Digital Image Processing

Basic Methods for Image Segmentation

Christophoros Nikou

cnikou@cs.uoi.gr

Images taken from:
R. Gonzalez and R. Woods. Digital Image Processing, Prentice Hall, 2008.
Computer Vision course by Svetlana Lazebnik, University of North Carolina at Chapel Hill (http://www.cs.unc.edu/~lazebnik/spring10/).

Image Segmentation

- Obtain a compact representation of the image to be used for further processing.
- Group together similar pixels
- Image intensity is not sufficient to perform *semantic* segmentation
  - Object recognition
    - Decompose objects to simple tokens (line segments, spots, corners)
  - Finding buildings in images
    - Fit polygons and determine surface orientations.
  - Video summarization
    - Shot detection

Image Segmentation (cont.)

Goal: separate an image into “coherent” regions.

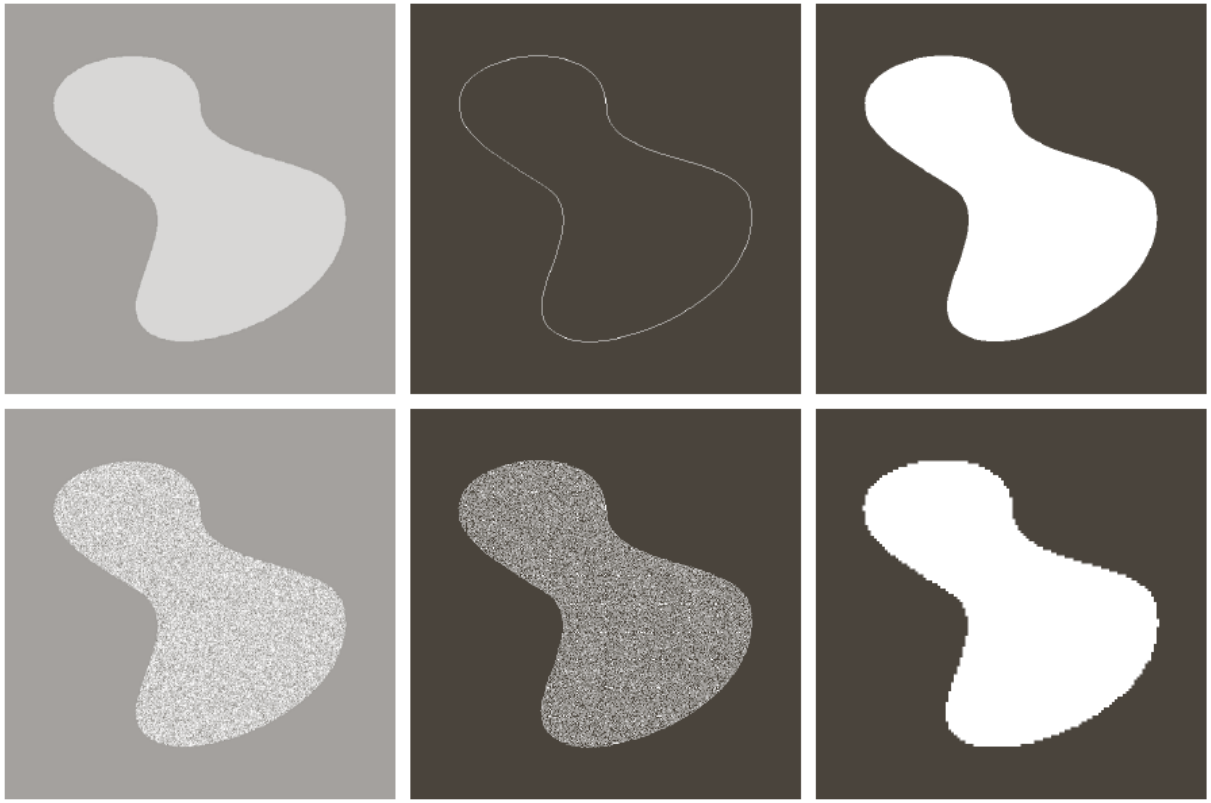
- Basic methods
  - point, line, edge detection
  - thresholding
  - region growing
  - morphological watersheds
- Advanced methods
  - clustering
  - model fitting.
  - probabilistic methods.
  - …

Fundamentals

- Edge information is in general not sufficient.

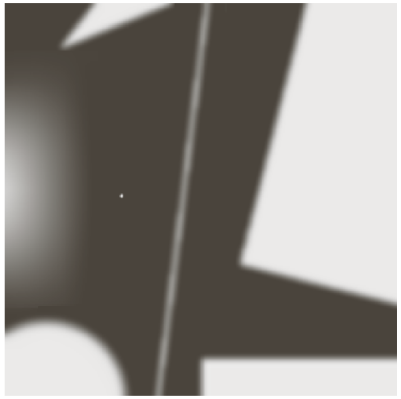
Constant intensity (edge-based segmentation)

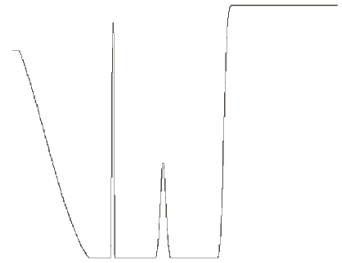
Textured region (region-based segmantation)

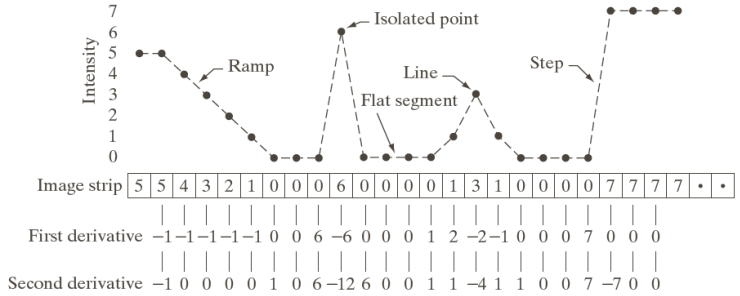


Point, line and edge detection

- First order derivatives produce thick edges at ramps.
- Second order derivatives are non zero at the onset and at the end of a ramp or step edge (sign change).
- Second order derivatives respond stronger at isolated points and thin lines than first order derivatives.





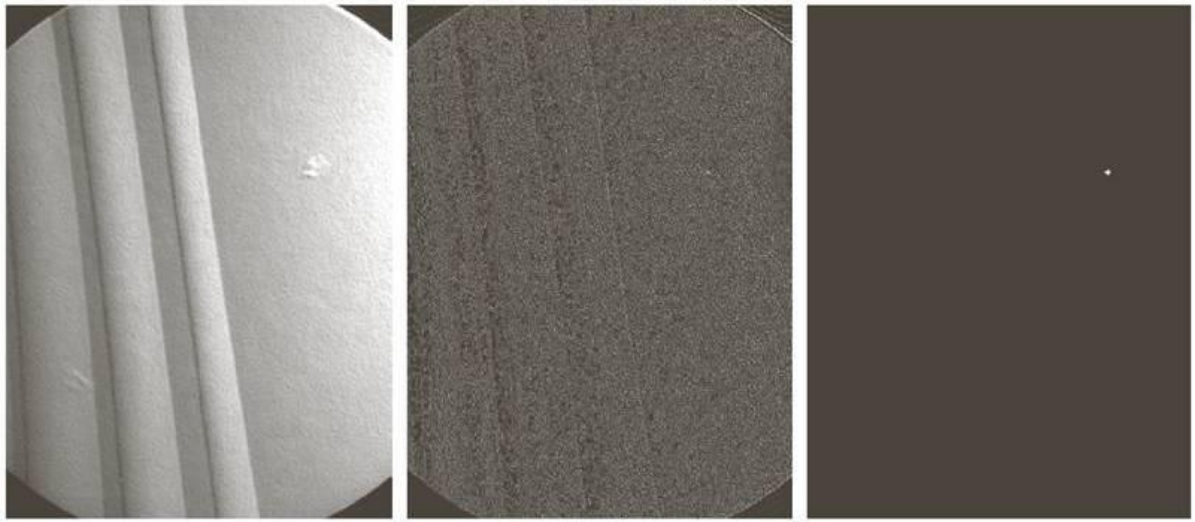


Detection of isolated points

- Simple operation using the Laplacian.

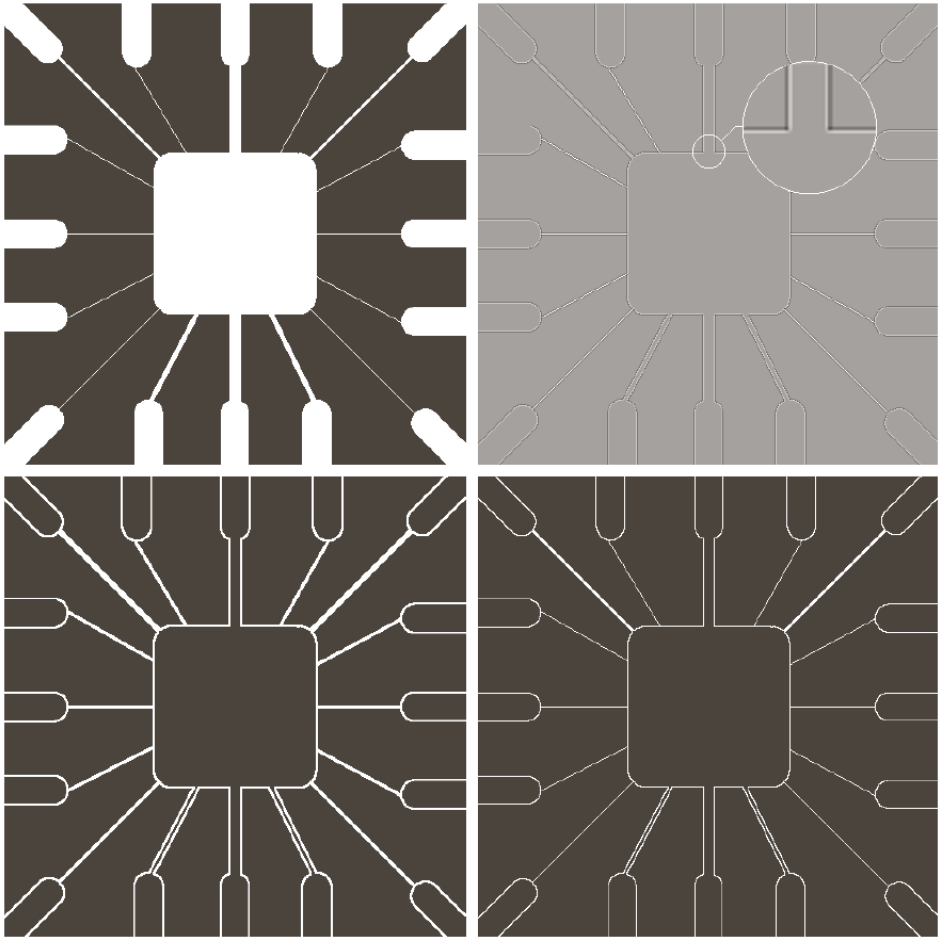
$$g(x,y) = \begin{cases} 1 & \text{if } \left|\nabla^2 f(x,y)\right| > T \\ 0 & \text{otherwise} \end{cases}$$

1	1	1
1	−8	1
1	1	1



Line detection

- The Laplacian is also used here.
- It has a double response
  - Positive and negative values at the beginning and end of the edges.
- Lines should be thin with respect to the size of the detector



Absolute value Positive value

Line detection (cont.)

- The Laplacian is isotropic.
- Direction dependent filters localize 1 pixel thick lines at other orientations (0, 45, 90).

−1	−1	−1
2	2	2
−1	−1	−1

Horizontal

2	−1	−1
−1	2	−1
−1	−1	2

+45°

−1	2	−1
−1	2	−1
−1	2	−1

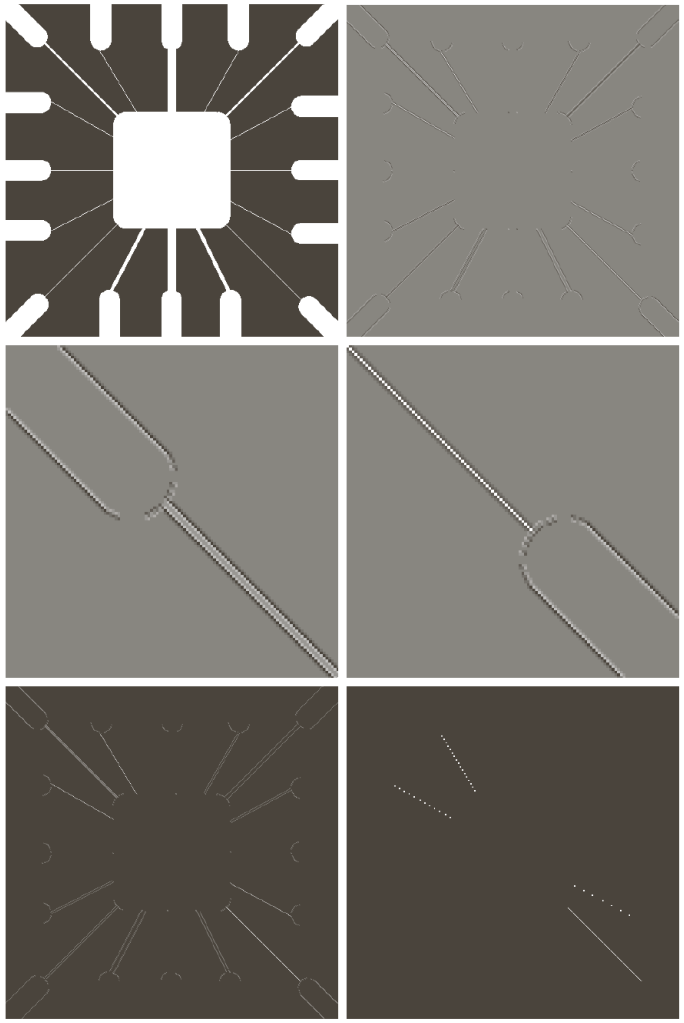
Vertical

−1	−1	2
−1	2	−1
2	−1	−1

−45°

Line detection (cont.)

- The Laplacian is isotropic.
- Direction dependent filters localize 1 pixel thick lines at other orientations (0, 45, 90).

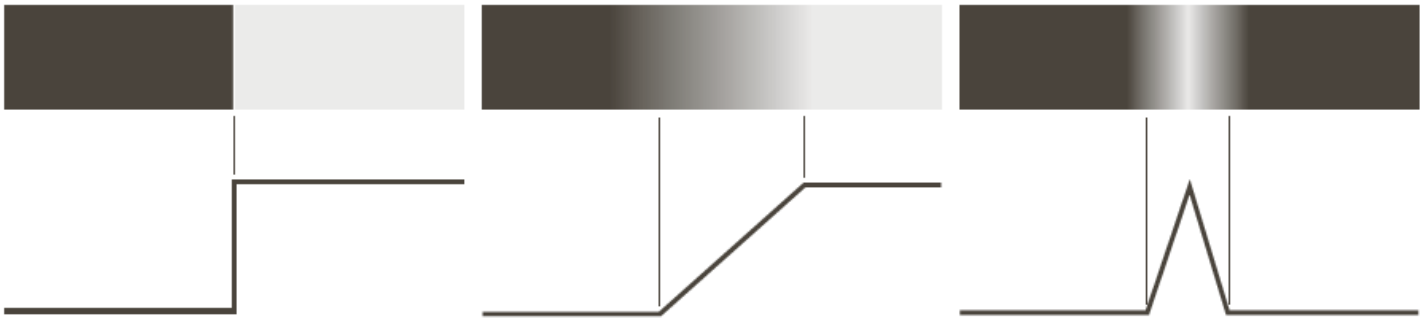


a b
c d
e f

FIGURE 10.7
(a) Image of a wire-bond template. (b) Result of processing with the +45° line detector mask in Fig. 10.6. (c) Zoomed view of the top left region of (b). (d) Zoomed view of the bottom right region of (b). (e) The image in (b) with all negative values set to zero. (f) All points (in white) whose values satisfied the condition $g \geq T$, where g is the image in (e). (The points in (f) were enlarged to make them easier to see.)

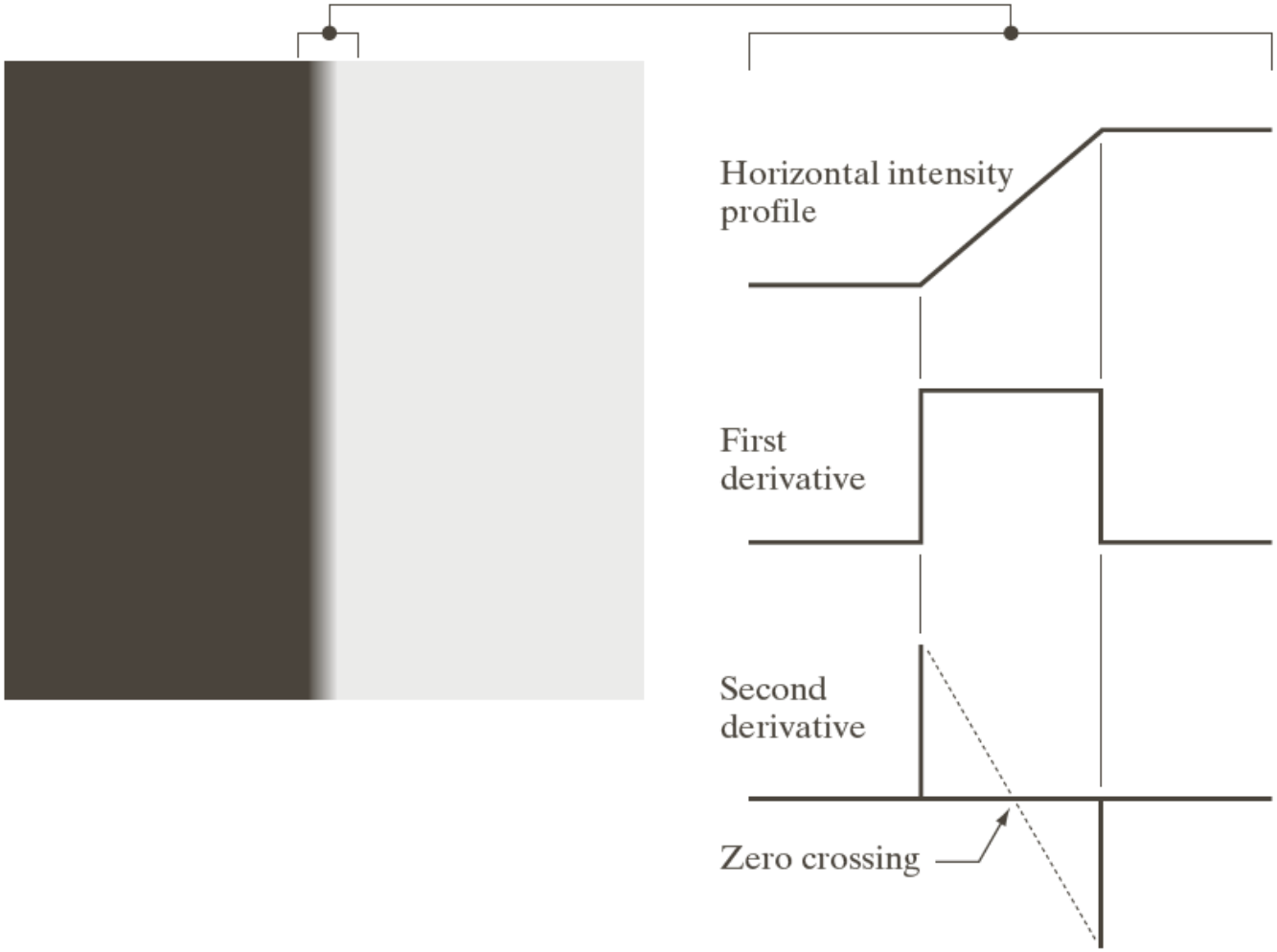
Edge detection

- Edge models
  - Ideally, edges should be 1 pixel thin.
  - In practice, they are blurred and noisy.

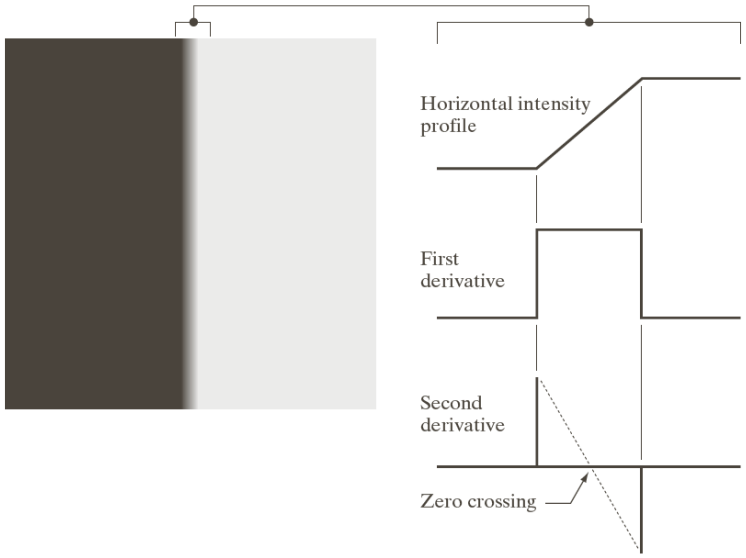


Step edge Ramp edge Roof edge

Edge detection (cont.)



Edge detection (cont.)



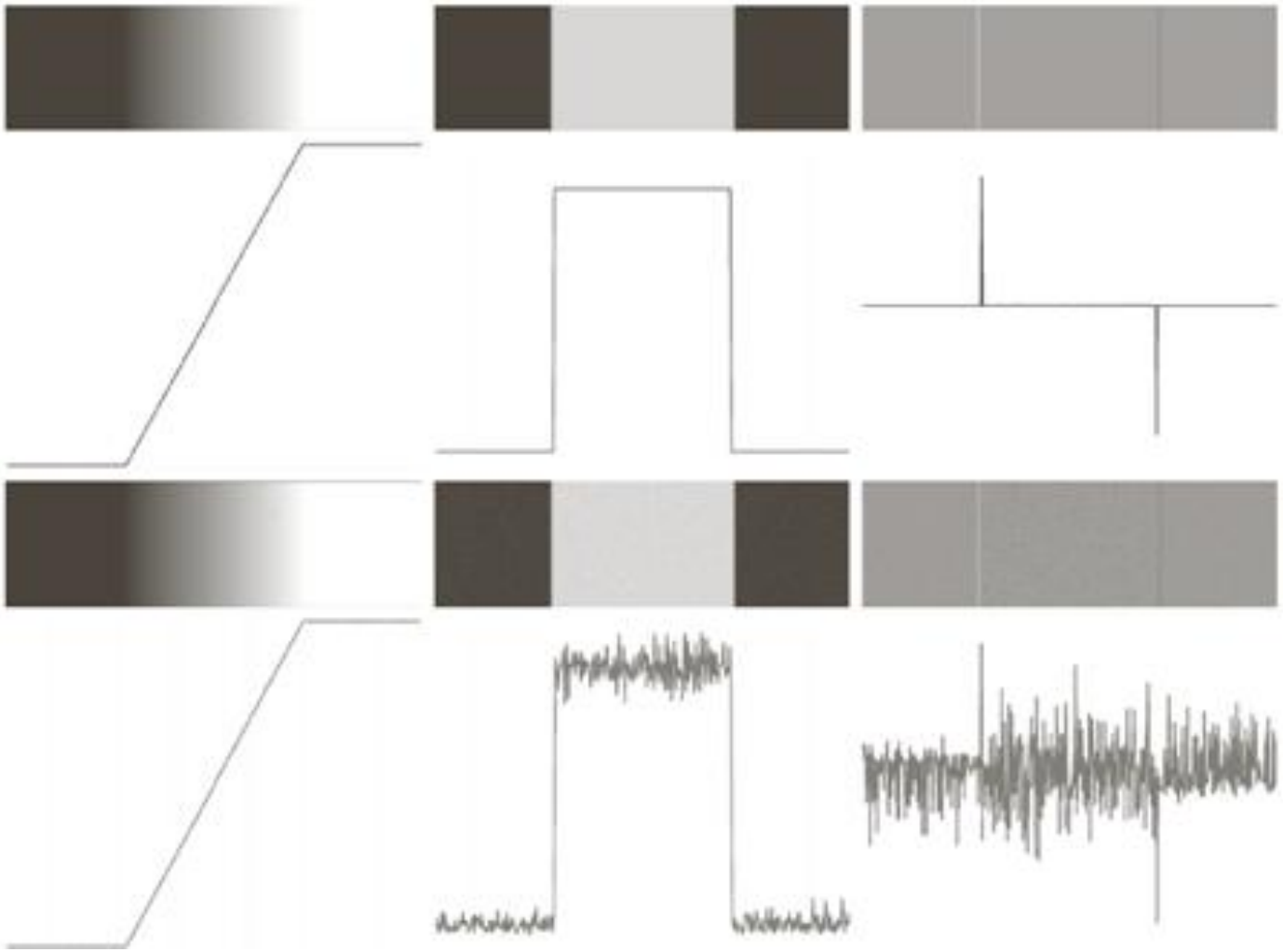
Edge point detection

- Magnitude of the first derivative.
- Sign change of the second derivative.

