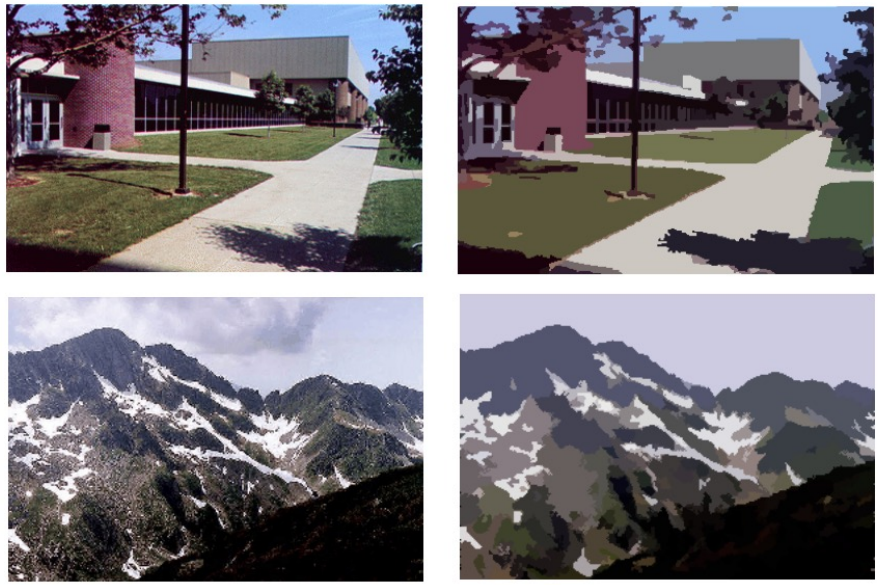
Segmentation – other* methods

Three points from the topic:

1. Segmentation by region growing
2. Segmentation by watershed
3. Segmetnation by mean-shift



Monday: split-and-merge

* - other than thresholding

Region growing (agglomerative clustering)

- Start from seed pixel(s)
- Compare neighborhood pixels (4-NB or 8-NB etc) to feature from region.
  - *Feature* can be mean of intensitiy values from pixels in region etc.
- Include if NB *pixel feature* is *”close to” feature* from region.
  - *”Close to”* can be a distance function and a threshold.
- Continue until no new NB pixels satisfies criteria.

Region growing – implementation with boundary stack

- Inputs: Image I(x,y), sixe MXN, Seed = (xs,ys),
- Initialize: L=zeros(M,N), L(xs,ys)=1, stack.push(xs,ys), f(region) = f(xs,ys)

Keep track of pixels at boundary of region (for example in a stack)

While (stack is not empty) do:

- Look at NB to a boundary pixel popped from the stack: (xn,yn)
- Is feature vector f(xn,yn) similar to f(region)?
- If yes-> pixel I(xn,yn) is added to the region, L(xn,yn)=1, (xn,yn) is pushed to the boundary stack, and f(region) is updated.

Region growing, gaussian model

- Let the model be gaussian, the feature is intensity value, but normalized with the st.dev :

Region growing – more on feature choices

- Feature can be intensity value and mean, but van also include different frequency bands, variance over a local neighbourhood, gradient values, different color bands etc. -> feature vector **f**.
- The feature vecort of the region can be found as the mean of each of the features in the vector, over the region **fm**,.
- Similarity can be measured as inner product:

torsdag 31. oktober 2019 12:13

Example of region growing

O = seed

final region

0	1	0	0	1	1	0	2	1
1	1	0	0	3	4	3	1	0
0	2	3	4	5	4	3	0	1
0	0	0	4	5	4	4	1	1
1	0	1	3	3	2	0	0	2
0	1	0	2	0	1	1	2	5
1	0	0	1	0	0	1	3	4

$\bar{f}_0 = 4$

Let the intensitylevel be our feature $f(\underline{q})$ = intensity

similarity: $d = | f(\underline{q}) - \bar{f} |$

$\underline{q}$: position coord.

$\bar{f}$: mean over region

include q in region

if $d \leq T$, $T=2$

Use 4NB 

here: Go through entire boundary before update $\bar{f}$

torsdag 31. oktober 2019 12:13

Example of region growing

○ = seed

final region

0	1	0	0	1	1	0	2	1
1	1	0	0	3	4	3	1	0
0	2	3	4	5	4	3	0	1
0	0	0	4	5	4	4	1	1
1	0	1	3	3	2	0	0	2
0	1	0	2	0	1	1	2	5
1	0	0	1	0	0	1	3	4

$\bar{f}_0 = 4$

Let the intensitylevel be our feature $f(q)$ = intensity

similarity: $d = |f(q) - \bar{f}|$

q : position coord.

