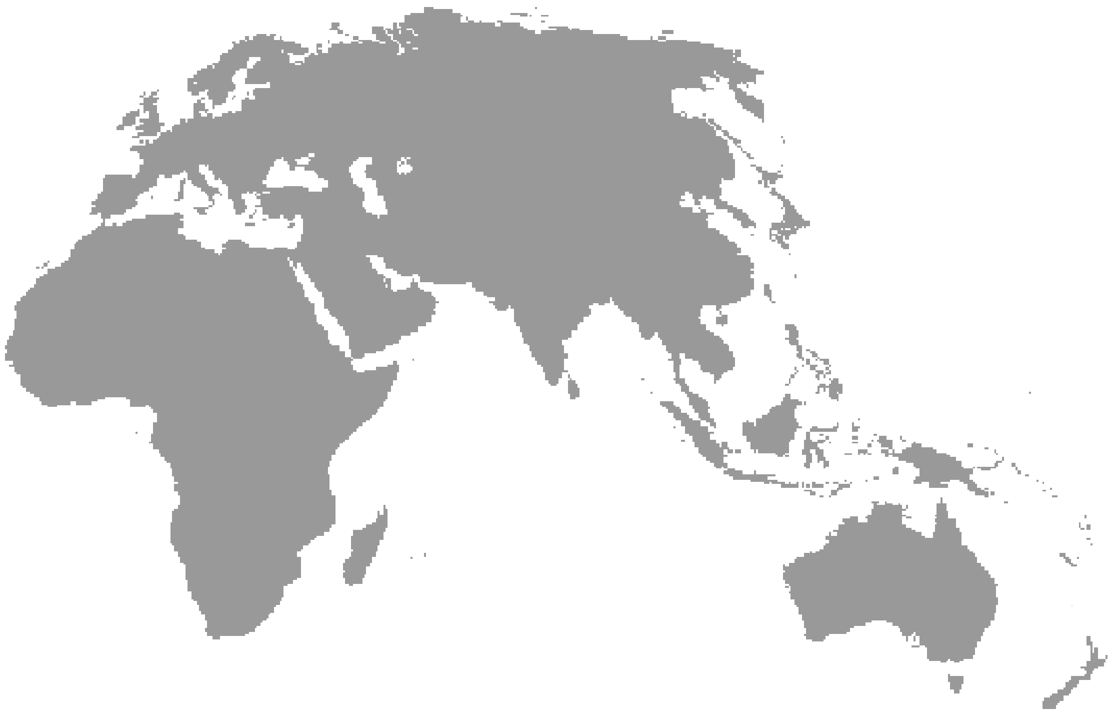
Segmentation

Segmentation is the task of isolating the objects of interest in an image.

Typically, given an image R, the segmented image consists of regions R_1, R_2, $\ldots$, R_n such that,

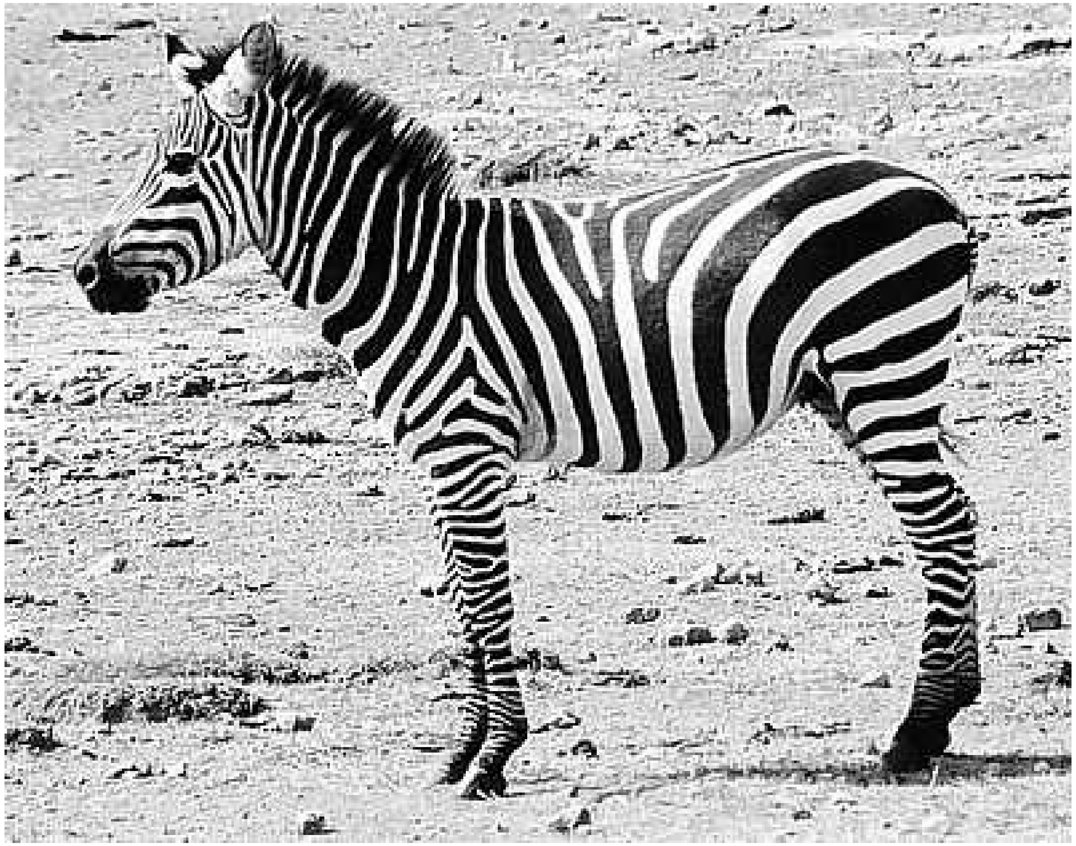
1. R_i's are disjoint (i.e. $R_i \cap R_j = \emptyset$ if $i \neq j$),
2. R_i's are connected,
3. R_i's partition the image (i.e. $\cup R_i = R$).

Segmentation



Segmentation

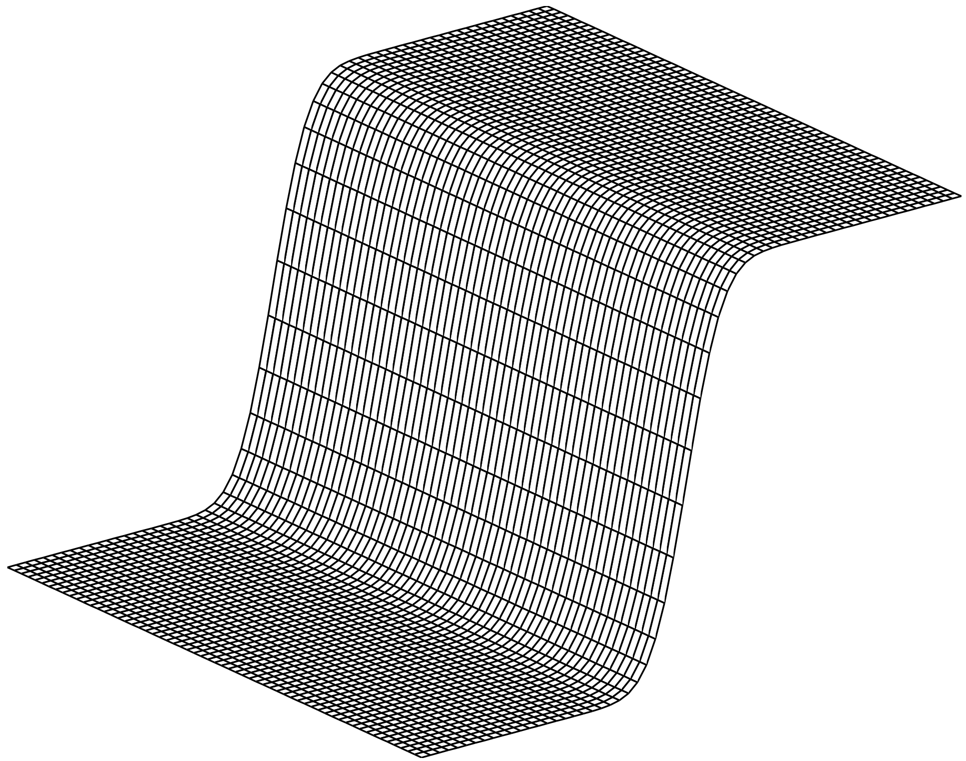
Segmentation



What is an Edge?

