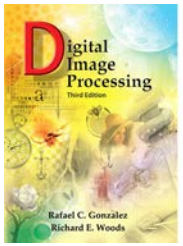
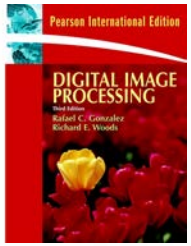
Lecture 6 – Segmentation

Wednesday, November 15

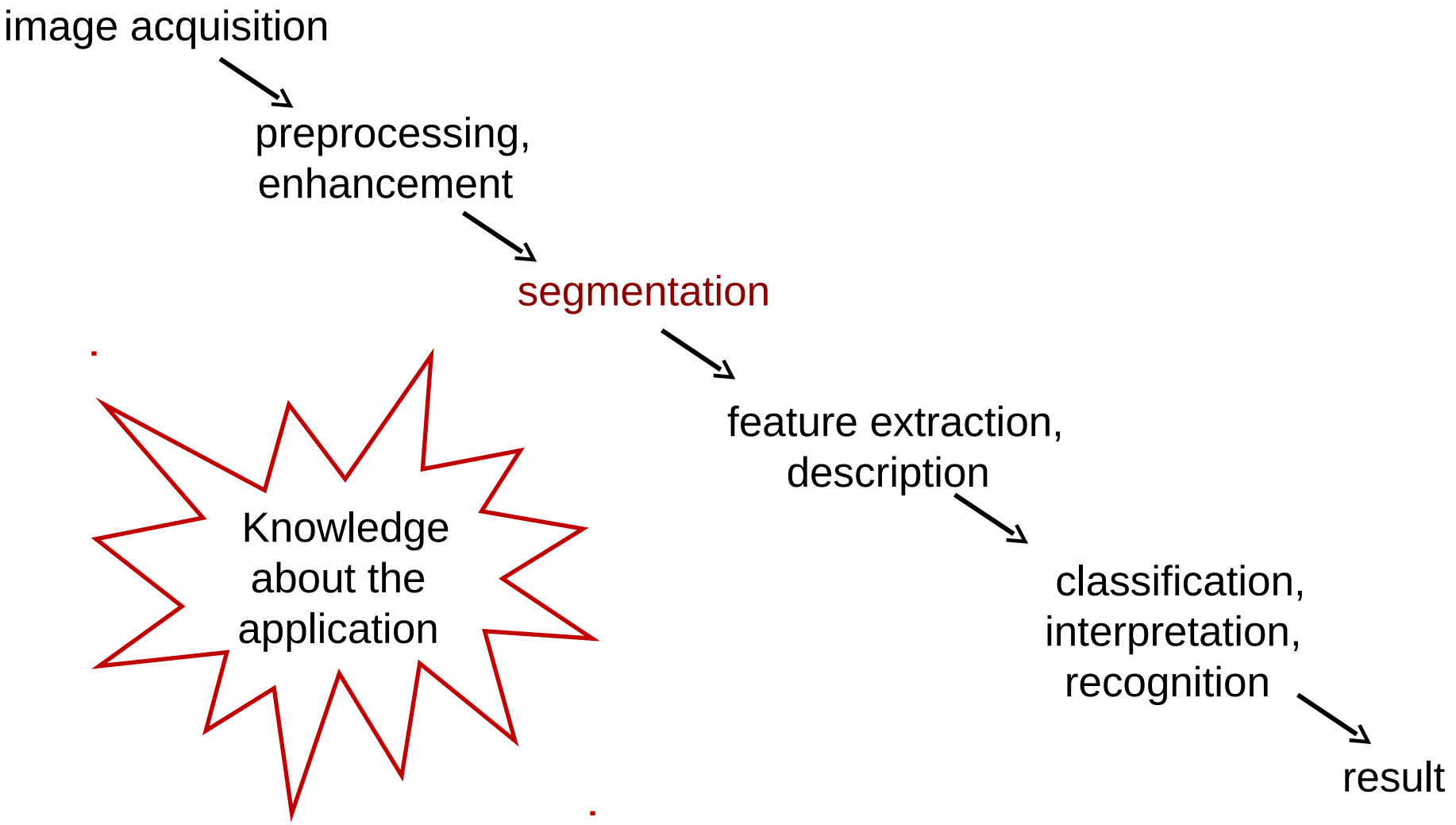
Ch. **10.1-10.2.5 and**
10.3-10.5
Gonzales & Woods





Damian Matuszewski
damian.matuszewski@it.uu.se
Centre for Image analysis
Uppsala University

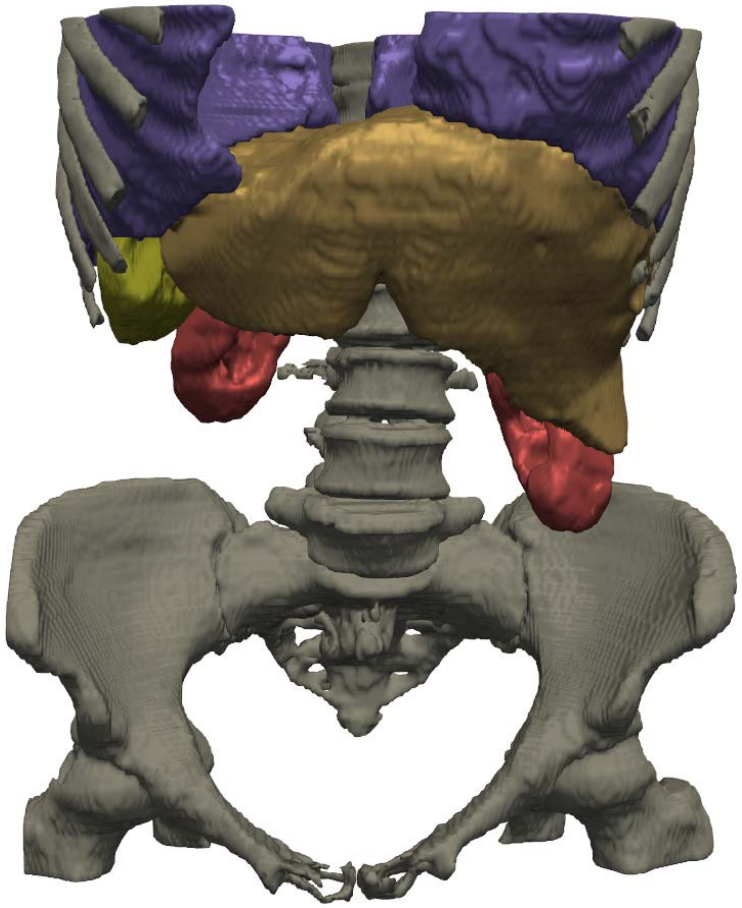
Problem solving using image analysis: fundamental steps



Today

- What is image segmentation?
- An overview of methods for image segmentation:
  - Thresholding
  - Edge-based segmentation
  - Hough transform
  - Region-based segmentation
  - Watershed
  - Match-based segmentation

What is segmentation?



Dividing the image into different regions.

Separating objects from background and giving them individual ID numbers (labels).

Wikipedia on segmentation

-”In computer vision, Segmentation is the process of partitioning a digital image into multiple segments”

-”More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain visual characteristics.”

-”Each of the pixels in a region are similar with respect to some characteristic or computed property, such as color, intensity, or texture. Adjacent regions are significantly different with respect to the same characteristic(s).”

Why segmentation?

Accurate segmentation of objects of interest in an image greatly facilitates further analysis of these objects. For example, it allows us to:

- Count the number of objects of a certain type.
- Measure geometric properties (e.g., area, perimeter) of objects in the image.
- Study properties of an individual object (intensity, texture, etc.)
- ...

Segmentation

Segmentation is often the most difficult problem to solve in image analysis.

There is no universal solution!

Targeted Segmentation

Segmentation is an ill-posed problem...

What is a correct segmentation of this image?

Targeted Segmentation

...unless we specify a segmentation target.

"Segment the orange car from the background"

"Segment all road signs from the background"

Targeted Segmentation

A segmentation can also be defined as a mapping from the set of pixels to some application dependent target set, e.g.

- {Object, Background}
- {Humans, Other objects}
- {1,2,3,4,...}
- {Healthy tissue, Tumors}

To perform accurate segmentation, we (or our algorithms) need to somehow know how to differentiate between different elements of the target set.

