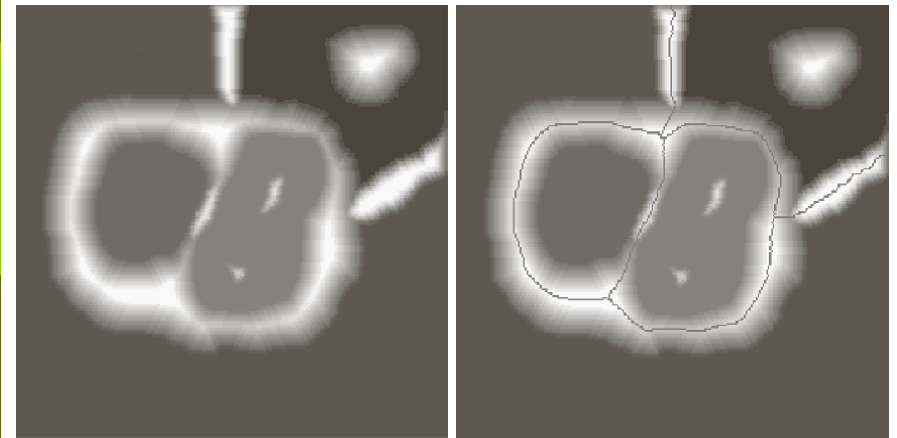# Watersheds

From the book:
Digital Image Processing
by Gonzales & Woods

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

---

Watershed segmentation, introduction

a)   h)

Fig. 10.54

Watershed segmentation, introduction

- Fig. 10.54 a) The watershed algorithm regards a gray scale image as a landscape, where the lakes are preferably dark pixels and the ridges and mountains are preferably bright pixels.
- A watershed = a ridge between individual lakes and rivers.
- A catchments basin = a geographic area whose water flows to a certain lake.
- Fig. 10.54 h) The watershed algorithm can find the watersheds in the image. They form closed curves and the area inside each closed curve, the catchments basin, can be regarded as a segmented, individual object.

Watershed segmentation, description
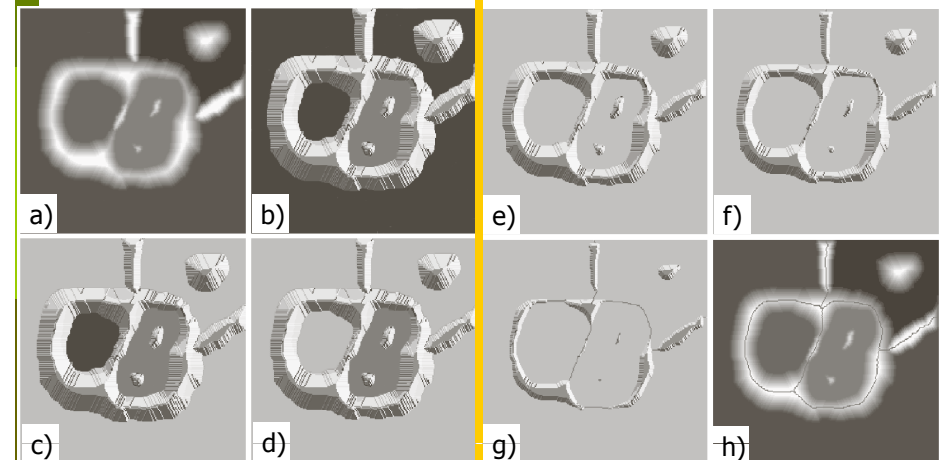

a)   b)   e)   f)
c)   d)   g)   h)

Fig. 10.54

Watershed segmentation, description

- Fig. 10.54 b) is a topographic view of 10.54 a). It is intended to give a 3D-feeling.
- In the beginning, the landscape in a) and b) is totally dry. Local minima are detected and holes are punched in each of them.
- Then the entire landscape is flooded from below by letting water rise through the holes at a uniform rate.
- In c), there is water (light gray) in the background.
- In d), there is also water in the left lake.
- In e), there is water in both lakes.
- In f), there is the first sign on overflow between two catchments basins. Therefore, a short dam has been built between them.
- In g), there are rather long dams.
- Finally, the whole landscape is over flooded with water, separated by high dams, watersheds.
- In h), these watersheds are shown overlaid on the original image.