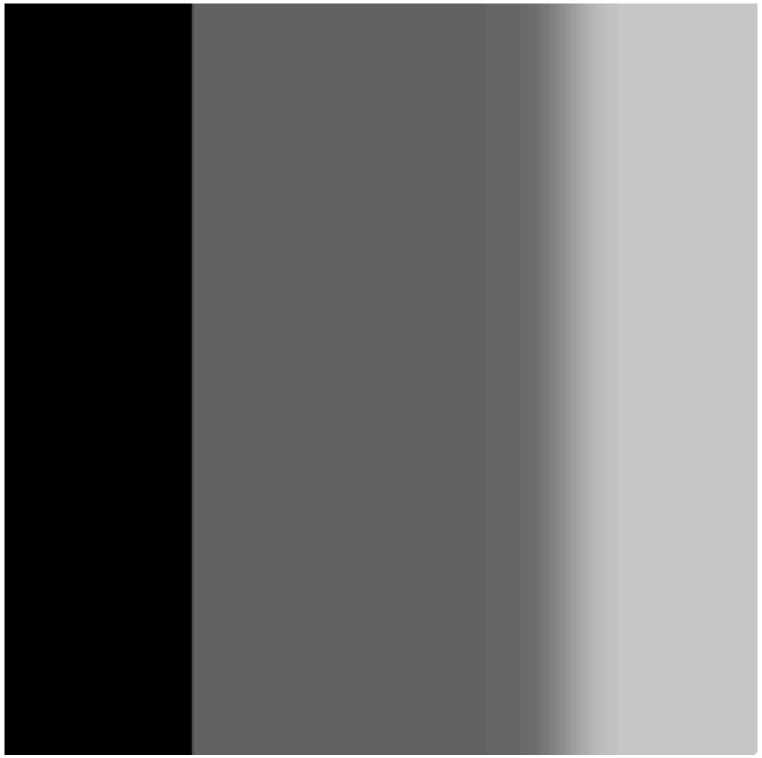
An edge involves a transition from an intensity level to another intensity level.

Edge Detection



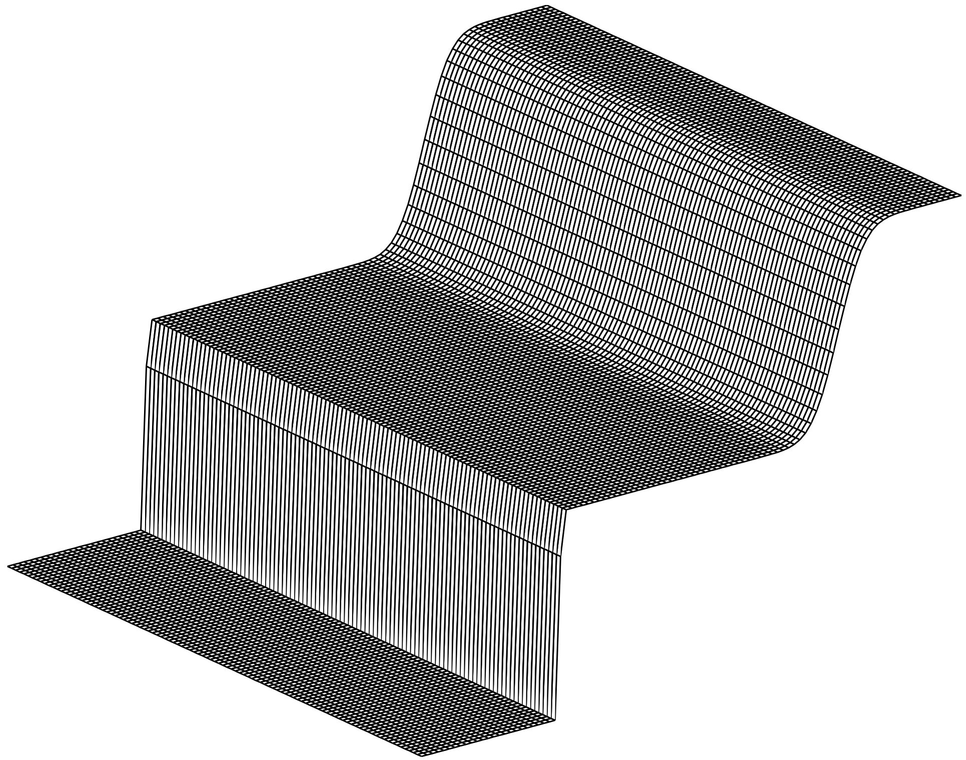
Edge Models

Edges may be sharp (step edge) or diffuse (ramp edge).



Edge Models

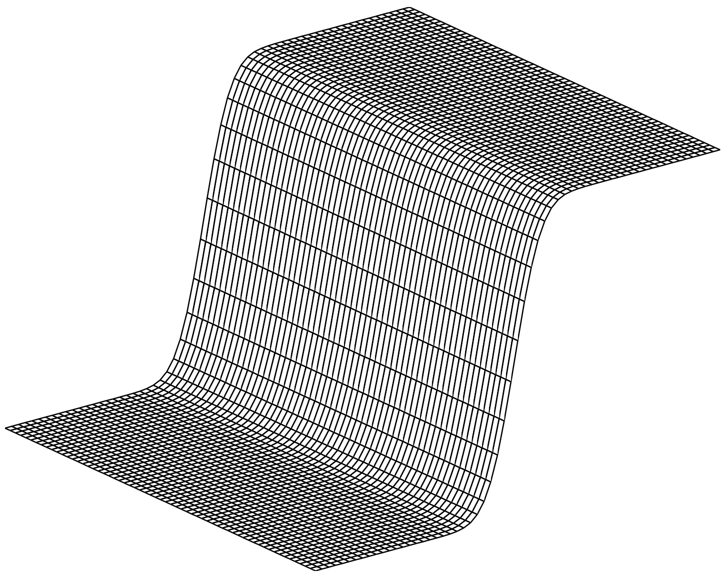
Edges may be sharp (step edge) or diffuse (ramp edge).



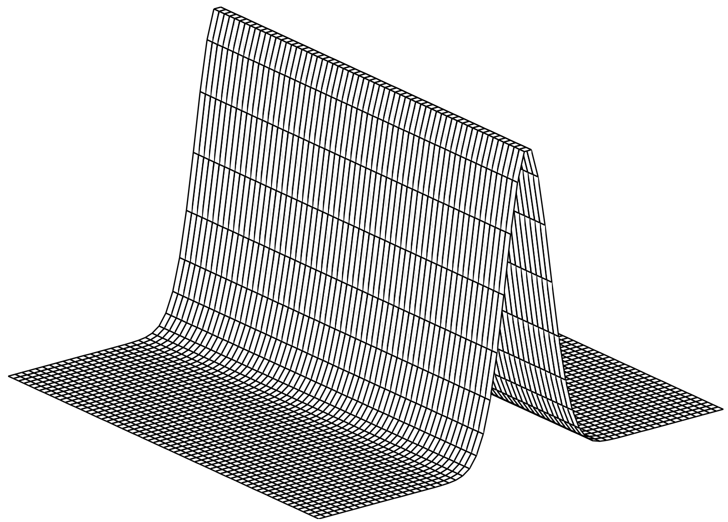
Edge Models

Edges may be sharp (step edge) or diffuse (ramp edge).