Segmentation

Segmentation algorithms are often based on one of the following two basic properties of intensity values:

Similarity
Partitioning an image into regions that are similar according to a set of predefines criteria.

Discontinuity
Detecting boundaries of regions based on local discontinuity in intensity.

Five types of segmentation algorithms

Thresholding - *Similarity*
Based on pixel intensities (shape of histogram is often used for automation).

Edge-based - *Discontinuity*
Detecting edges that separate regions from each other.

Region-based - *Similarity*
Grouping similar pixels (with e.g. region growing or merge & split).

Watershed segmentation - *Discontinuity*
Find regions corresponding to local minima in intensity.

Match-based - *Similarity*
Comparison to a given template.

Thresholding

Thresholding

Which pixels belong to the object?

A **threshold** ***T***, a gray level intensity, classifies every pixel as belonging to objects (foreground) or background. (Or rather, {dark objects, bright objects}).

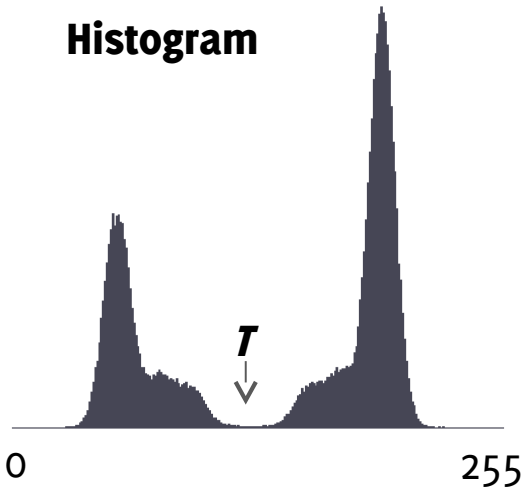
Thesholding

Global threshold
The same value is used for the whole image.

Optimal global threshold
Based on the shape of the current image histogram. Searching for valleys, Gaussian distribution etc.

Local (or dynamic) threshold
The image is divided into non-overlapping sections, which are thresholded one by one.

Global thesholding



We chose a **threshold *T*** midway between the two gray value distributions.

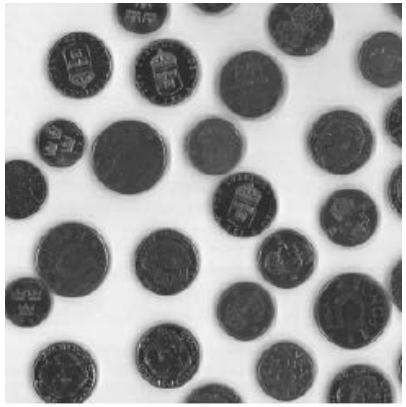
Here: In the tresholded binary image, pixel values below *T* belong to the object (black), pixels above *T* are background (white).

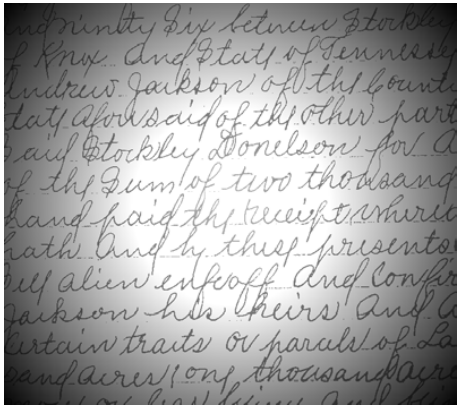
How to find a global threshold

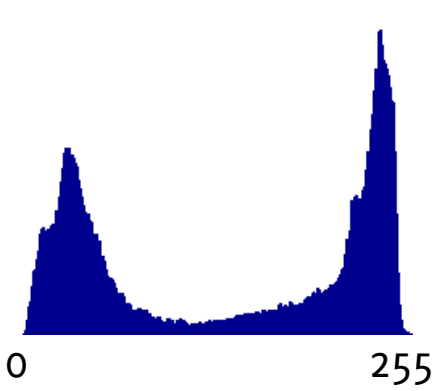
Example method

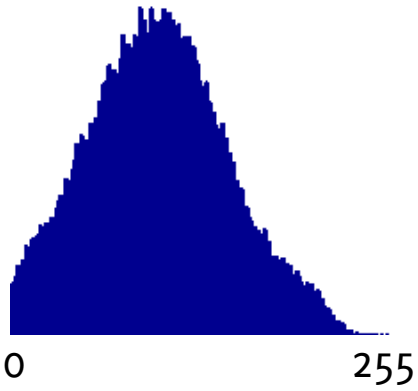
1. Choose initial threshold T_0
2. Define $f(x,y) > T_0$ as background and $f(x,y) < T_0$ as foreground
3. Calculate mean for background μ_{bg} and foreground μ_{fg}
4. Set next threshold $T_i = (\mu_{bg} + \mu_{fg})/2$
5. Repeat 2.-4. until stopping criteria, $T_i = T_i\text{-}1$, is fulfilled

When does intensity based thresholding work?



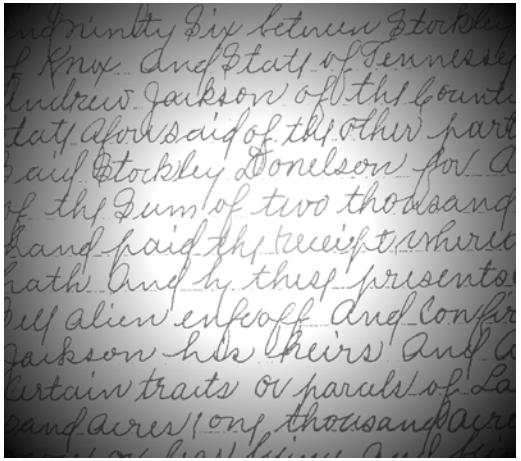


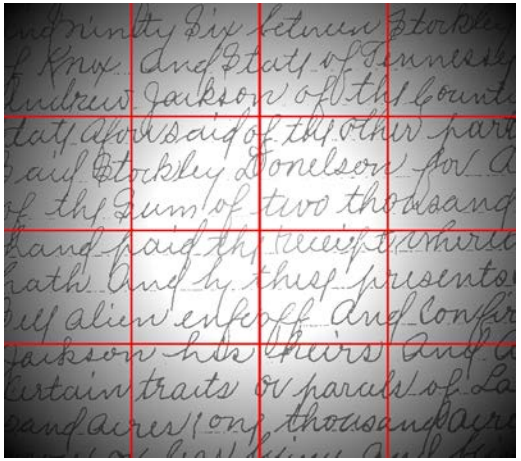


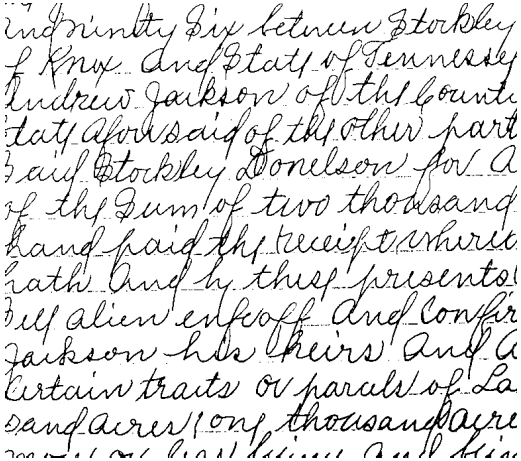


What happens when the target object is not present?

Local thresholding







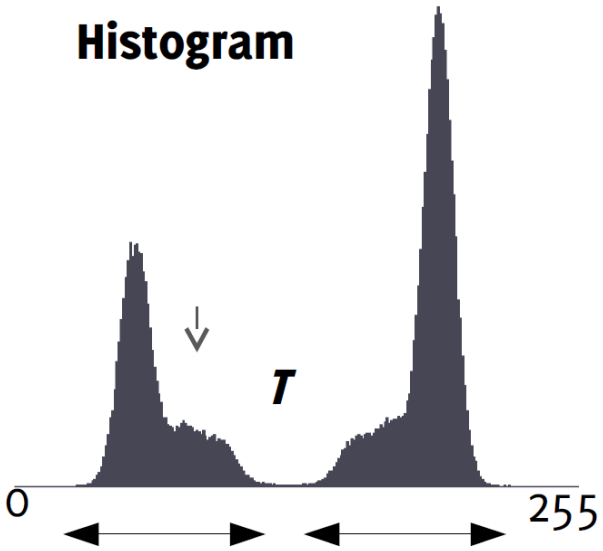
Example method

Subdivide image into non-overlapping rectangles. These rectangles are chosen small enough so that the illumination of each is approximately uniform. Then determine a global threshold for each subimage.

