Digital Image Processing Lecture 11 “Various topics”

- Homomorphic filtering
- Gray-scale morphology
- Representation and description, various things in Ch. 11
- Numbers according to Gonzales & Woods, Global Edition, 4th edition. (Numbers in other editions may vary).
- Gonzales & Woods:
  - Chapter 4, pp. 293-296, 328-332
  - Chapter 9, pp. 674-691
  - Chapter 11 pp. 828-830, 840

Maria Magnusson, CVL, Institutionen för Systemteknik, Tekniska Högskolan, Linköpings Universitet

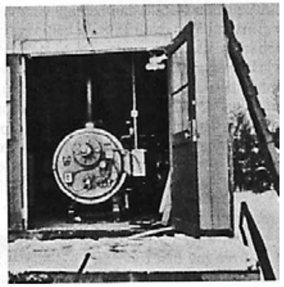
Homomorphic filtering, ex1)

From G&W 2:nd edition, Fig.4.33. G&W 4:th edition, Fig.4.60 contains another example.

- Problem: slow varying illumination in a scene a(x,y)
- Desired: the reflectivity b(x,y)
- The camera delivers f(x,y)=a(x,y)·b(x,y)
- How can a(x,y) be removed?

Before homo-mor-phic filte-ring

$f(x,y) = a(x,y) \cdot b(x,y)$



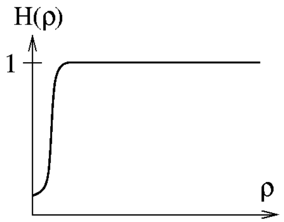
$\approx b(x,y)$

After homo-mor-phic filte-ring

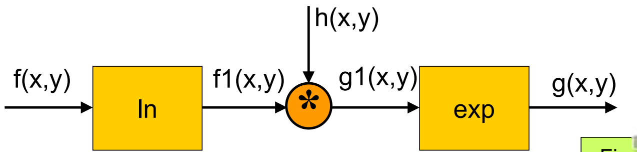
Homomorphic filtering

This filter removes much of the slow varying a (and a1).

- f1(x,y) = ln[f(x,y)]
- g(x,y)=exp[g1(x,y)]
- If h(x,y)=δ(x,y): g(x,y)=f(x,y), i.e. nothing happens
- f(x,y)=a(x,y)b(x,y) gives f1(x,y)=a1(x,y)+ b1(x,y), where b1=ln b and a1=ln a



$\rho = \sqrt{u^2 + v^2}$



≈Fig. 4.58

Homomorphic filtering, ex2)

- A laser gives multiplicative noise (speckle)
- Multiplicative noise (speckle) can be made additive by homomorphic filtering
- The noise can then be suppressed with some kind of filtering.
- Ex) from K. Jain: Fundamentals of Digital Image Processing

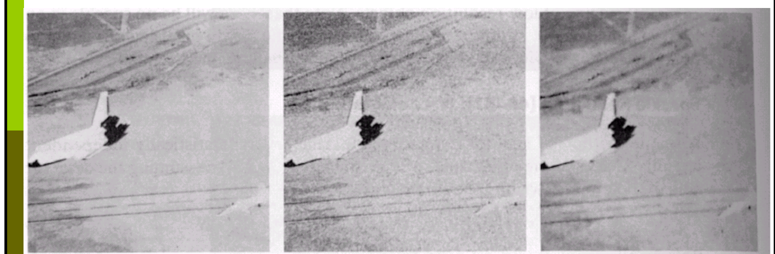


Image without noise, b(x,y) | Image with noise f(x,y)=b(x,y)·n(x,y) | Filtered image g(x,y)

Order-statistics (ranking) filter (basis for gray scale morphology)