Edge Detection



(a)



(b)

$\mathcal{Q}$: How do we go from (a) to (b)?

Derivative Images

$\mathcal{Q}$: How do we go from (a) to (b)?

$\mathcal{A}$: Compute the derivative in the horizontal direction.

In practice, we can convolve with

1	0	−1

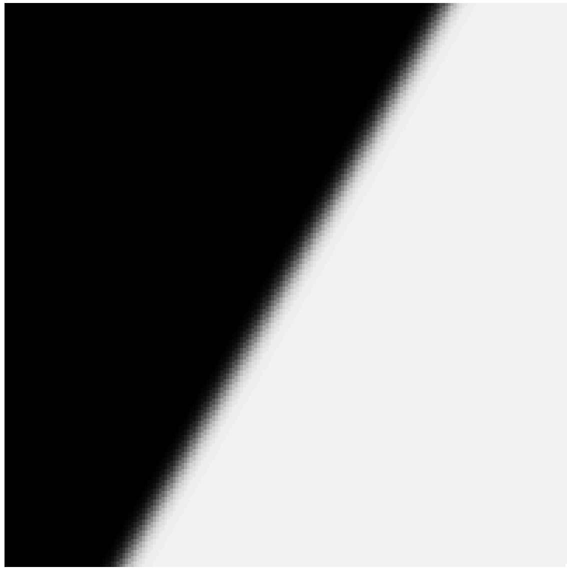
or

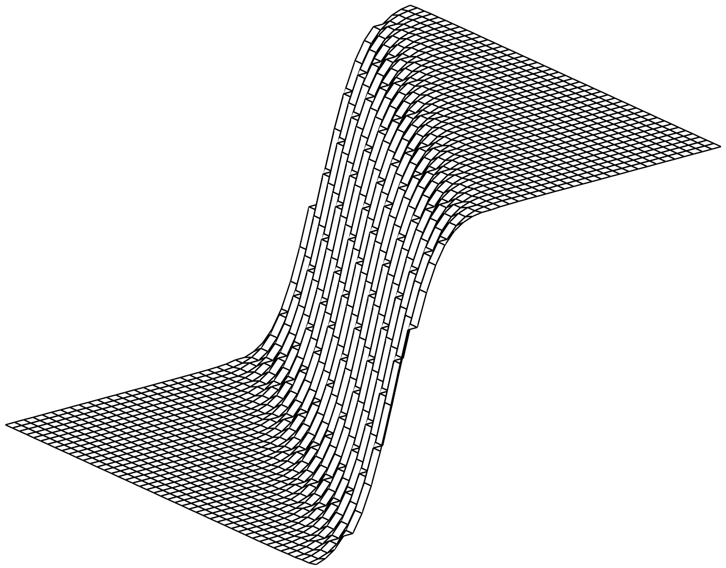
1	0	−1
1	0	−1
1	0	−1

or

1	0	−1
2	0	−2
1	0	−1

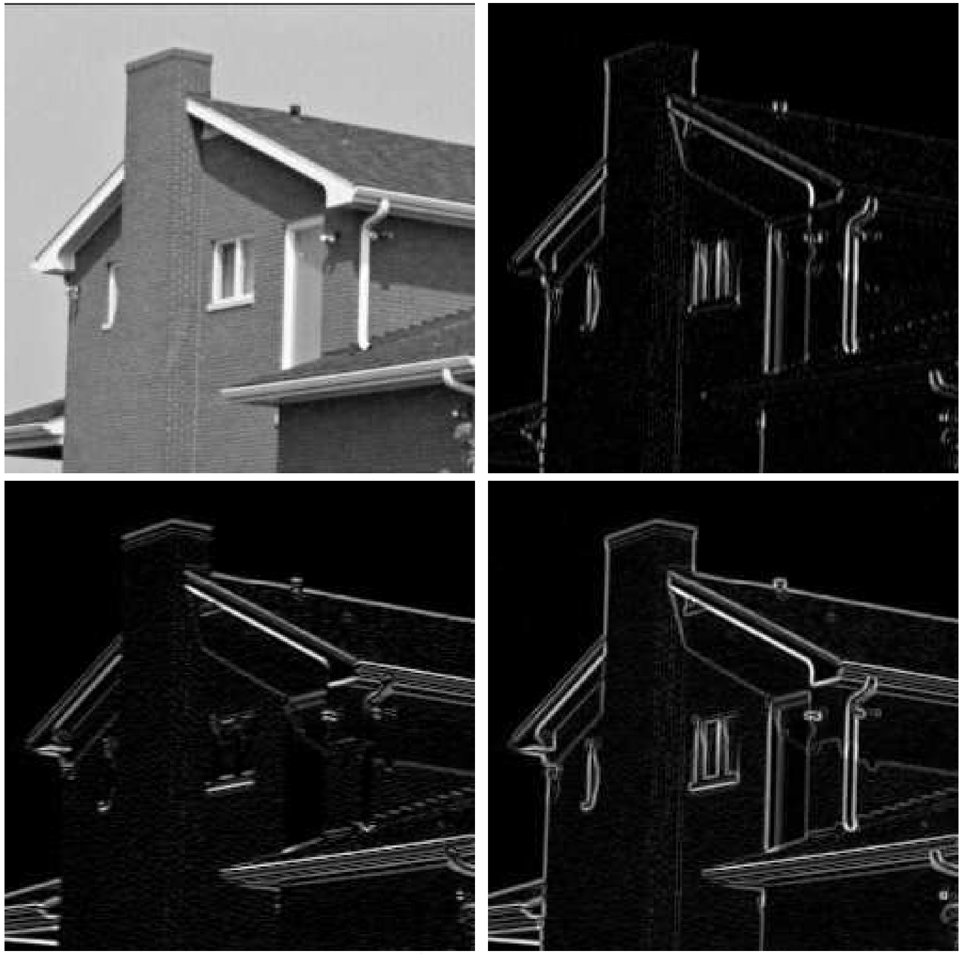
Directional Derivative





$\mathcal{Q}$: What if the edge is not vertical?

Gradient Magnitude



f	f_x
f_y	$\sqrt{f_x^2 + f_y^2}$

Gradient Magnitude

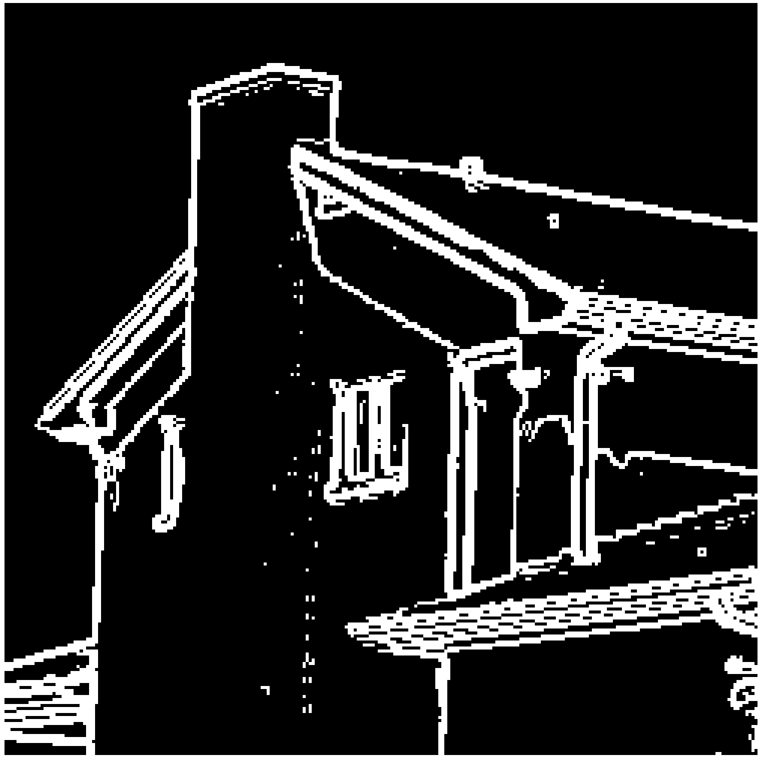
Given the gradient magnitude, how do we determine the edges?