Thresholding in matlab

```
> im = imread('coins.png');
> T = graythresh(im); % Gives value in [0,1]
> BW = im>(T*max(im(:)));
> imagesc(BW)
```

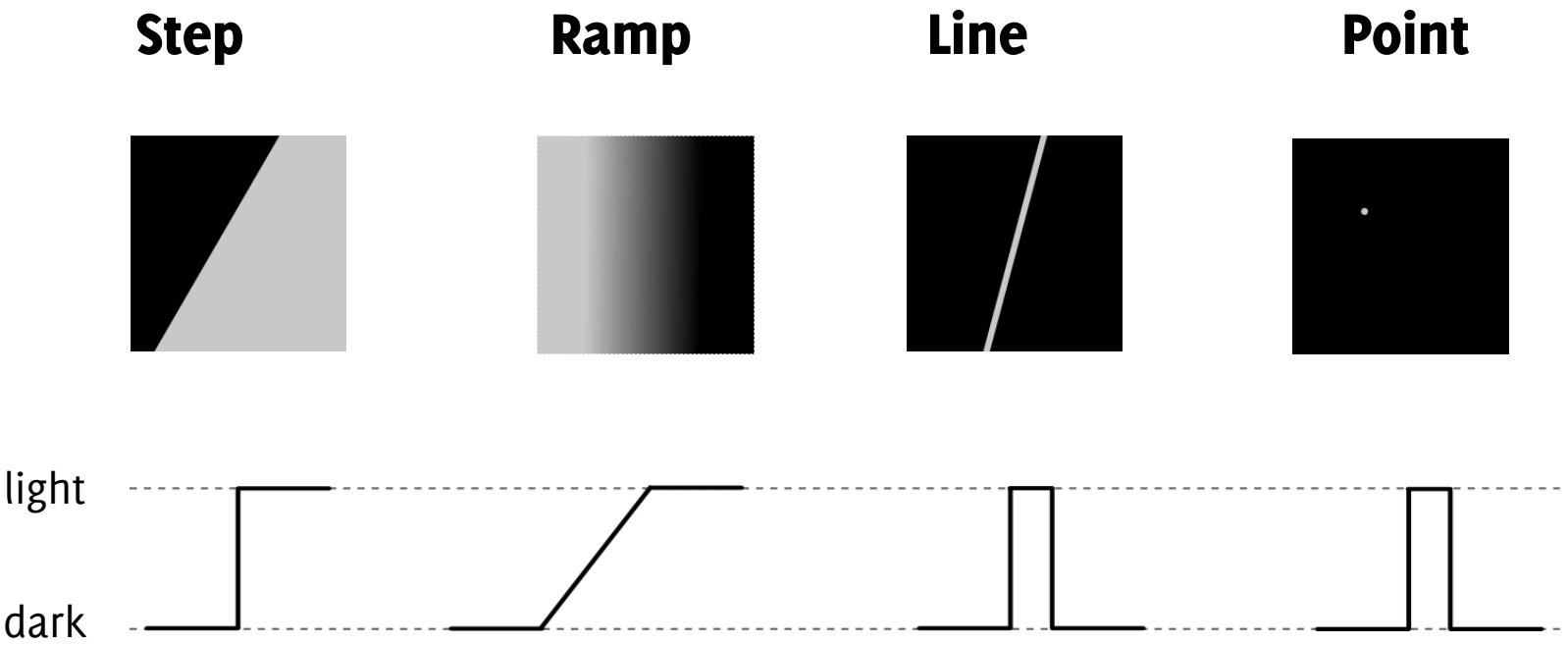
Graythresh uses Otsu's method for finding the threshold. Otsu's method minimizes the intraclass variance.



Edge-based segmentation

Edge-based segmentation

Detection of sharp, local changes in intensity.

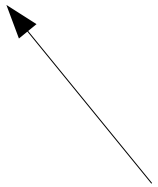


Edge-based segmentation

General workflow

1. Detect edges, i.e., mark each pixel as ”edge” or ”not edge”.

2. Divide the image into regions, based on the detected edges. (Edge linking, Hough transform)



This part is non-trivial!

Edge detection

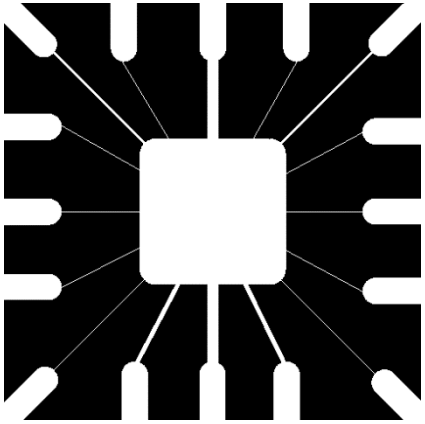
Edge detection typically consists of two steps:

1. **Enhance edges**
   Apply an edge/point detector (e.g. Sobel, Laplace)

2. **Extract edges**
   Segment edges of interest (e.g. thresholding)

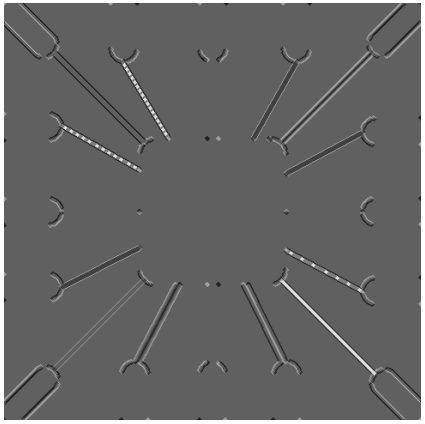
Example

Task: Find lines in this image, that are 1 pixel thick and have an orientation of -45°.



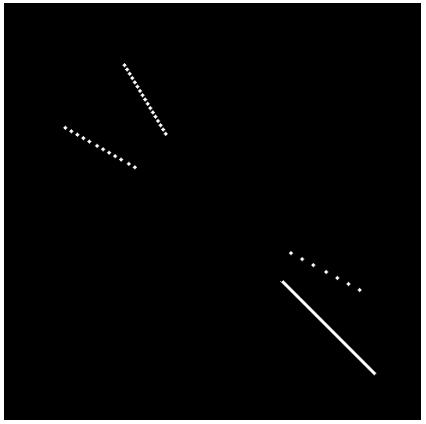
Original image

(Wire-bond mask of an electric circuit)



Filtered

2	-1	-1
-1	**2**	-1
-1	-1	**2**



Thresholded

Hough transform

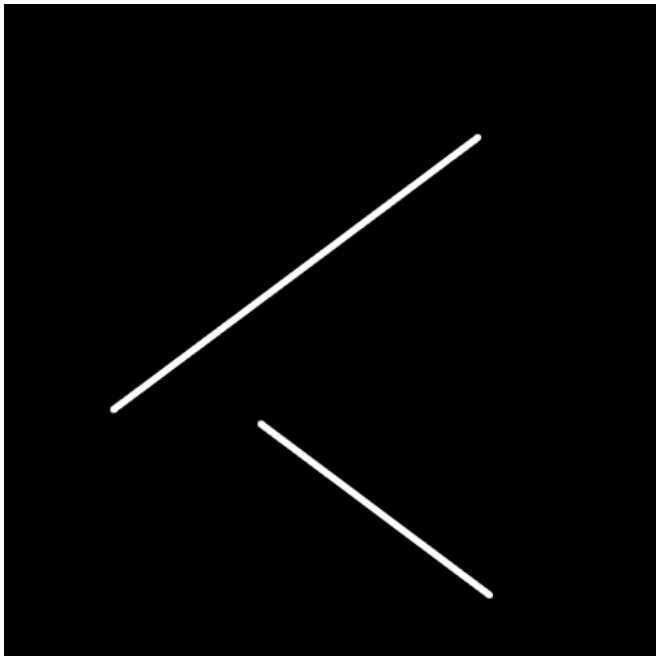
- Consider a point (x_i, y_i) in the plane. Infinitely many lines pass through (x_i, y_i), all satisfying the equation $y_i = ax_i + b$ for varying a and b.
- We can rewrite this equation as $b = -x_i a + y_i$. The set of lines passing through the fixed point (x_i, y_i) in the xy-plane form a single line in the ab-plane (parameter space).
- A second point (x_j, y_j) is also associated with a line in parameter space. This line intersects the line associated with (x_i, y_i) at a point

  (a', b') corresponding to the line containing both (x_i, y_i) and (x_j, y_j). In fact, all points on this line have lines in parameter space that intersect at (a', b').