$\bar{f}$: mean over region

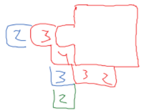
include q in region if $d \leq T$, $T = 2$

Use 4NB 

here: Go through entire boundary before update $\bar{f}$

$\bar{f}_0 = 4$ ⇒ include all $\bar{f}_1 = \dfrac{4+4+4+5+3}{5} = \dfrac{20}{5} = 4$

$\bar{f}_1 = 4$ ⇒ $\bar{f}_2 = \dfrac{20 + \overbrace{3+3+5+4+4}^{\text{new}}}{5+5} = \dfrac{39}{10} = 3.9$

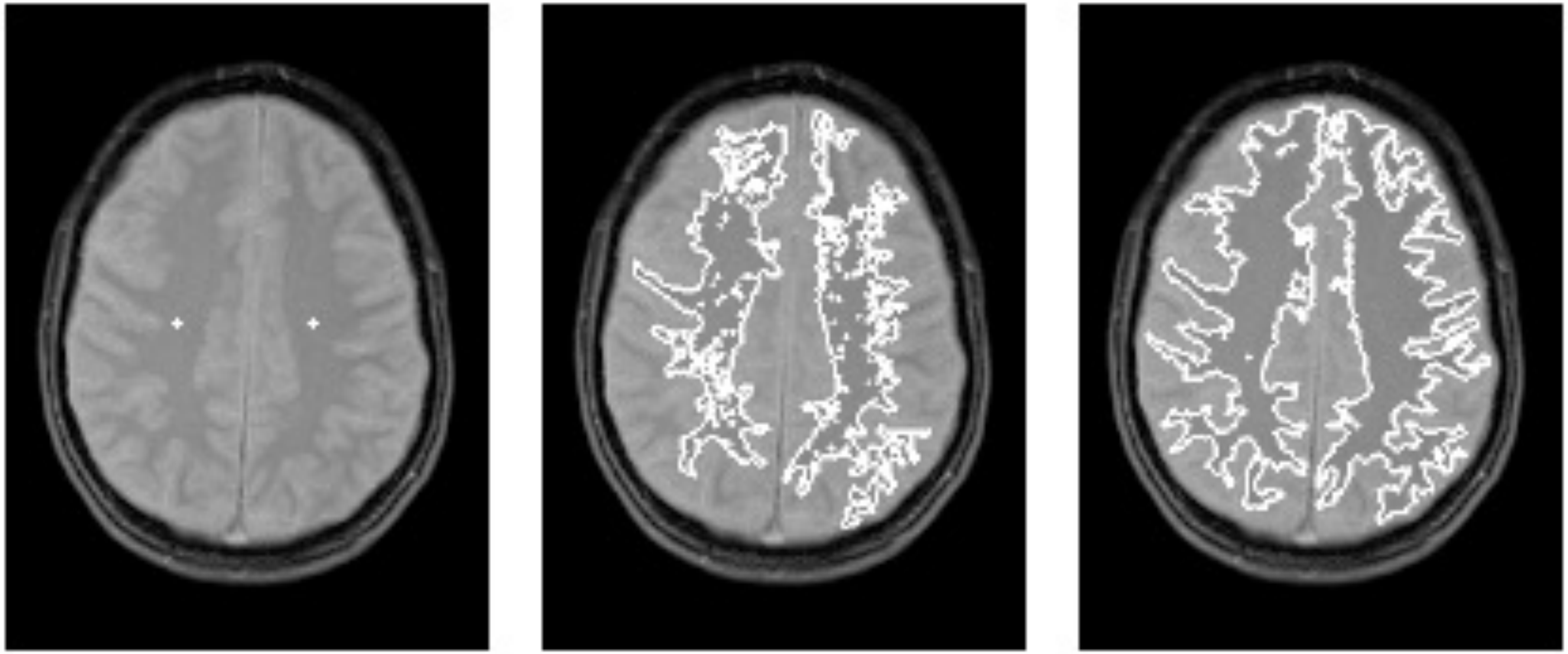


⇒ $\bar{f}_3 = \dfrac{39+3+4+3+2}{10+4} = \dfrac{51}{14} = 3{,}64$

$\bar{f}_4 = \dfrac{51+2+3}{14+2} = \dfrac{56}{16} = 3.5$

$\bar{f}_5 = \dfrac{56+2}{16+1} = 3.41$

Region growing



Distance transform – recap from morphology

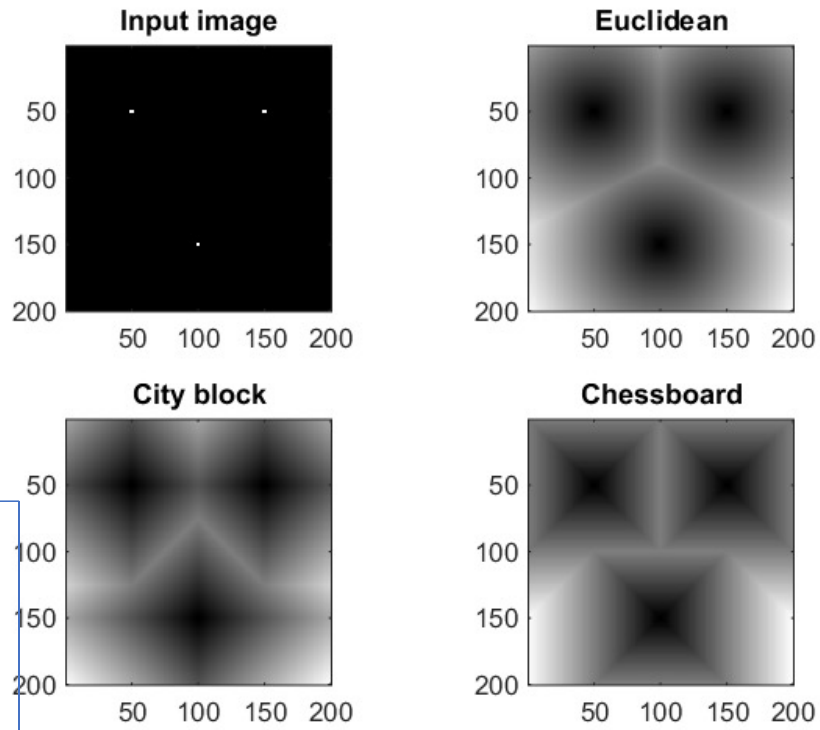
