mean filtering. (c) Result of geometric mean filtering. (d) Result of adaptive noise reduction filtering. All filters were of size 7 × 7.

Geometric mean filtered

Adaptive noise reduction filtered

Digital Image Processing: Gonzalez and Woods Book: Filtering Chapter

Adaptive Contrast Enhancement (ACE) Filter

$$I_{ACE}(r,c) = k_1\left[\frac{m_{I(r,c)}}{\sigma_{loc(r,c)}}\right][I(r,c) - m_{loc}(r,c)] + k_2 m_{loc}(r,c)$$

This term relates to $Coeff\ of\ Variation = \frac{\sigma}{m}$

where $m_{I(r,c)}$ = mean of the entire image I(r,c)

σ_{loc} = local standard deviation (in the window under consideration

m_{loc} = local mean (average in the window under consideration)

k_1 ,k_2 = constants, vary between 0 and 1

Here, the goal is to enhance rather than denoise!

Areas of low contrast (low standard variation) are boosted.
The mean is then added back, to restore local average brightness.

In practice, it is often helpful to shrink the histogram of the image before applying this filter

Image Filtering and Enhancement

Note: Always check the range of the resulting, i.e. the filtered or the enhanced, image intensity:

$$I_{filtered}(x,y) = \begin{cases} 0 & if \quad I_{filtered}(x,y) < 0 \\ 255 & if \quad I_{filtered}(x,y) > 255 \\ I_{filtered}(x,y) & otherwise \end{cases}$$

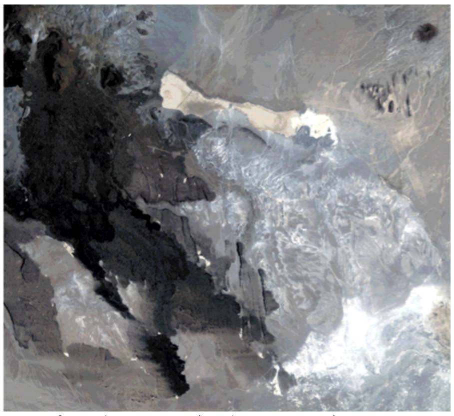
Color Image Filtering

Typically apply the filter either to:
1. each R,G,B channel separately; or
2. only the Intensity (brightness) component in another color space, e.g: H,S,I color space
(we did not cover different color spaces in this course)

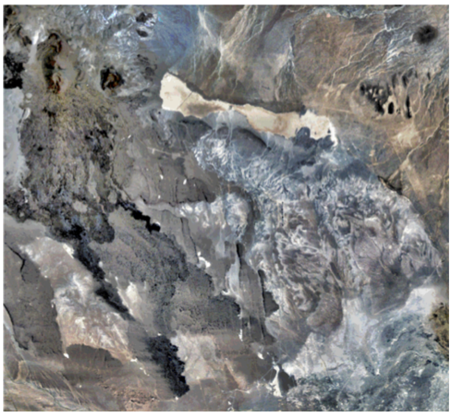
The original image

After a kind of Adaptive Contrast Enhancement

Adaptive Contrast Enhancement Filter example



Portion of a Landsat TM scene (Bands 3, 2, 1 as RGB)



Left: Dark lava flows and bright salt flats reveal little local detail.
Right: Same scene with a kind of RGB adaptive contrast enhancement filter applied
The final image is a weighted average of the LACE filter output (90%) and the original image (10%).
Enhanced contrast in dark and light areas brings out significant surface detail throughout the image.

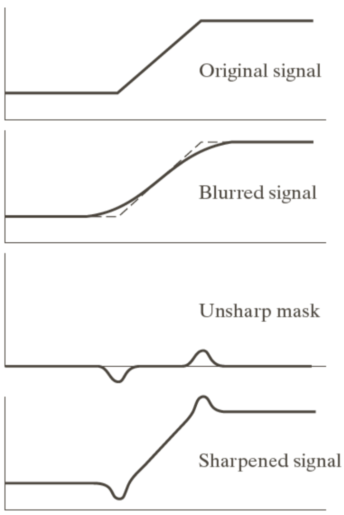
http://www.microimages.com/documentation/TechGuides/56lace10n.pdf

Unsharp Masking and High-Boost Filtering

The unsharp filter is a simple sharpening operator which derives its name from the fact that it enhances edges (and other high frequency components in an image) via a procedure which subtracts an unsharp, or smoothed, version of an image from the original image.