Watershed segmentation, dam construction

a)

$d^{(8)} =$

☐ 1:st dilation
■ 2:nd dilation
⊠ Dam points

b)

c)

3:rd dilation   Fig. 10.55
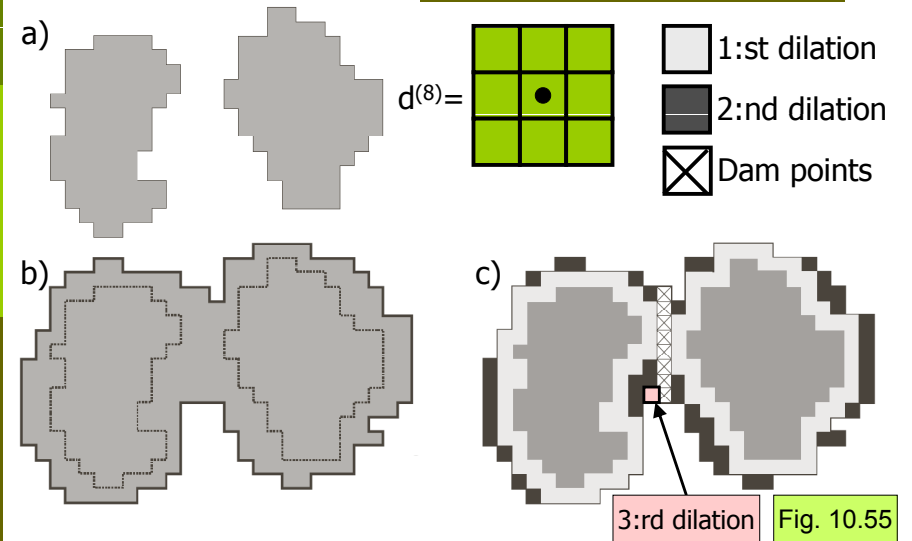
Watershed segmentation, dam construction

- Let $C_{n-1}(M_1)$ denote the pixels in lake M_1 at flooding step n-1. These are shown to the left in a).
- Let $C_{n-1}(M2)$ denote the pixels in lake M_2 at flooding step n-1. These are shown to the right in a).
- Flooding step n is shown in b).
- Let C(n-1) denote the union of $C_{n-1}(M_1)$ and $C_{n-1}(M_2)$. There are 2 objects in C(n-1) in a) and 1 object i C(n) in b). This indicates that something have happened in step n and that some action must be performed.
- Dilate the pixels in a) with the $d^{(8)}$-structuring element considering the following constraints:
  - Dilation is not allowed outside b).
  - Dilation is not allowed so that the pixels in M_1 and M_2 are united.
  - Instead, if the pixels in M_1 and M_2 are going to be united, denote: dam pixel.
- The first dilation in c) gives the light gray pixels.
- The second dilation in c) gives the dark gray pixels and the X-marked dam pixels.

Watershed segmentation, Ex 1)


a)

b)

c)

d)