(i) Set the pixels below a threshold to zero.

(ii) Determine the local maxima – (how?)

Gradient Magnitude

Set the pixels below a threshold to zero.



Laplacian Operator

Determine the local maxima by looking at the zero crossings of second order derivatives.

The Laplacian operator is frequently used for this purpose.

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$$

The discrete approximation is :

0	1	0
1	-4	1
0	1	0

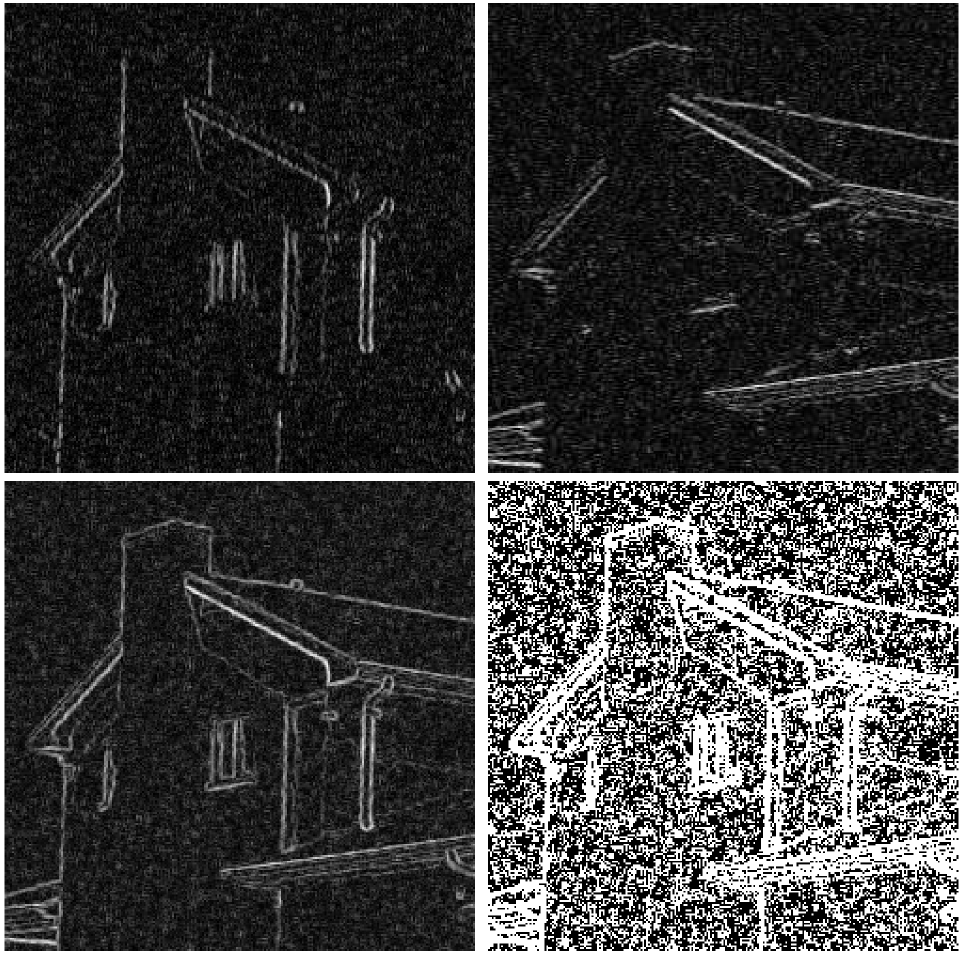
Laplacian Operator

Zero crossings of the discrete Laplacian.

Noisy Images

What if there is noise ?

Noisy Images



We end up with a very dirty edge map.

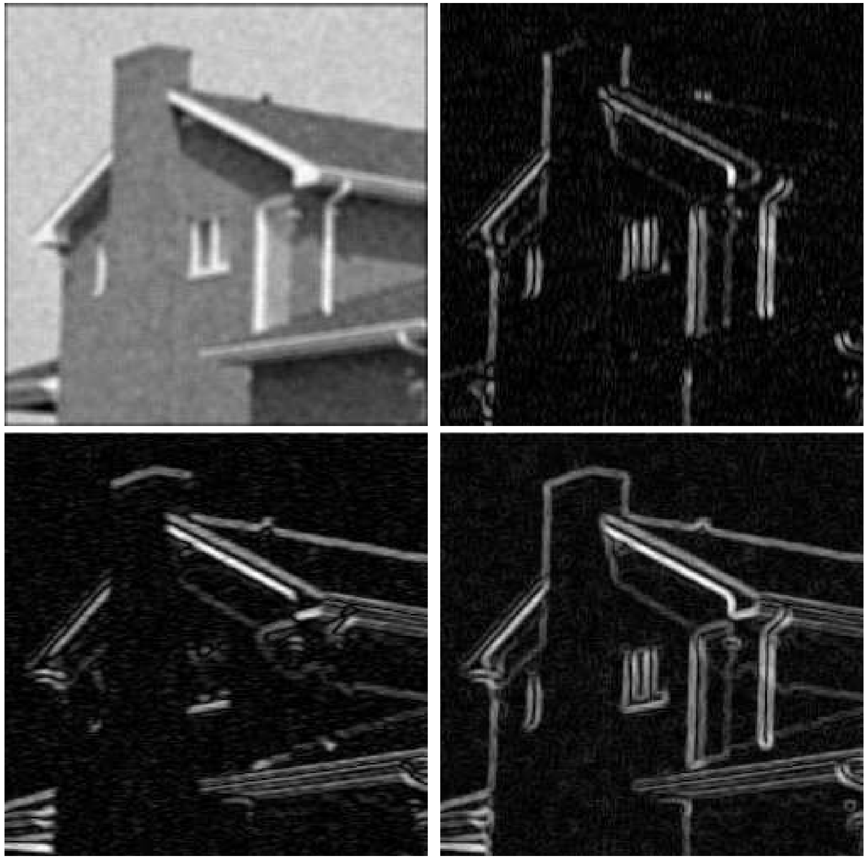
Noisy Images

Effects of using different thresholds.

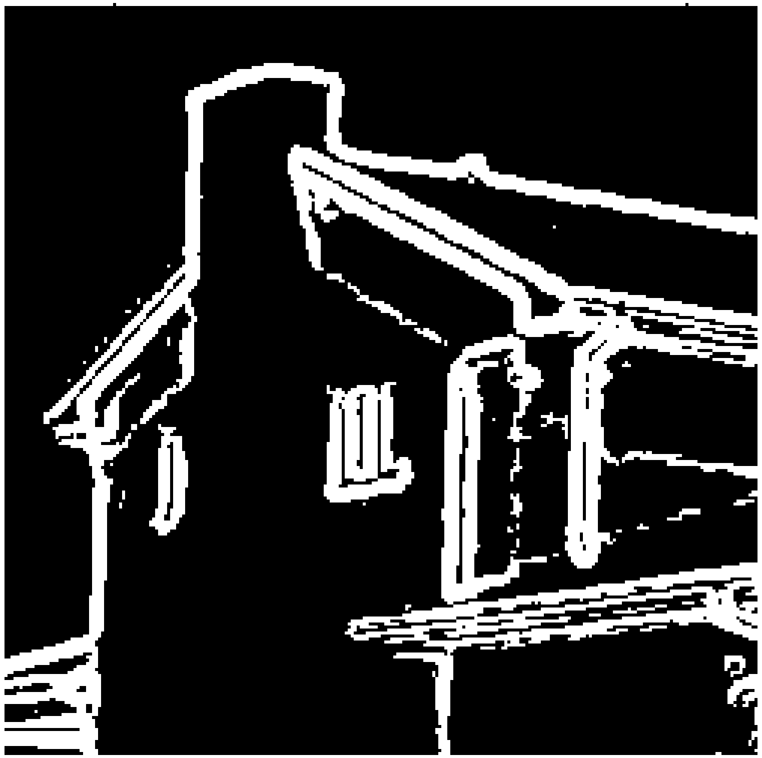
Noisy Images

Apply a lowpass filter in order to suppress noise.