The unsharp filtering technique is commonly used in the photographic and printing industries for crispening edges.

Characteristics of an unsharp filter

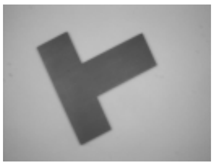


a
b
c
d

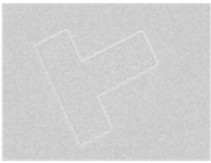
FIGURE 3.39 1-D illustration of the mechanics of unsharp masking. (a) Original signal. (b) Blurred signal with original shown dashed for reference. (c) Unsharp mask. (d) Sharpened signal, obtained by adding (c) to (a).

This edge image can be used for sharpening if we add it back into the original signal.

Consider the simple image object whose strong edges have been slightly blurred by camera focus.



In order to extract a sharpened view of the edges, we smooth this image using a mean filter (kernel size 3×3) and then subtract the smoothed result from the original image



Produce an edge image g(x,y) from an input image f(x,y) via

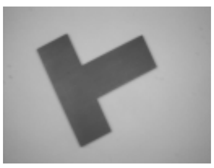
$$g\left(x,y\right) = f\left(x,y\right) - f_{smooth}(x,y)$$

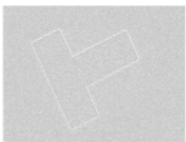
where $f_{smooth}(x,y)$ is a smoothed version of f(x,y)

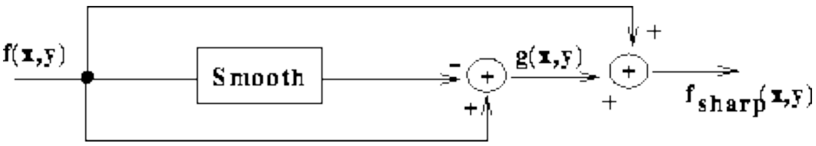
Because we subtract all low frequency components from the original image (*i.e.*, we highpass filtered the image) we are left with only high frequency edge descriptions.

Desired thing: a sharpening operator give us back our original image with the high frequency components enhanced.

In order to achieve this effect,
we now add some proportion of this high frequency image back onto our original image.







The complete unsharp filtering operator.

$$g\left(x,y\right) = f\left(x,y\right) - f_{smooth}(x,y)$$

$$f_{sharp}\left(x,y\right) = f\left(x,y\right) + k * g(x,y)$$

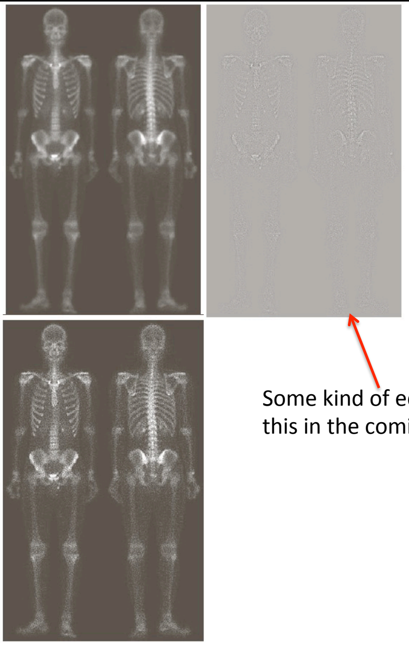
a
b
c
d
e

FIGURE 3.40 (a) Original image. (b) Result of blurring with a Gaussian filter. (c) Unsharp mask. (d) Result of using unsharp masking. (e) Result of using highboost filtering.

k=1

k=4.5

Digital Image Processing: Gonzalez and Woods Book: Filtering Chapter



a b
c d

FIGURE 3.43 (a) Image of whole body bone scan. (b) Laplacian of (a). (c) Sharpened image obtained by adding (a) and (b).