Hough transform

- Assume that we are given n points in an image (detected edge points).
- Each of these points can be considered as ”evidence” for the family of lines passing through that point (a line in parameter space).
- In principle, the parameter space line corresponding to each point could be plotted, and principal lines in the image could be found by identifying points in parameter space where a large number of lines intersect.
- A practical problem is that a (the slope of the line) approaches infinity as the line approaches the vertical direction. This can be solved by using the following representation of a line: $x \cos \Theta + y \sin \Theta = \rho$.

Hough transform



Original image

Hough transform
(Matlab: hough(image);)

Hough transform

The Hough transform groups edge points into object candidates. This is performed by an explicit voting procedure over a set of parameterized image objects.

What is it good for?
“Holes“ (missing pixels) in the desired curves.
Noisy edge points (detected by the edge detector)

The Hough transform can also be extended to find other parametric objects (curves, ellipses, etc.)

Hough transform in MATLAB

```
im = imread('circuit.tif');
BW = edge(im,'canny');
[H,theta,rho] = hough(BW);
P = houghpeaks(H,5,'threshold',ceil(0.3*max(H(:))));
lines = houghlines(BW,theta,rho,P,'FillGap',5,'MinLength',7);
figure, imshow(im), hold on
xy = [lines(1).point1; lines(1).point2];
plot(xy(:,1),xy(:,2),'LineWidth',2,'Color','green');
```

`hough` computes the hough transform.

`houghpeaks` find maxima in the hough transform H.

`houghlines` find start and end points of line segments.

Region-based segmentation

Region-based segmentation

Region splitting and merging *(Top-down approach)*

1. Set up criteria for what is a uniform area (e.g. mean, variance, bi-modality of histogram, texture, etc.).
2. Start with the full image and **split** it into four sub-images.
3. Check each sub-image. If it is not uniform, **split** it again into four sub-images.
4. Repeat 3. until no more splitting is performed.
5. Compare sub-images with the neighboring regions and **merge**, if they are uniform.
6. Repeat 5. until no more merging is performed.

The method is also called **quad-tree** division.

Region-based segmentation

Region growing *(Bottom-up approach)*

1. Find starting points.
2. Include neighboring pixels with similar feature (gray level, texture, color, etc.).
3. Continue until all pixels have been associated with one of the starting points.

Problems

Non trivial to find good starting points, difficult to automate and needs good criteria for similarity.

Watershed segmentation

Watershed -drop of water analogy

Think of the gray level image as a **landscape**. A drop of water landing at any point in the landscape will flow down to a local minimum in the landscape.

For any local minimum in the landscape, there is a set of points, called the *catchment basin,* from which a drop of water will flow to that given minimum.

The boundaries between adjacent catchment basins form the watershed.

Watershed -flooding analogy

Think of the gray level image as a **landscape**.

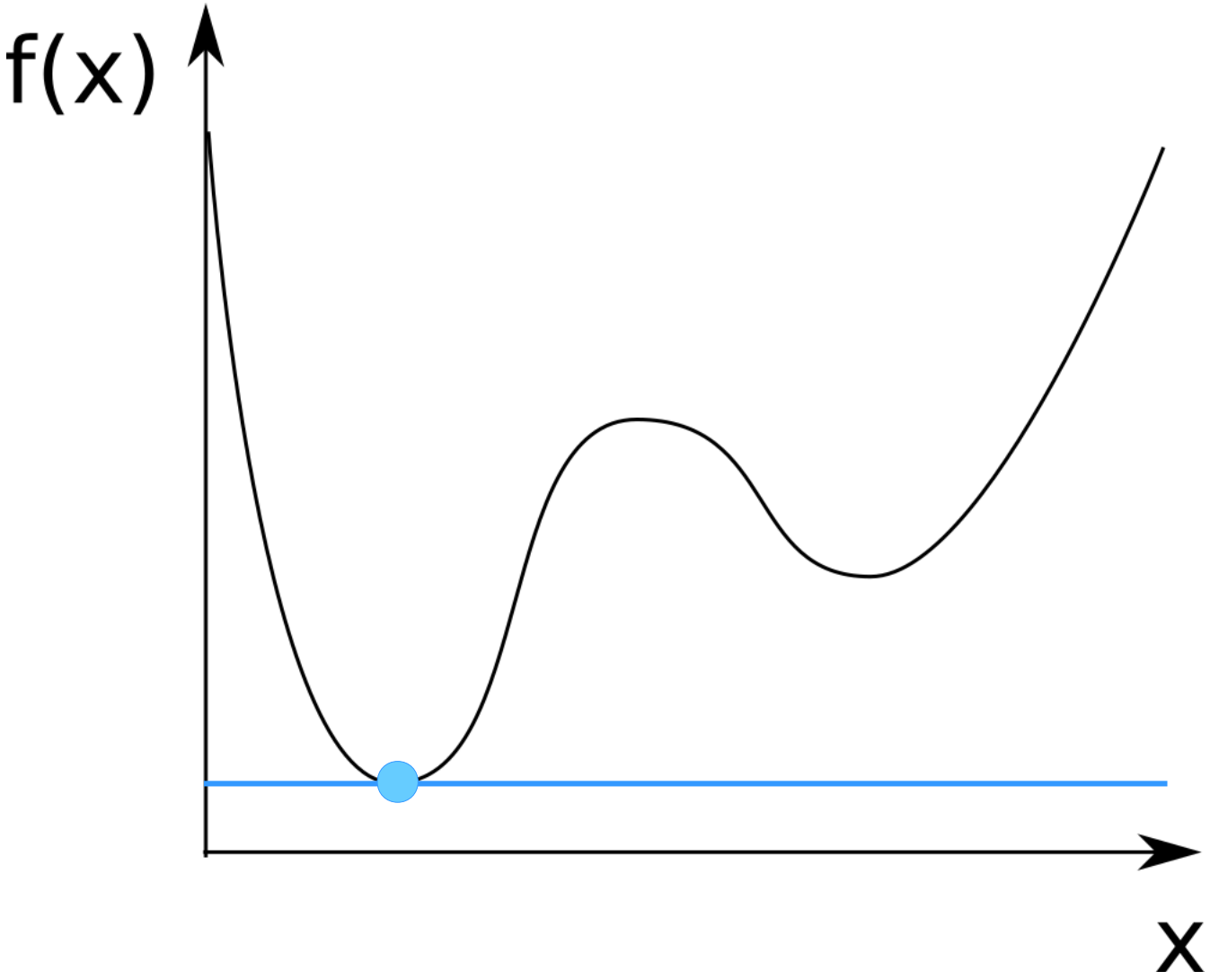
Let **water rise** from the bottom of each valley (the water from the valley it given its own label).

As soon as the water from two valleys meet, build a dam, or a **watershed**.