Following the lowpass filter, perform edge detection.



Noisy Images



Lowpass filter followed by gradient magnitude thresholding.

Laplacian of Gaussian (LoG)

The lowpass filter and the laplacian operator may be combined.

The Marr-Hildreth edge detector considers the zero crossings of the image after applying a Laplacian of a Gaussian.

In this case, the idea is to

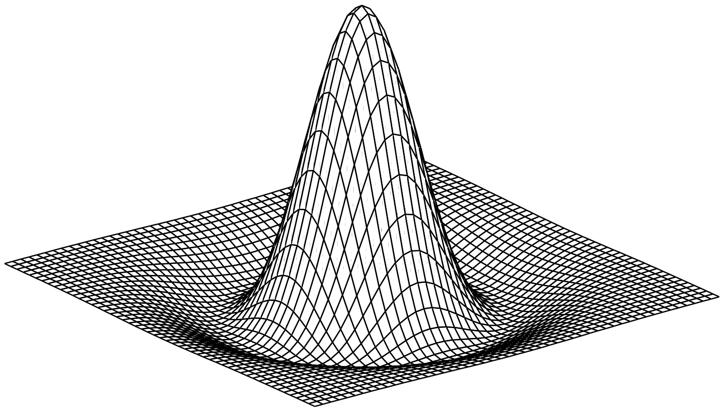
(i) Filter with $\nabla^2 \left(e^{-(x^2+y^2)/2\sigma^2} \right)$

(ii) Determine the zero crossings of the resulting image.

By modifying σ, one selects the right scale for detecting edges.

Laplacian of Gaussian (LoG)

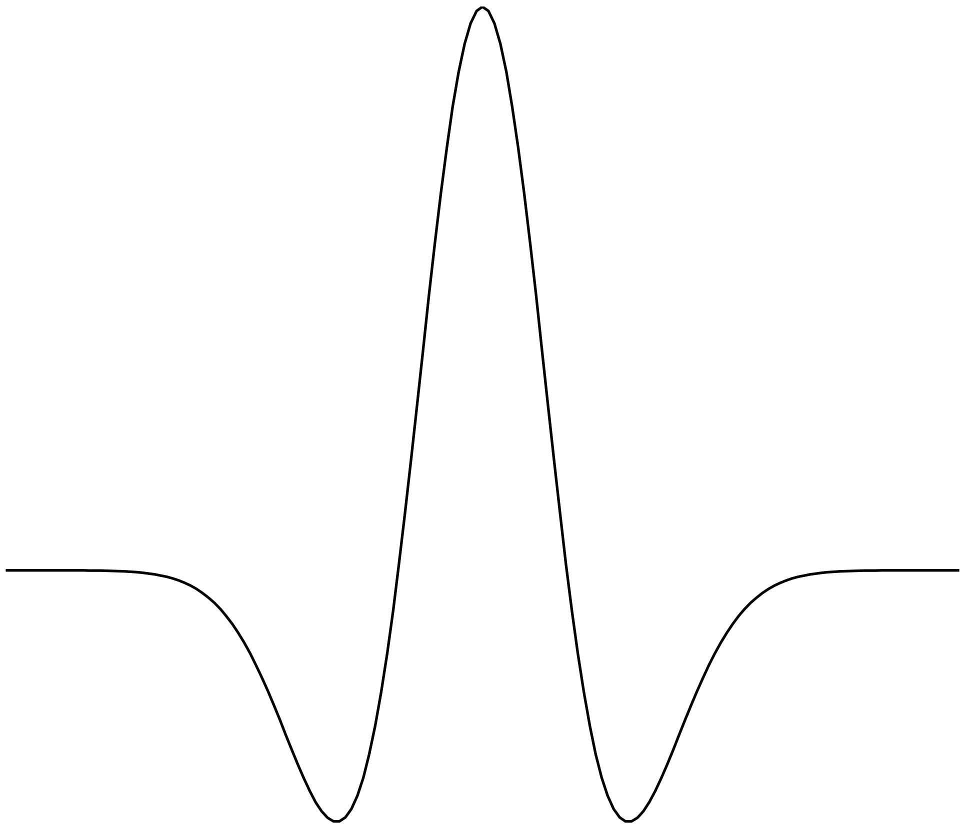
The Laplacian of Gaussian and its 5×5 approximation.



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

Laplacian of Gaussian (LoG)

Cross section from $y = 0$.



'Mexican Hat'

Difference of Gaussians (DoG)

The Laplacian of Gaussian can be approximated by a difference of Gaussians :

$$\text{DoG} = \frac{1}{2\pi\sigma_1^2}\exp\left(\frac{x^2+y^2}{2\sigma_1^2}\right) - \frac{1}{2\pi\sigma_2^2}\exp\left(\frac{x^2+y^2}{2\sigma_2^2}\right)$$

with $\sigma_1 > \sigma_2$.

Canny Edge Detector

The edges obtained by thresholding the gradient magnitude are usually too thick.

If the threshold is selected too high, the edges that have relatively low gradient magnitudes are eliminated – also, gaps may form.

Canny edge detector produces thin edges that are connected.

How can we overcome these problems?

Canny Edge Detector

To thin the edges, apply 'non-maxima suppression' :

(i) For each putative edge pixel, determine whether the magnitude of the gradient is a local maximum in the gradient direction.

(ii) Remove the pixel if it is not a local maximum.

Canny Edge Detector

Suppose $g_N(x, y)$ is thresholded gradient magnitude image with the non-maxima suppressed.

To connect the edges,

(i) Determine two edge maps

$$g_{NH}(x, y) = \big(g_N(x, y) > T_H\big)$$
$$g_{NL}(x, y) = \big(T_H > g_N(x, y) > T_L\big)$$

(ii) For each non-zero pixel of g_{NH}, add a neighbor if it is non-zero in g_{NL}.