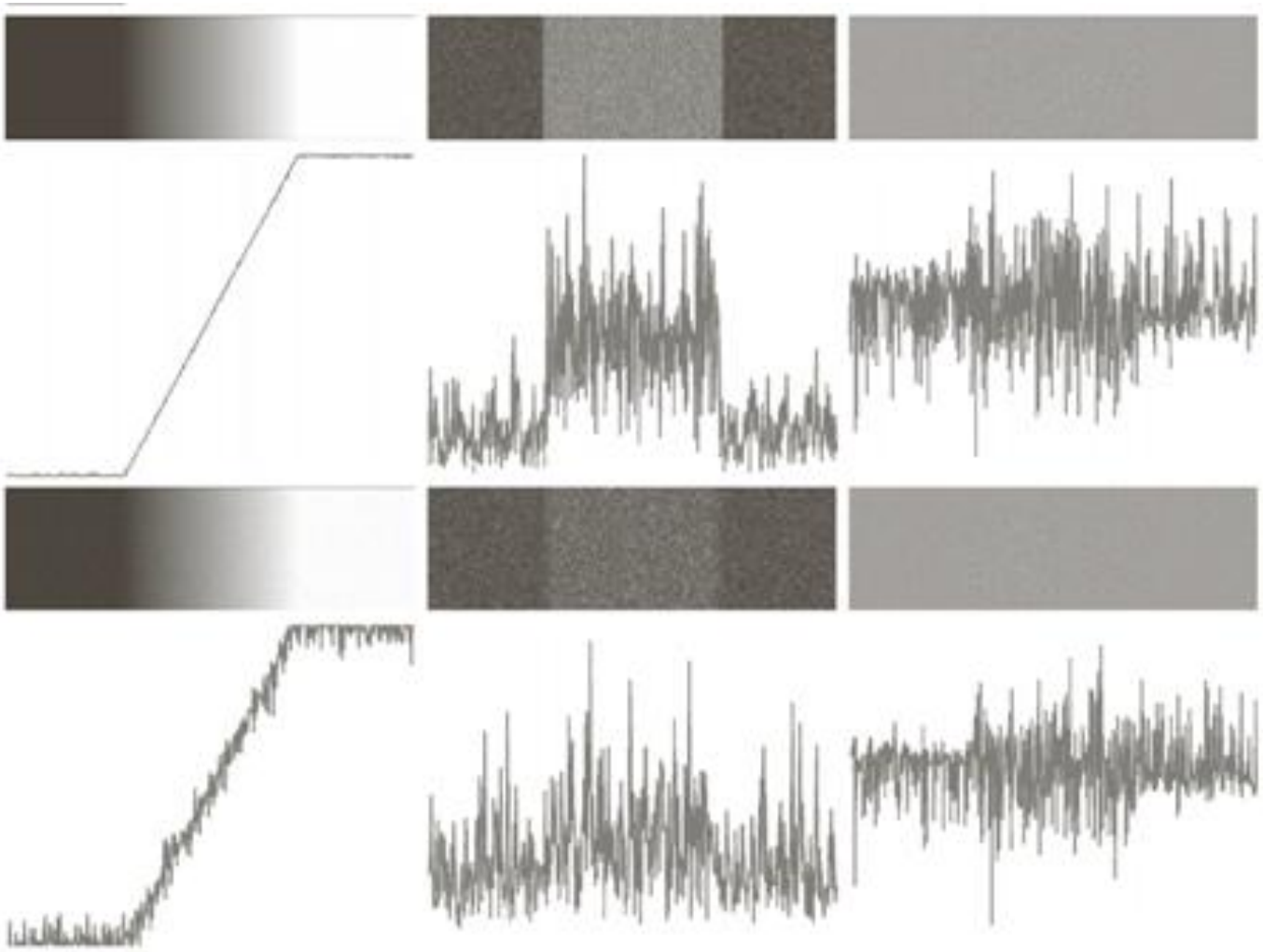
Observations:

- Second derivative produces two values for an edge (undesirable).
- Its zero crossings may be used to locate the centres of thick edges.

Edge model and noise



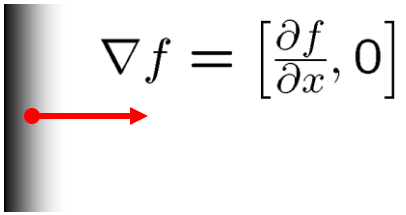
Edge model and noise (cont.)

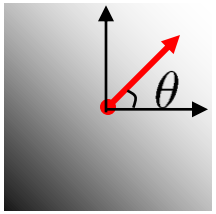


Fundamental steps in edge detection

- Image smoothing for noise reduction.
  - Derivatives are very sensitive to noise.
- Detection of candidate edge points.
- Edge localization.
  - Selection of the points that are true members of the set of points comprising the edge.

Image gradient

- The gradient of an image: $\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$
- 



The gradient points in the direction of most rapid increase in intensity.

The gradient direction is given by $\theta = \tan^{-1}\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x}\right)$

The *edge strength* is given by the gradient magnitude

$$\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

Source: Steve Seitz

Gradient operators

$$\frac{\partial f(x,y)}{\partial x} = f(x+1,y) - f(x,y)$$

$$\frac{\partial f(x,y)}{\partial y} = f(x,y+1) - f(x,y)$$

Roberts operators

−1
1

−1	1

−1	0	0	−1
0	1	1	0

Roberts

Gradient operators (cont.)

Integrates image smoothing

−1	−1	−1
0	0	0
1	1	1

−1	0	1
−1	0	1
−1	0	1

Prewitt

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

Sobel

Gradient operators (cont.)

Diagonal edges

0	1	1	−1	−1	0
−1	0	1	−1	0	1
−1	−1	0	0	1	1

Prewitt

0	1	2	−2	−1	0
−1	0	1	−1	0	1
−2	−1	0	0	1	2

Sobel

Gradient operators (cont.)

Image

Sobel $|g_y|$

Sobel $|g_x|$

Sobel $|g_x|+|g_x|$

Gradient operators (cont.)

Image smoothed prior to edge detection.

The wall bricks are smoothed out.

Image

Sobel $|g_y|$

Sobel $|g_x|$

Sobel $|g_x|+|g_x|$

Gradient operators (cont.)

Diagonal Sobel filters

Gradient operators (cont.)

Thresholded Sobel gradient amplitydes at 33% of max value

Thresholded gradient

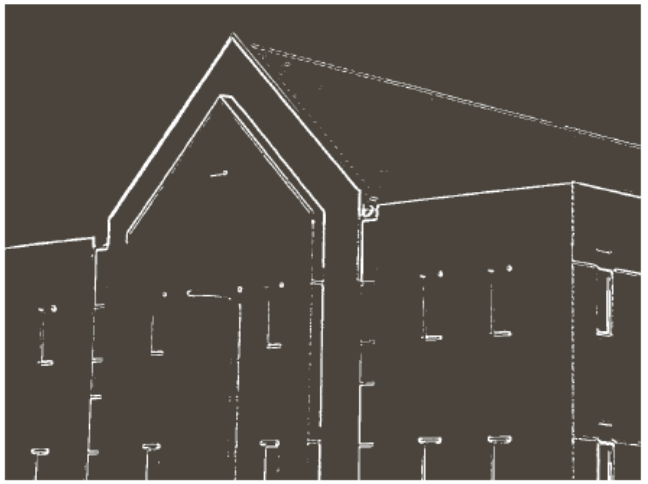


Image smoothing prior to gradient thresholding

The LoG operator

- A good place to look for edges is the maxima of the first derivative or the zeros of the second derivative.
- The 2D extension approximates the second derivative by the Laplacian operator (which is rotationally invariant):

$$\nabla^2 f(x,y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

- Marr and Hildreth [1980] argued that a satisfactory operator that could be tuned in scale to detect edges is the Laplacian of the Gaussian (LoG).

The LoG operator (cont.)

$$\nabla^2\left(G(x,y)*f(x,y)\right)=\nabla^2 G(x,y)*f(x,y)$$

- The LoG operator is given by:

$$\nabla^2 G(x,y)=\left(\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}\right)G(x,y)=$$

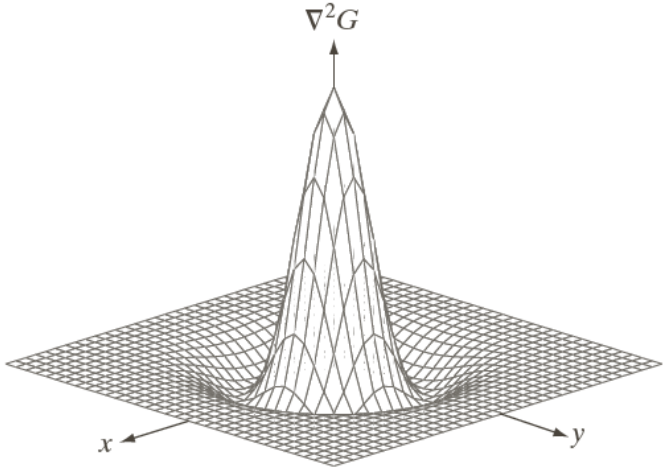
$$\left(\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}\right)e^{-\frac{x^2+y^2}{2\sigma^2}}=\left(\frac{x^2+y^2-2\sigma^2}{\sigma^4}\right)e^{-\frac{x^2+y^2}{2\sigma^2}}$$

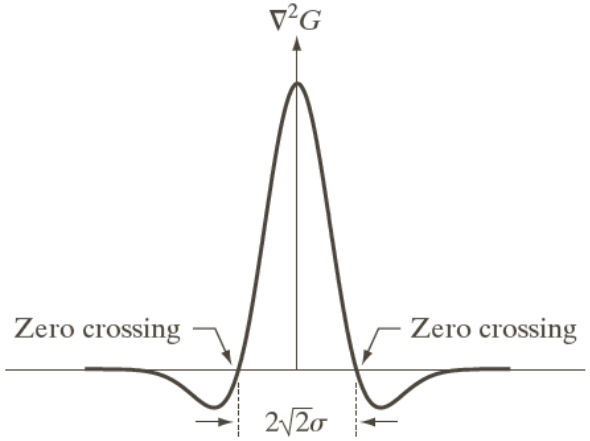
The LoG operator (cont.)

The zero crossings are at:

$$x^2 + y^2 = 2\sigma^2$$

$$\nabla^2 G(x,y) = \frac{\partial^2 G}{\partial x^2} + \frac{\partial^2 G}{\partial y^2} = \left(\frac{x^2 + y^2 - 2\sigma^2}{\sigma^4} \right) e^{-\frac{x^2+y^2}{2\sigma^2}}$$





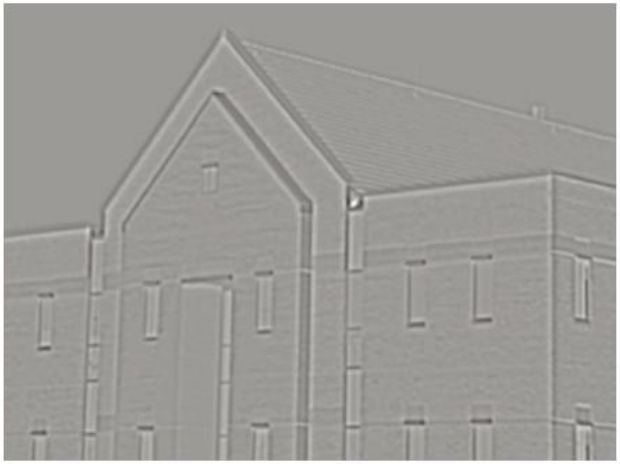
0	0	−1	0	0
0	−1	−2	−1	0
−1	−2	16	−2	−1
0	−1	−2	−1	0
0	0	−1	0	0

The LoG operator (cont.)

- Fundamental ideas
  - The Gaussian blurs the image. It reduces the intensity of structures at scales much smaller than σ.
  - The Laplacian is isotropic and no other directional mask is needed.
- The zero crossings of the operator indicate edge pixels. They may be computed by using a 3x3 window around a pixel and detect if two of its opposite neighbors have different signs (and their difference is significant compared to a threshold).

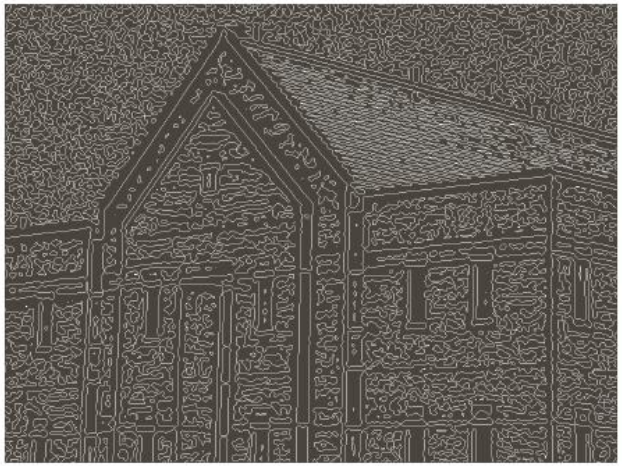
The LoG operator (cont.)

Image



LoG

Zero crossings



Zero crossings with a threshold of 4% of the image max

The LoG operator (cont.)

- Filter the image at various scales and keep the zero crossings that are common to all responses.
- Marr and Hildreth [1980] showed that LoG may be approximated by a difference of Gaussinas (DOG):

$$\mathrm{DoG}(x,y) = \frac{1}{2\pi\sigma_1^2} e^{-\frac{x^2+y^2}{2\sigma_1^2}} - \frac{1}{2\pi\sigma_2^2} e^{-\frac{x^2+y^2}{2\sigma_2^2}}, \quad \sigma_1 > \sigma_2$$

- Certain channels of the human visual system are selective with respect to orientation and frequency and can be modeled by a DoG with a ratio of standard deviations of 1.75.

The LoG operator (cont.)

- Meaningful comparison between LoG and DoG may be obtained after selecting the value of σ for LoG so that LoG has the same zero crossings as DoG:

$$\sigma^2 = \frac{\sigma_1^2 \sigma_2^2}{\sigma_1^2 - \sigma_2^2} \ln\left(\frac{\sigma_1^2}{\sigma_2^2}\right)$$

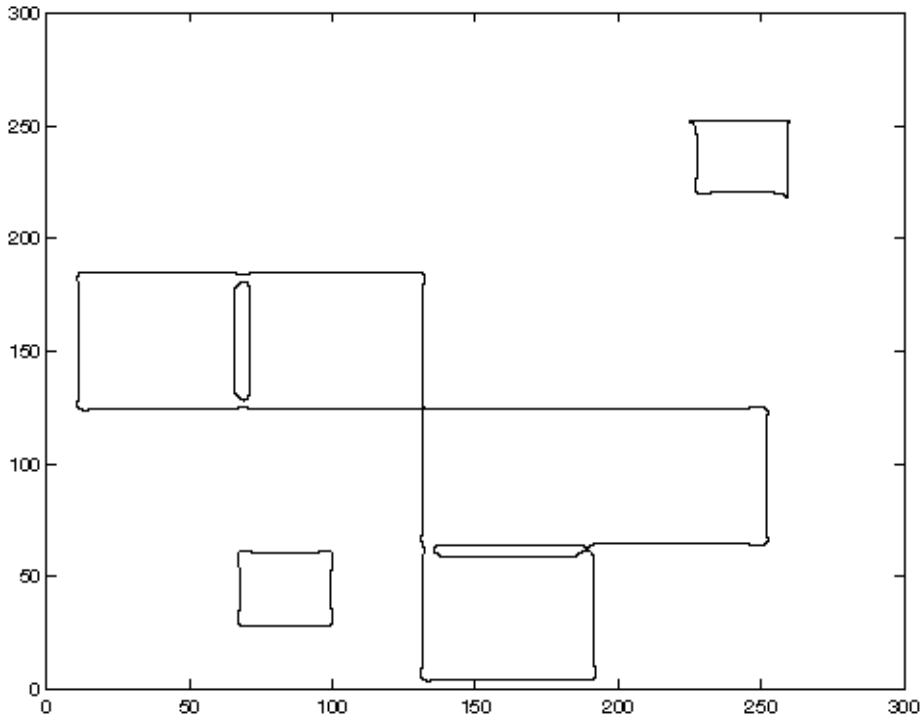
- The two functions should also be scaled to have the same value at the origin.

The LoG operator (cont.)

- LoG has fallen to some disfavour.
- It is not oriented and its response averages the responses across the two directions.
  - Poor response at corners.
  - Difficulty in recording the topology of T-junctions (trihedral vertices).
- Several studies showed that image components along an edge contribute to the response of LoG to noise but not to a true edge. Thus, zero crossings may not lie exactly on an edge.

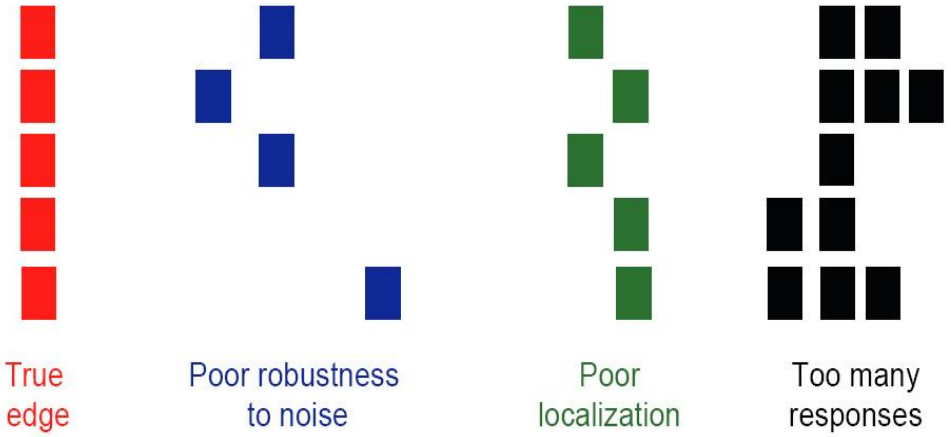
The LoG operator (cont.)

Poor corner and trihedral vertices detection



Designing an optimal edge detector

- Criteria for an “optimal” edge detector [Canny 1986]:
  - **Good detection:** the optimal detector must minimize the probability of false positives (detecting spurious edges caused by noise), as well as that of false negatives (missing real edges).
  - **Good localization:** the edges detected must be as close as possible to the true edges
  - **Single response:** the detector must return one point only for each true edge point; that is, minimize the number of local maxima around the true edge.



Canny edge detector

- Probably the most widely used edge detector.
- Theoretical model: step-edges corrupted by additive Gaussian noise.
- Canny has shown that the first derivative of the Gaussian closely approximates the operator that optimizes the product of *signal-to-noise ratio* and localization.

$$\frac{d}{\partial x} e^{-\frac{x^2}{2\sigma^2}} = -\frac{x}{\sigma^2} e^{-\frac{x^2}{2\sigma^2}}$$

Canny edge detector

- Generalization to 2D by applying the 1D operator in the direction of the edge normal.
- However, the direction of the edge normal is unknown beforehand and the 1D filter should be applied in all possible directions.
- This task may be approximated by smoothing the image with a circular 2D Gaussian, computing the gradient magnitude and use the gradient angle to estimate the edge strength.

Canny edge detector (cont.)

1. Filter image with derivative of Gaussian.
2. Find magnitude and orientation of gradient.
3. Non-maximum suppression:
   - Thin multi-pixel wide “ridges” down to single pixel width.
4. Linking and thresholding (hysteresis):
   - Define two thresholds: low and high
   - Use the high threshold to start edge curves and the low threshold to continue them.

Canny edge detector (cont.)

original image

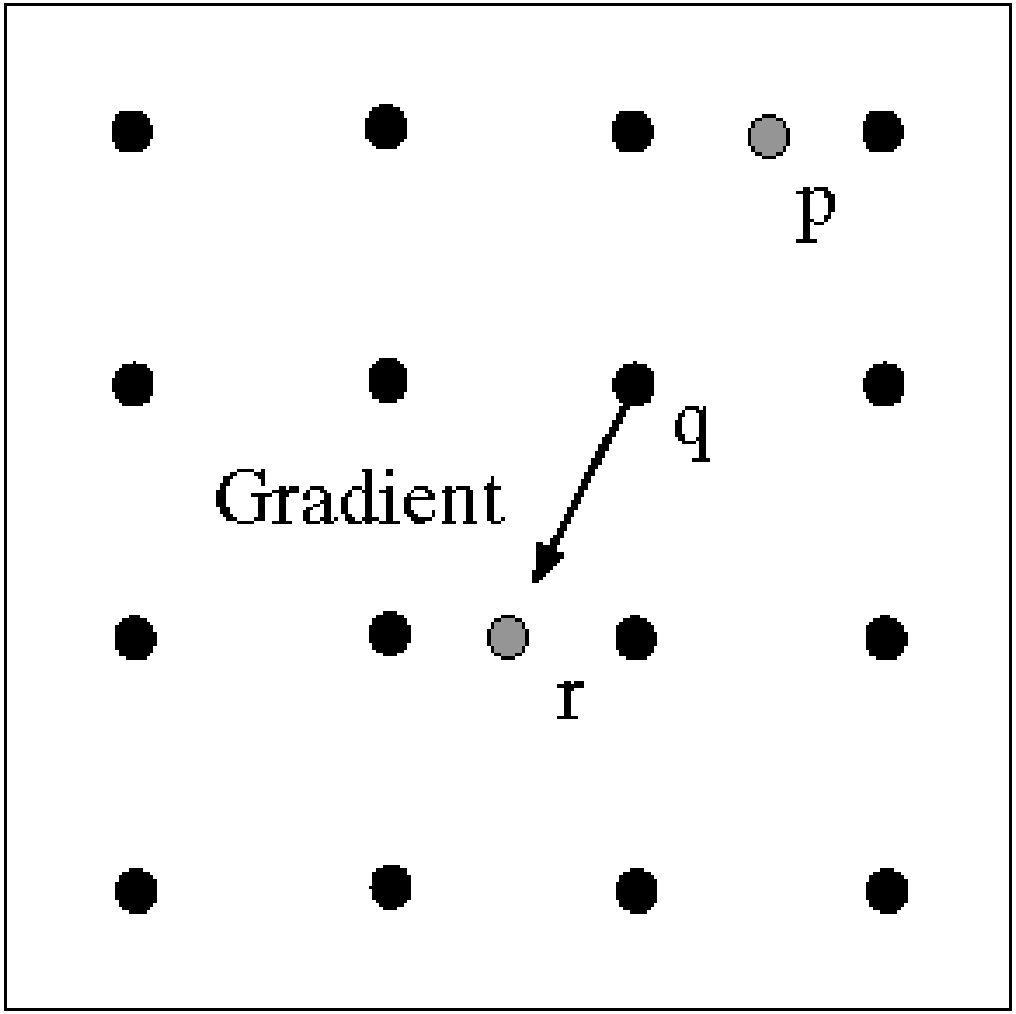
Canny edge detector (cont.)

Gradient magnitude

Canny edge detector (cont.)

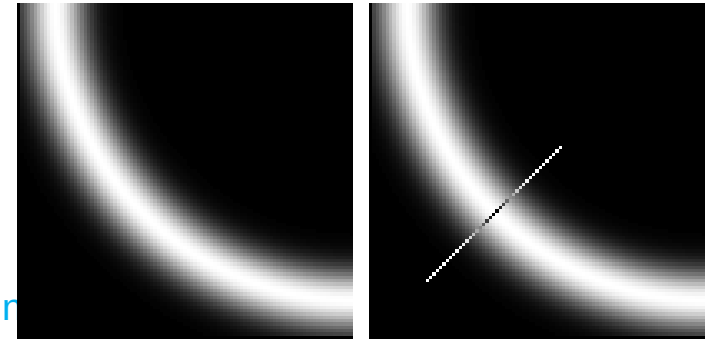
Non-maximum suppression and hysteresis thresholding

Non-maximum suppression (cont.)