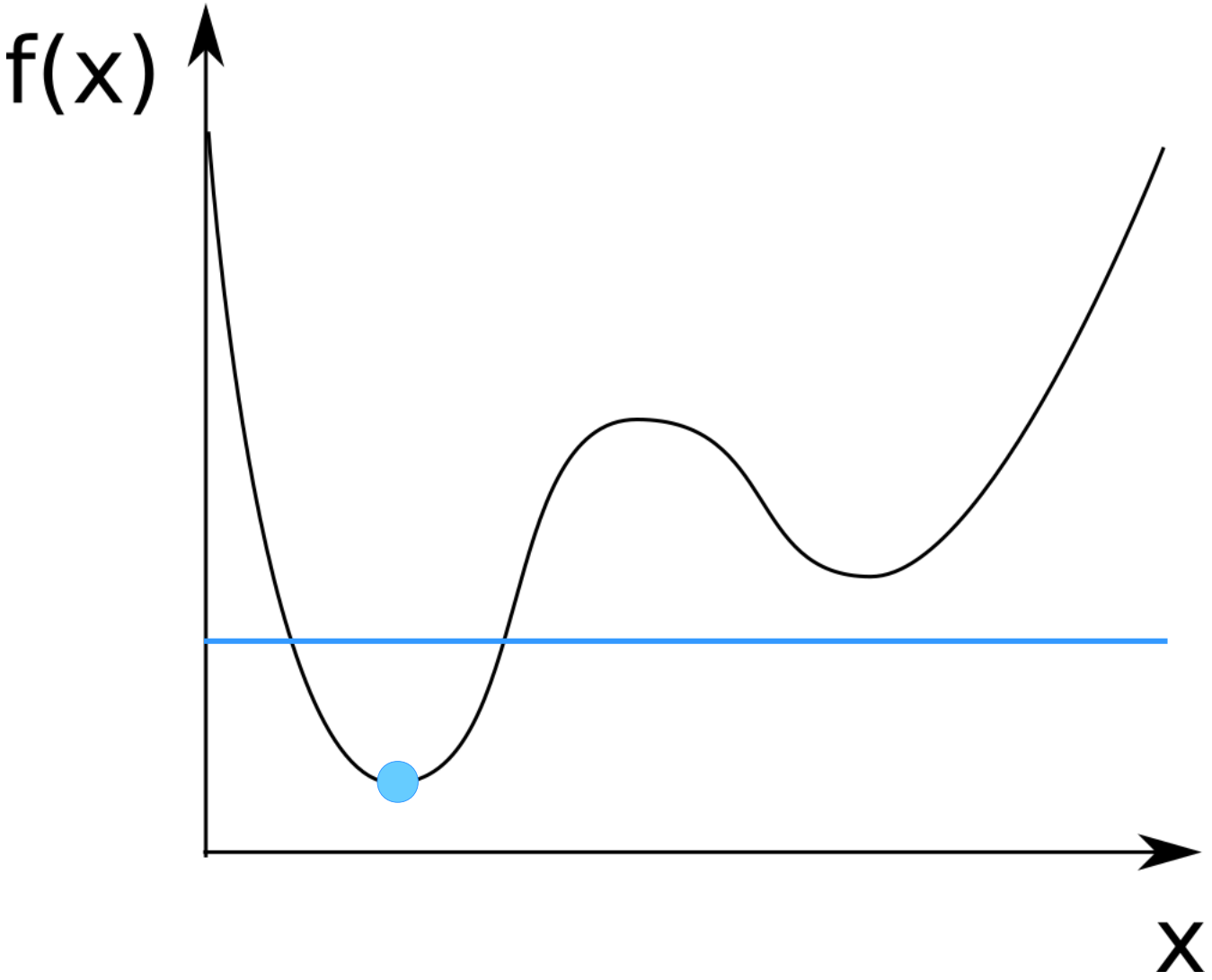
These watersheds will define the **borders** between different regions in the image.

The watershed algorithm can be used directly on the image, on an edge enhanced image or on a distance transformed image.