- An order-statistic filter orders the values according to size in a neighborhood. The order numbers are $k \in \{1,\ldots,N\}$, where N is the number of pixels in the neighborhood.
- The out-value is set to the in-value number k.
- $k=(N+1)/2$: median filter
- $k=1$: min filter
- $k=N$: max filter

$$\hat{f}(x,y) = \underset{(s,t)\in S_{xy}}{\text{median}}\{g(s,t)\} \quad (5\text{-}27)$$

$$\hat{f}(x,y) = \max_{(s,t)\in S_{xy}}\{g(s,t)\} \quad (5\text{-}28)$$

$$\hat{f}(x,y) = \min_{(s,t)\in S_{xy}}\{g(s,t)\} \quad (5\text{-}29)$$

Gray scale morphology

- The dilation of a flat structuring element b at any location (x,y) is defined as the maximum value in a neighborhood defined by the size of b

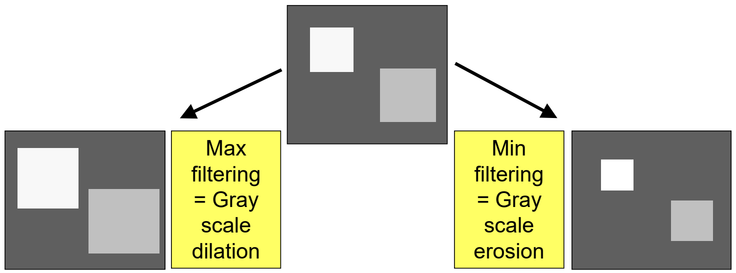
$$[f \oplus b](x,y) = \max_{(s,t)\in b}\{f(x-s, y-t)\} \quad (9\text{-}50)$$

- The erosion of a flat structuring element b at any location (x,y) is defined as the minimum value in a neighborhood defined by the size of b

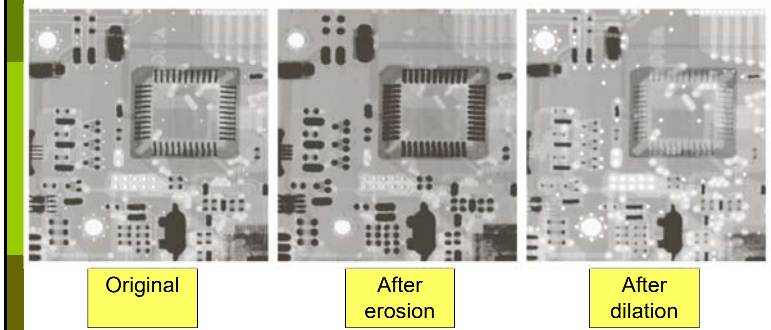
$$[f \ominus b](x,y) = \min_{(s,t)\in b}\{f(x+s, y+t)\} \quad (9\text{-}49)$$

- Note the similarity with the order-statistic max- and min filters!

Max and min filter Gray scale dilation and erosion



Ex) Max and min filter Gray scale dilation and erosion



Small bright dots are reduced, image darker

Small dark dots are reduced, image brighter

Fig. 9.37

Nonflat structuring elements (SE) for grey scale morphology

$$[f \oplus b_N](x,y) = \max_{(s,t)\in b_N} \{f(x-s, y-t) + b_N(s,t)\}$$ (9-52)

$$[f \ominus b_N](x,y) = \min_{(s,t)\in b_N} \{f(x+s, y+t) - b_N(s,t)\}$$ (9-51)

Also called Sternberg's operator or "rolling ball"

Nonflat SE:s are not so common in practice.