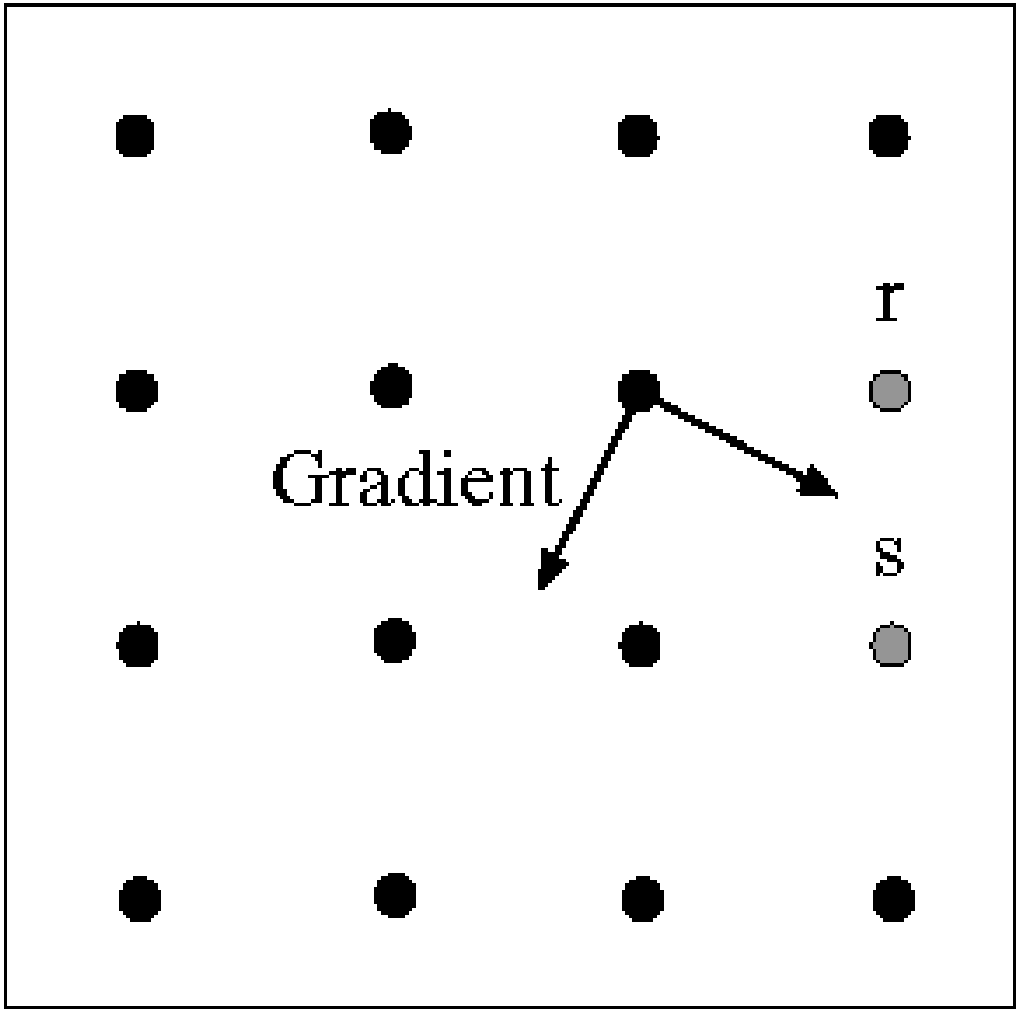
At pixel q, we have a maximum if the value of the gradient magnitude is larger than the values at both p and at r.

Interpolation provides these values.

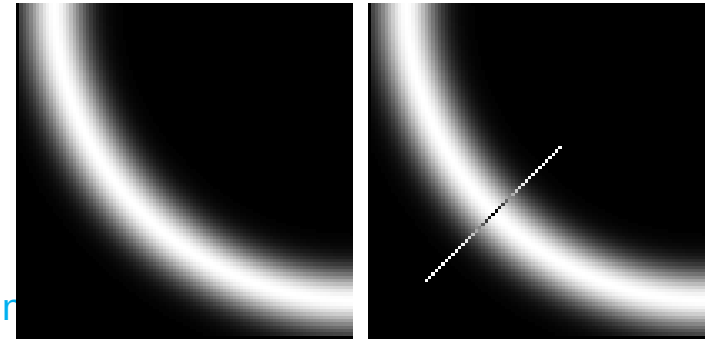


Source: D. Forsyth

Predict the next edge point



Assume the marked point is an edge point. Then we construct the tangent to the edge curve (which is normal to the gradient at that point) and use this to predict the next points (here either r or s).



Source: D. Forsyth

Hysteresis Thresholding

Reduces false edge pixels. It uses a low (T_L) and a high threshold (T_H) to create two additional images from the gradient magnitude image $g(x,y)$:

$$g_L(x,y) = \begin{cases} g(x,y) & g(x,y) \geq T_L \\ 0 & \text{otherwise} \end{cases}, \quad g_H(x,y) = \begin{cases} g(x,y) & g(x,y) \geq T_H \\ 0 & \text{otherwise} \end{cases}$$

$g_L(x,y)$ has more non zero pixels than $g_H(x,y)$.

We eliminate from $g_L(x,y)$ all the common non zero pixels:

$$g_L(x,y) = g_L(x,y) - g_H(x,y)$$

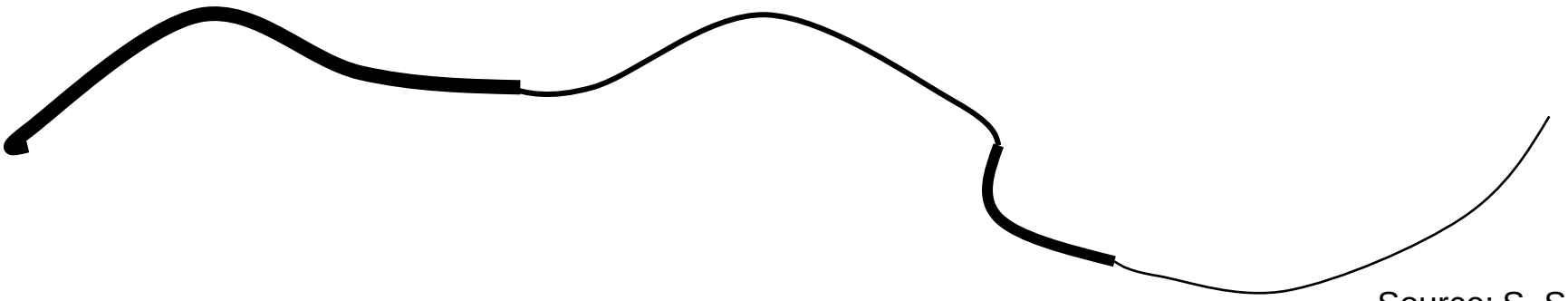
$g_L(x,y)$ and $g_H(x,y)$ may be viewed as *weak* and *strong* edge pixels.

Canny suggested a ratio of 2:1 to 3:1 between the thresholds.

Source: S. Seitz

Hysteresis Thresholding (cont.)

- After the thresholdings, all strong pixels are assumed to be valid edge pixels. Depending on the value of T_H, the edges in $g_H(x,y)$ typically have gaps.
- All pixels in $g_L(x,y)$ are considered valid edge pixels if they are 8-connected to a valid edge pixel in $g_H(x,y)$.



Source: S. Seitz

Hysteresis thresholding (cont.)

high threshold
(strong edges)

low threshold
(weak edges)

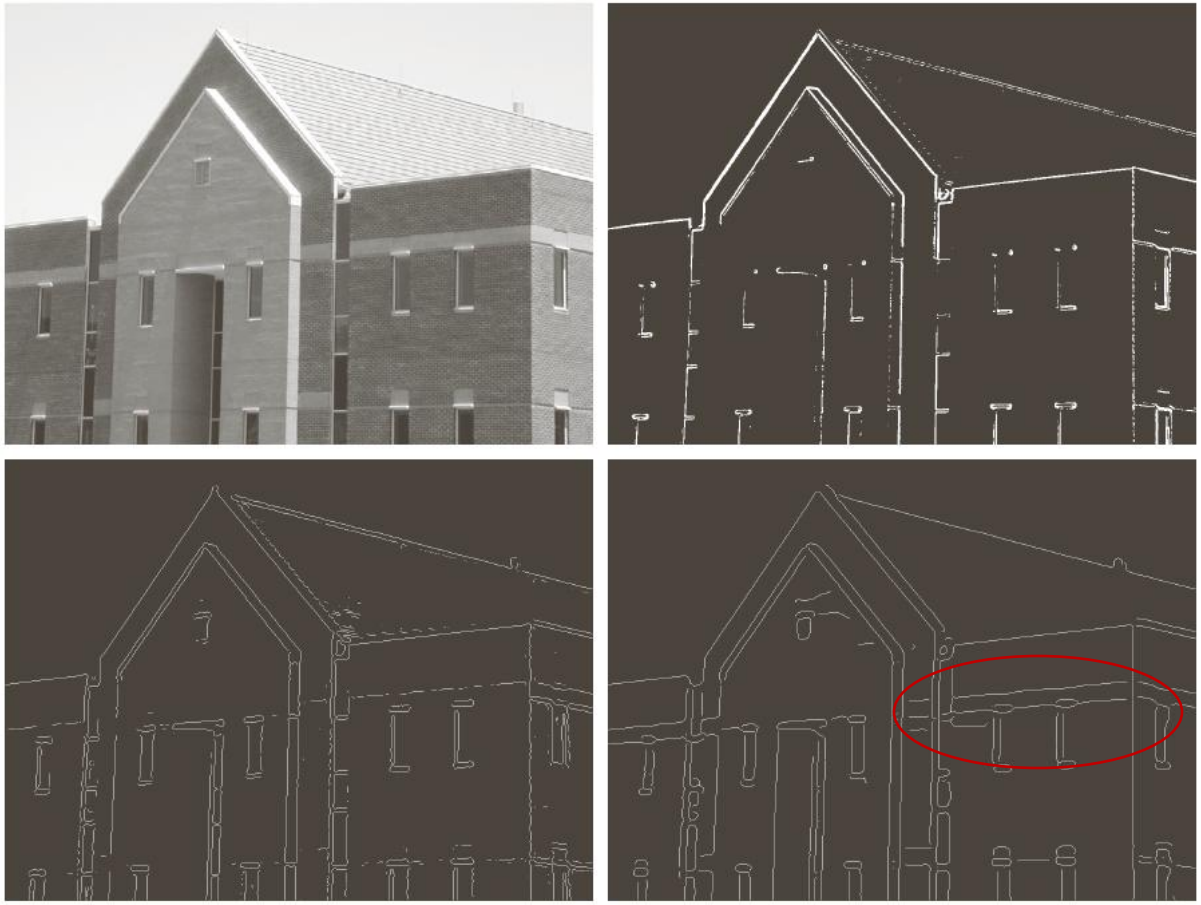
hysteresis threshold

Source: L. Fei-Fei

Canny vs LoG

Image

Thresholded gradient



LoG

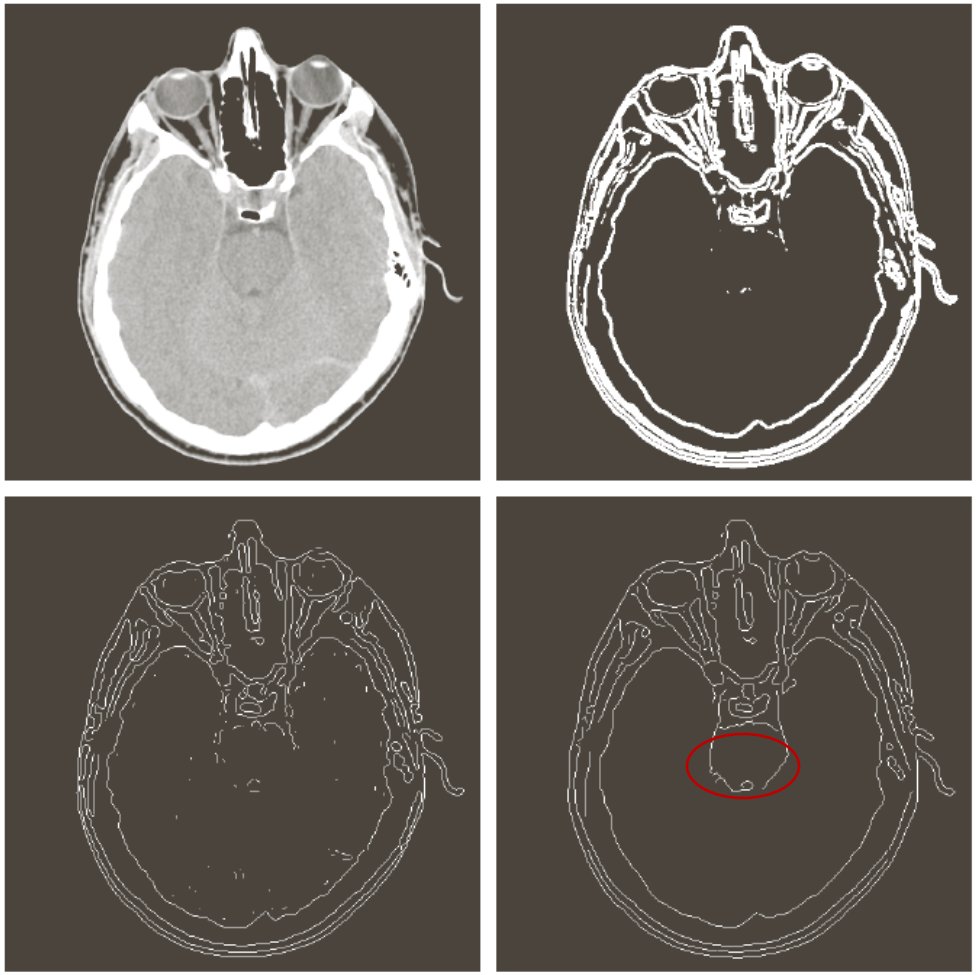
Canny

Both edges of the concrete band lining the bricks were preserved.

Canny vs LoG

Image

Thresholded gradient



LoG

Canny

The posterior boundary of the brain was preserved.

Edge Linking

- Even after hysteresis thresholding, the detected pixels do not completely characterize edges completely due to occlusions, non-uniform illumination and noise. Edge linking may be:
  - **Local**: requiring knowledge of edge points in a small neighborhood.
  - **Regional**: requiring knowledge of edge points on the boundary of a region.
  - **Global**: the Hough transform, involving the entire edge image.

Edge Linking by Local Processing

A simple algorithm:

1. Compute the gradient magnitude and angle arrays $M(x,y)$ and $a(x,y)$ of the input image $f(x,y)$.
2. Let S_{xy} denote the neighborhood of pixel (x,y).
3. A pixel (s,t) in S_{xy} in is linked to (x,y) if:

$$|M(x,y)-M(s,t)| \le E \text{ and } |a(x,y)-a(s,t)| \le A$$

A record of linked points must be kept as the center of S_{xy} moves.

Computationally expensive as all neighbors of every pixel should be examined.

Edge Linking by Local Processing (cont.)

A faster algorithm:

1. Compute the gradient magnitude and angle arrays $M(x,y)$ and $a(x,y)$ of the input image $f(x,y)$.
2. Form a binary image:

$$g(x,y)=\begin{cases}1 & M(x,y)\geq T_M \text{ and } a(x,y)\in[A-T_A, A+T_A] \\ 0 & \text{otherwise}\end{cases}$$

3. Scan the rows of $g(x,y)$ (for $A=0$) and fill (set to 1) all gaps (zeros) that do not exceed a specified length K.
4. To detect gaps in any other direction $A=\theta$, rotate $g(x,y)$ by θ and apply the horizontal scanning.

Edge Linking by Local Processing (cont.)

Image | Gradient magnitude | Horizontal linking

We may detect the licence plate from the ratio width/length (2:1 in the USA)

Vertical linking | Logical OR | Morphological thinning

Edge Linking by Regional Processing

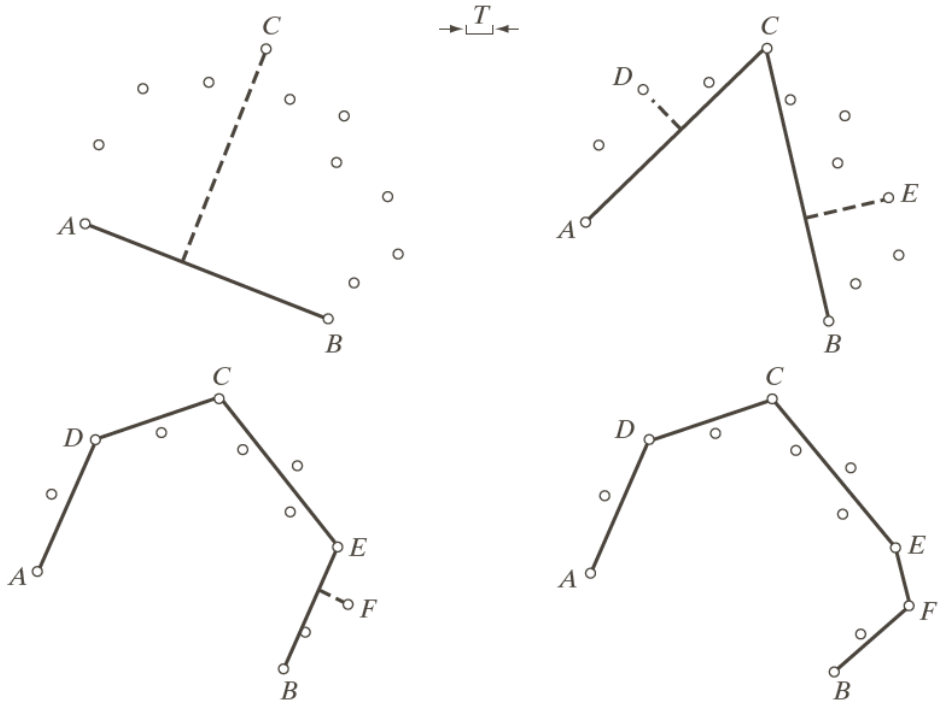
- Often, the location of regions of interest is known and pixel membership to regions is available.
- Approximation of the region boundary by fitting a polygon. Polygons are attractive because:
  - They capture the essential shape.
  - They keep the representation simple.
- Requirements
  - Two starting points must be specified (e.g. rightmost and leftmost points).
  - The points must be ordered (e.g. clockwise).

Edge Linking by Regional Processing (cont.)

- Variations of the algorithm handle both open and closed curves.
- If this is not provided, it may be determined by distance criteria:
  - Uniform separation between points indicate a closed curve.
  - A relatively large distance between consecutive points with respect to the distances between other points indicate an open curve.
- We present here the basic mechanism for polygon fitting.

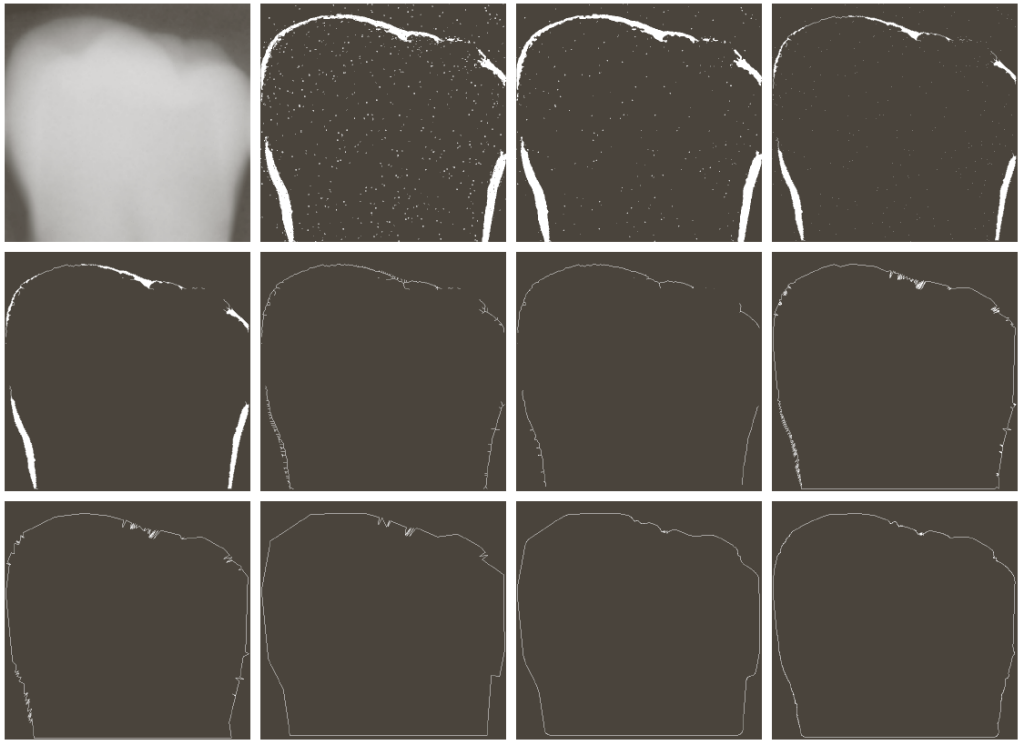
Edge Linking by Regional Processing (cont.)

- Given the end points A and B, compute the straight line AB.
- Compute the perpendicular distance from all other points to this line.
- If this distance exceeds a threshold, the corresponding point C having the maximum distance from AB is declared a vertex.
- Compute lines AC and CB and continue.



Edge Linking by Regional Processing (cont.)

Regional processing for edge linking is used in combination with other methods in a chain of processing.



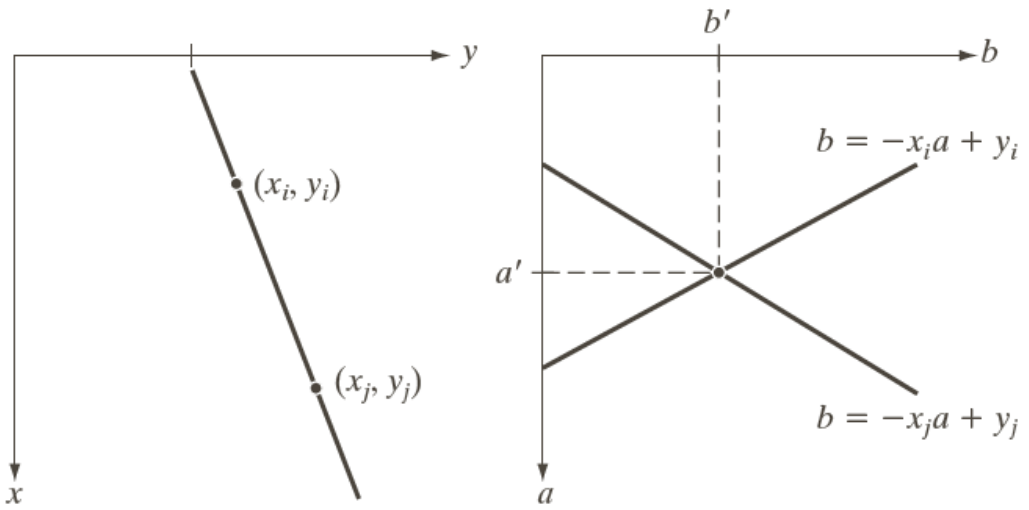
a b c d
e f g h
i j k l

FIGURE 10.30 (a) A 550 × 566 X-ray image of a human tooth. (b) Gradient image. (c) Result of majority filtering. (d) Result of morphological shrinking. (e) Result of morphological cleaning. (f) Skeleton. (g) Spur reduction. (h)–(j) Polygonal fit using thresholds of approximately 0.5%, 1%, and 2% of image width (T = 3, 6, and 12). (k) Boundary in (j) smoothed with a 1-D averaging filter of size 1 × 31 (approximately 5% of image width). (l) Boundary in (h) smoothed with the same filter.

Hough transform

- An early type of voting scheme.
- Each line is defined by two points (x_i,y_i) and (x_j,y_j).
- Each point (x_i,y_i) has a line parameter space (a,b) because it belongs to an infinite number of lines $y_i=ax_i+b$.
- All the points (x,y) on a line $y=a'x+b'$ have lines in parameter space that intersect at $(a'x+b')$.