For each pixel in the binary image find the distance to the nearest non-zero pixel (or any give feature).

Let two points be defined: $(x_1, y_1), (x_2, y_2)$

Euclidean distance: $$D_e = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

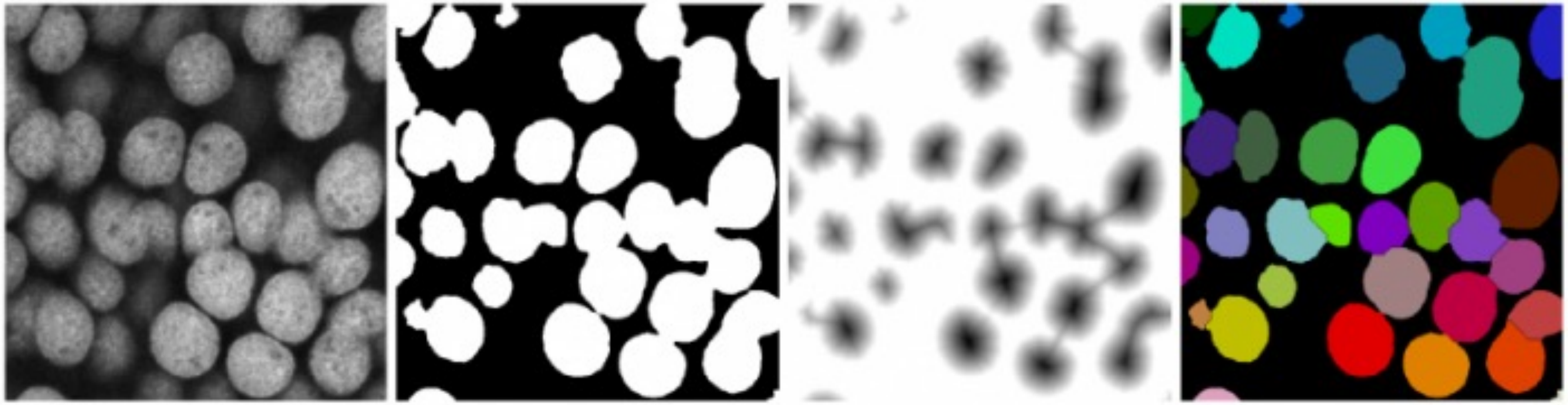
Cityblock distance: $$D_{cb} = |x_1 - x_2| + |y_1 - y_2|$$

Chessboard distance: $$D_{chess} = \max(|x_1 - x_2|, |y_1 - y_2|)$$

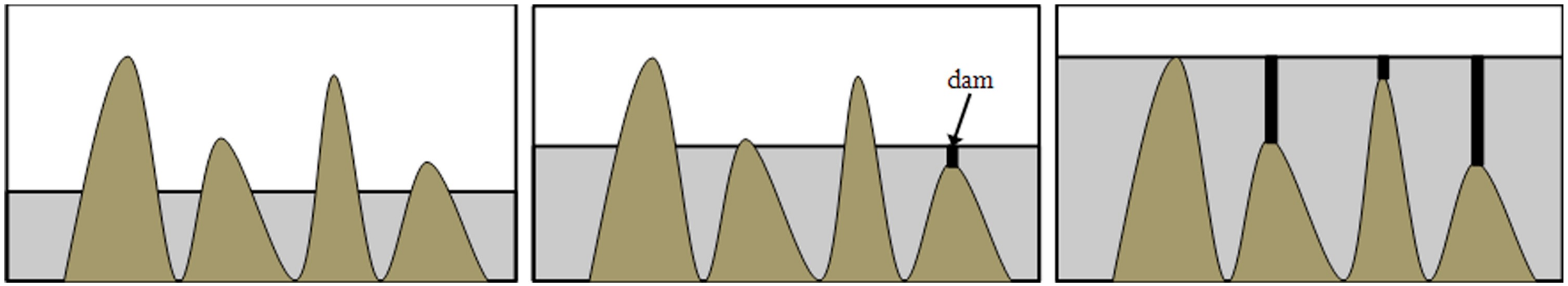


Distance transform

The distance transform can be used for example for skeletonization, or as input for watershed segmentation



Watershed- idea



(a) (b) (c)

Can not do watershed directly on image, have to find *gradient, or distance transform* of image first. Looks at the preprocessed image (gradient or distance) as a surface.