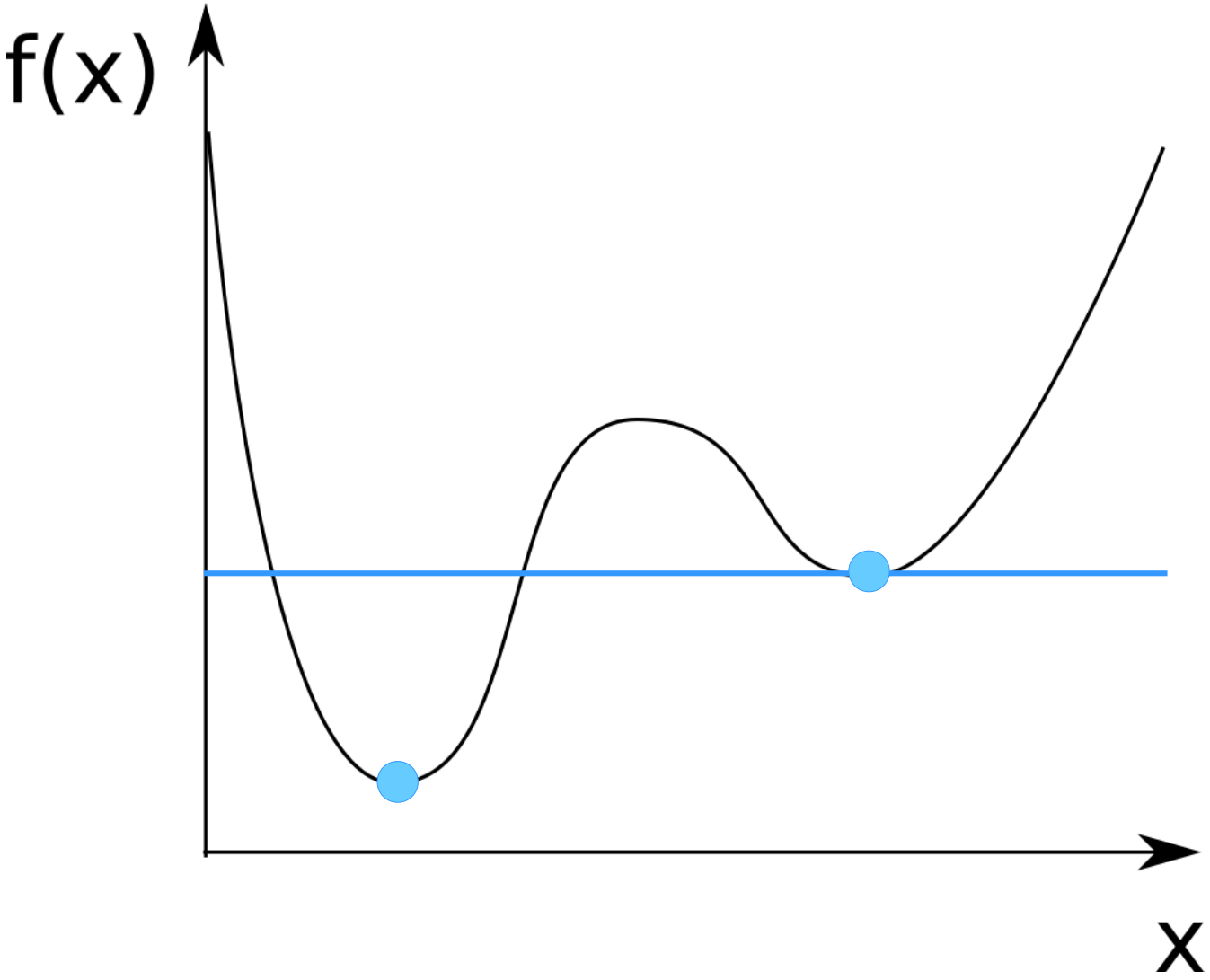
Watershed process



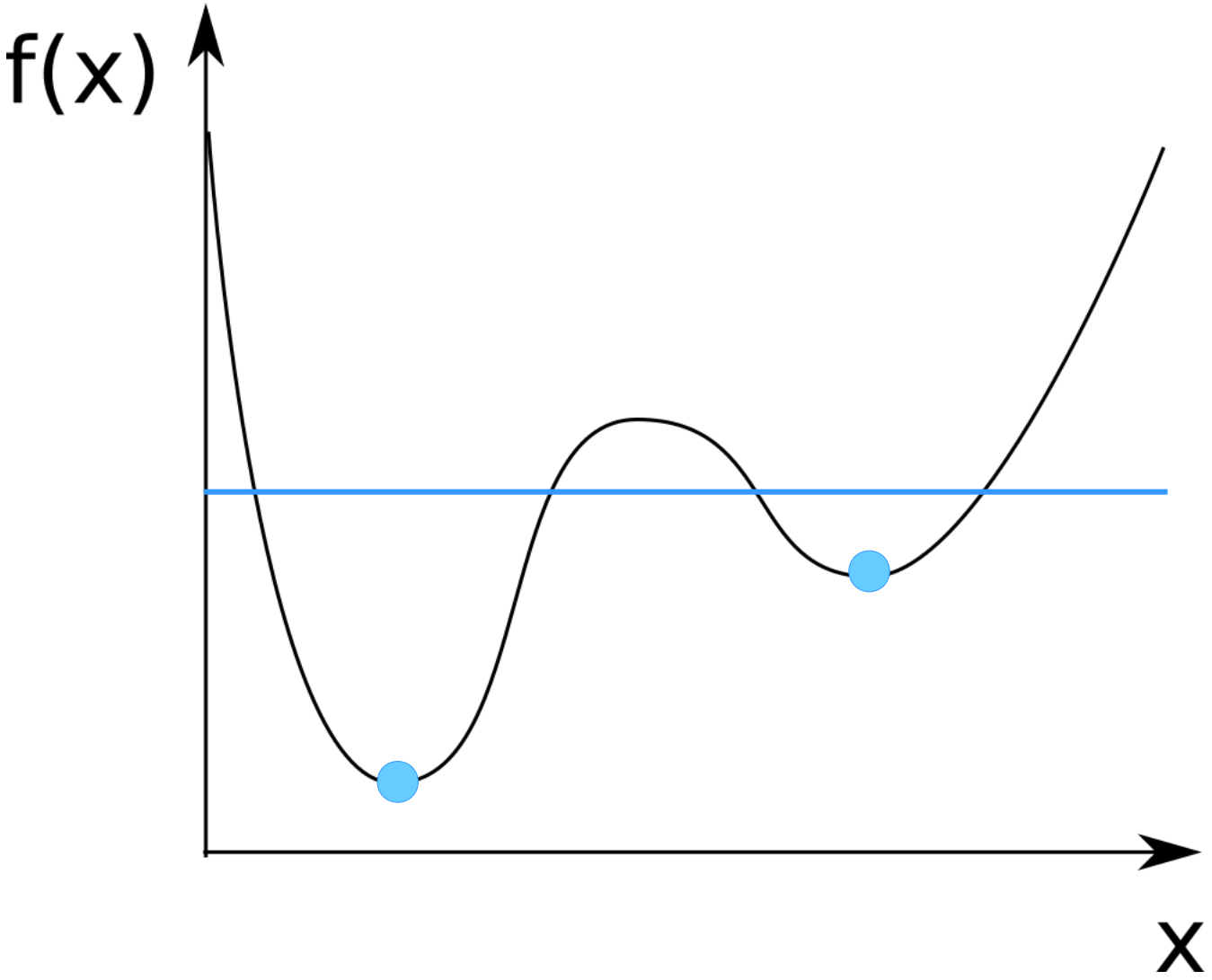
Watershed process



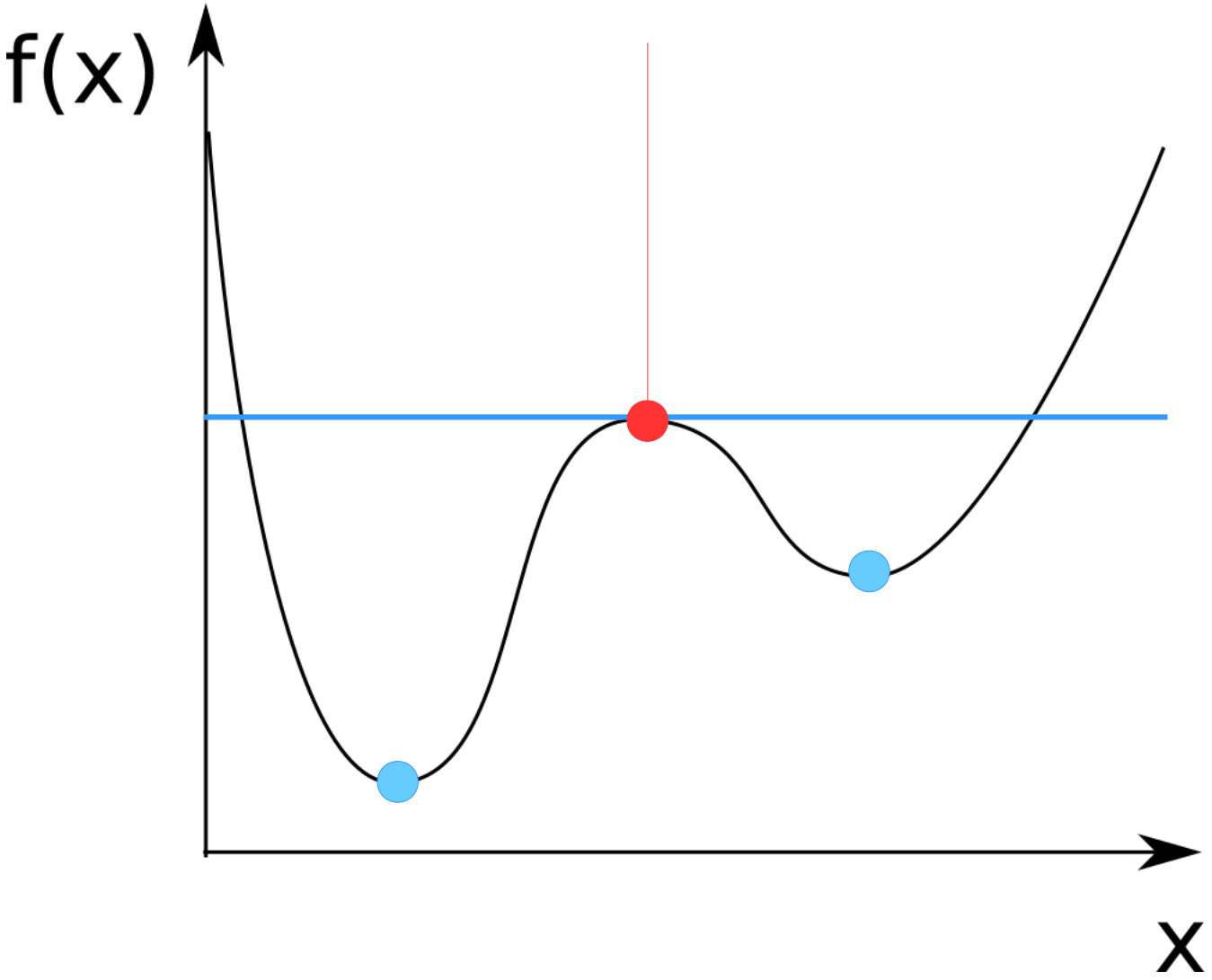
Watershed process



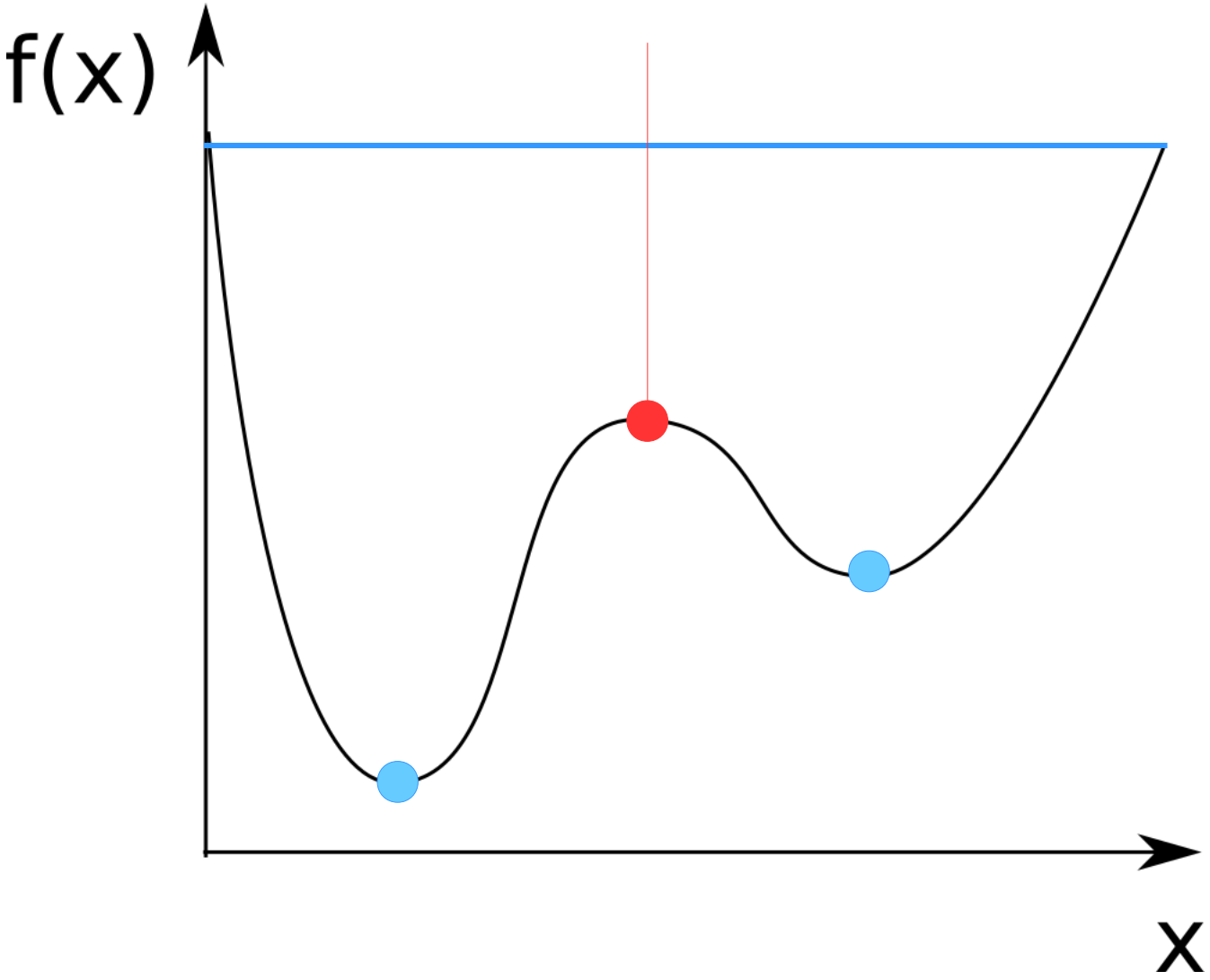
Watershed process



Watershed process

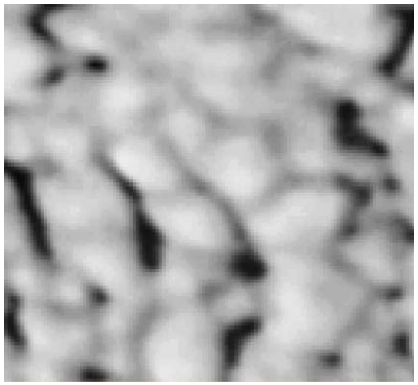


Watershed process

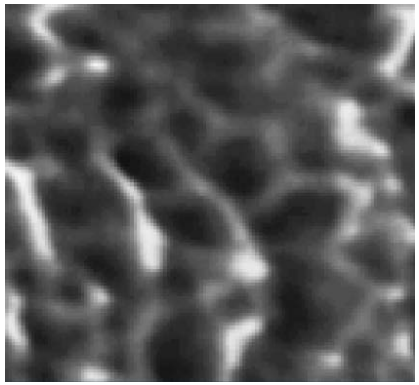


Watershed

Example of watershed directly applied on gray level image:

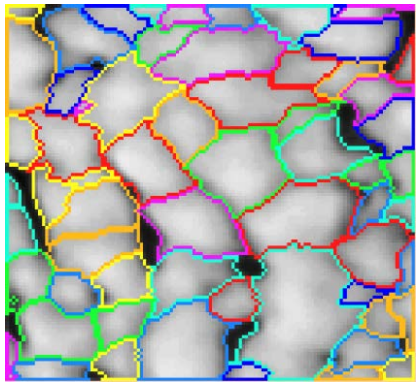


Original image



Inverted image

(starting points are in valleys)

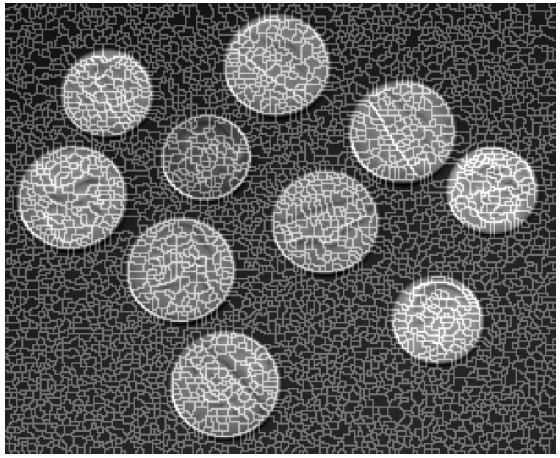


Segmentation result

Watershed