Image space

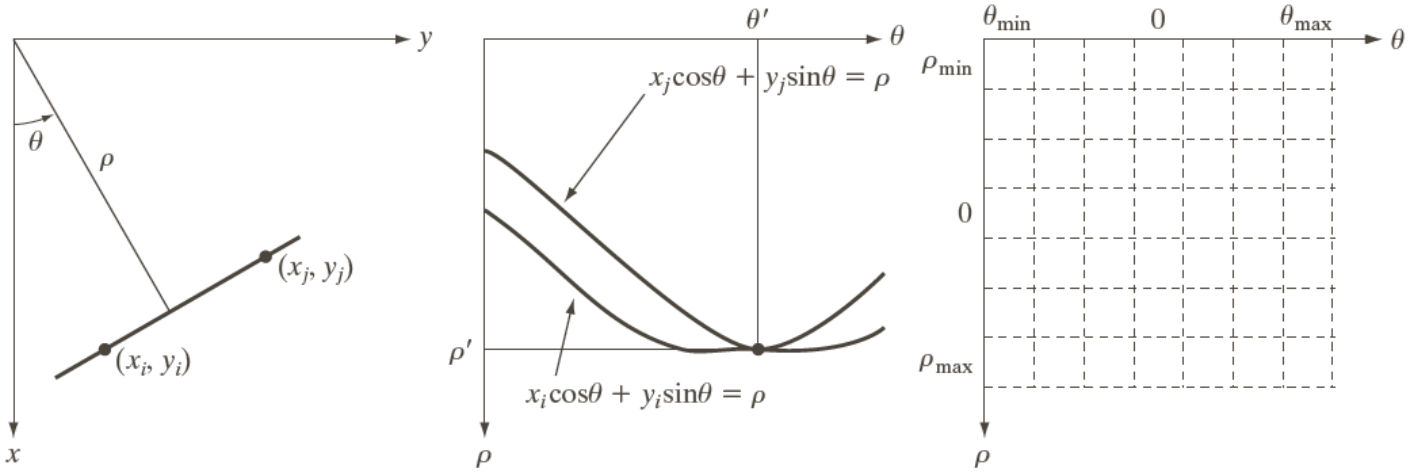


Parameter space

Hough transform (cont.)

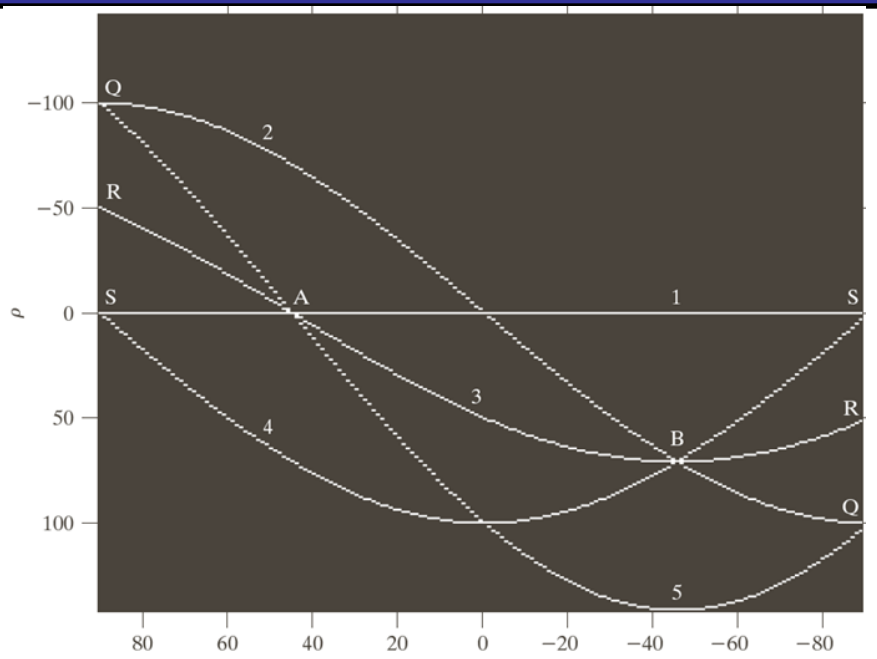
- Problems with the (a,b) parameter space:
  - Unbounded parameter domain
  - Vertical lines require infinite a.
- Polar (normal) representation of a line:

$$x\cos\theta + y\sin\theta = \rho$$



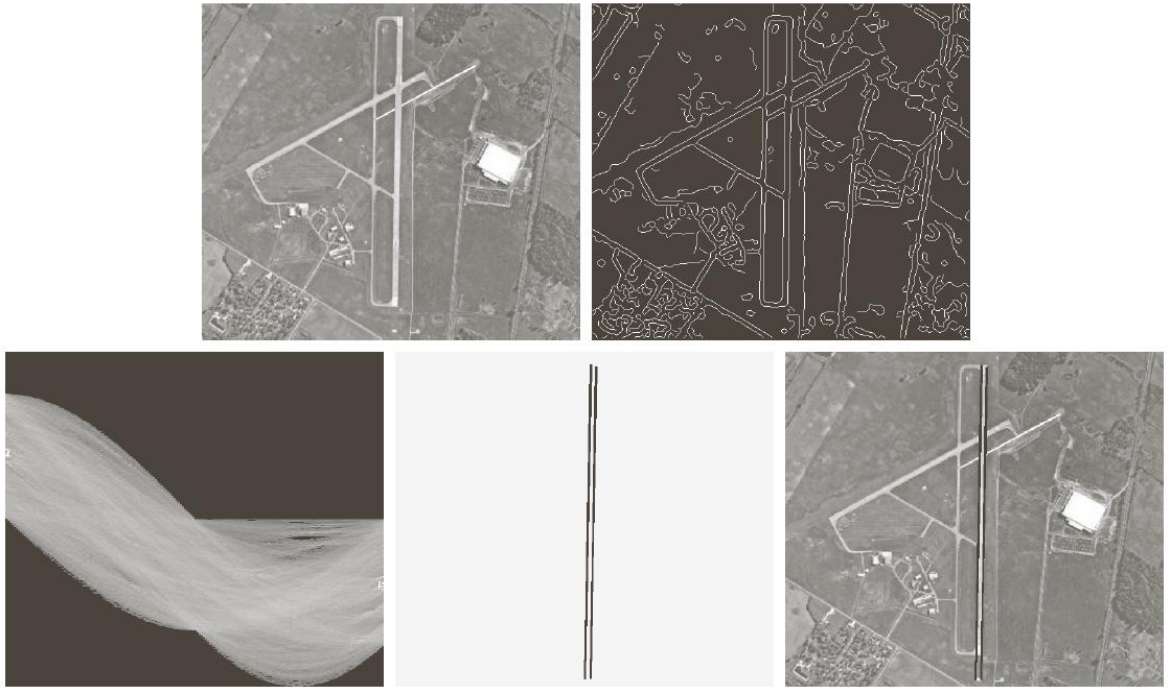
Accumulator array

Hough transform (cont.)



- A: intersection of curves corresponding to points 1, 3, 5.
- B: intersection of curves corresponding to points 2, 3, 4.
- Q, R and S show the reflective adjacency at the edges of the parameter space. They do not correspond to points.

Hough transform (cont.)



- We only know the orientation of the runway (around 0 deg) and the observer's position relative to it (GPS, flight charts etc.).
- We look for the peak at the accumulator array around 0 deg and join gaps below 20% of the image height.
- Applications in autonomous navigation.

Practical details

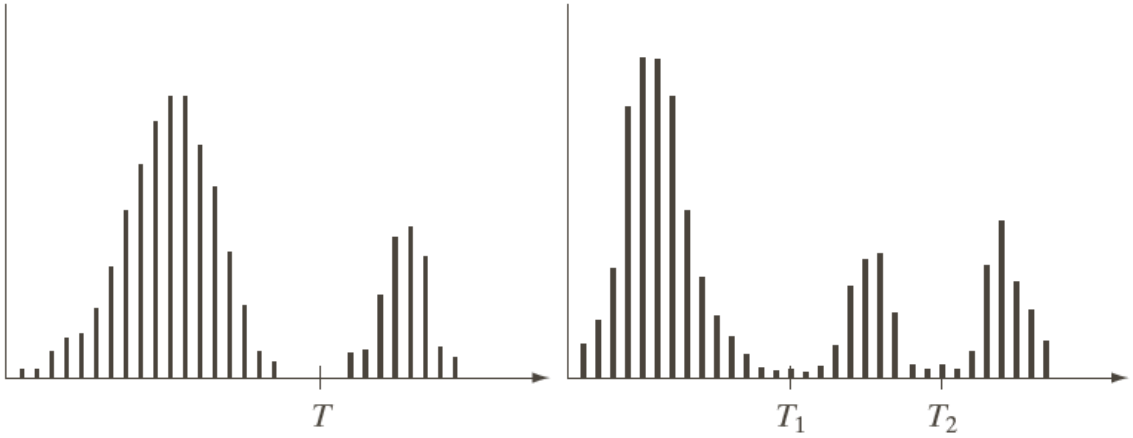
- Try to get rid of irrelevant features
  - Take only edge points with significant gradient magnitude.
- Choose a good grid / discretization
  - Too coarse: large votes obtained when too many different lines correspond to a single bucket.
  - Too fine: miss lines because some points that are not exactly collinear cast votes for different buckets.
- Increment also neighboring bins (smoothing in accumulator array).

Thresholding

- Image partitioning into regions directly from their intensity values.

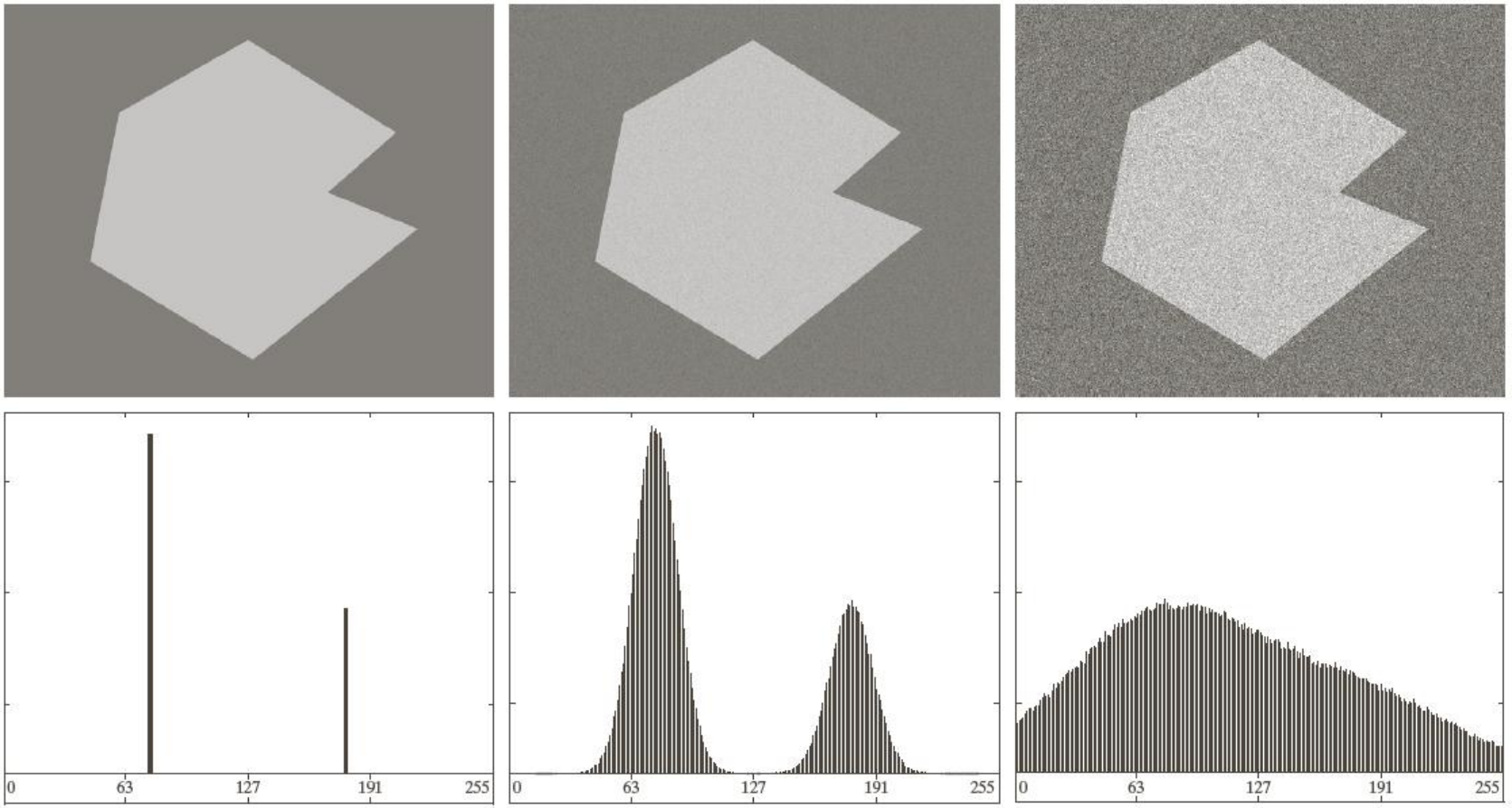
$$g(x,y)=\begin{cases}1 & \text{if } f(x,y)>T\\ 0 & \text{if } f(x,y)\le T\end{cases}$$

$$g(x,y)=\begin{cases}0 & \text{if } f(x,y)\le T_1\\ 1 & \text{if } T_1<f(x,y)\le T_2\\ 2 & \text{if } f(x,y)>T_2\end{cases}$$



Noise in Thresholding

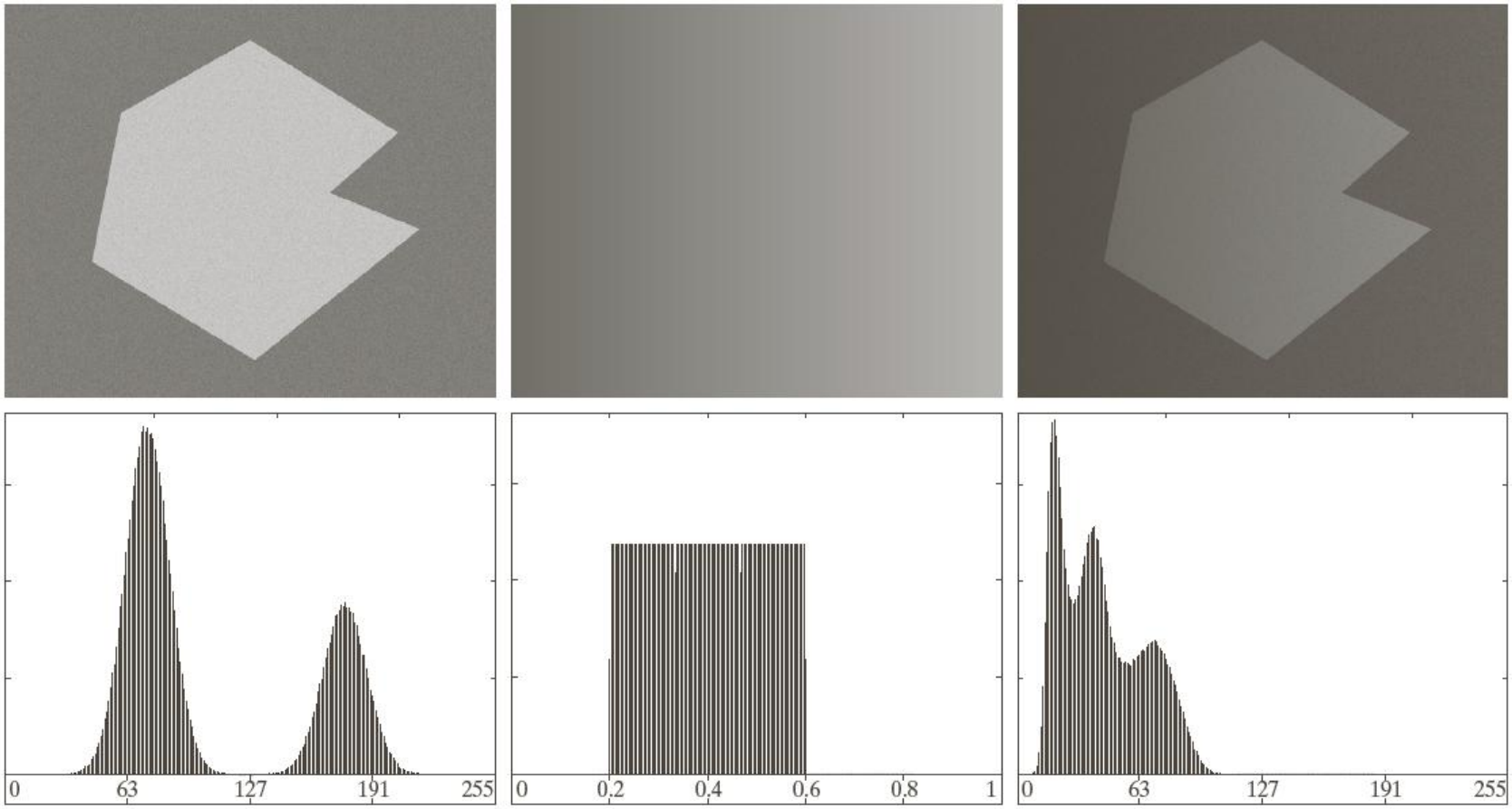
Difficulty in determining the threshold due to noise



Noiseless | Gaussian (μ=0, σ=10) | Gaussian (μ=0, σ=10)

Illumination in Thresholding

Difficulty in determining the threshold due to non-uniform illumination

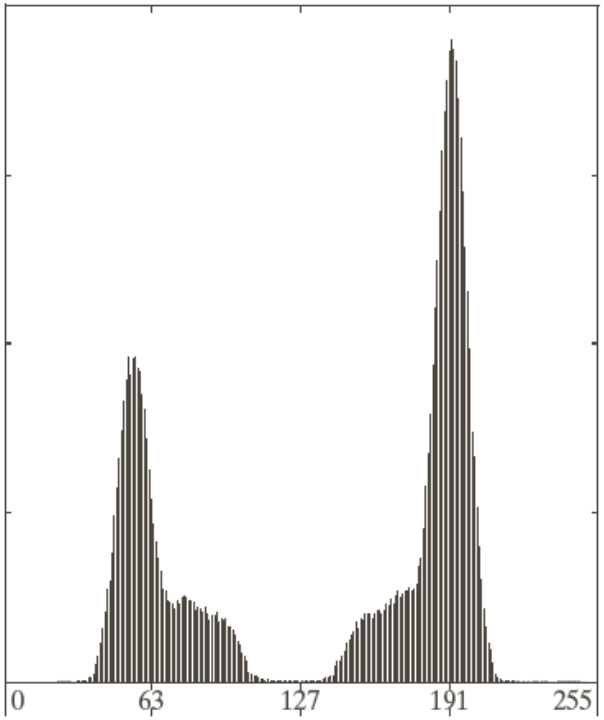


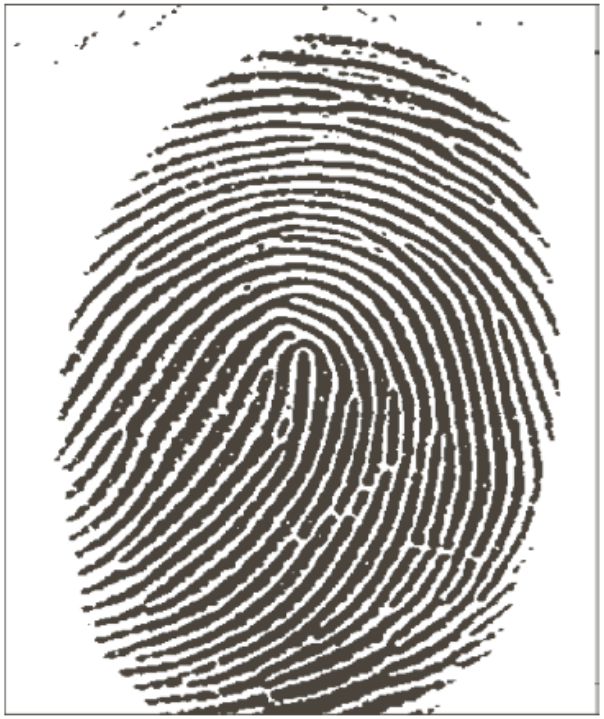
(a) Noisy image (b) Intensity ramp image Multiplication

Basic Global Thresholding

- **Algorithm**
  - Select initial threshold estimate T.
  - Segment the image using T
    - Region G_1 (values $> T$) and region G_2 (values $< T$).
  - Compute the average intensities m_1 and m_2 of regions G_1 and G_2 respectively.
  - Set $T=(m_1+m_2)/2$
  - Repeat until the change of T in successive iterations is less than ΔT.

Basic Global Thresholding (cont.)





T=125

Optimum Global Thresholding using Otsu's Method

- The method is based on statistical decision theory.
- Minimization of the average error of assignment of pixels to two (or more) classes.
- Bayes decision rule may have a nice closed form solution to this problem provided
  - The pdf of each class.
  - The probability of class occurence.
- Pdf estimation is not trivial and assumptions are made (Gaussian pdfs).
- Otsu (1979) proposed an attractive alternative maximizing the *between-class variance*.
  - Only the histogram of the image is used.

Otsu's Method (cont.)

- Let $\{0, 1, 2,\ldots, L\text{-}1\}$ denote the intensities of a MxN image and n_i the number of pixels with intensity i.
- The normalized histogram has components:

$$p_i = \frac{n_i}{MN}, \quad \sum_{i=0}^{L-1} p_i = 1, \quad p_i \geq 0$$

- Suppose we choose a threshold k to segment the image into two classes:
  - C_1 with intensities in the range $[0, k]$,
  - C_2 with intensities in the range $[k\text{+}1, L\text{-}1]$.

Otsu's Method (cont.)

- The probabilities of classes C_1 and C_2:

$$P_1(k) = \sum_{i=0}^{k} p_i, \quad P_2(k) = \sum_{i=k+1}^{L-1} p_i = 1 - P_1(k)$$

- The mean intensity of class C_1 :

$$m_1(k) = \sum_{i=0}^{k} iP(i \mid C_1) = \sum_{i=0}^{k} i \frac{P(C_1 \mid i)P(i)}{P(C_1)} = \frac{1}{P_1(k)} \sum_{i=0}^{k} ip_i$$

Intensity i belongs to class C_1 and $P(C_1 \mid i) = 1$

Otsu's Method (cont.)

- Similarly, the mean intensity of class C_2:

$$m_2(k) = \sum_{i=k+1}^{L-1} iP(i \mid C_2) = \sum_{i=k+1}^{L-1} i \frac{P(C_2 \mid i)P(i)}{P(C_2)} = \frac{1}{P_2(k)} \sum_{i=k+1}^{L-1} ip_i$$

- Mean image intensity:

$$m_G = \sum_{i=0}^{L-1} ip_i = P_1(k)m_1(k) + P_2(k)m_2(k)$$

- Cumulative mean image intensity (up to k):

$$m(k) = \sum_{i=0}^{k} ip_i$$

Otsu's Method (cont.)

- Between class variance:

$$\sigma_B^2(k) = P_1(k)[m_1(k) - m_G]^2 + P_2(k)[m_2(k) - m_G]^2$$

- With some manipulation we get:

$$\sigma_B^2(k) = \frac{[m_G P_1(k) - m(k)]^2}{P_1(k)[1 - P_1(k)]}$$

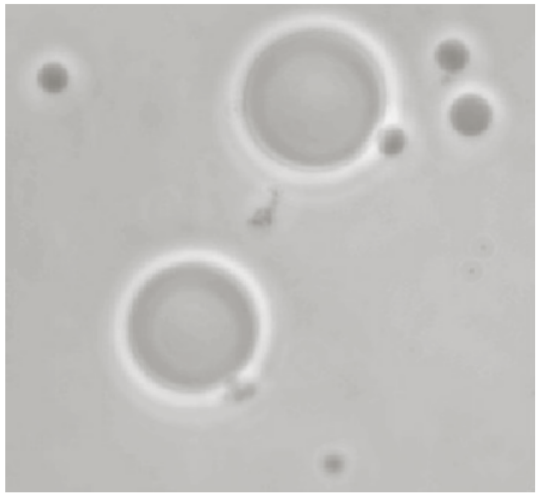
- The value of k is selected by sequential search as the one maximazing:

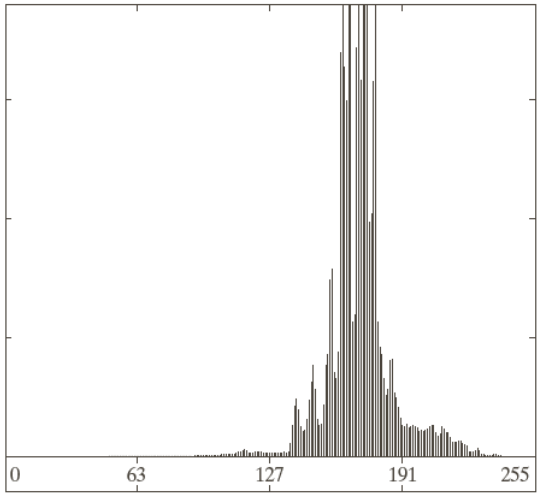
$$k^* = \max_{0 \le k \le L-1} \{\sigma_B^2(k)\}$$

Otsu's Method (cont.)