1) Find gradient or distance image
2) Find valleys (local minimas)
3) ”pinch holes” in the local minimas and «sink surface into water», i.e. flood the valleys.
4) Put up a ”watershed” (dam) when a basin wants to join another basin.

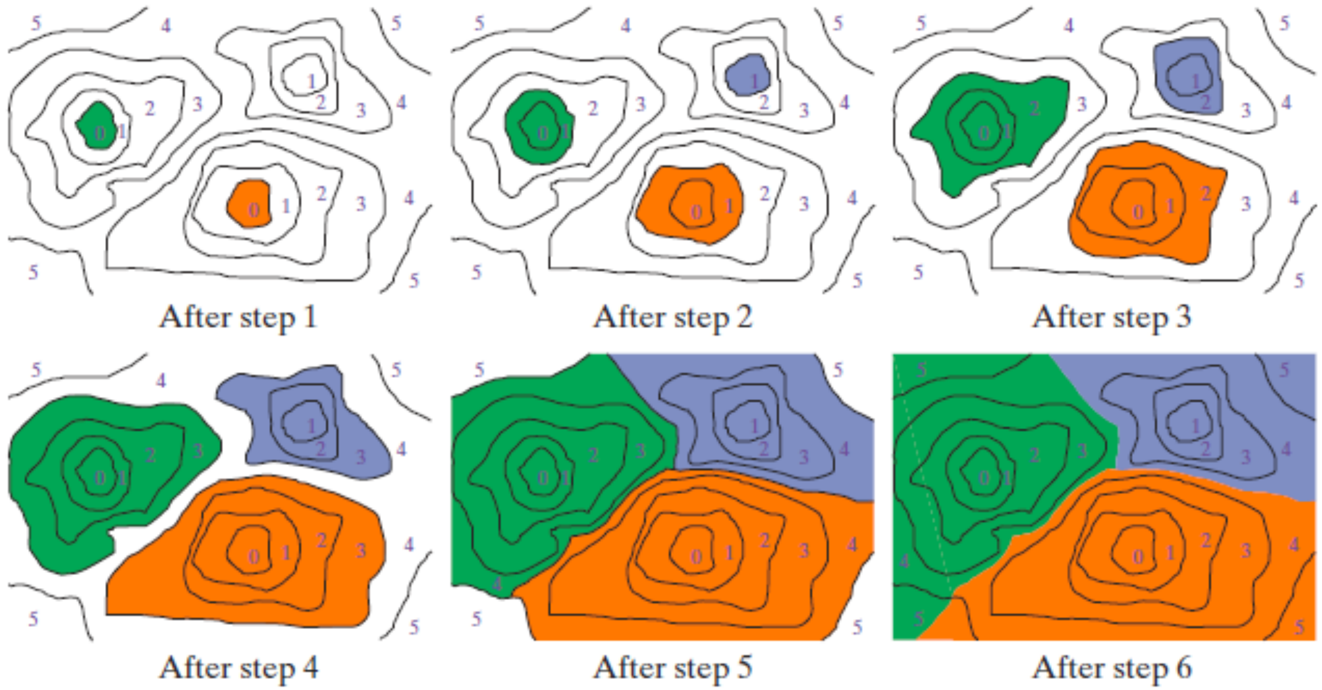
Can use markers: only set up sheds if basins from different markers tries to merge.