Some kind of edge detector (we will study this in the coming weeks)

Digital Image Processing: Gonzalez and Woods Book: Filtering Chapter

A slight further generalization of unsharp masking is called *high-boost filtering*

$$g(x,y) = A\,f(x,y) - f_{smooth}(x,y)$$

where $A \geq 1$.

0	−1	0
−1	$A + 4$	−1
0	−1	0

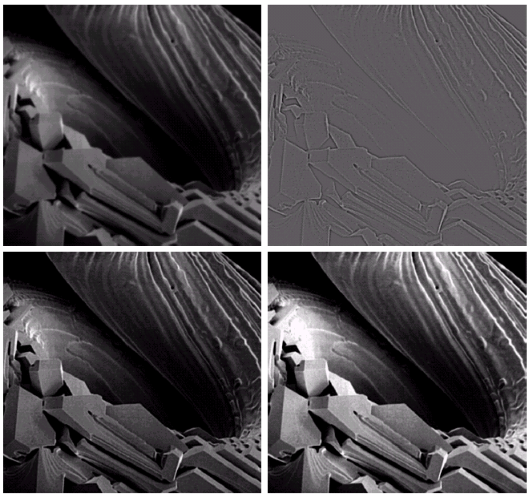
−1	−1	−1
−1	$A + 8$	−1
−1	−1	−1

a b

FIGURE 3.42 The high-boost filtering technique can be implemented with either one of these masks, with $A \geq 1$.

a b
c d

FIGURE 3.43
(a) Same as Fig. 3.41(c), but darker.
(a) Laplacian of (a) computed with the mask in Fig. 3.42(b) using $A = 0$.
(c) Laplacian enhanced image using the mask in Fig. 3.42(b) with $A = 1$. (d) Same as (c), but using $A = 1.7$.

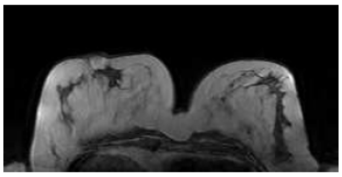


A=1 A=1.7

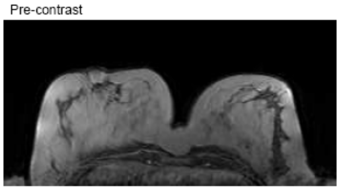
Some kind of edge detector (we will study this in the coming weeks)

Digital Image Processing: Gonzalez and Woods Book: Filtering Chapter

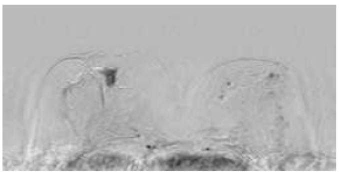
Enhancement Using Subtraction



Pre-contrast



Post-contrast



Subtraction of pre- and post-contrast

MR Mammography (D. Rueckert)

Another Nonlinear Kind of Filtering: MORPHOLOGICAL IMAGE FILTERING

Basic idea: Uses two basic Operations

1. Erode (~ Min Filter)

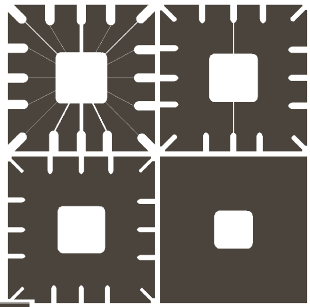
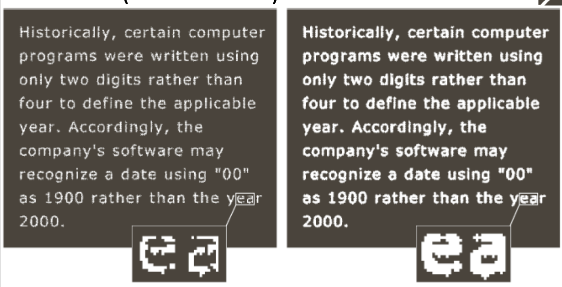


FIGURE 9.5 Using erosion to remove image components. (a) A 486 × 486 binary image of a wire-bond mask. (b)–(d) Image eroded using square structuring elements of sizes 11 × 11, 15 × 15, and 45 × 45, respectively. The elements of the SEs were all 1s.