Example (no obvious valley in the histogram)

Image





Histogram

Basic method

Otsu's method

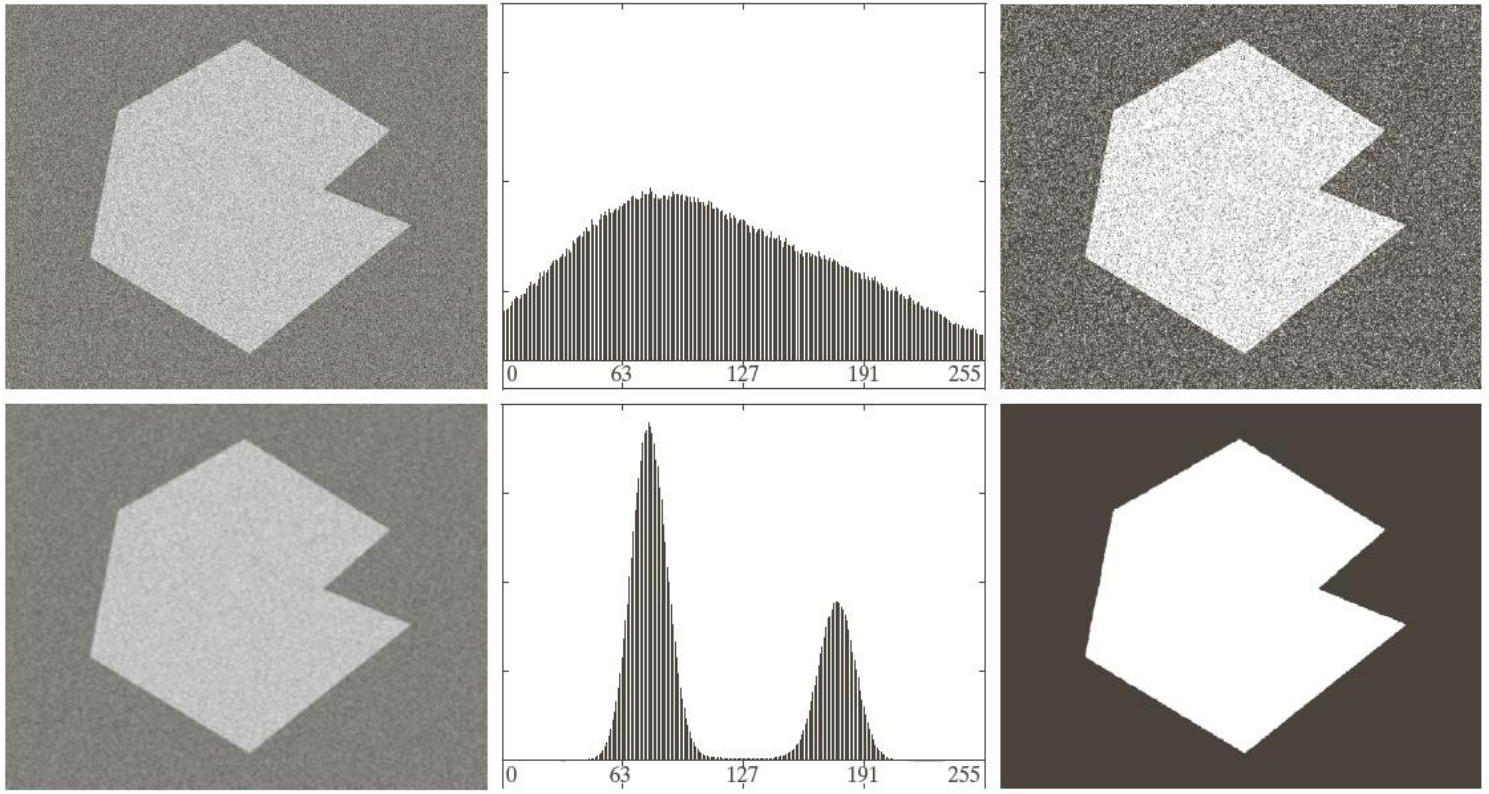
Otsu's Method (cont.)

Smoothing helps thresholding (may create histogram valley)

Noisy image

Histogram

Otsu no smoothing



Noisy image smoothed

Histogram

Otsu with smoothing

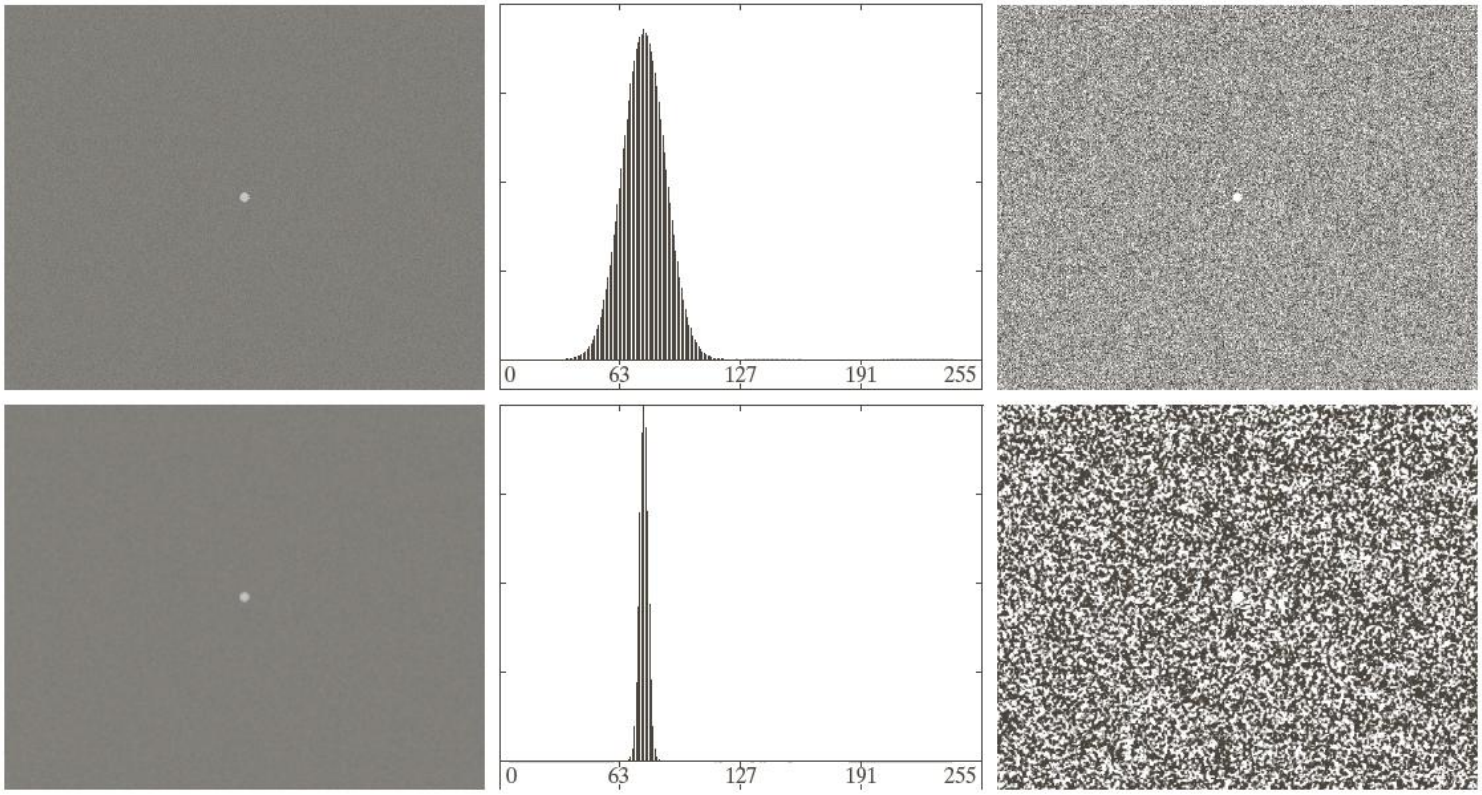
Otsu's Method (cont.)

Otsu's method, even with smoothing cannot extract small structures

Noisy image, σ=10

Histogram

Otsu no smoothing



Noisy image smoothed

Histogram

Otsu with smoothing

Otsu's Method (cont.)

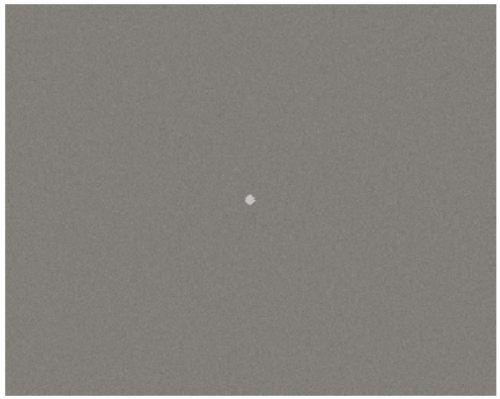
- Dealing with small structures
  - The histogram is unbalanced
    - The background dominates
    - No valleys indicating the small structure.
  - Idea: consider only the pixels around edges
    - Both structure of interest and background are present equally.
    - More balanced histograms
  - Use gradient magnitude or zero crossings of the Laplacian.
    - Threshold it at a percentage of the maximum value.
    - Use it as a mask on the original image.
    - Only pixels around edges will be employed.

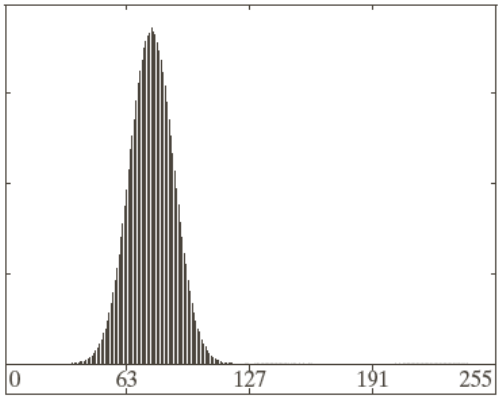
Otsu's Method (cont.)

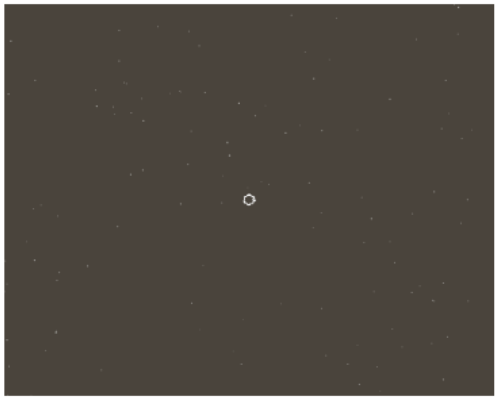
Noisy image, σ=10

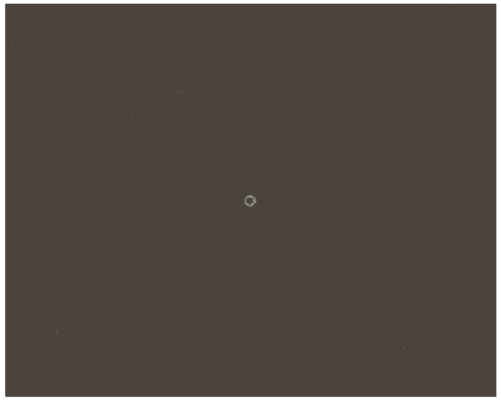
Histogram

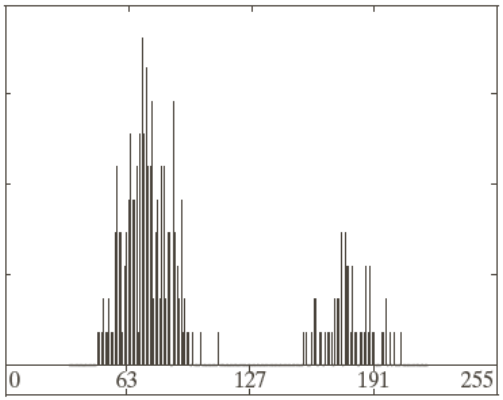
Gradient magnitude mask (at 99.7% of max)

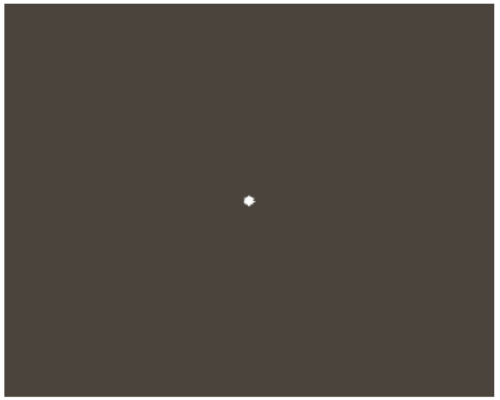












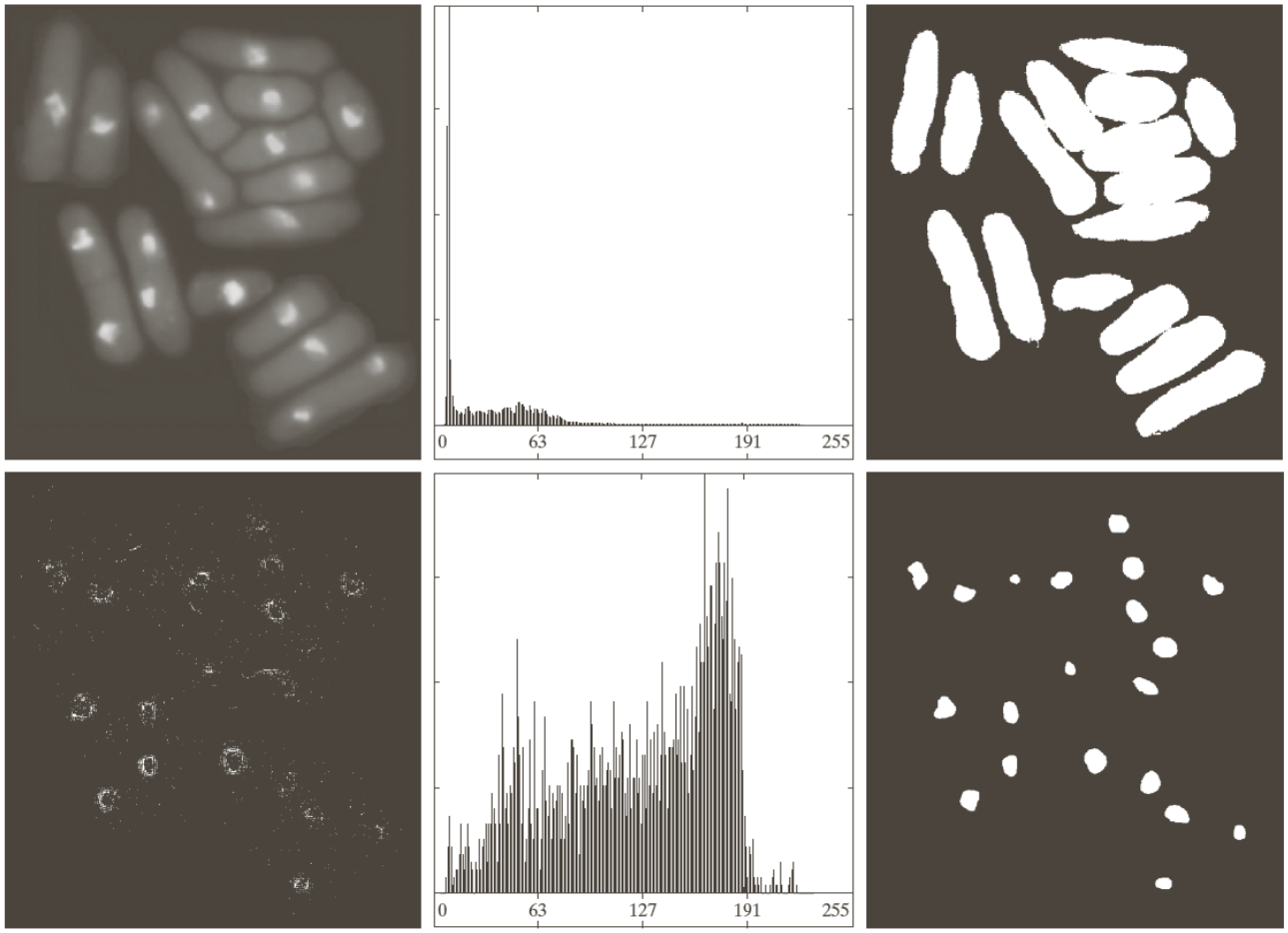
Multiplication (Mask x Image)

Histogram

Otsu

Otsu's Method (cont.)

Image | Histogram | Otsu on original histogram



We wish to extract the bright spots (nuclei) of the cells

Thresholded absolute Laplacian | Histogram | Otsu

Otsu's Method (cont.)

- The method may be extended to multiple thresholds
- In practice for more than 2 thresholds (3 segments) more advanced methods are employed.
- For three classes, the between-class variance is:

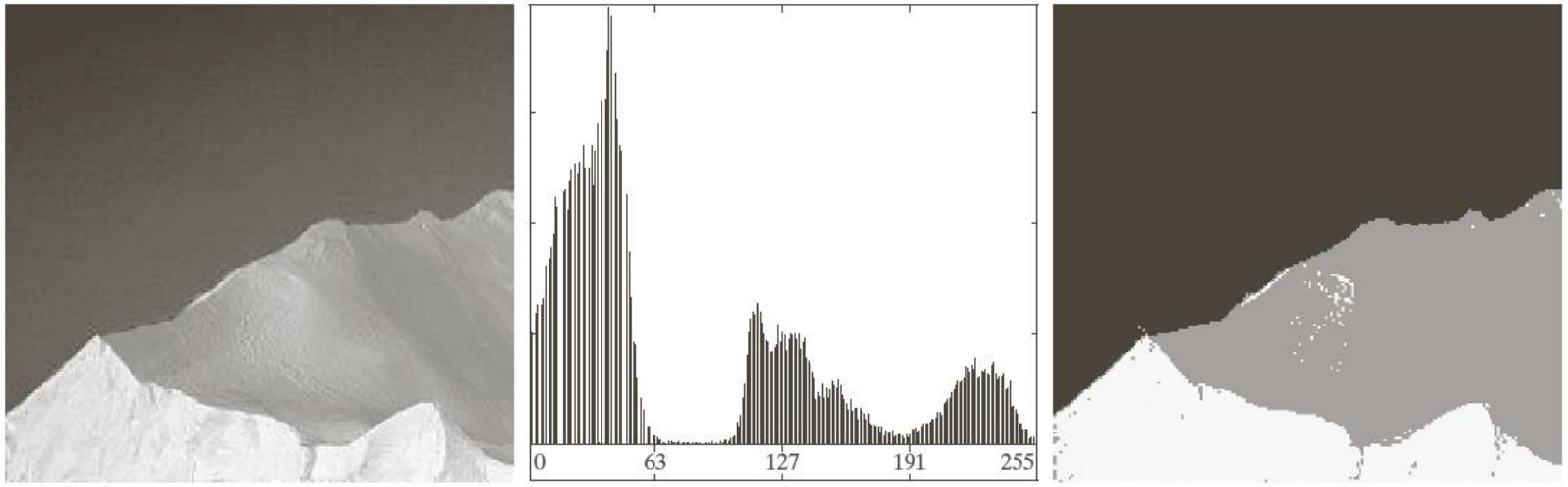
$$\sigma_B^2(k_1,k_2) = P_1(k_1)[m_1(k_1) - m_G] + P_2(k_1,k_2)[m_2(k_1,k_2) - m_G] + P_3(k_2)[m_3(k_2) - m_G]$$

- The thresholds are computed by searching all pairs for values:

$$(k_1^*, k_2^*) = \max_{0 \le k_1 < k_2 \le L-1} \{\sigma_B^2(k_1,k_2)\}$$

Otsu's Method (cont.)

Image of iceberg segmented into three regions.



k_1=80, k_2=177

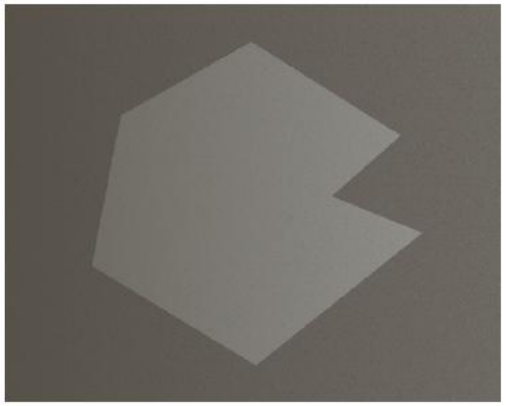
Variable Thresholding

Image partitioning.

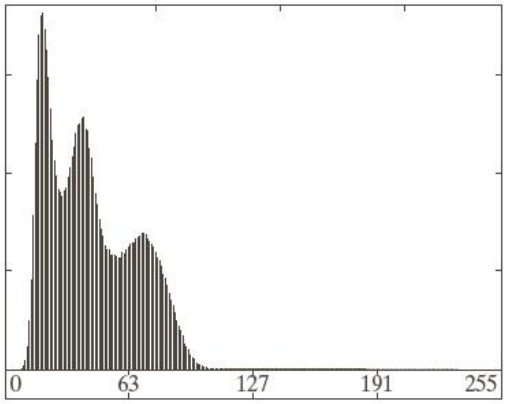
The image is sub-divided and the method is applied to every sub-image.

Useful for illumination non-uniformity correction.

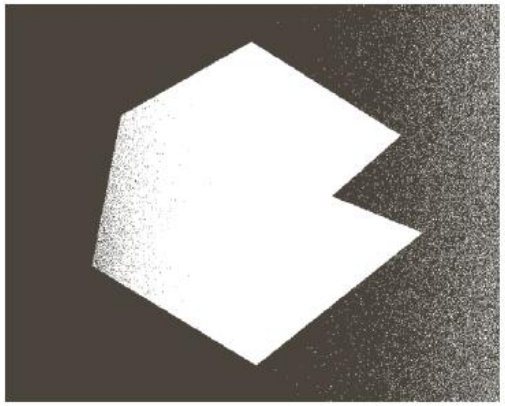
Shaded image

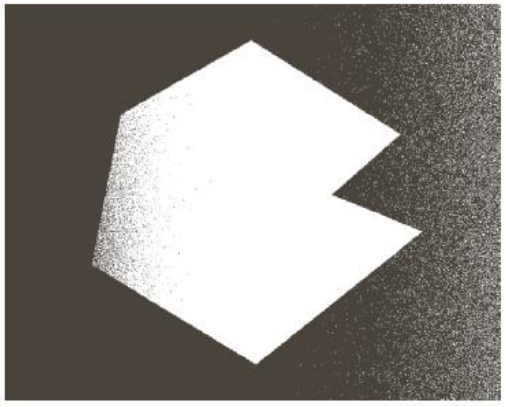


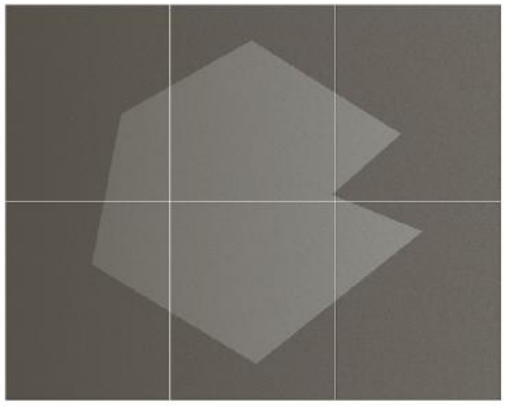
Histogram

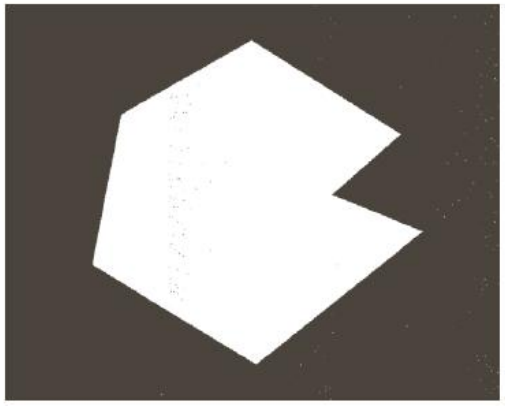


Simple thresholding









Otsu

Subdivision

Otsu at each sub-image

Variable Thresholding(cont.)

Use of local image properties.