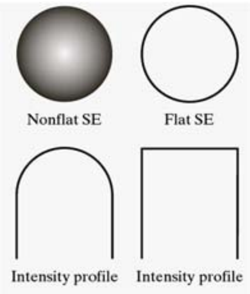
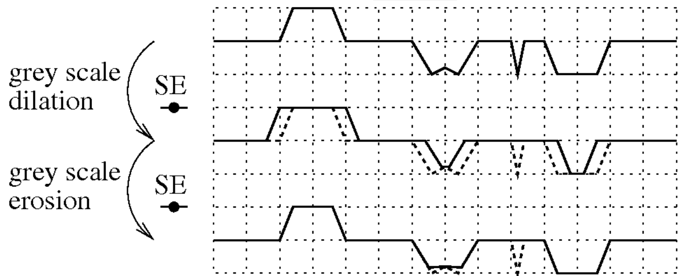


Fig. 9.36

Grey scale closing in 1D = dilation + erosion

$(f \oplus b) \ominus b$ (9-57)



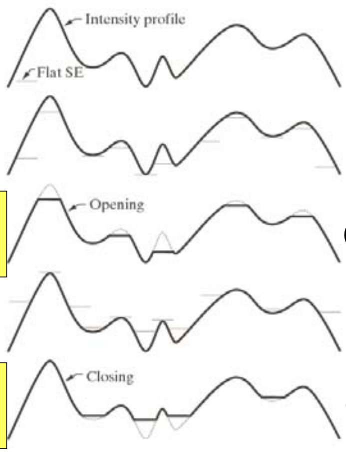
Closing removes small dark details, while leaving the other function values rather undisturbed.

It looks like the SE has moved above the curve

Note that the small local minima disappear

Grey scale opening and closing in 1D

Fig. 9.38



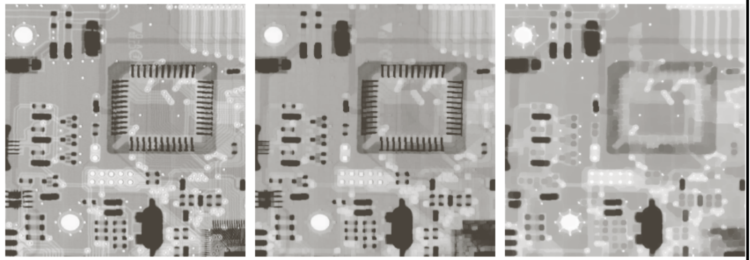
It looks like the SE has moved below the curve

$(f \ominus b) \oplus b$ (9-56)

It looks like the SE has moved above the curve

$(f \oplus b) \ominus b$ (9-57)

Ex) Opening and closing



Original

After opening

Small bright dots are reduced

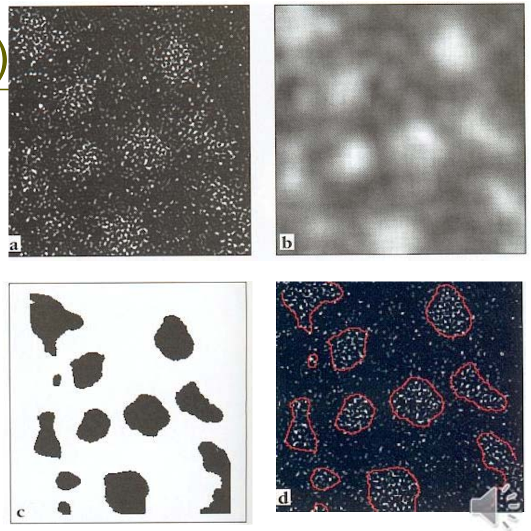
After closing

Small dark dots are reduced

Fig. 9.39

Practical Example)

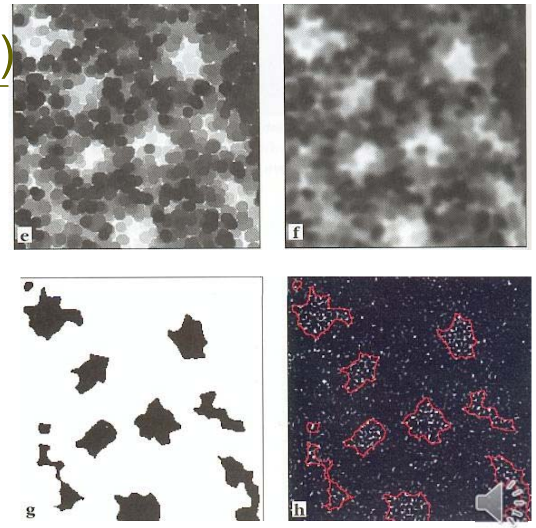
- From Russ: Image Process-ing Handbook, Ch. 3, Fig. 25
- The grains cor-responds to a drug submitted to cells. Task: Delineate cells that incorpora-ted the drug.
- a) Original
- b) LP-filtered
- c) Thresh-holding
- d) Overlaid