Canny Edge Detector

In summary, the Canny Edge Detector consists of

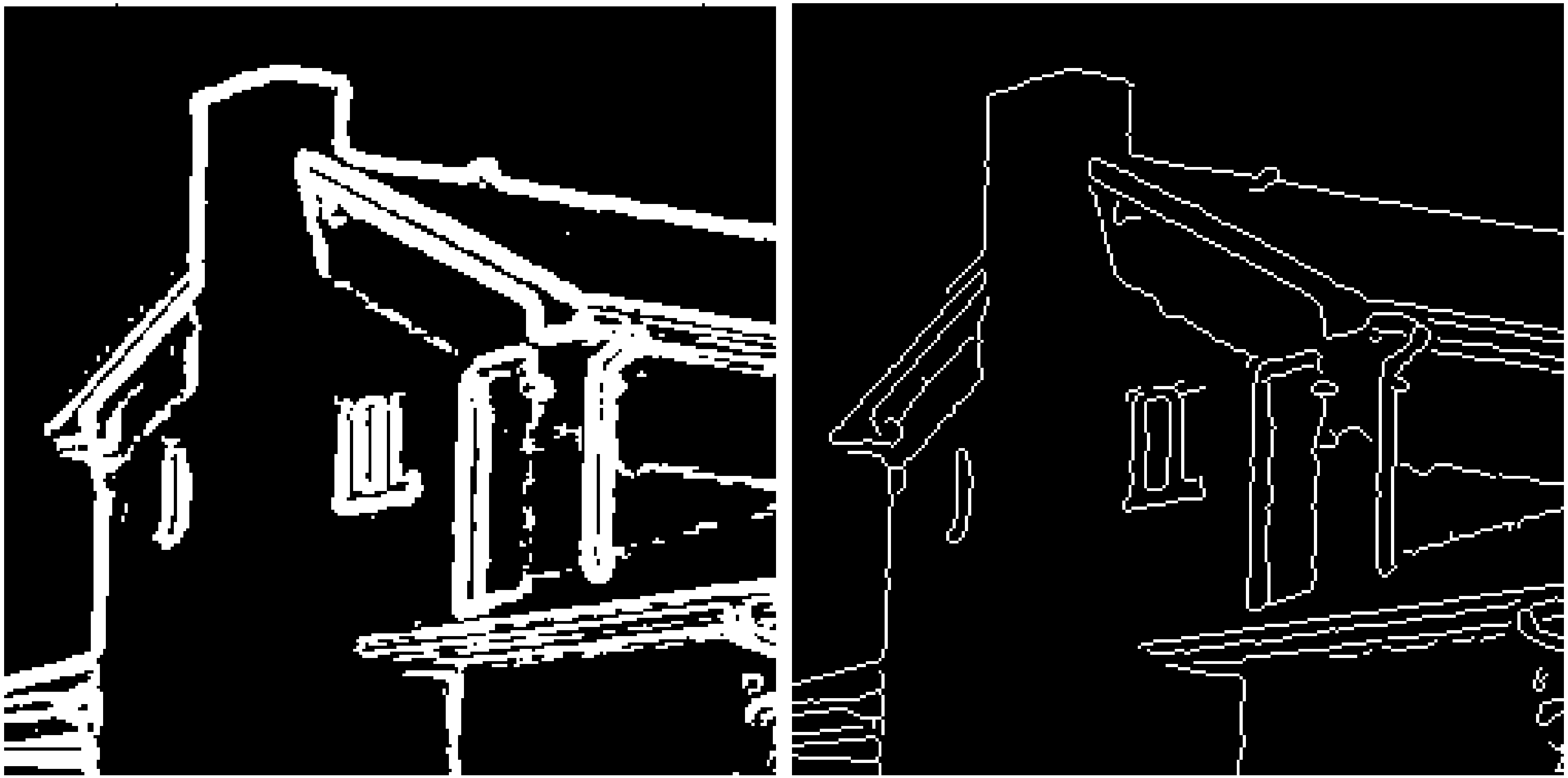
(i) Smooth the image with a Gaussian filter.

(ii) Compute the gradient magnitude image.

(iii) Suppress the non-maxima from the gradient magnitude image.

(iv) Link the edges.

Canny Edge Detector



Left : Threshold applied to the gradient magnitude of the lowpass filtered image.

Right : Output of Canny edge detector with manually chosen parameters.

Segmentation[1]

$\mathcal{Q}$: How can we separate the object from the background?

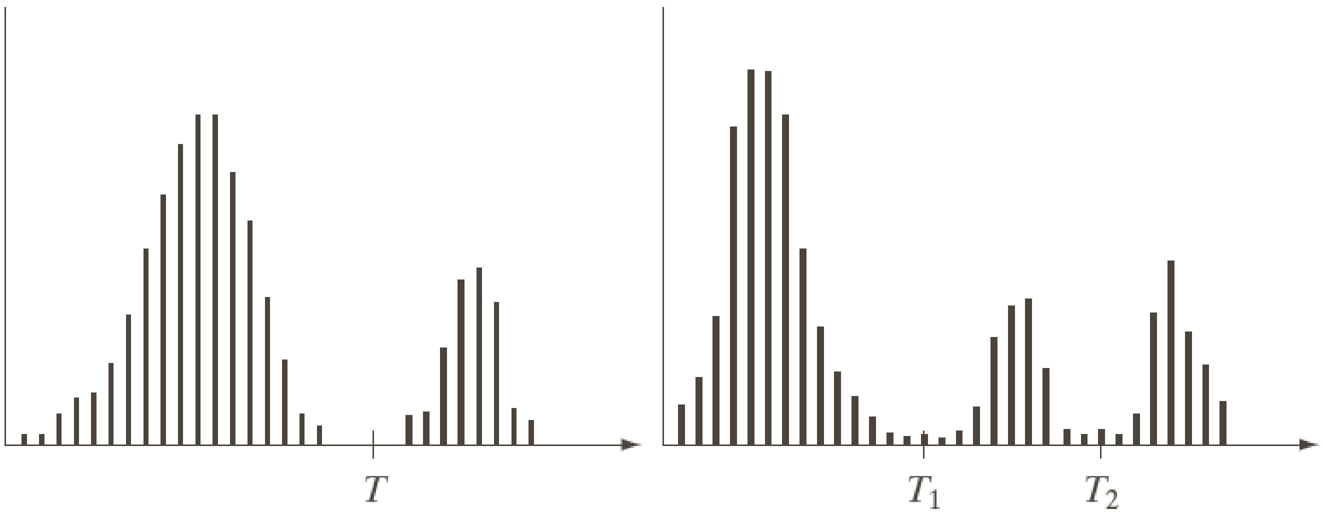
[1]The images in the rest of the slides are from Gonzalez & Woods' book.

Thresholding

$\mathcal{Q}$: How can we separate the object from the background?

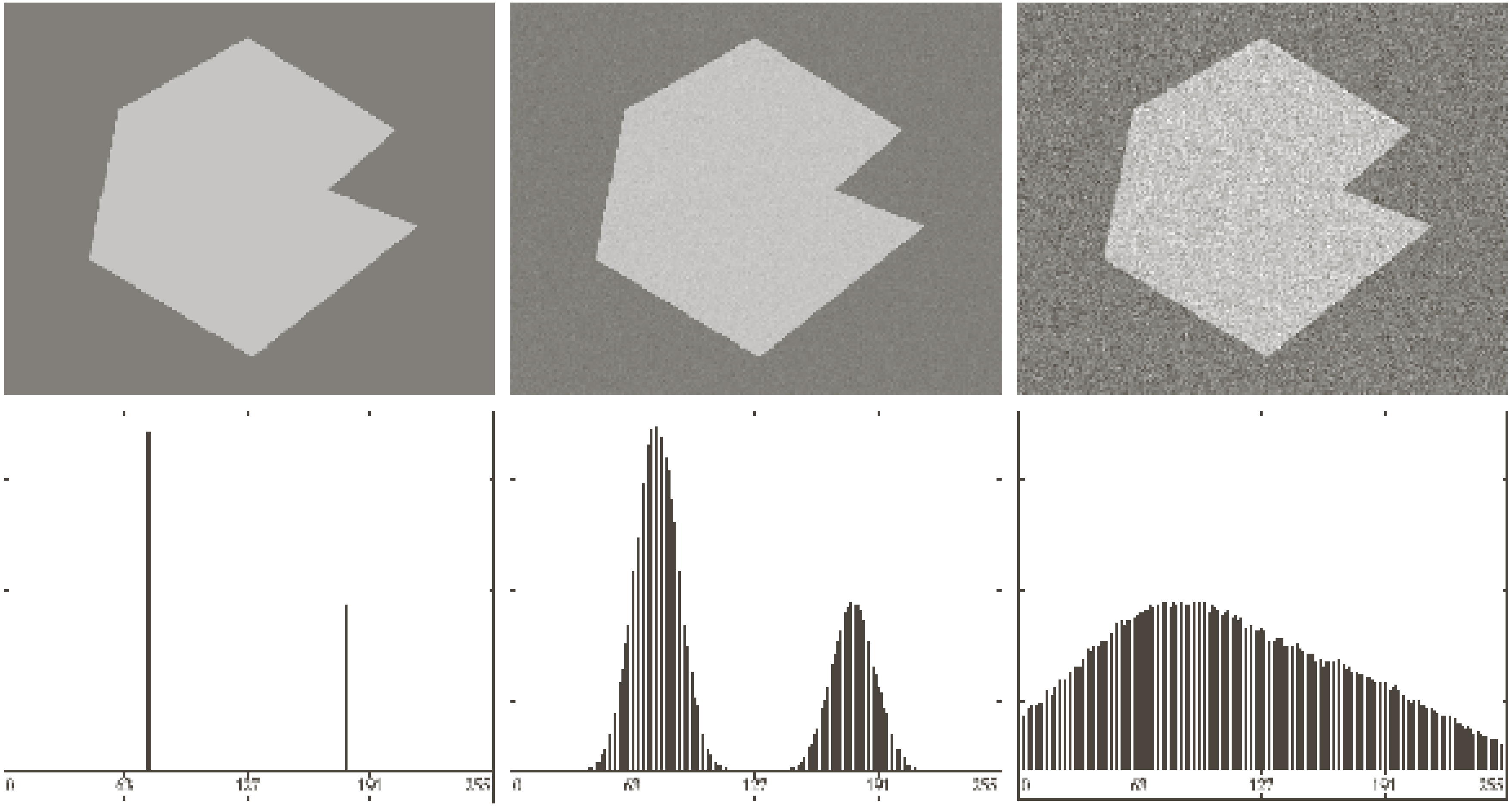
$\mathcal{A}$: Threshold the image.