2. Dilate (~ Max Filter)



0	1	0
1	1	1
0	1	0

FIGURE 9.7 (a) Sample text of poor resolution with broken characters (see magnified view). (b) Structuring element. (c) Dilation of (a) by (b). Broken segments were joined.

Digital Image Processing: Gonzalez and Woods Book

Dilation, in general, causes objects to dilate or grow in size;

Erosion causes objects to shrink.

The amount and the way that they grow or shrink depend upon the choice of the structuring element, which in a sense relates to the filter size and shape

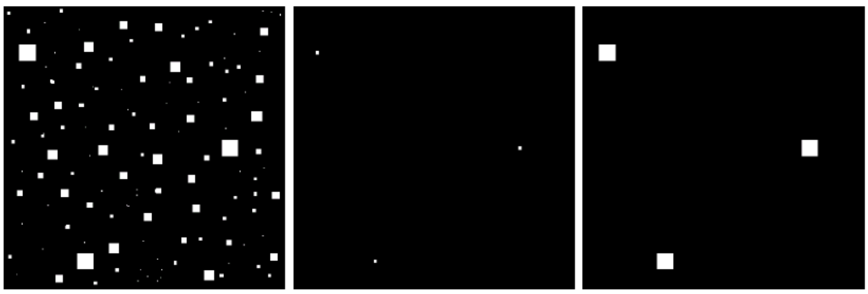


FIGURE 9.7 (a) Image of squares of size 1, 3, 5, 7, 9, and 15 pixels on the side. (b) Erosion of (a) with a square structuring element of 1's, 13 pixels on the side. (c) Dilation of (b) with the same structuring element.

Digital Image Processing: Gonzalez and Woods Book

Next Generation Filters: Learnable

Idea: Let the algorithm learn the filters through various kinds of Artificial Neural Networks, lately known as Deep Learning

Not covered in our class, where you are learning hand-crafted filters, and feature engineering, which is important to know before you work with Learning in (Visual) Data Processing/Science

END OF LECTURE

Recall Learning Objective (LO) for Week 4: Students will be able to:

LO2. Design and implement various image transforms: neighborhood operation-based spatial filters

In the next Assignment:
You will work on Spatial Filters

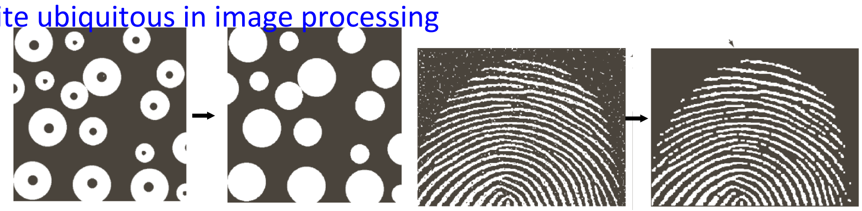
Reading Assignments:

Study this week's topics from your lecture notes

NEXT TIME: We will study Edge Detection

MORPHOLOGICAL IMAGE FILTERING

EXTRA Material: You are not responsible from this and the following slides, but you may certainly (need to) use morphological filters as they are quite ubiquitous in image processing



Goals:

Nonlinear Filtering to clean noise, or emphasize desired features

Extract image components that are useful in representation and description of objects/ shapes in images:

- regions
- boundaries
- skeletons
- convex hulls

Connectivity analysis,

Extract blobs and do a blob analysis for counting etc, …

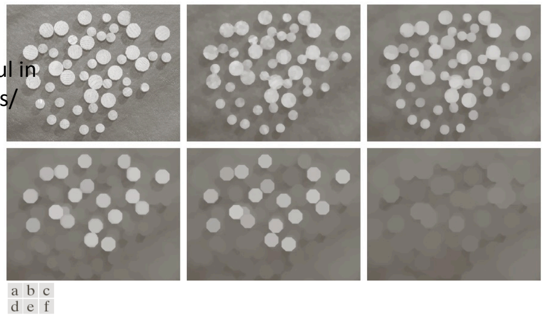
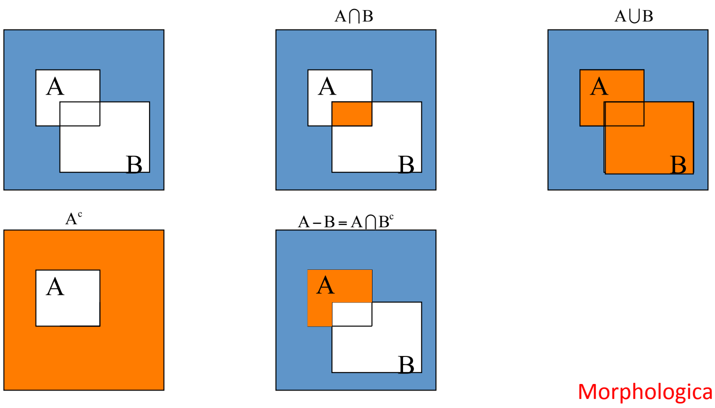


FIGURE 9.41 (a) 531 × 675 image of wood dowels. (b) Smoothed image. (c)–(f) Openings of (b) with disks of radii equal to 10, 20, 25, and 30 pixels, respectively. (Original image courtesy of Dr. Steve Eddins, The MathWorks, Inc.)

Morphological Image Filtering is based on Set Operations



Morphological Image Processing

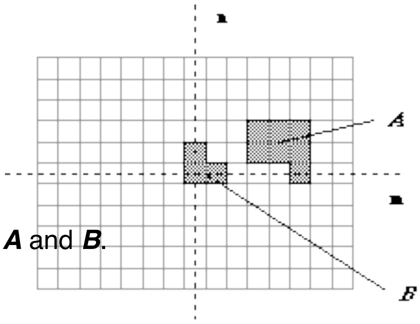
- Binary Morphology: assumes objects are represented in images using only two “color” values, say black and white.
- The coordinates of the black (or white) pixels form a complete description of the objects in the image.
- Objects form the Sets in images
- Another branch of morphology: Grayscale morphological operations

Morphology-based Operations

We defined an image as an (amplitude) function of two, real (coordinate) variables $I(x,y)$ or two, discrete variables $I[m,n]$.

An alternative definition of an image can be based on the notion that an image consists of a set (or collection) of either continuous or discrete coordinates.

In a sense, the set corresponds to the points or pixels that belong to the objects in the image.

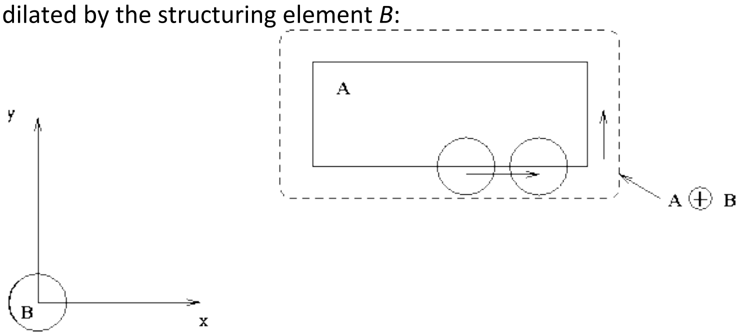


A binary image containing two object sets ***A*** and ***B***.

DILATION

Consider the example where *A* is a rectangle and *B* is a disc centered on the origin.