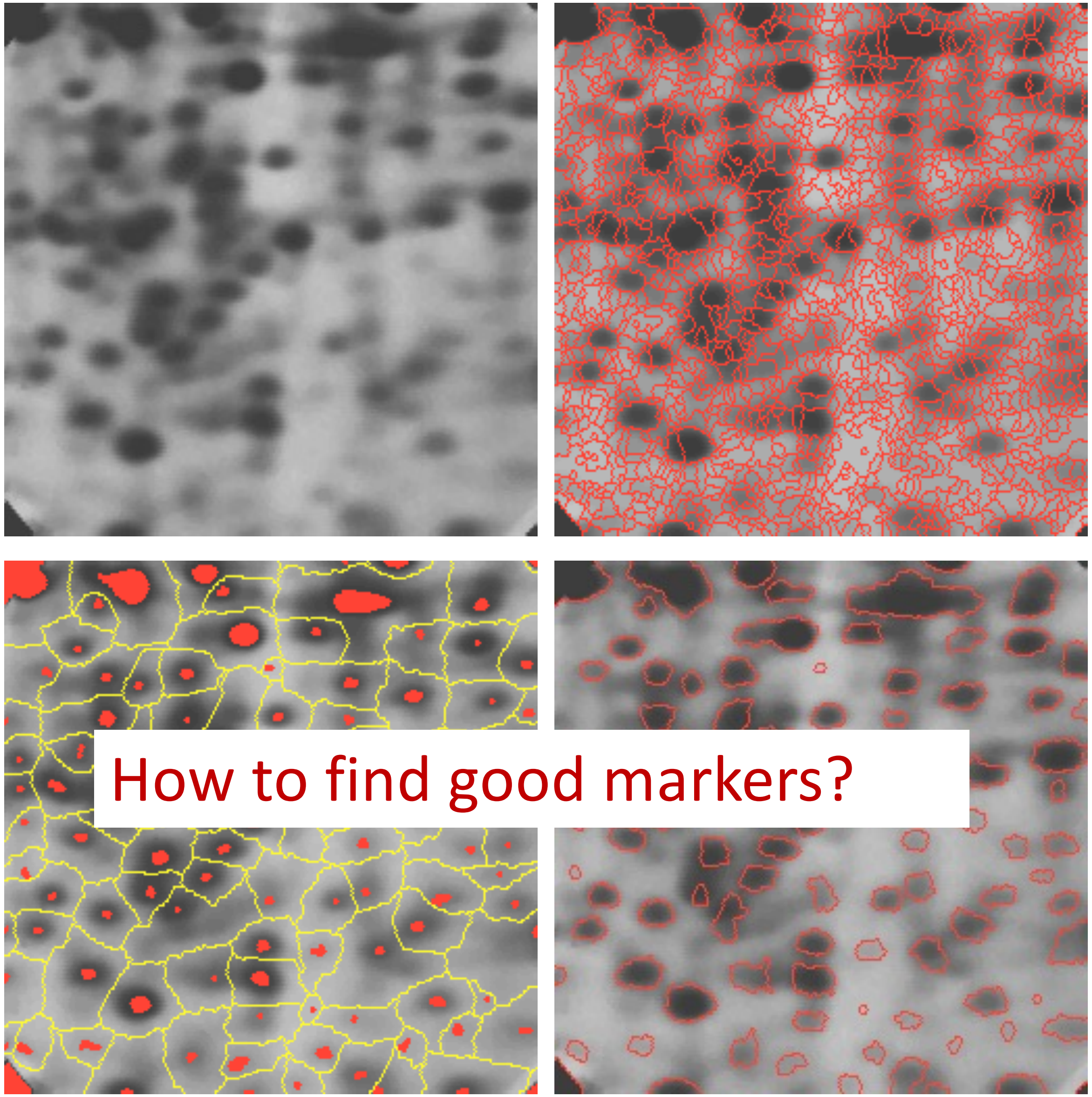
Geodesic influence zone: At each level this must be determined for each existing catchment basin. It is the set of unlabeled pixels connected to the basin and closer to that basin than any other.

The regions are growed in a breath first manner to ensure regions are met halfway when necessary .

Watershed leads to oversegmentation. Common solution is **marker-based** watershed segmentation.

Figure 10.28 Step-by-step results of immersion-based watershed segmentation on a segmentation function with 6 levels (0 through 5). The different colors indicate the unique labels of the three different regions. The contour lines of the segmentation function are shown, with numbers indicating the levels of the pixels.





How to find good markers?

Watershed can produce high degree of oversegmentation

Marker controlled Watershed can improve this. Only when basins from different markers are flooded, a watershed is made.

Top left: original image

Top right: watershed of gradient image, oversegmented.

Bottom left: red is blob marker, yellow is background marker.

Bottom right: Result after marker controlled watershed.