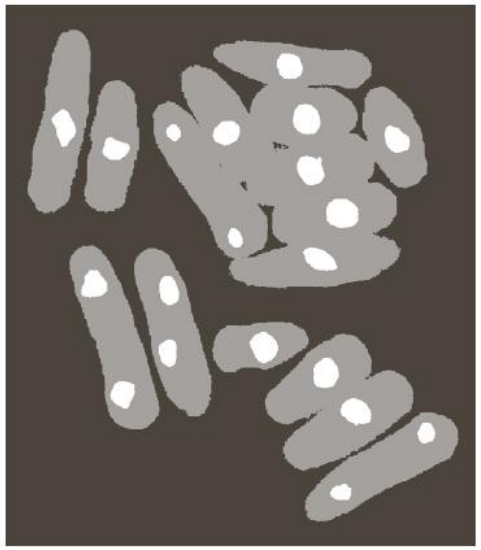
- Compute a threshold for every single pixel in the image based on its neighborhood (m_{xy}, σ_{xy},…).

$$g(x,y) = \begin{cases} 1 & Q(\text{local properties}) \text{ is true} \\ 0 & Q(\text{local properties}) \text{ is false} \end{cases}$$

$$Q(\sigma_{xy}, m_{xy}) = \begin{cases} \text{true} & f(x,y) > a\sigma_{xy} \;\; AND \;\; f(x,y) > bm_{xy} \\ \text{false} & \text{otherwise} \end{cases}$$

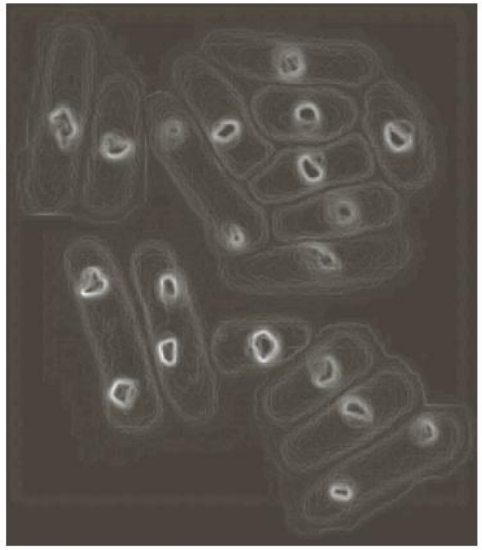
Variable Thresholding (cont.)

Image



Otsu with
2 thresholds

Image of local
standard deviations



Local
Thresholding

More accurate
nuclei extraction.

Variable Thresholding (cont.)

Moving averages.

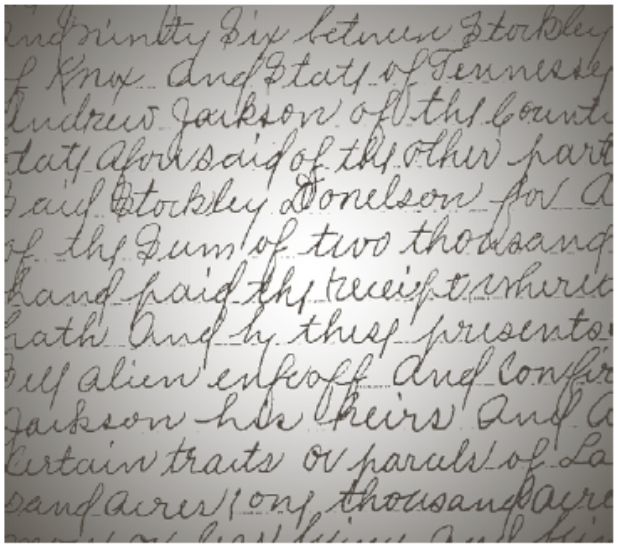
- Generally used along lines, columns or in zigzag .
- Useful in document image processing.
- Let z_{k+1} be the intensity of a pixel in the scanning sequence at step $k+1$. The moving average (mean intensity) at this point is:

$$m(k+1) = \frac{1}{n}\sum_{i=k+2-n}^{k+1} z_i = m(k) + \frac{1}{n}(z_{k+1} - z_{k-n})$$

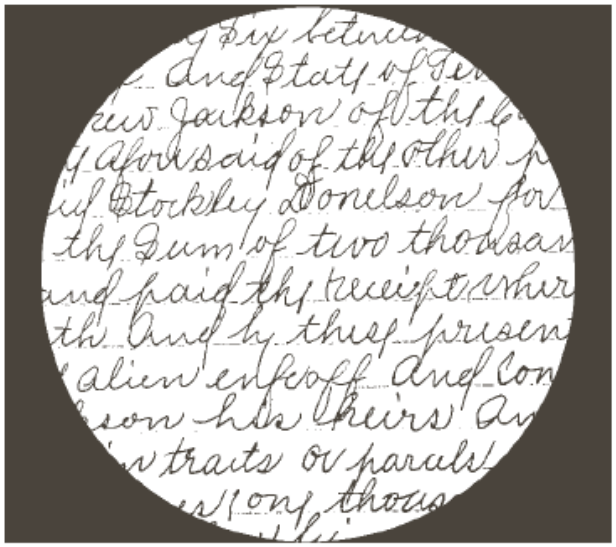
n is the number of points used in the average

- Segmentation is then performed at each point comparing the pixel value to a fraction of the moving average.

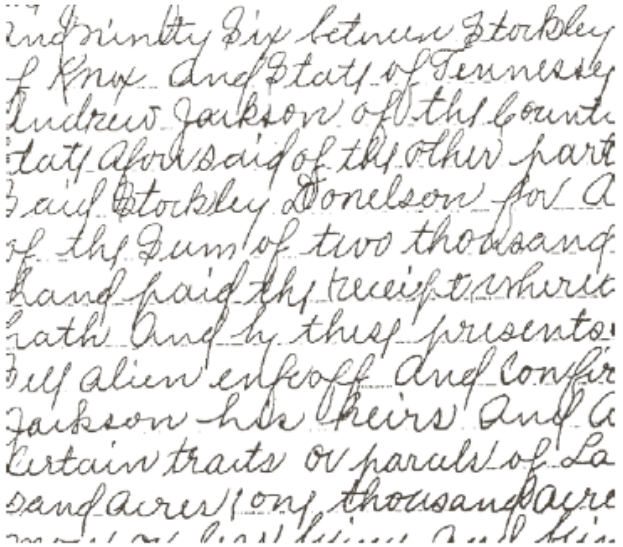
Variable Thresholding (cont.)



Shaded text images occur typically from photographic flashes

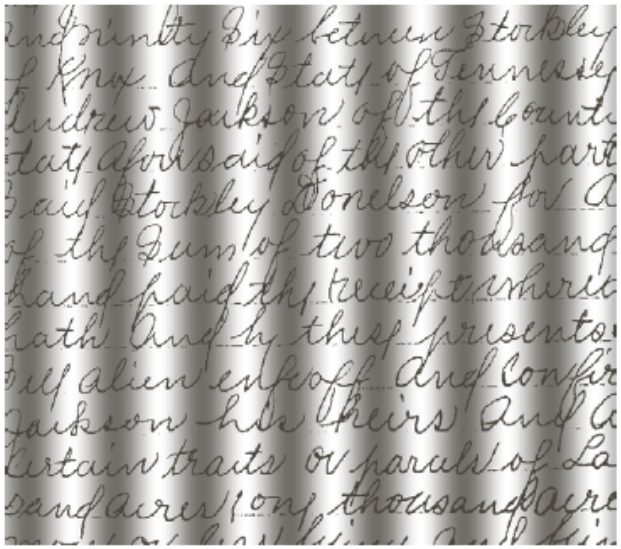


Otsu



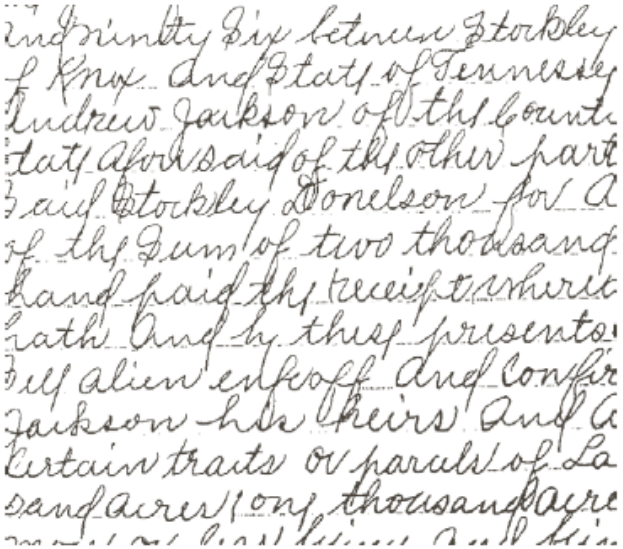
Moving averages

Variable Thresholding (cont.)



Sinusoidal variations (the power supply of the scanner not grounded properly)

Otsu



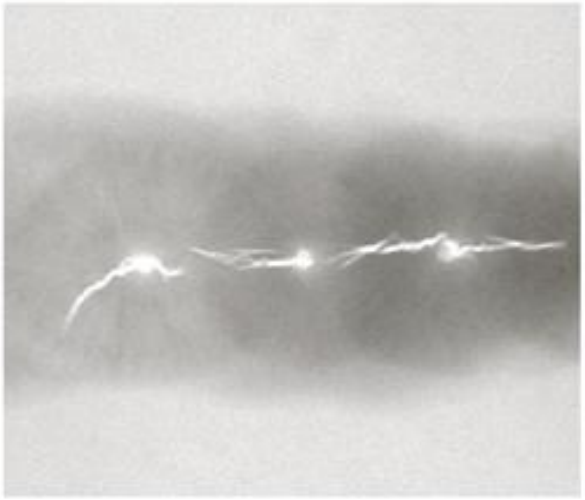
Moving averages

The moving average works well when the structure of interest is small with respect to image size (handwritten text).

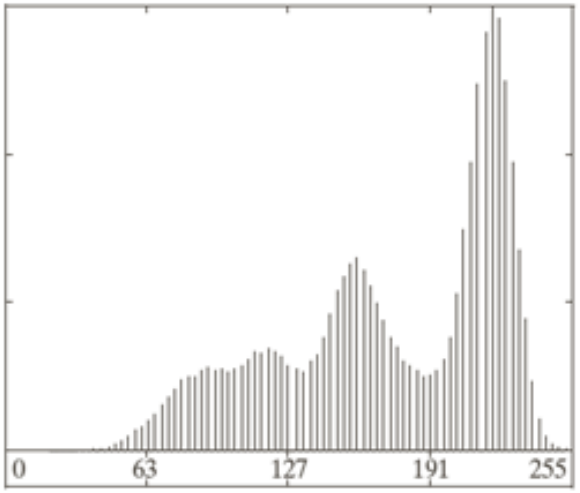
Region Growing

- Start with seed points $S(x,y)$ and grow to larger regions satisfying a predicate.
- Needs a stopping rule.
- **Algorithm**
  - Find all connected components in $S(x,y)$ and erode them to 1 pixel.
  - Form image $f_q(x,y)=1$ if $f(x,y)$ satisfies the predicate.
  - Form image $g(x,y)=1$ for all pixels in $f_q(x,y)$ that are 8-connected with to any seed point in $S(x,y)$.
  - Label each connected component in $g(x,y)$ with a different label.

Region Growing (cont.)



X-Ray image of weld with a crack we want to segment.



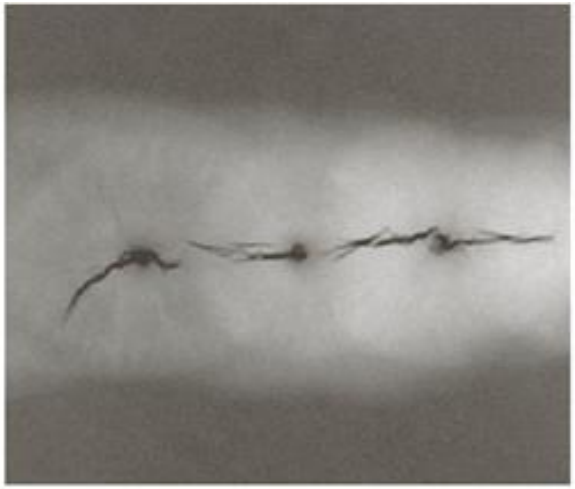
Histogram

Seed image (99.9% of max value in the initial image).
Crack pixels are missing.

The weld is very bright. The predicate used for region growing is to compare the absolute difference between a seed point and a pixel to a threshold. If the difference is below it we accept the pixel as crack.

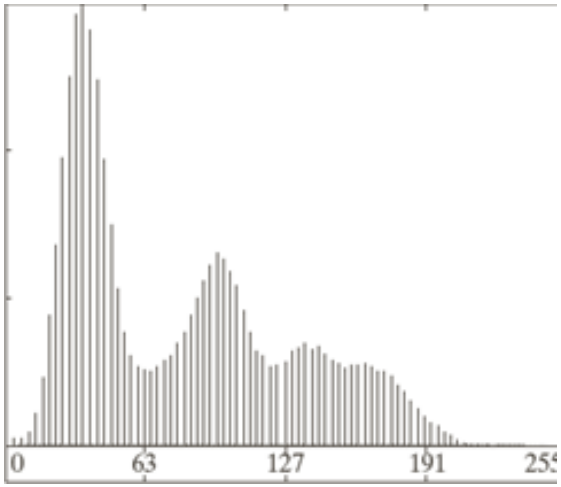
Region Growing (cont.)

Seed image eroded to 1 pixel regions.



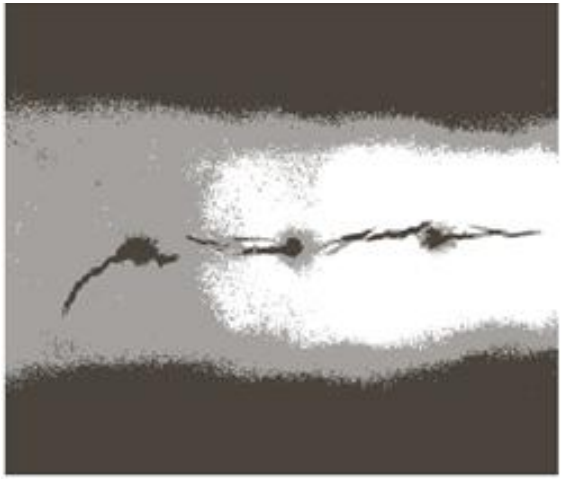
Difference between the original image and the initial seed image.

The pixels are ready to be compared to the threshold.



Histogram of the difference image. Two valleys at 68 and 126 provided by Otsu.

Region Growing (cont.)



Otsu thresholding of the difference image to 3 regions (2 thresholds).

Thresholding of the difference image with the lowest of the dual thresholds.

Notice that the background is also considered as crack.

Segmentation result by region growing.

The background is not considered as crack.
It is removed as it is not 8-connected to the seed pixels.

Region Splitting and Merging

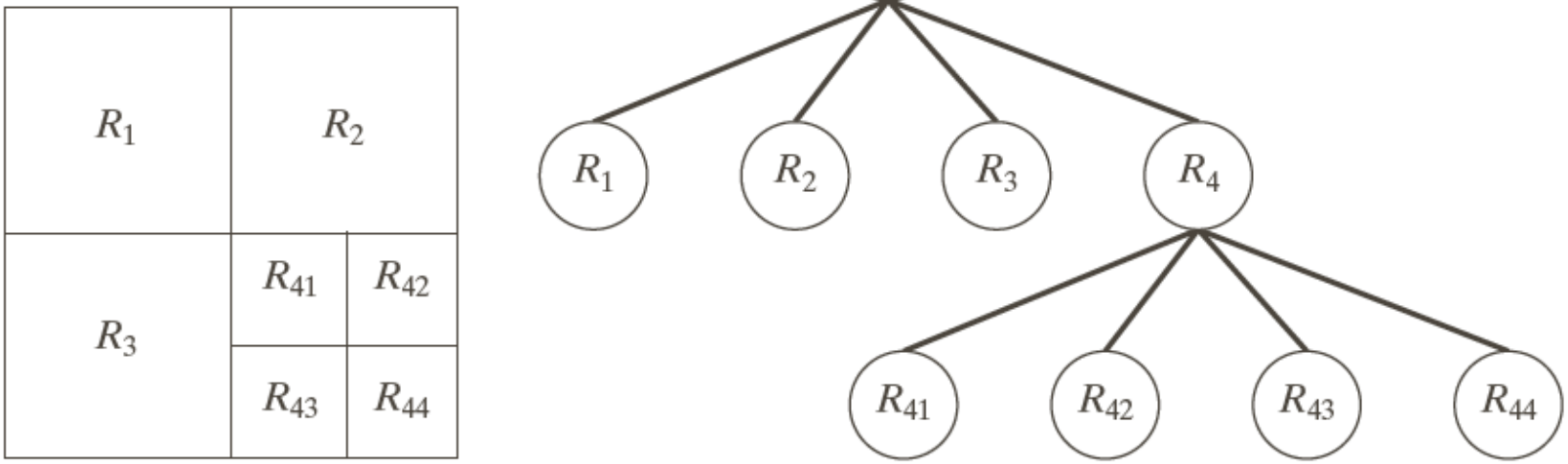
- Based on quadtrees (*quadimages*).
- The root of the tree corresponds to the image.
- Split the image to sub-images that do not satisfy a predicate Q.
- If only splitting was used, the final partition would contain adjacent regions with identical properties.
- A merging step follows merging regions satisfying the predicate Q.

Region Splitting and Merging (cont.)

- **Algorithm**
  - Split into four disjoint quadrants any region R_i for which $Q(R_i)$=FALSE.
  - When no further splitting is possible, merge any adjacent regions R_i and R_k for which $Q(R_i \cup R_k)$=TRUE.
  - Stop when no further merging is possible.
- A maximum quadregion size is specified beyond which no further splitting is carried out.
- Many variations heve been proposed.
  - Merge any adjacent regions if each one satisfies the predicate individually (even if their union does not satisfy it).

Region Splitting and Merging (cont.)

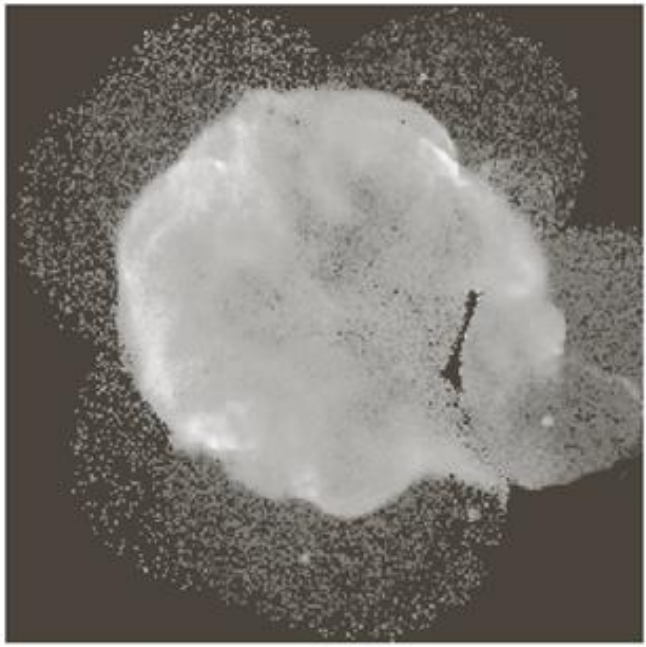
Quadregions resulted from splitting.



Merging examples:

- R_2 may be merged with R_{41}.
- R_{41} may be merged with R_{42}.

Region Splitting and Merging (cont.)



- Image of the Cygnus Loop. We want to segment the outer ring of less dense matter.
- Characteristics of the region of interest:
  - Standard deviation grater than the background (which is near zero) and the central region (which is smoother).
  - Mean value greater than the mean of background and less than the mean of the central region.
  - Predicate:

$$Q = \begin{cases} \text{true} & \sigma > \alpha \; AND \;\; 0 < m < b \\ \text{false} & \text{otherwise} \end{cases}$$

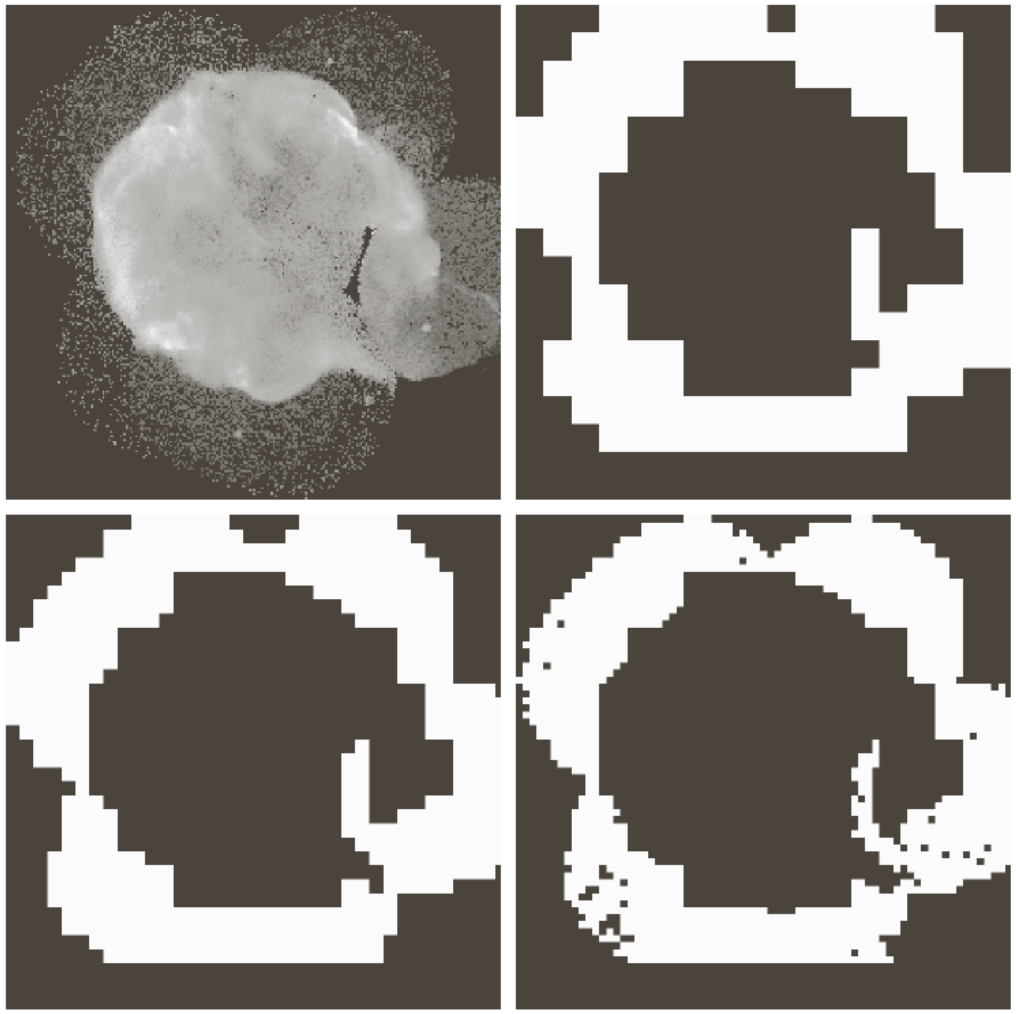
Region Splitting and Merging (cont.)

Varying the size of the smallest allowed *quadregion*.

Larger *quadregions* lead to block-like segmentation.

Smaller *quadregions* lead to small black regions.

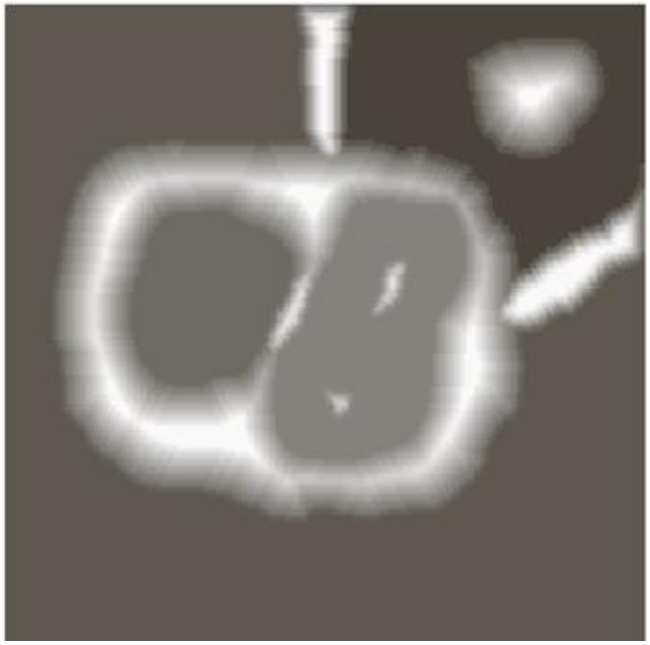
16x16 seems to be the best result.