Thresholding with Histograms



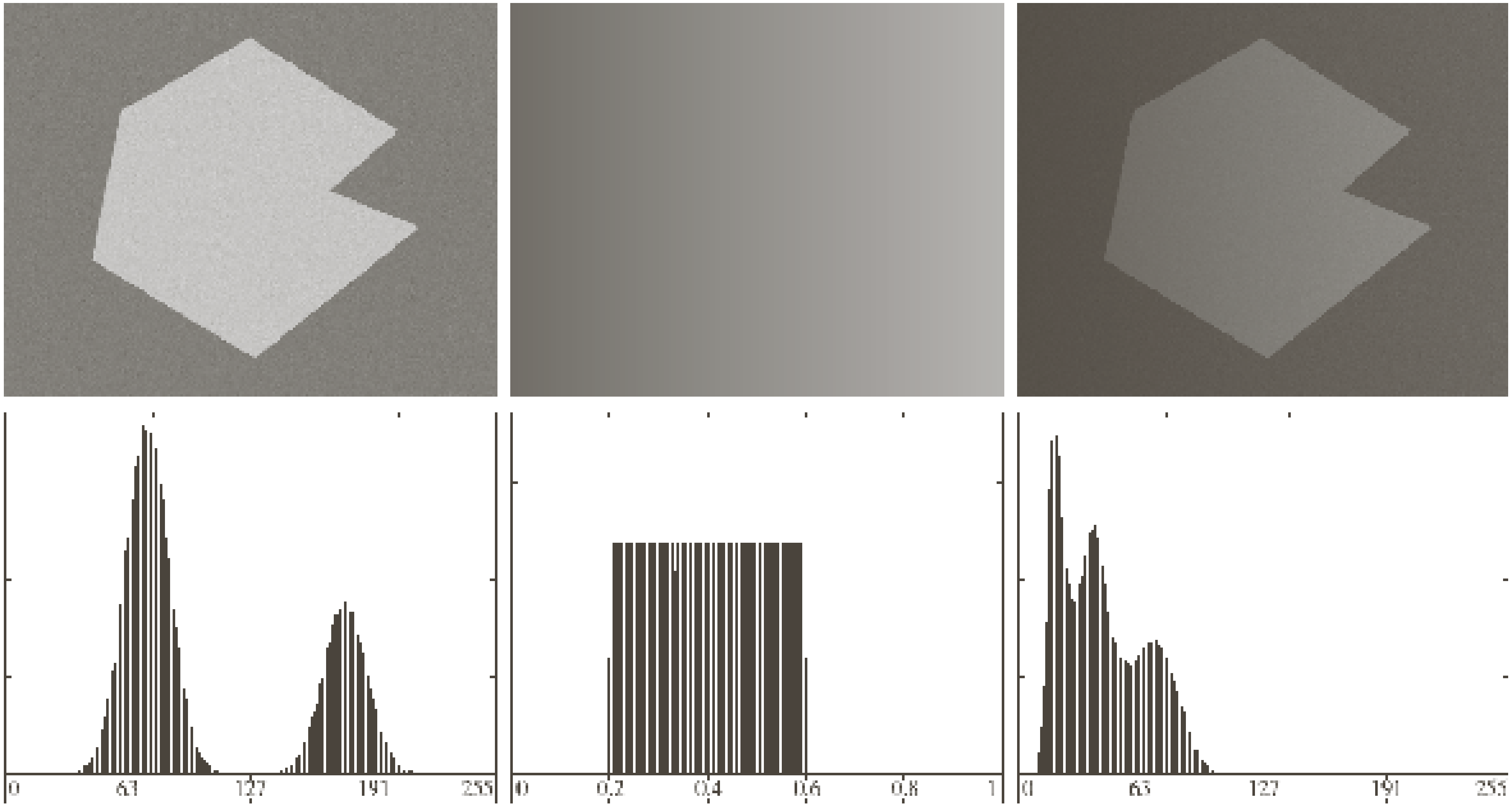
(a) Object and background (b) ?

Effects of Noise



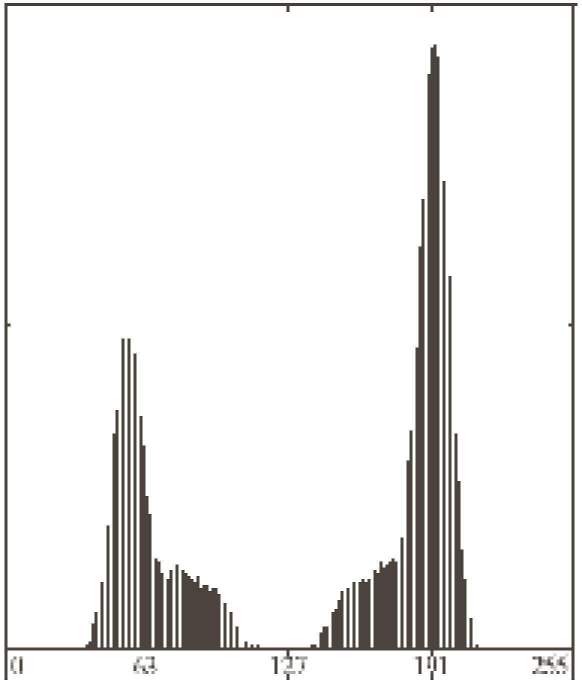
Additive noise smooths the histogram. (Why?)

The Role of Illumination and Reflectance



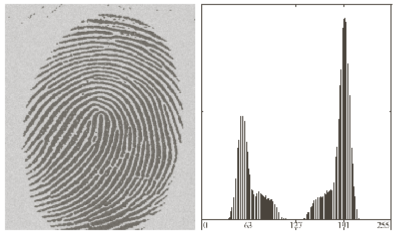
Suppose we have the image in the third column, which is obtained by imaging the source in the first column with a spatially varying illumination source.

Global Thresholding



Given the histogram of the image, how do we select the threshold automatically?

Global Thresholding



(1) Select an initial estimate, T.