Practical Example)

- From Russ: Image Process-ing Handbook, Ch. 3, Fig. 25, cont.
- e) Gray scale opening & closing of a)
- f) LP-filtered
- g) Thresh-holding
- h) Overlaid

Result: Better delineation of cells!



Practical Example)

- a) b)
  c) d)
- a) Original Hubble telescope image
- b-d) Opening and closing
- b) disk radii=1
- c) disk radii=3
- d) disk radii=5

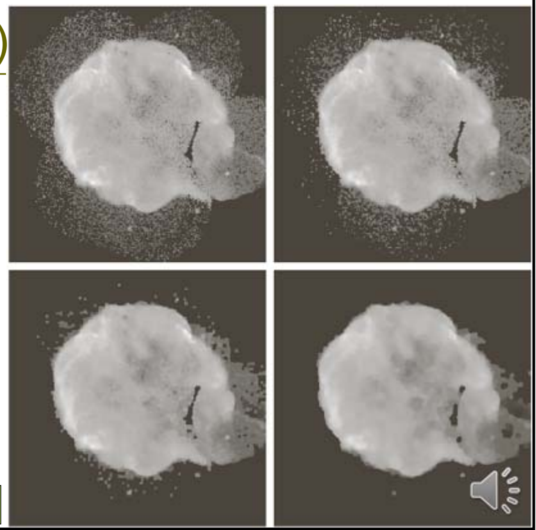


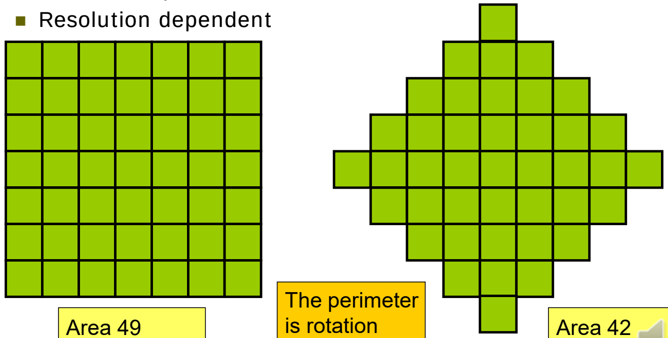
Fig. 9.40

Various things from Chapter 11: Representation and description

- We can represent an object in terms of
  - its external characteristics (its boundary)
  - its internal characteristics (the pixels inside the region)
- Examples of features that we can measure
  - Length of the object and length of the boundary
  - Area
  - Form factor, i.e. “length of the boundary”^2/area
  - Orientation (find the main axes by calculating eigenvectors, lecture 9)

Measuring the perimeter

- Just follow the boundary and add? No!
  - Rotation dependent
  - Resolution dependent



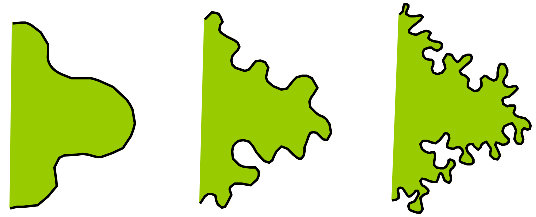
Area 49
Perimeter 28

The perimeter is rotation dependent!