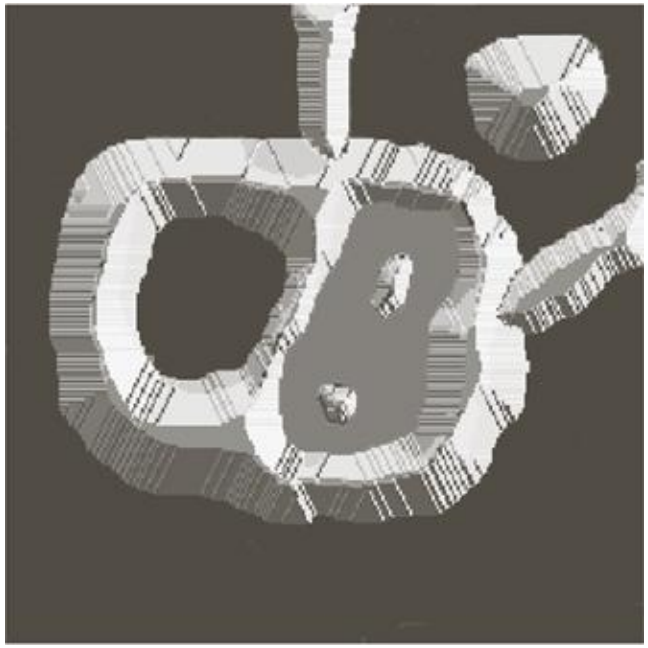
Morphological Watersheds

- Visualize an image topographically in 3D
  - The two spatial coordinates and the intensity (relief representation).
- Three types of points
  - Points belonging to a regional minimum.
  - Points ta which a drop of water would fall certainly to a regional minimum (*catchment basin*).
  - Points at which the water would be equally likely to fall to more than one regional minimum (crest lines or *watershed lines*).
- Objective: find the watershed lines.

Morphological Watersheds (cont.)

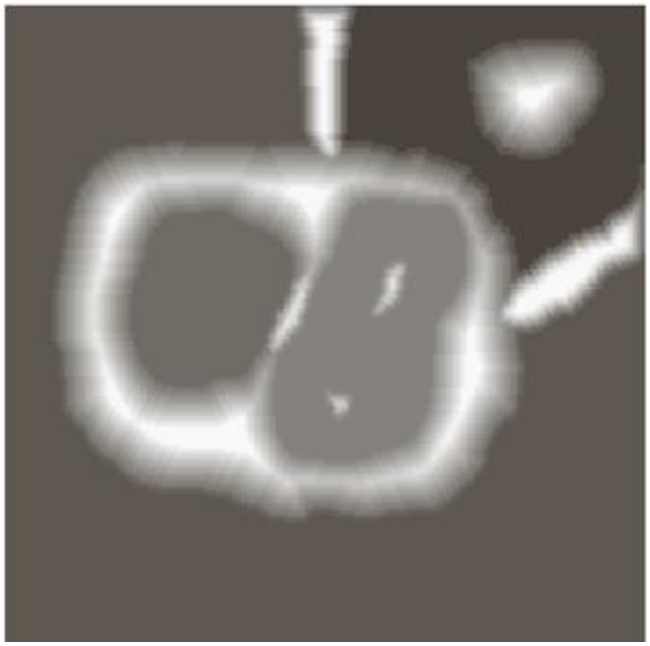


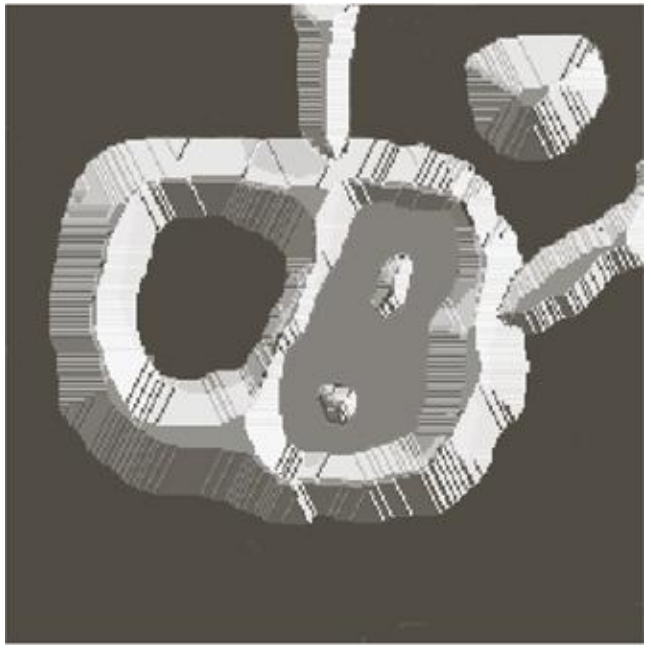
Image



- Topographic representation.
- The height is proportional to the image intensity.
- Backsides of structures are shaded for better visualization.

Morphological Watersheds (cont.)

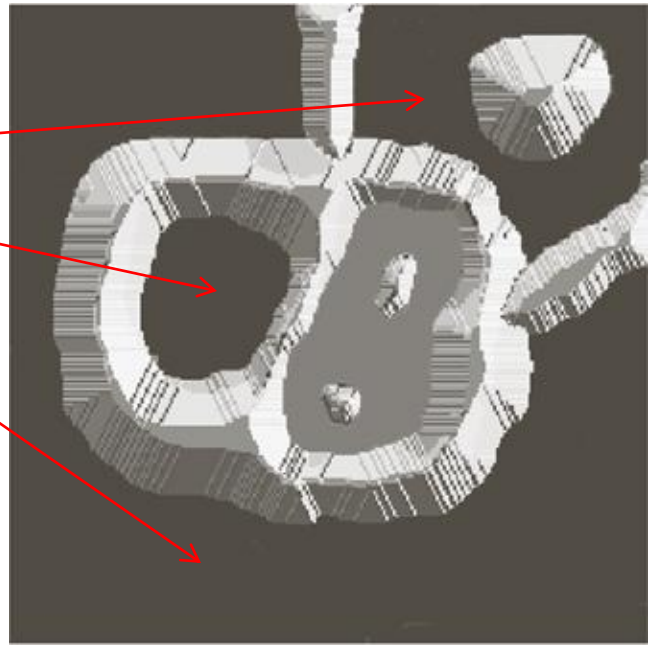




- A hole is punched in each regional minimum and the topography is flooded by water from below through the holes.
- When the rising water is about to merge in catchment basins, a dam is built to prevent merging.
- There will be a stage where only the tops of the dams will be visible.
- These continuous and connected boundaries are the result of the segmentation.

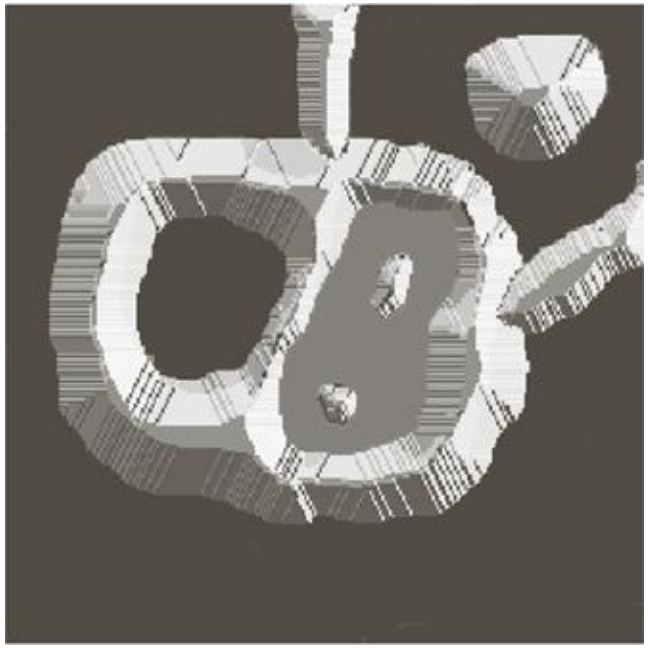
Morphological Watersheds (cont.)

Regional minima



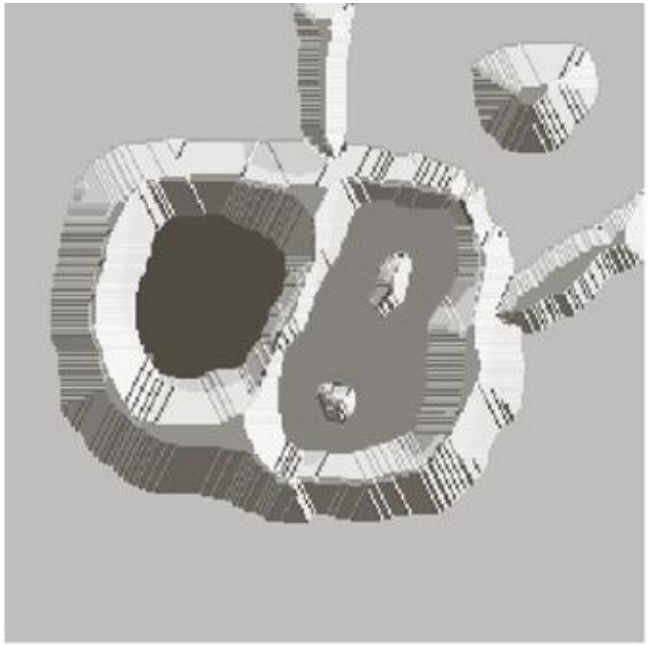
- Topographic representation of the image.
- A hole is punched in each regional minimum (dark areas) and the topography is flooded by water (at equal rate) from below through the holes.

Morphological Watersheds (cont.)



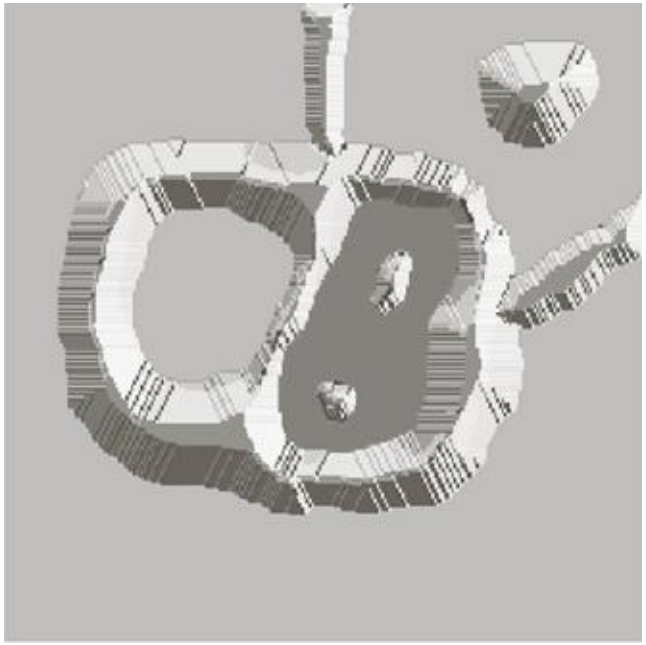
- Before flooding.
- To prevent water from spilling through the image borders, we consider that the image is surrounded by dams of height greater than the maximum image intensity.

Morphological Watersheds (cont.)



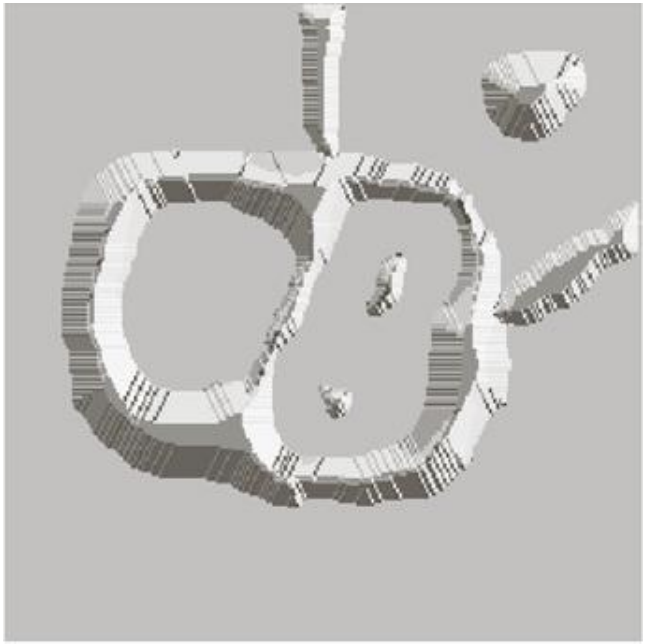
- First stage of flooding.
- The water covered areas corresponding to the dark background.

Morphological Watersheds (cont.)



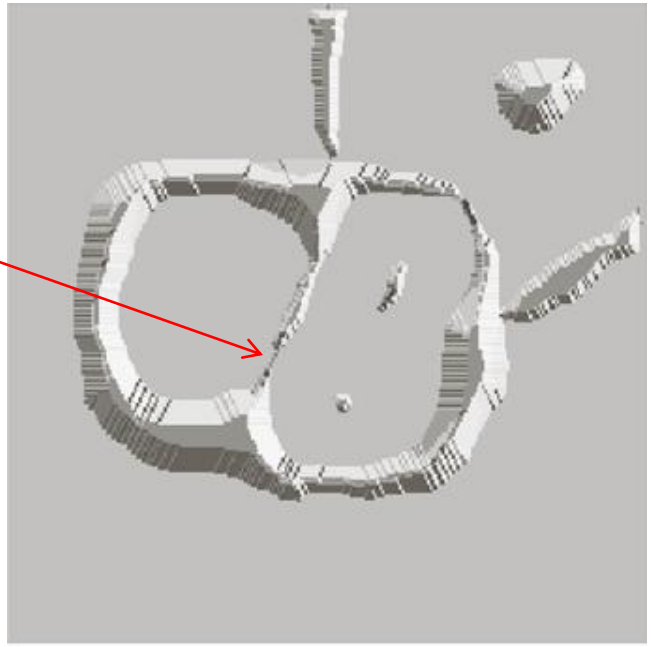
- Next stages of flooding.
- The water has risen into the other catchment basin.

Morphological Watersheds (cont.)



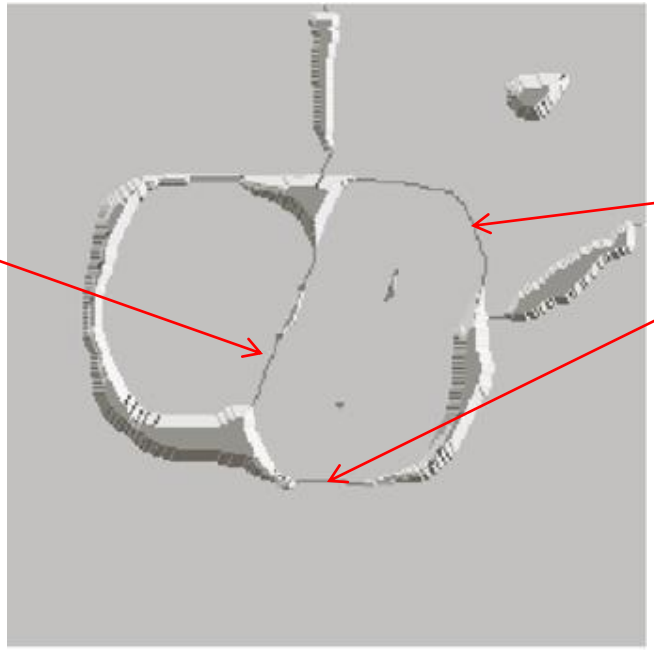
- Further flooding. The water has risen into the third catchment basin.

Morphological Watersheds (cont.)



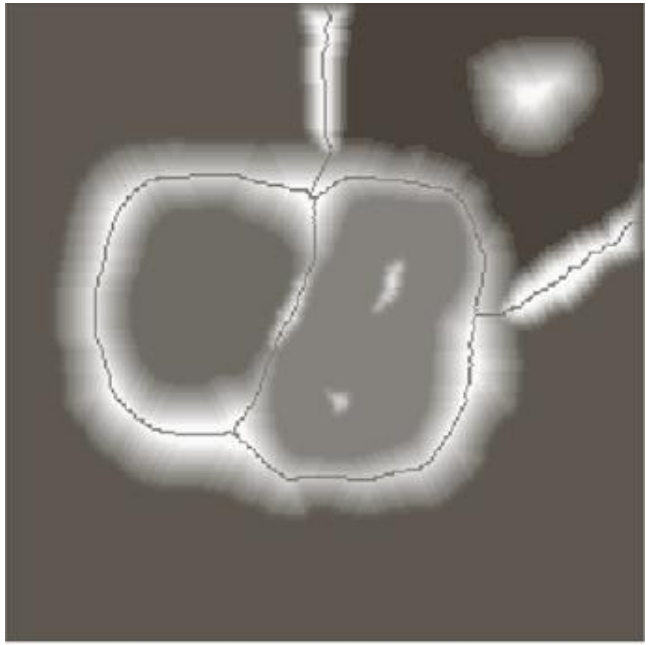
- Further flooding.
- The water from the left basin overflowed into the right basin.
- A short dam is constructed to prevent water from merging.

Morphological Watersheds (cont.)



- Further flooding.
- The effect is more pronounced.
- The first dam is now longer.
- New dams are created.

Morphological Watersheds (cont.)



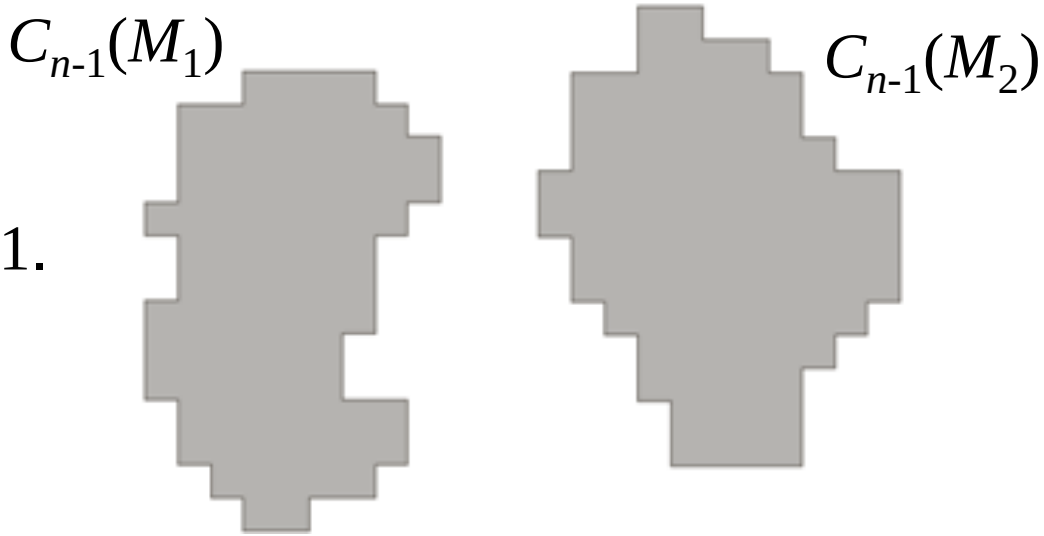
Final watershed lines superimposed on the image.

- The process continues until the maximum level of flooding is reached.
- The final dams correspond to the watershed lines which is the result of the segmentation.
- Important: continuous segment boundaries.

Morphological Watersheds (cont.)

- Dams are constructed by morphological dilation.

Flooding step n-1.

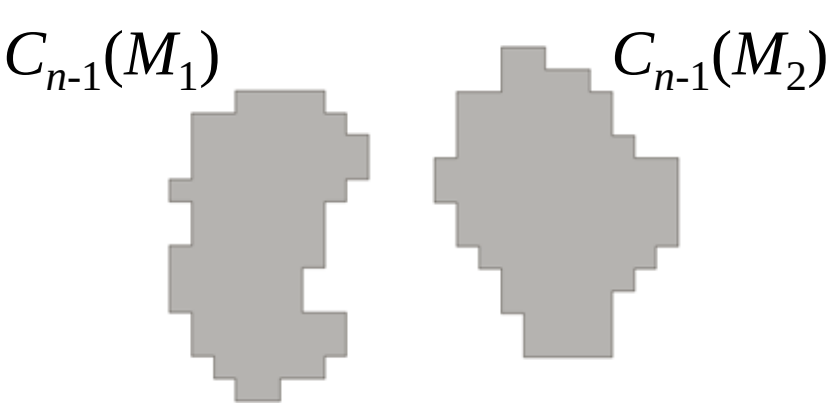


Regional minima: M_1 and M_2.
Catchment basins associated: $C_{n-1}(M_1)$ and $C_{n-1}(M_2)$.

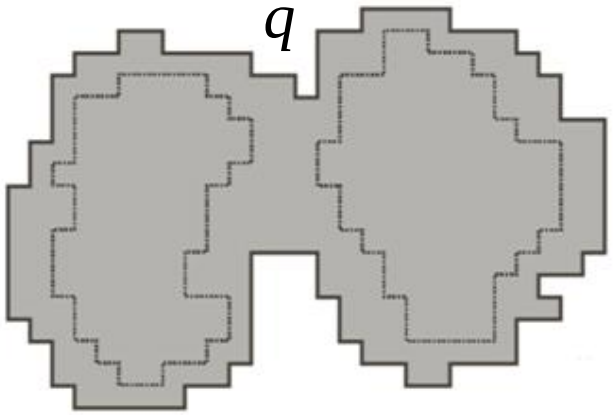
$$C[n-1] = C_{n-1}(M_1) \cup C_{n-1}(M_2)$$

$C[n\text{-}1]$ has two connected components.

Morphological Watersheds (cont.)



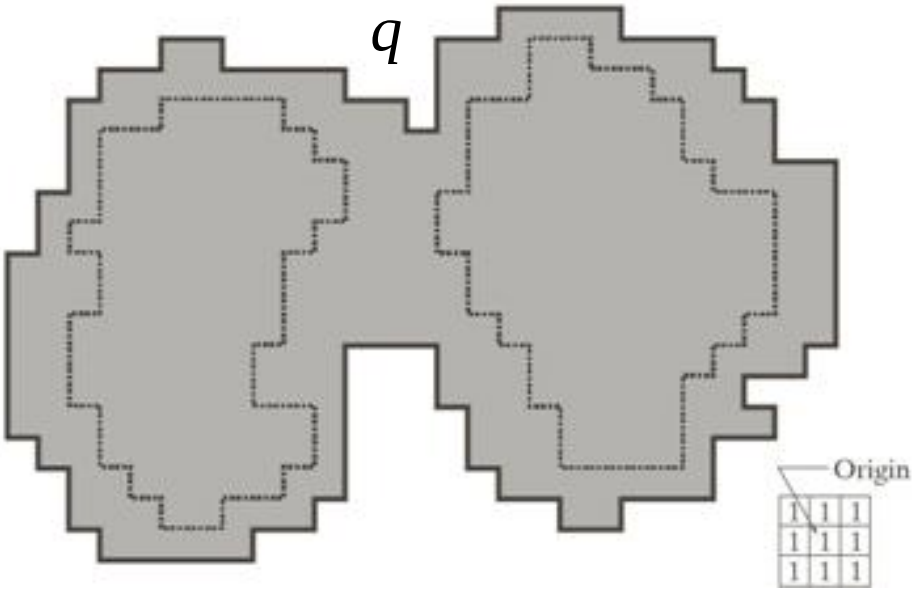
Flooding step n-1.



Flooding step n.

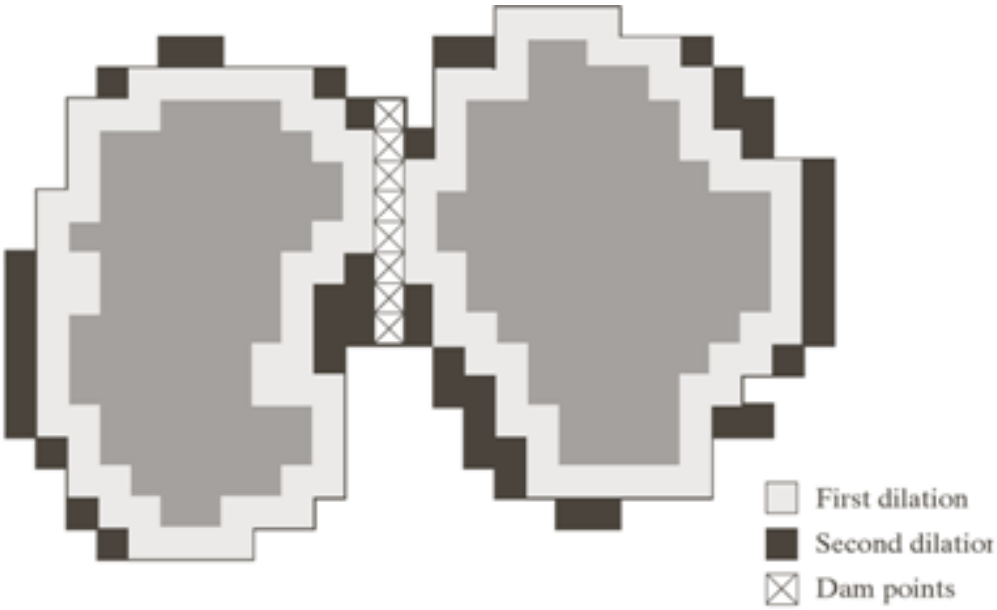
- If we continue flooding, then we will have one connected component.
- This indicates that a dam must be constructed.
- Let q be the merged connected component if we perform flooding a step n.

Morphological Watersheds (cont.)