Example of watershed directly applied on gray level image:

Original image

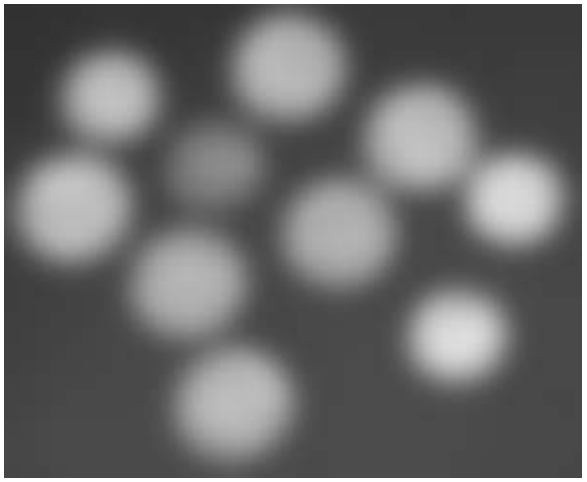


Segmentation result

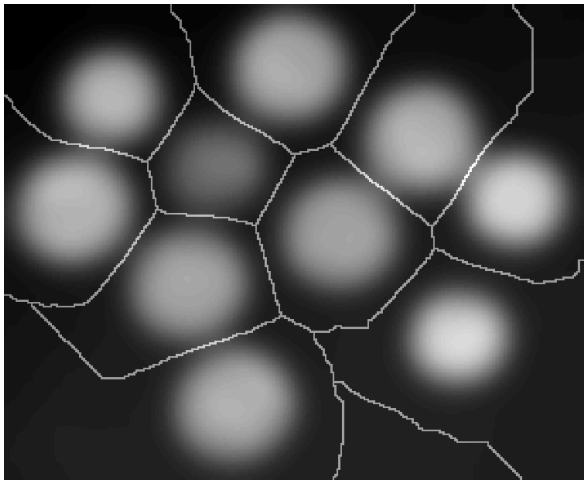
```
>> image=imread('coins.png');
>> ws=watershed(image);
>> imagesc(ws);
>>
```

Watershed

Example of watershed directly applied on gray level image:



Blurred image

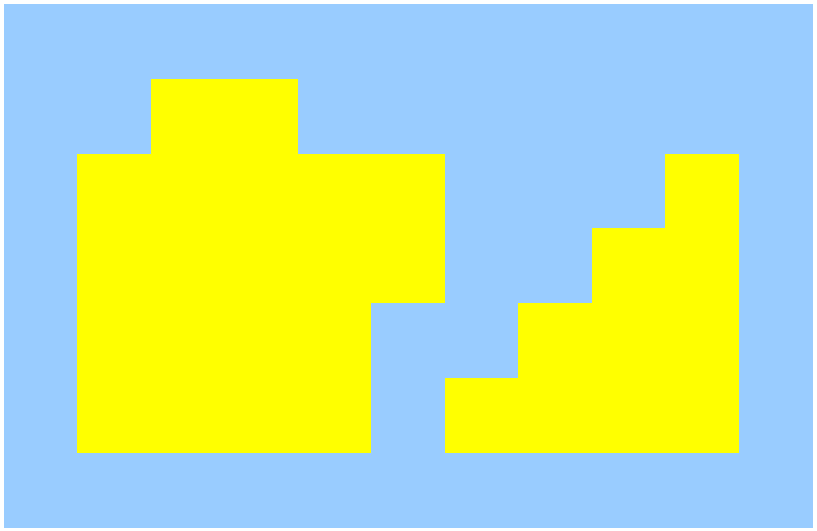


Segmentation result

Distance transform

Distance measure

2	1	2
1	0	1
2	1	2



Original image

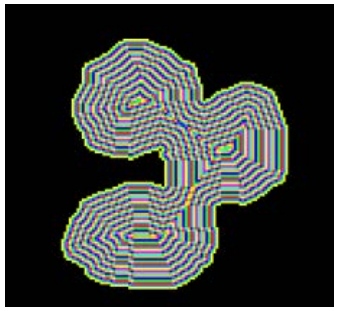
0	0	0	0	0	0	0	0	0	0	0
0	0			0	0	0	0	0	0	0
0						0	0	0	1	0
0						0	0	1	1	0
0					0	0	1	2	1	0
0					0	1	1	1	1	0
0	0	0	0	0	0	0	0	0	0	0