A is dilated by the structuring element *B*:



The result is a new set (whose outer border is marked by dashed points) made up of all points generated by first:

- Shifting B set over A
- Assigning the intersection of shifted B and A to each center pixel point

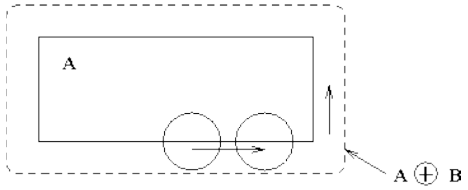
DILATION

Dilation of A by Structuring element B

Mathematically

$$A \oplus B = \{x : (\hat{B}_x \cap A) \neq \varnothing\}$$

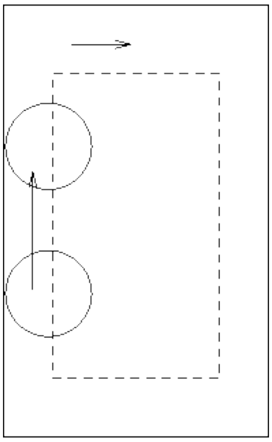
$\hat{B}_x$: Reflection of B about its origin



Since *B* is symmetric: $\hat{B}_x = B_x$

EROSION

Erosion of the object *A* by a structuring element *B* is given by



Interpretation of Erosion: Set of all points x such that B translated by x is completely contained in A
=> no common elements with background A^c

$$A \ominus B = \{x : B_x \subseteq A\}.$$

A is eroded by the structuring element *B* to give the internal dashed shape.

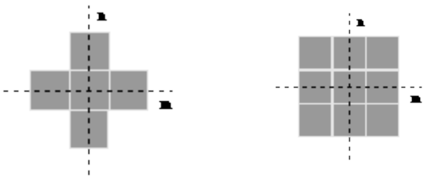
Dilation and erosion are duals of each other with respect to set complementation and reflection. That is,

$$(A \ominus B)^c = A^c \oplus \hat{B}.$$

Dilation, in general, causes objects to dilate or grow in size;

Erosion causes objects to shrink.

The amount and the way that they grow or shrink depend upon the choice of the structuring element.



The standard structuring elements (a) *N_4* **(b) *N_8***

Typically the structuring element *B* is a circular disc in the plane, but it can be any shape.

The image and structuring element sets need not be restricted to sets in the 2D plane, but could be defined in 1, 2, 3 (or higher) dimensions.

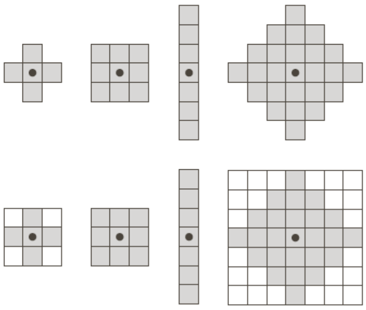


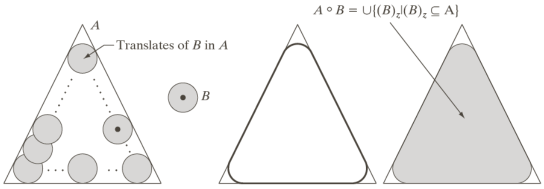
FIGURE 9.2 First row: Examples of structuring elements. Second row: Structuring elements converted to rectangular arrays. The dots denote the centers of the SEs.

Applications of morphological operations

Erosion and dilation can be used in a variety of ways, in parallel and series, to give other transformations including thickening, thinning, skeletonisation and many others.

Now intuitively, dilation expands an image object and erosion shrinks it. We can combine the two operations to obtain new different operations.

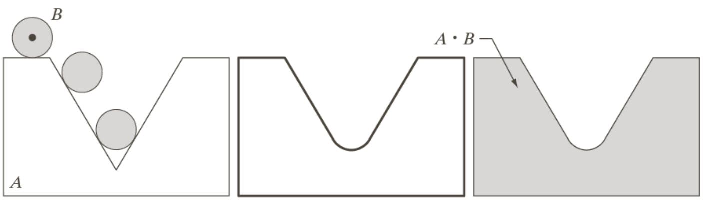
Opening: Erosion followed by dilation



a b c d

FIGURE 9.8 (a) Structuring element B "rolling" along the inner boundary of A (the dot indicates the origin of B). (b) Structuring element. (c) The heavy line is the outer boundary of the opening. (d) Complete opening (shaded). We did not shade A in (a) for clarity.