cmm.ensmp.fr/~beucher/wtshed.html

One example solution

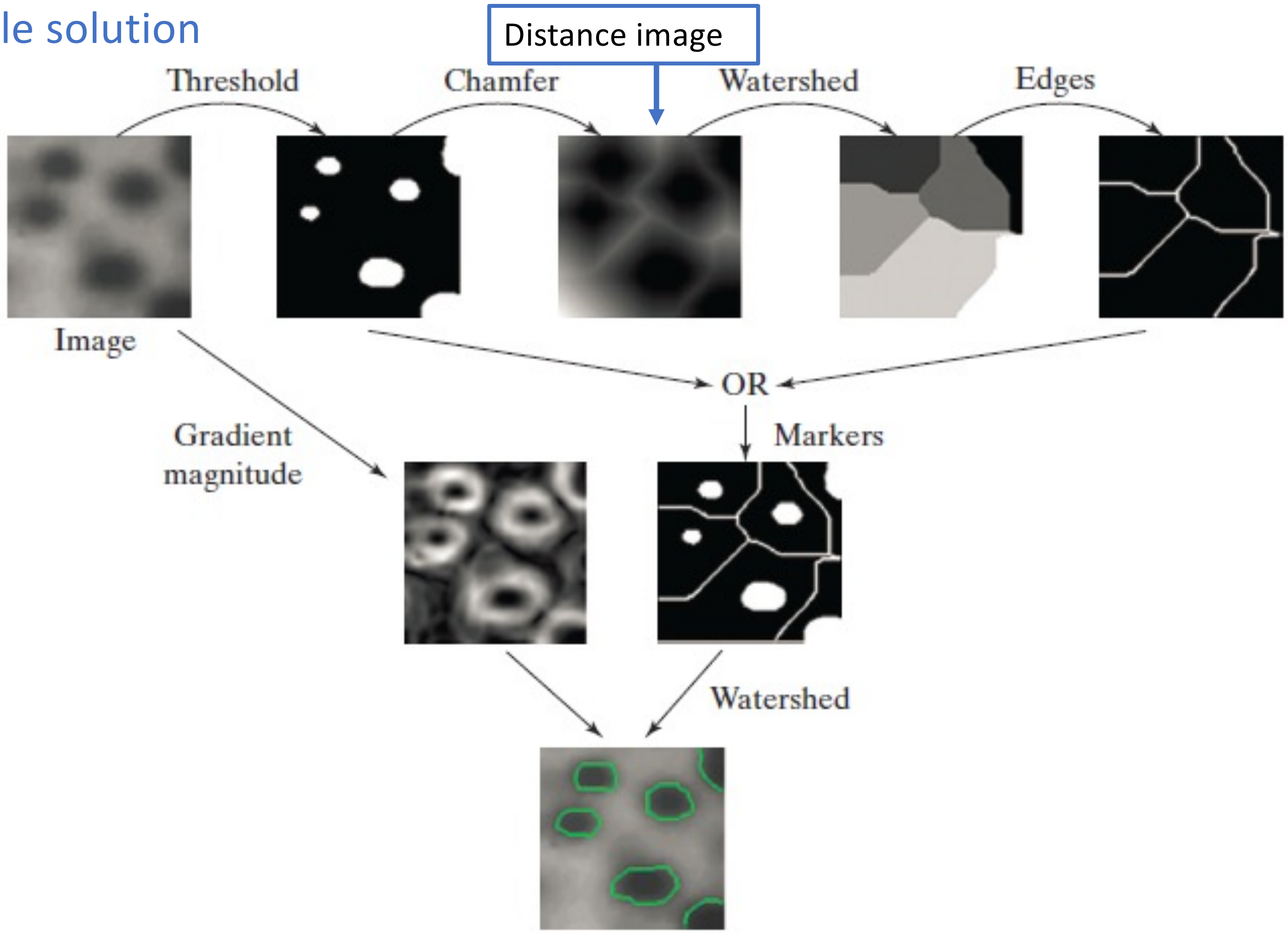


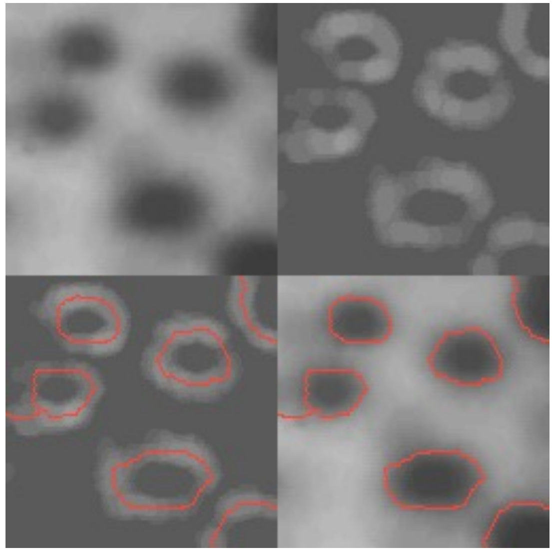
Figure 10.30 Flowchart of the end-to-end watershed procedure.