Distance transform

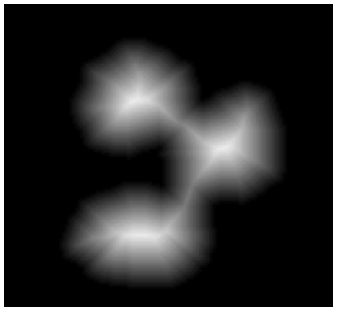
Watershed

Example of watershed on distance transformed image:

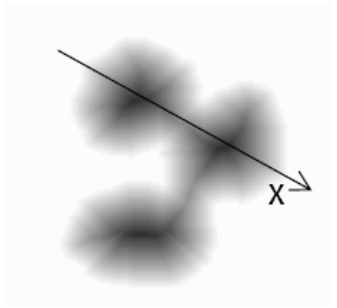
Original image



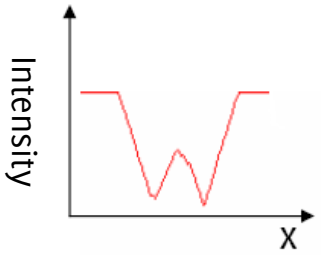
Distance transform



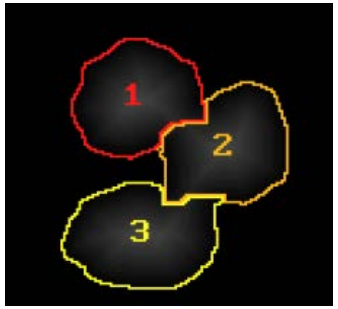
DT as intensity



DT inverse

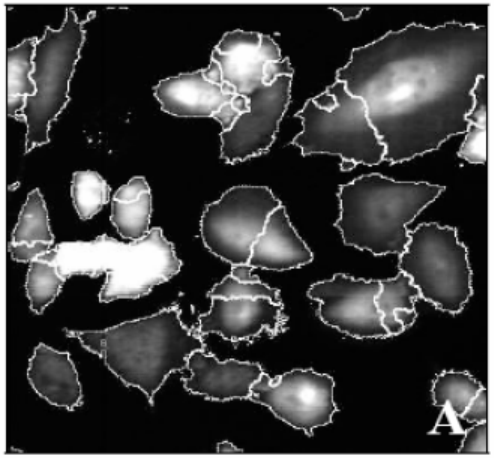


“Intensity landscape“

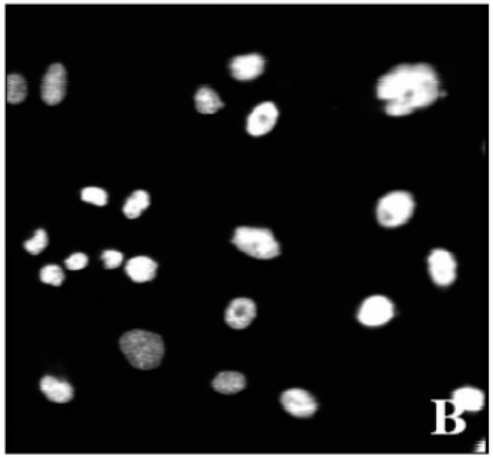


Segmentation result

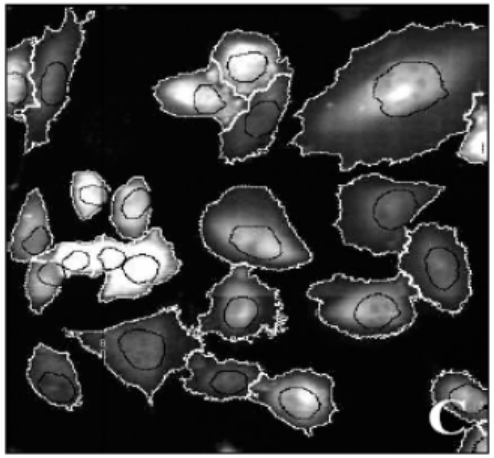
Seeded watershed



Oversegmentation



Seeds (nuclei)



Seeded watershed result

Example for seeded watershed:

Every cell has a cell nucleus, which can be detected by thresholding and watershed segmentation. Using these nuclei as seeds, the cytoplasms are easy to find.

Computing watersheds

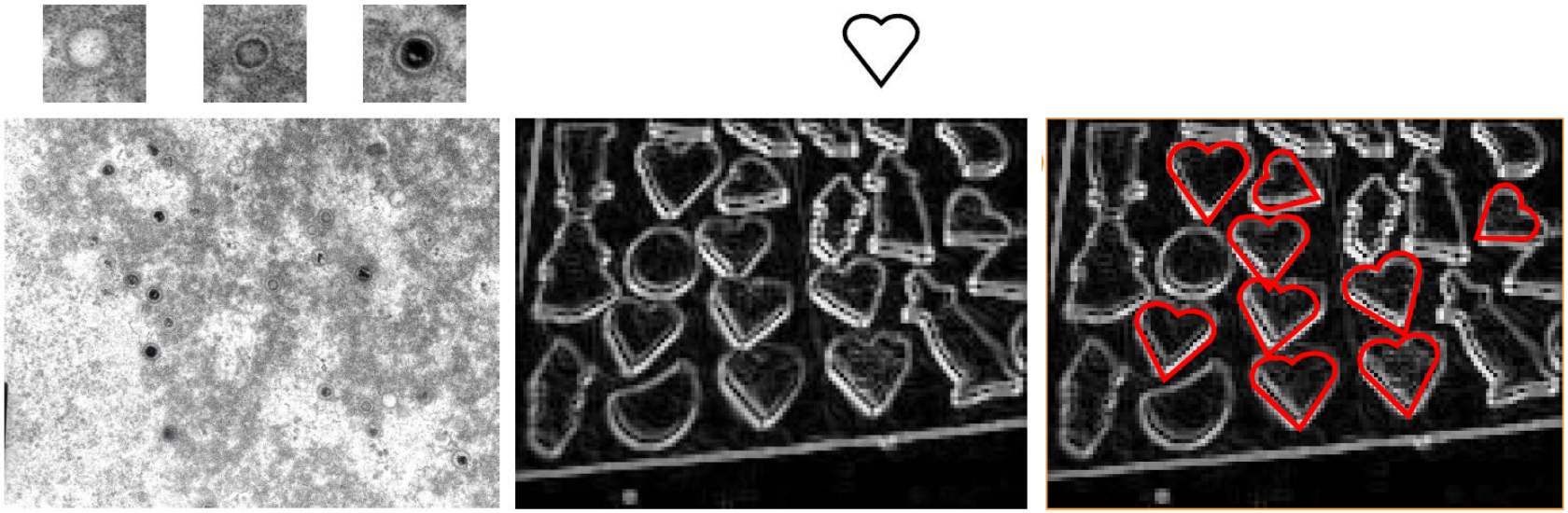
Algorithm by F. Meyer (early 90's)

1. A set of markers, pixels where the flooding shall start, are chosen. Each is given a different label.
2. The neighboring pixels of each marked area are inserted into a priority queue with a priority level corresponding to the gray level of the pixel.
3. The pixel with the highest priority level is extracted from the priority queue. If the neighbors of the extracted pixel that have already been labeled all have the same label, then the pixel is labeled with their label. All non-marked neighbors that are not yet in the priority queue are put into the priority queue.
4. Redo step 3 until the priority queue is empty.

Match-based segmentation

Match-based segmentation

Compare a **template** to the underlying image to find objects with a certain intensity distribution or shape.



Match-based segmentation

Computational problem: Testing all possible transformation (translation, rotation, scaling, etc. -> depends on the application) of the template / model.

Simplify segmentation by experimental design

When possible:

- Make sure the illumination is even
- Avoid shading
- Have a uniform background (in a different color)
- Avoid reflection (glittering)
- Use a standardized position (industry)