Closing: Dilation followed by erosion

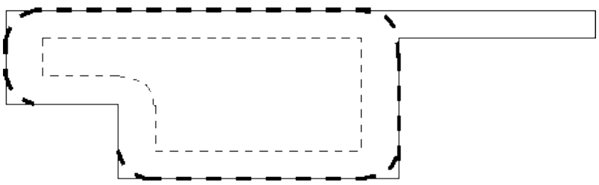


a b c

FIGURE 9.9 (a) Structuring element B "rolling" on the outer boundary of set A. (b) The heavy line is the outer boundary of the closing. (c) Complete closing (shaded). We did not shade A in (a) for clarity.

Opening generally smooths a contour in an image,
breaking narrow isthmuses (a narrow passage connecting two larger structures) and eliminating thin protrusions.

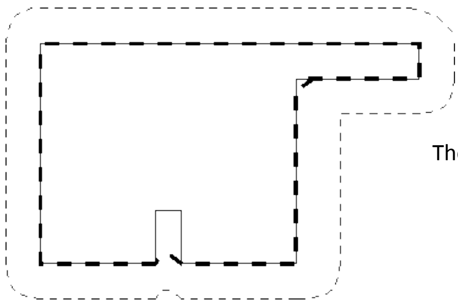
Closing tends to narrow smooth sections of contours, fusing narrow breaks and long thin gulfs, eliminating small holes, and filling gaps in contours.



The opening (given by the dark dashed lines) of *A* (given by the solid lines. The structuring element *B* is a disc.
The internal dashed structure is *A* eroded by *B*.

Closing is the dual operation of opening

$$A \bullet B = (A \oplus B) \ominus B.$$



The closing of *A* by the structuring element *B*.

This is like `smoothing from the outside'. Holes are filled in and narrow valleys are `closed'.

Morphological Filtering:

The morphological filter $(A \circ B) \bullet B$ can be used to eliminate `salt and pepper' noise.

a b
d c
e f

FIGURE 9.11
(a) Noisy image.
(b) Structuring element.
(c) Eroded image.
(d) Opening of A.
(e) Dilation of the opening.
(f) Closing of the opening.
(Original image courtesy of the National Institute of Standards and Technology.)

The important thing to note is that

- morphological operations preserve the main geometric structures of the object.
- Only features `smaller than' the structuring element are affected by transformations.
- All other features at `larger scales' are not degraded.

(This is not the case with linear transformations, such as given by convolution).

Boundary Extraction by Morphology

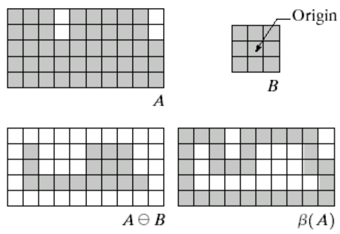
The *boundary* of a set A, denoted by ∂A, can be obtained by first eroding A with B, where B is a suitable structuring element, and then performing the set difference between A and its erosion. That is:

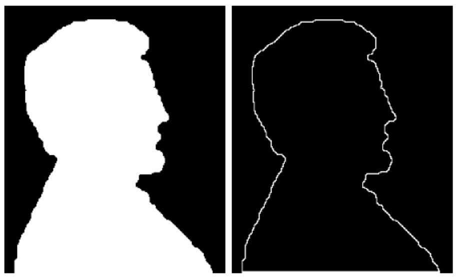
$$\partial A = A - (A \ominus B).$$

Typically, B would be a 3 x 3matrix of 1s.

a b
c d

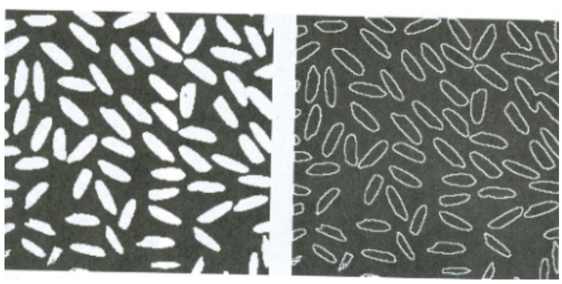
FIGURE 9.13 (a) Set A. (b) Structuring element B. (c) A eroded by B. (d) Boundary, given by the set difference between A and its erosion.





a b

FIGURE 9.14 (a) A simple binary image, with 1's represented in white. (b) Result of using Eq. (9.5-1) with the structuring element in Fig. 9.13(b).