Fig. 10.56
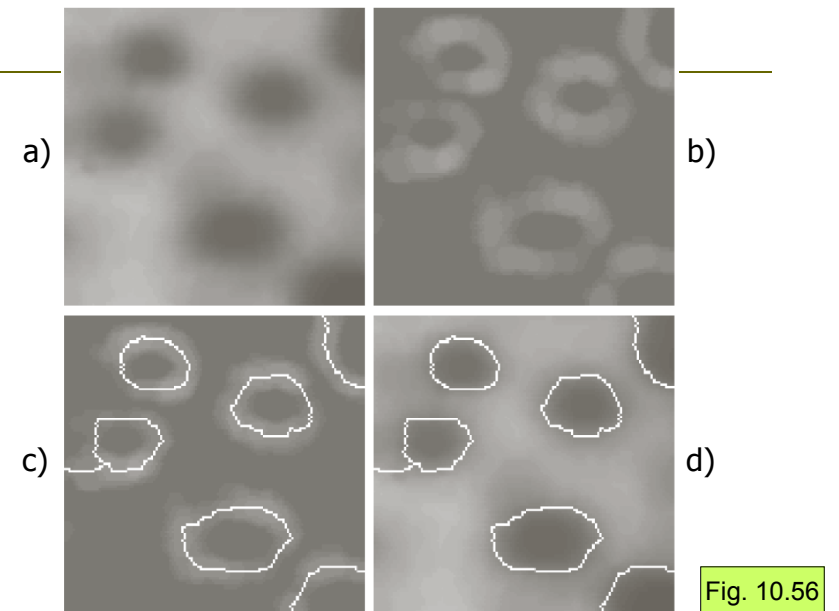
Watershed segmentation, Ex 1)

- a) Original image with dark objects that are going to be segmented.
- b) A usual pre-processing step is to calculate the magnitude of the gradient.
- c) The result of the watershed algorithm overlaid on the gradient image.
- d) The result of the watershed algorithm overlaid on the original image.

Watershed segmentation, Ex 2) for overlapping binary objects

- The watershed algorithm can also be used to segment overlapping binary objects.
- Distance mapping is used to construct a landscape with lakes.
- Computation: See this and the following 2 slides.
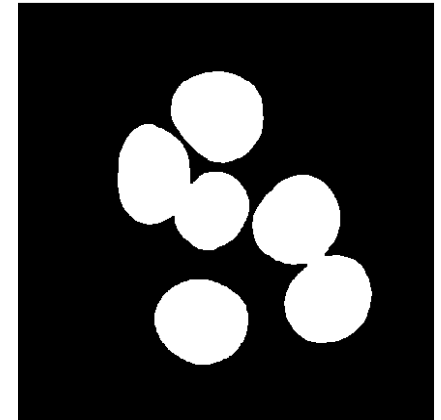
Binary input image

Ex2) Generate a distance map of the objects and "invert" it

Multiply with -1
Background=>-100

Distance map   "Inverted" distance map
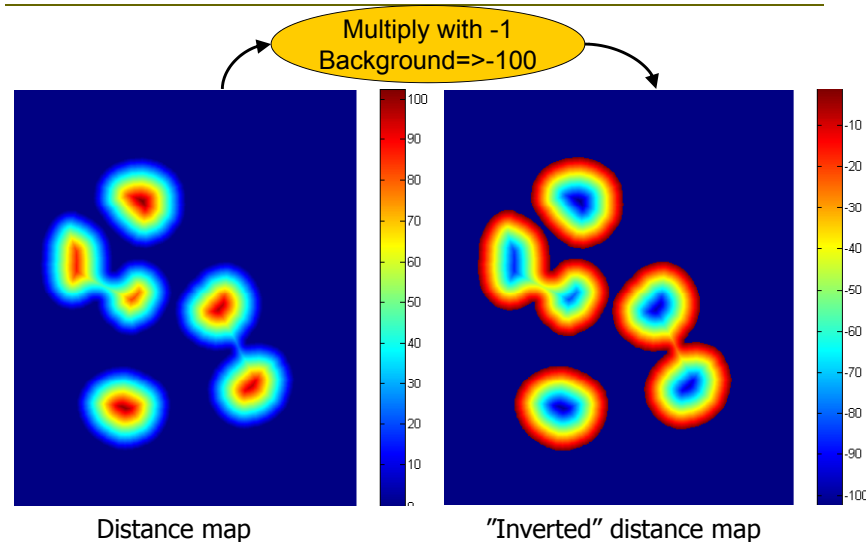
Ex2) Detect local min & apply the watershed algorithm on the "inverted" distance map

Labeled local min   Result of the watershed algorithm