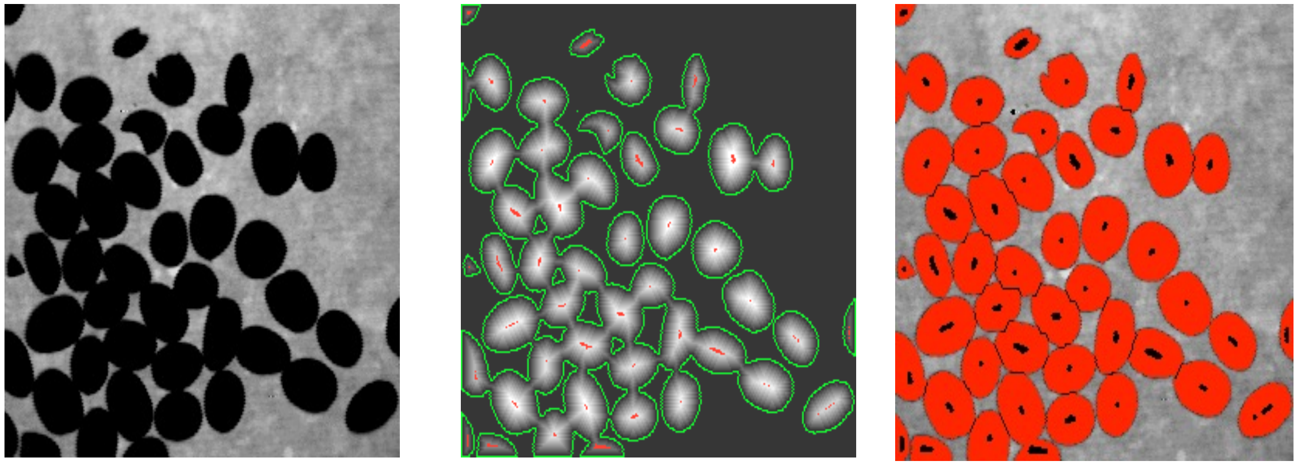
Watershed example

Watersehd from gradient image:

Original image, gradient image

Watershed of the gradient image, Final contours.





Watershed from distance image (with markers)

Left to right: Original image, Distance function, segmentation

cmm.ensmp.fr/~beucher/wtshed.html

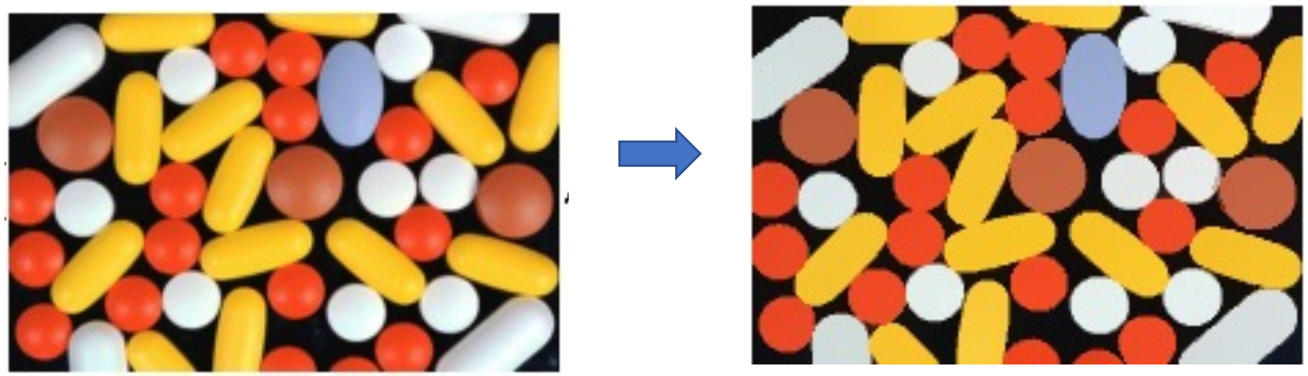
Mean-shift segmentation

- **Mean-shift segmentation algorithm**:
- involves first running the mean-shift filter, then applying some post processing steps to the result.

- Mean-shift clustering is a non-parametric, density-based clustering algorithm that can be used to identify clusters in a dataset.
- Particularly useful for datasets where the clusters have arbitrary shapes and are not well-separated by linear boundaries.
- mean-shift does not require specifying the number of clusters in advance.

Recap - from non-linear filters

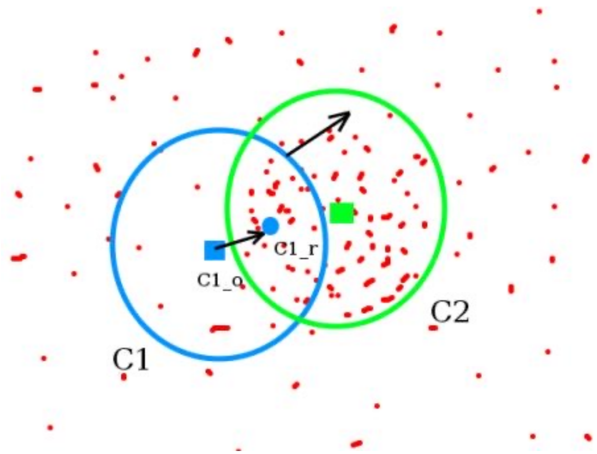
- Mean-shift filter
  - Clustering both in color values and in distance. Group pixels near in color values AND distance, and recalcualte mean according to mean-shift. Iteratively replaces each pixel with the mean of the pixels in a range-r neighborhood and whose value is within a distance d until mean-shift =0.