Region filling can be accomplished by Closing operation

or iteratively using dilations, complementation, and intersections

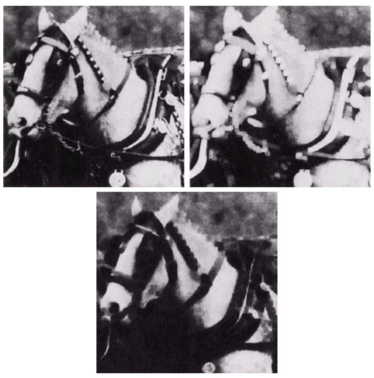
Not covered in class, you can look at Gonzalez Woods Chapter 9 for different kinds of morphological operations

a b c

FIGURE 9.16 (a) Binary image (the white dot inside one of the regions is the starting point for the hole-filling algorithm). (b) Result of filling that region. (c) Result of filling all holes.

Gray-value Morphology

Just: Replace binary sets with gray-valued images
Replace AND operation with MIN operation
OR operation with MAX operation



a b
c

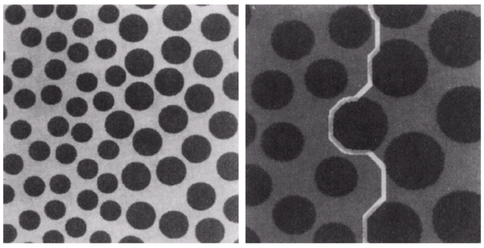
FIGURE 9.29 (a) Original image. (b) Result of dilation. (c) Result of erosion. (Courtesy of Mr. A. Morris, Leica Cambridge, Ltd.)

a) Dilation **b)** Erosion **c)** Smoothing

Texture Segmentation

a b

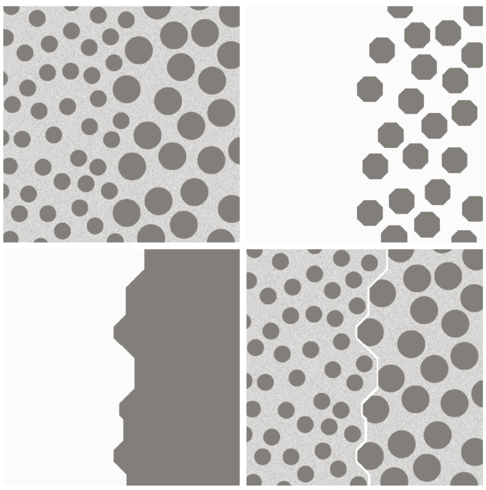
FIGURE 9.35 (a) Original image. (b) Image showing boundary between regions of different texture. (Courtesy of Mr. A. Morris, Leica Cambridge, Ltd.)



Close the image by successively using larger structuring elements.

a b
c d

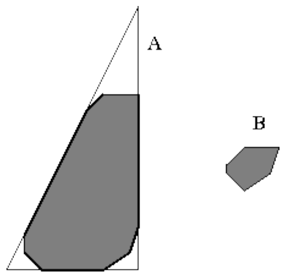
FIGURE 9.43 Textural segmentation. (a) A 600 × 600 image consisting of two types of blobs. (b) Image with small blobs removed by closing (a). (c) Image with light patches between large blobs removed by opening (b). (d) Original image with boundary between the two regions in (c) superimposed. The boundary was obtained using a morphological gradient operation.



(c) Open image b with SE > separation dist btw blobs

Opening is like `rounding from the inside': the opening of *A* by *B* is obtained by taking the union of all translates of *B* that fit inside *A*. Parts of *A* that are smaller than *B* are removed. Thus

$$A \circ B = (A \ominus B) \oplus B.$$



The opening of *A* by the structuring element *B*.