Area 42
Perimeter 36

The boundary is resolution dependent

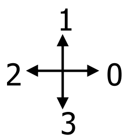
- Real objects have a perimeter that increases depending on the resolution
- Example: Regard the coast line of Sweden

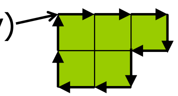


Boundary (border) following along the pixel border

Not in G&W!

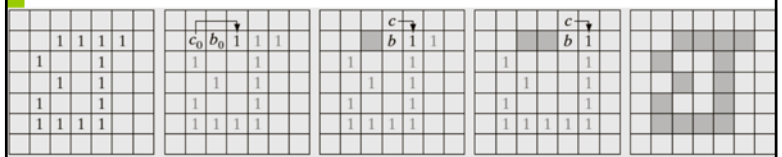
- Take place along the pixel border, clockwise
- Take place in four directions: 0,1,2,3
- Start pixel and every step are denoted => gives chain code
- Was for example received by the RB algorithm in lecture 7
- Example: (x,y)0003232211





Boundary (border) following in-side the pixels along the border

- Similar algorithm as the previous, for details, see the G&W book.



Error here in 3rd edition.
Corrected in 4th edition.

Fig. 11.1

Chain code inside the skeleton pixels, 4-points and 8-points

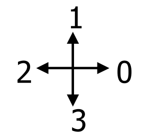
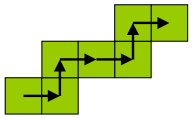
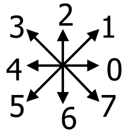
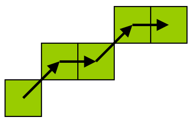


Fig. 11.3



(x,y)010010



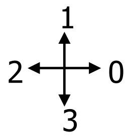


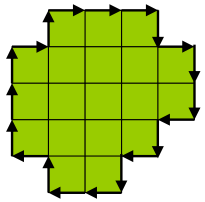
(x,y)1010

Gives a better value of the length.

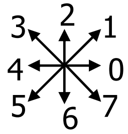
Chain code along the pixel border, 4-points and 8-points

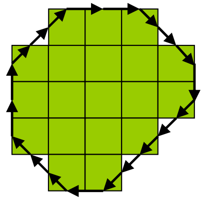
Not in G&W!





(x,y)00030332323
221211101





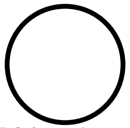
(x,y)00777655555
433322111

Gives a better value of the perimeter.

Form factor

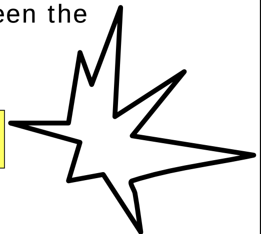
- A form factor measures the compactness of a region, defined as the ratio between the perimeter^2 and the area

Compact:



$\mathrm{P2A} = 1$

Non-compact:



$\mathrm{P2A} = \mathrm{big}$

$\mathrm{P2A} = \frac{P^2}{4\pi A}$ Similar to (11-18)

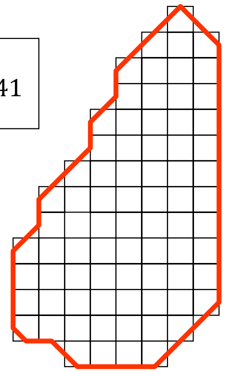
$P = \mathrm{boundary}$

$A = \mathrm{area}$

Example) Form factor P2A

- Note that P2A is resolution dependent for objects like the ”coast line of Sweden”

$$\mathrm{P2A} = \frac{P^2}{4\pi A} = \frac{\left(20 + 24/\sqrt{2}\right)^2}{4\pi \cdot 77} \approx 1.41$$



PhD-thesis 2019 by Bertil Grelsson

- Last year we had a guest lecture by Bertil Grelsson from Saab. His PhD-thesis from 2019 was about vision based navigation in the archipelagia in Sweden.
- He uses many different image processing techniques. Some of them we have talked about in the course, ex)
  - Canny edge detector
  - A variant of the Hough transform
  - Correllation with Gaussian based phase filter.