- Each of the connected components is dilated by the SE shown, subject to:
  1. The center of the SE has to be contained in q.
  2. The dilation cannot be performed on points that would cause the sets being dilated to merge.

Morphological Watersheds (cont.)

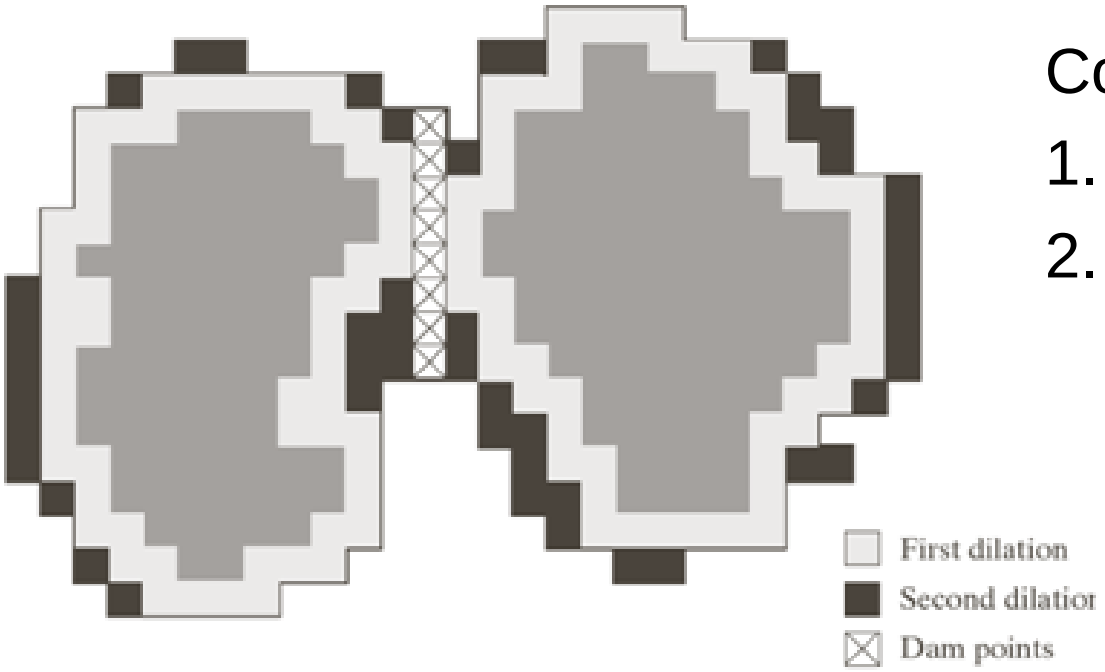


Conditions

1. Center of SE in q.
2. No dilation if merging.

- In the first dilation, condition 1 was satisfied by every point and condition 2 did not apply to any point.
- In the second dilation, several points failed condition 1 while meeting condition 2 (the points in the perimeter which is broken).

Morphological Watersheds (cont.)



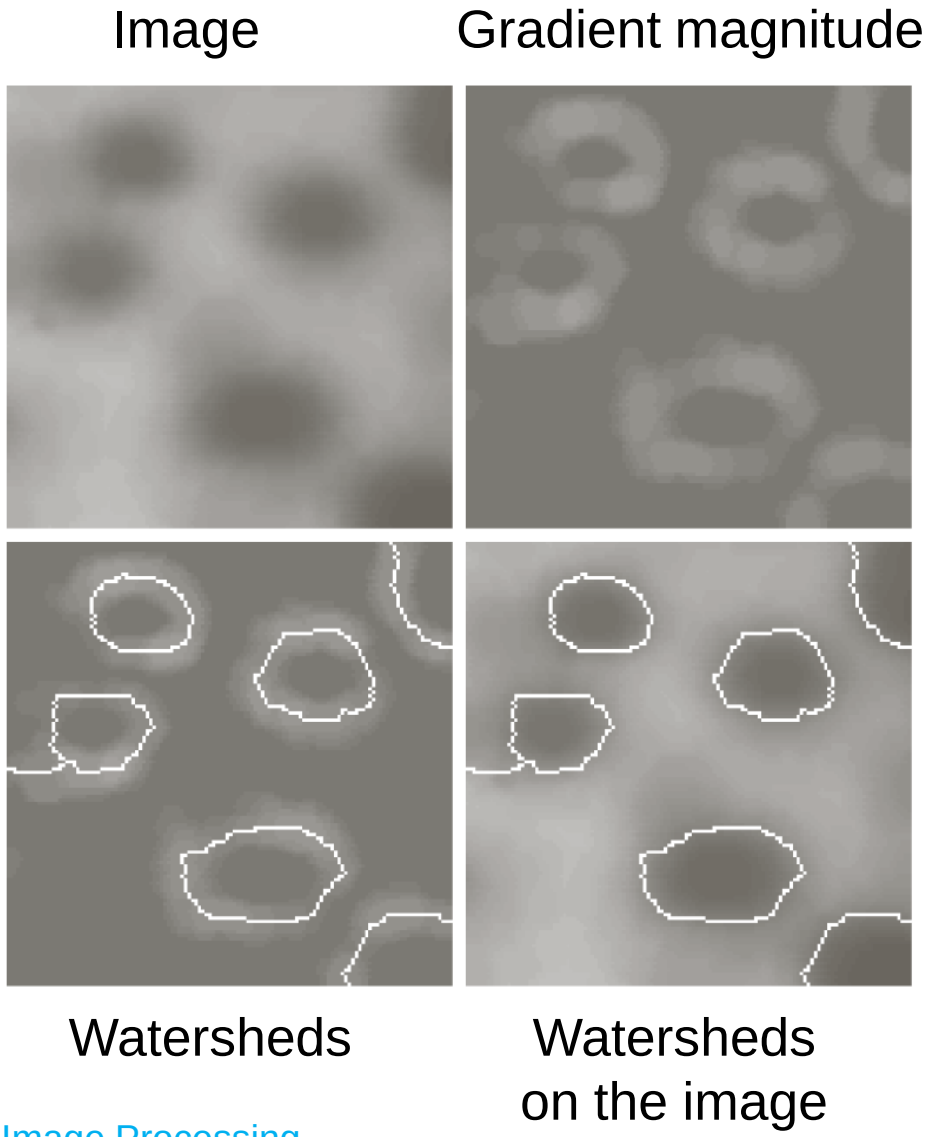
Conditions

1. Center of SE in q.
2. No dilation if merging.

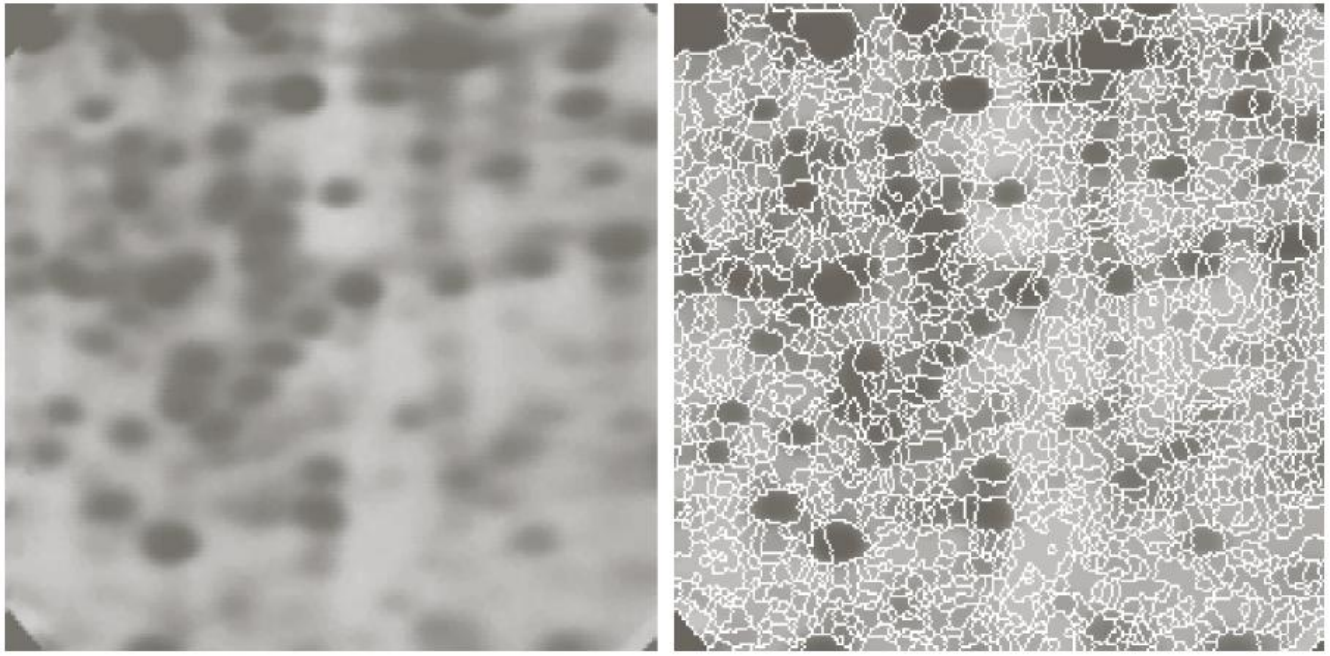
- The only points in q that satisfied both conditions form the 1-pixel thick path.
- This is the dam at step n of the flooding process.
- The points should satisfy both conditions.

Morphological Watersheds (cont.)

- A common application is the extraction of nearly uniform, blob-like objects from their background.
- For this reason it is generally applied to the gradient of the image and the catchment basins correspond to the blob like objects.



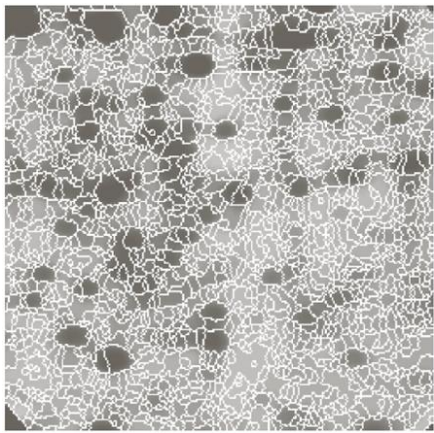
Morphological Watersheds (cont.)



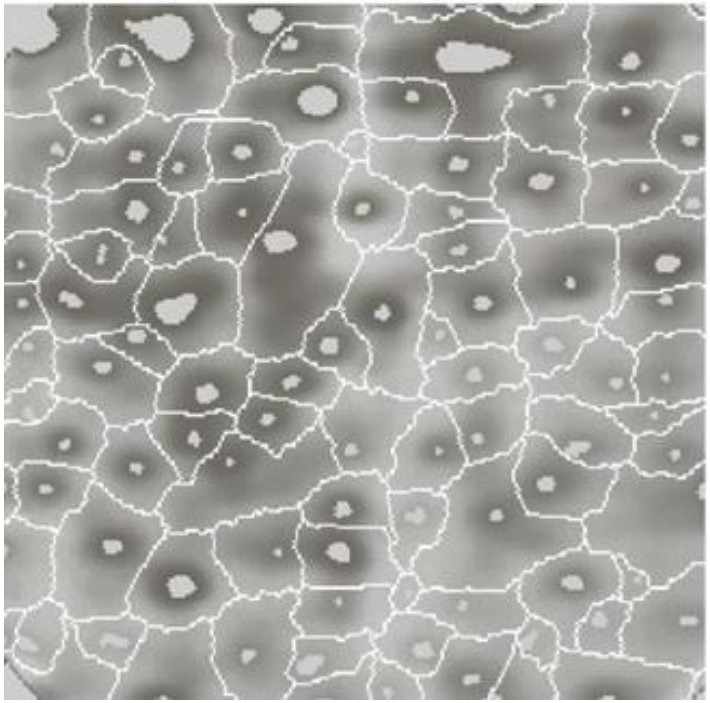
- Noise and local minima lead generally to *oversegmentation*.
- The result is not useful.
- Solution: limit the number of allowable regions by additional knowledge.

Morphological Watersheds (cont.)

- **Markers (connected components)**:
  - *internal*, associated with the objects
  - *external*, associated with the background.
- Here the problem is the large number of local minima.
- Smoothing may eliminate them.
- Define an *internal* marker (after smoothing):
  - Region surrounded by points of higher altitude.
    - They form connected components.
    - All points in the connected component have the same intensity.

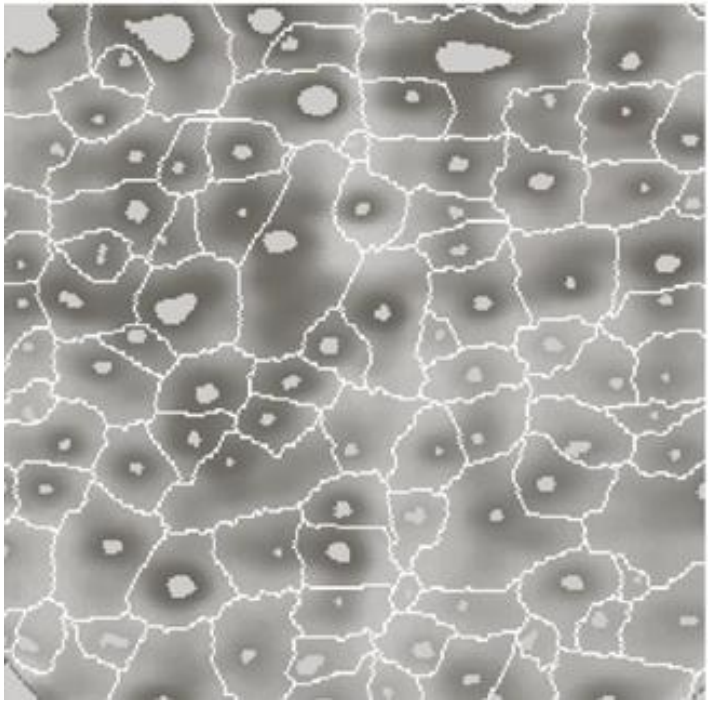


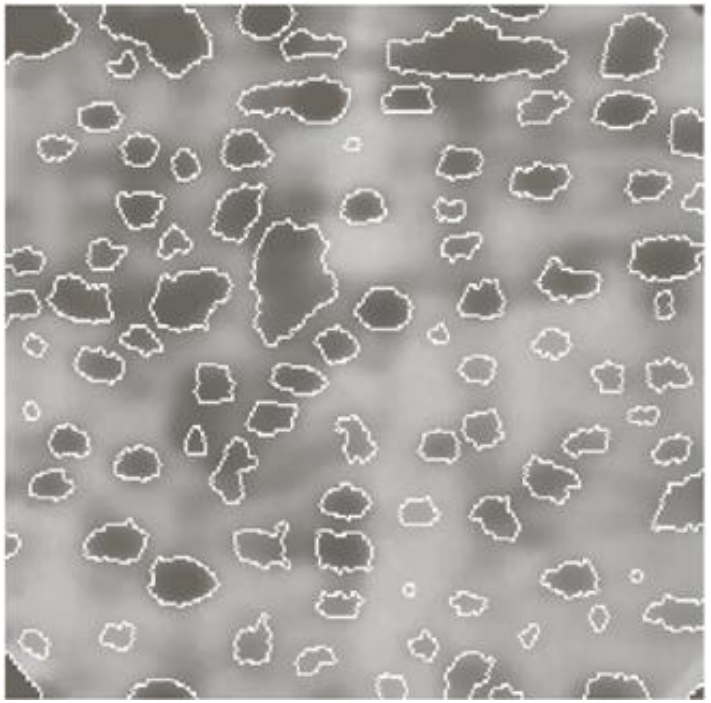
Morphological Watersheds (cont.)



- After smoothing, the internal markers are shown in light gray.
- The watershed algorithm is applied and the internal markers are the only allowable regional minima.
- The resulting watersheds are the external markers (shown in white).

Morphological Watersheds (cont.)



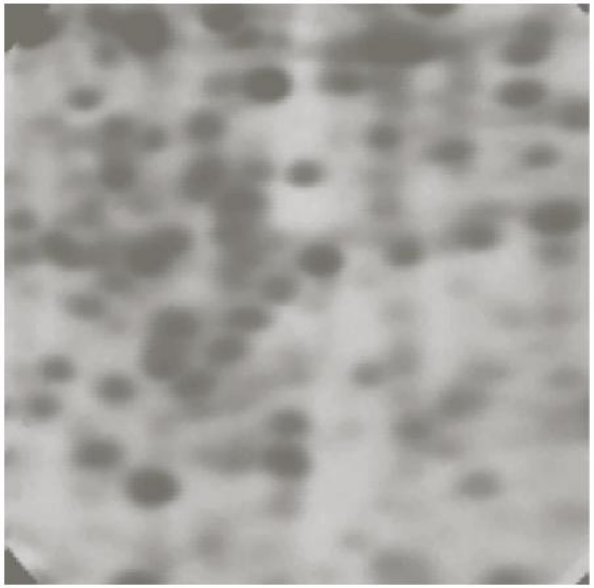


- Each region defined by the external marker has a single internal marker and part of the background.
- The problem is to segment each of these regions into two segments: a single object and background.
- The algorithms we saw in this lecture may be used (including watersheds applied to each individual region).

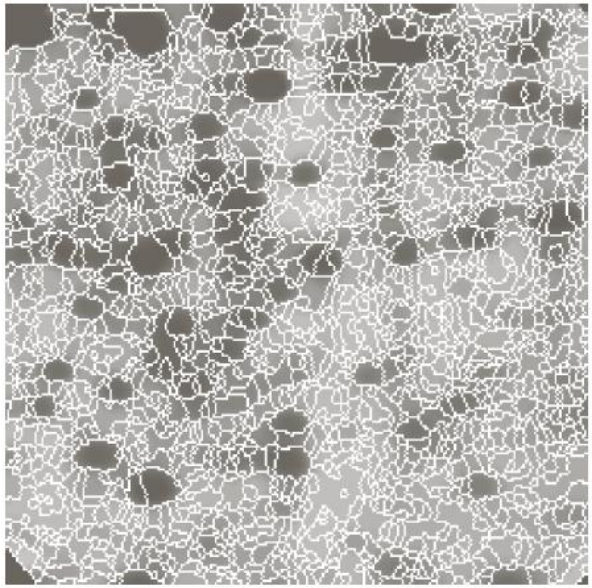
Morphological Watersheds (cont.)

Final segmentation.

Morphological Watersheds (cont.)



Image



Watersheds

Watersheds with markers

The use of motion in segmentation

- Powerful cue used by humans to extract objects and regions.
- Motion arises from
  - Objects moving in the scene.
  - Relative displacement between the sensing system and the scene (e.g. robotic applications, autonomous navigation).
- We will consider motion
  - Spatially.
  - In the frequency domain.

Basic spatial motion segmentation

- Difference image and comparison with respect to a threshold:

$$d_{ij}(x,y) = \begin{cases} 1 & \text{if } \left| f(x,y,t_i) - f(x,y,t_j) \right| > T \\ 0 & \text{otherwise} \end{cases}$$

- The images should be registered.
- Illumination should be relatively constant within the bounds defined by *T*.

Accumulative differences

- Comparison of a reference image with every subsequent image in the sequence.
- A counter is incremented every time a pixel in the current image is different from the reference image.
- When the k^{th} frame is being examined, the entry in a given pixel of the accumulative difference image (ADI) gives the number of times this pixel differs from its counterpart in the reference image.

Accumulative differences (cont.)

- Absolute ADI:

$$A_k(x,y) = \begin{cases} A_{k-1}(x,y)+1 & \text{if } \left|R(x,y) - f(x,y,t_k)\right| > T \\ A_{k-1}(x,y) & \text{otherwise} \end{cases}$$

- Positive ADI:

$$P_k(x,y) = \begin{cases} P_{k-1}(x,y)+1 & \text{if } R(x,y) - f(x,y,t_k) > T \\ P_{k-1}(x,y) & \text{otherwise} \end{cases}$$

- Negative ADI:

$$N_k(x,y) = \begin{cases} N_{k-1}(x,y)+1 & \text{if } R(x,y) - f(x,y,t_k) < -T \\ N_{k-1}(x,y) & \text{otherwise} \end{cases}$$

Accumulative differences (cont.)

- ADIs for a rectangular object moving to southeast.

Absolute Positive Negative

- The nonzero area of the positive ADI gives the size of the object.
- The location of the positive ADI gives the location of the object in the reference frame.
- The direction and speed may be obtained fom the absolute and negative ADIs.
- The absolute ADI contains both the positive and negative ADIs.

Accumulative differences (cont.)

- To establish a reference image in a non stationary background.
  - Consider the first image as the reference image.
  - When a non stationary component has moved out of its position in the reference frame, the corresponding background in the current frame may be duplicated in the reference frame. This is determined by the positive ADI:
    - When the moving object is displaced completely with respect to the reference frame the positive ADI stops increasing.

Accumulative differences (cont.)

- Subtraction of the car going from left to right to establish a reference image.

- Repeating the task for all moving objects may result in a static reference image.
- The method works well only in simple scenarios.

Frequency domain techniques

- Consider a sequence $f(x,y,t)$, $t=0,1,\ldots,K-1$ of size M x N.
- All frames have an homogeneous zero background except of a single pixel object with intensity of 1 moving with constant velocity.
- At time $t=0$, the object is at (x', y') and the image plane is projected onto the vertical (x) axis. This results in a 1D signal which is zero except at x'.
- If we multiply the 1D signal by $\exp[j2\pi\alpha_1 x\Delta t]$, for $x=0,1,\ldots,M-1$ and sum the results we obtain the single term $\exp[j2\pi\alpha_1 x'\Delta t]$.
- In frame $t=1$, suppose that the object moved to $(x'+1, y')$, that is, it has moved 1 pixel parallel to the x-axis. The same procedure yields $\exp[j2\pi\alpha_1(x'+1)+\Delta t]$.

Frequency domain techniques (cont.)

- Applying Euler's formula, for t=0,1,…,K-1:

$$e^{j2\pi\alpha_1(x'+t)\Delta t} = \cos[2\pi\alpha_1(x'+t)\Delta t] + j\sin[2\pi\alpha_1(x'+t)\Delta t]$$

- This is a complex exponential with frequency α_1.
- If the object were moving V_1 pixels (in the x-direction) between frames the frequency would have been $V_1\alpha_1$.
- This causes the DFT of the 1D signal to have wo peaks, one at $V_1\alpha_1$ and one at K-$V_1\alpha_1$.
- Division by a_1 yields the velocity V_1.
- These concepts may be generalized as follows.

Frequency domain techniques (cont.)

- For a sequence of K images of size M x N, the sum of the weighted projections onto the x-axis at an integer instant of time is:

$$g_x(t,\alpha_1) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y,t)e^{j2\pi\alpha_1 x\Delta t},\ t = 0,1,...,K-1$$

- Similarly, the sum of the weighted projections onto the y-axis at an integer instant of time is:

$$g_y(t,\alpha_2) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y,t)e^{j2\pi\alpha_2 y\Delta t},\ t = 0,1,...,K-1$$

Frequency domain techniques (cont.)

- The 1D DFT of the above signals are:

$$G_x(u_1,\alpha_1)=\sum_{t=0}^{K-1} g_x(t,\alpha_1)e^{j2\pi u_1 t/K},\ u_1=0,1,...,K-1$$

$$G_y(u_2,\alpha_2)=\sum_{t=0}^{K-1} g_y(t,\alpha_2)e^{j2\pi u_2 t/K},\ u_2=0,1,...,K-1$$

- The peaks in $G_x(u_1,\alpha_1)$ and $G_y(u_2,\alpha_2)$ provide the velocities:
  - $u_1=\alpha_1 V_1$
  - $u_2=\alpha_2 V_2$

Frequency domain techniques (cont.)

- The unit of velocity is in pixels per total frame rate. For example, $V_1=10$ is interpreted as a motion of 10 pixels in K frames.
- For frames taken uniformly, the actual physical speed depends on the frame rate and the distance between pixels. Thus, for $V_1=10$, $K=30$, if the frame rate is two images/sec and the distance between pixels is 0.5 m, the speed is:
  - $V_1=(10 \text{ pixels})(0.5 \text{ m/pixel})(2 \text{ frames/sec}) / (30 \text{ frames}) = 1/3 \text{ m/sec}.$

Frequency domain techniques (cont.)

- The sign of the x-component of the velocity is obtained by using Fourier properties of sinusoids:

$$S_{1x} = \left.\frac{d^2 \operatorname{Re}\left[g_x(t,\alpha_1)\right]}{dt^2}\right|_{t=n} \qquad S_{2x} = \left.\frac{d^2 \operatorname{Im}\left[g_x(t,\alpha_1)\right]}{dt^2}\right|_{t=n}$$

- The above quantities will have the same sign at an arbitrary time point $t=n$, if $V_1>0$.
- Conversely, they will have opposite signs if $V_1<0$.
- In practice, if either is zero, we consider the next closest point.