Mean Shift clustering

The mean shift algorithm seeks modes
of the given set of points

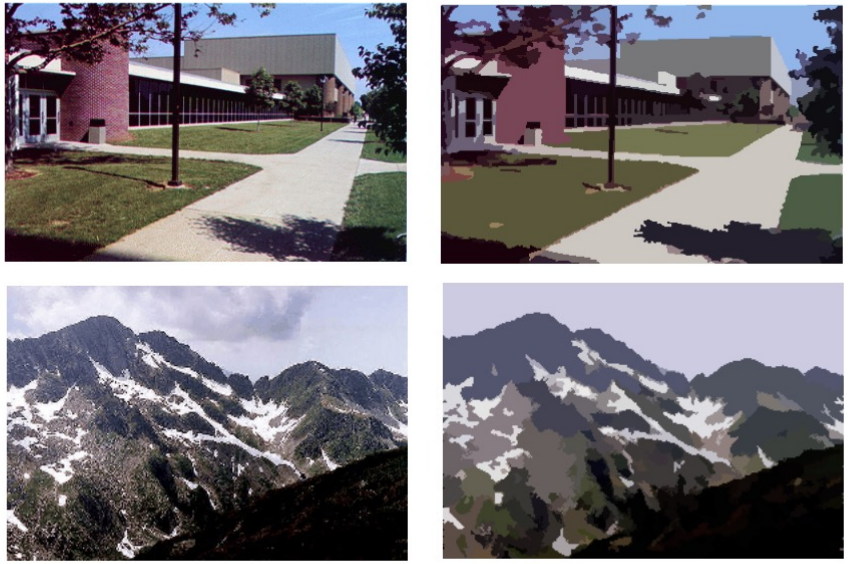
1. For each point:
a) Center a window on that point
b) Compute the mean of the data in the search window
c) Center the search window at the new mean location
d) Repeat (b,c) until convergence
1. Assign points that lead to nearby modes to the same cluster



Original window C1, center c1_o, but centroid is c1_r. -> move region with the "mean-shift" so that center of new region is at c1_r (C2).

Used for clustering, tracking, segmentation ..

Mean-shift filter



- Mean shift can be seen as locating the maxima –

the modes - of a density function given discrete data that function

- Let **x** = (x,y,v) =(x,y, I(x,y)) for each pixel

- Let a kernel function, K(**x**i-**x**) be given. This function determines the weight of nearby points for re-estimation of the mean. Typically a Gaussian kernel on the distance to the current estimate is used

- The weighted mean of the density in the window, determined by K in a neighborhood around the pixel , is found as m(**x**). The difference (m(**x**) –**x**) is called the mean shift . Now **x** is set equal to m(**x**), and the process is repeated until convergence.

$$m(\mathbf{x}) = \frac{\sum_{\mathbf{x}_i \in N} K(\mathbf{x}_i - \mathbf{x})\mathbf{x}_i}{\sum_{\mathbf{x}_i \in N} K(\mathbf{x}_i - \mathbf{x})}$$

- Output image: value at «start positon» (x´,y´) is set to the «end value» in m(x).

Mean-shift segmentation

- Postprocessing: Any two pixels that are nearby in both spatial coordinates and range are merged, and finally small regions are merged with nearby larger regions.

- Good results – computationally expensive.
- Links Segmentation to edge-preserving smoothing!   Any edge-preserving smoothing can be done instead.

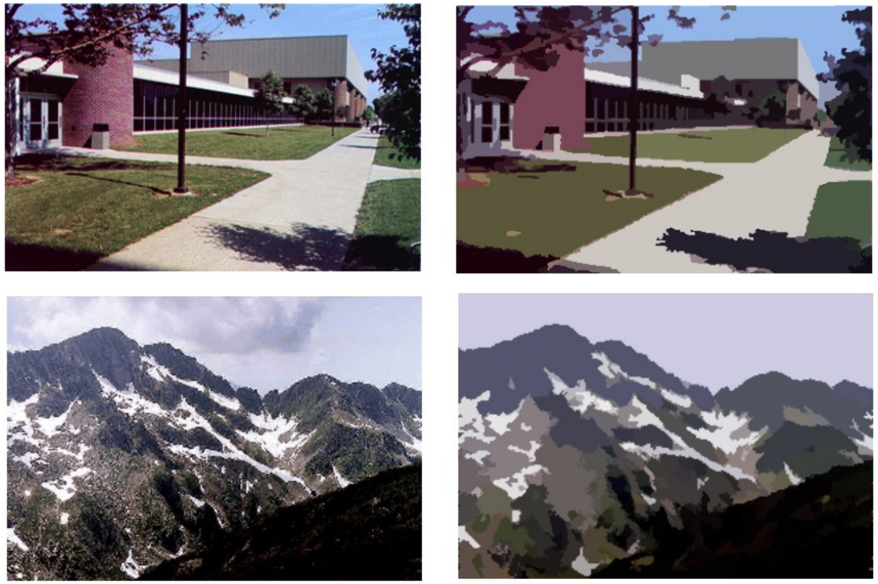
Figure 5.22 An image (left) and the cartoon-like result of mean-shift filtering (right), using $h_s = 32$ and $h_r = 16$.

Figure 10.31 Result of the mean-shift segmentation algorithm on an image of a clown.

(First run mean-shift filter, then apply postprocessing steps.)

Segmentation – other* methods

Three points from the topic:



1. Segmentation by region growing
2. Segmentation by watershed
3. Segmetnation by mean-shift

* - other than thresholding