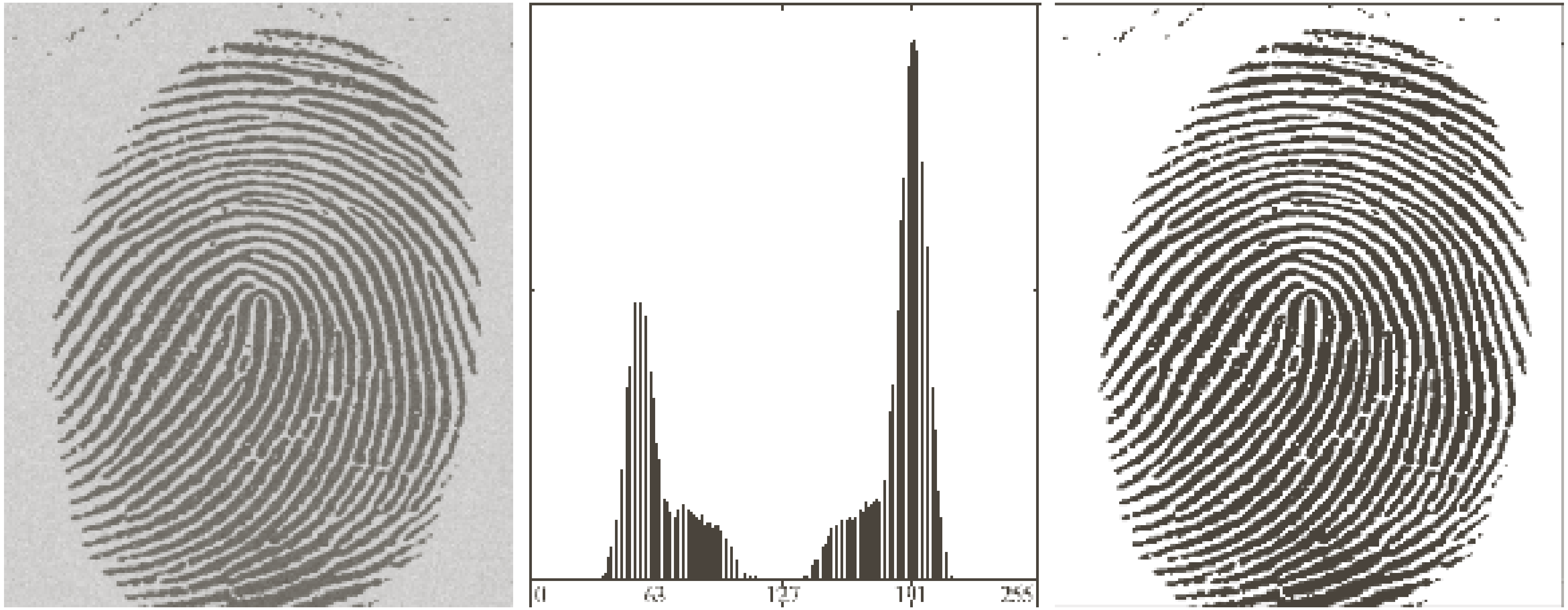
(2) Segment the image using T to create two images $G_1(x, y) = (g(x, y) > T)$, $G_2(x, y) = (g(x, y) \leq T)$.

(3) Compute the average intensity values m_1, m_2 for the pixels in G_1, G_2 respectively.

(4) Update $T = (m_1 + m_2)/2$.

(5) Repeat steps (2)–(4) until convergence.

Global Thresholding



Result of thresholding with the described algorithm.

Selection of the Threshold Using Otsu's Method

The method proposed by Otsu for threshold selection consists of finding the threshold that maximizes a certain metric, the so-called between-class variance, defined as,

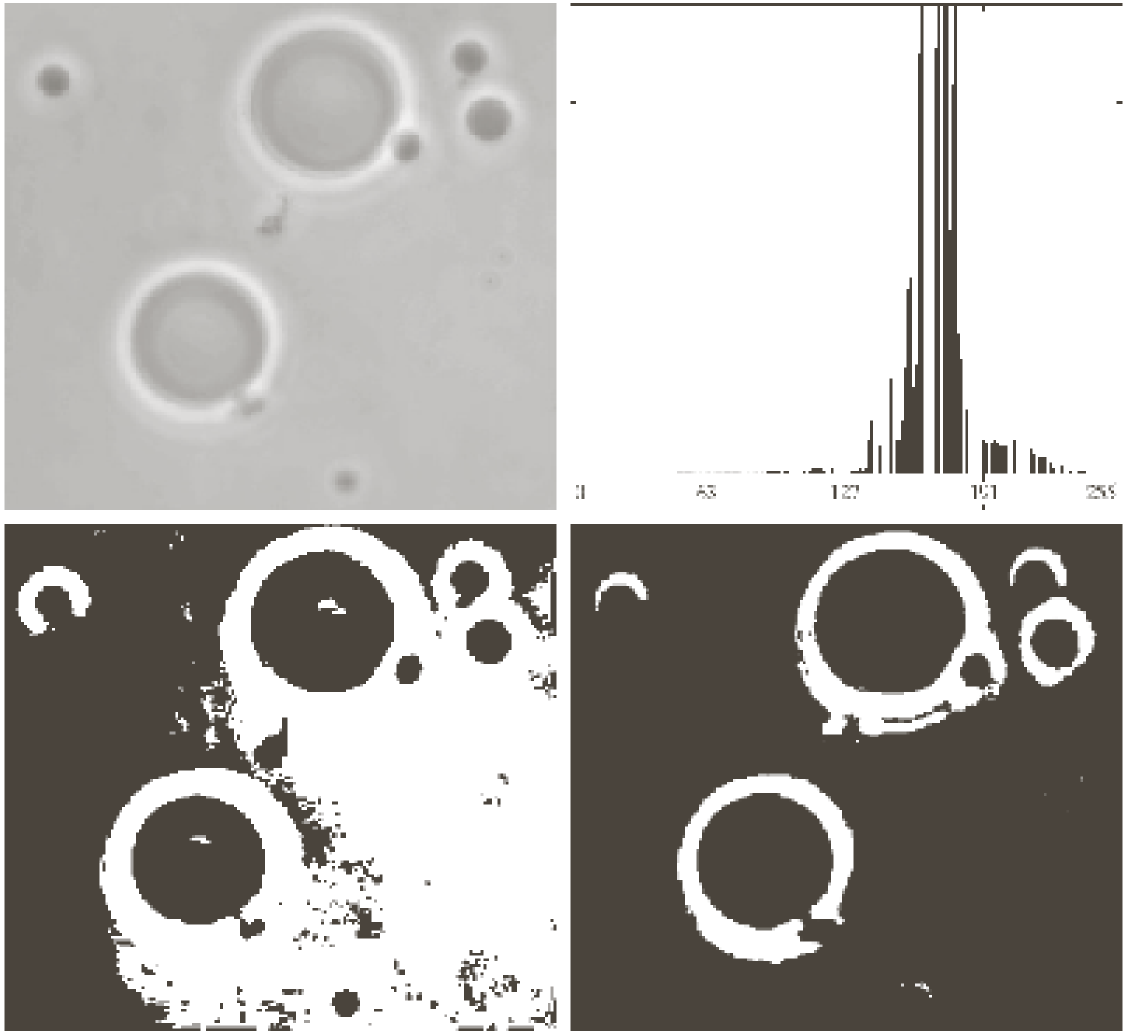
$$\sigma_B^2(T) = P_1(T)\left(m_1(T) - m\right)^2 + P_2(T)\left(m_2(T) - m\right)^2$$

where

- $P_1(T)$ is the probability that a pixel lies in G_1, using T as the threshold.
- $P_2(T)$ is the probability that a pixel lies in G_2, using T as the threshold.
- $m_1(T)$ is the mean value of the pixels from G_1, using T as the threshold.
- $m_2(T)$ is the mean value of the pixels from G_2, using T as the threshold.
- m is the mean of the whole image.

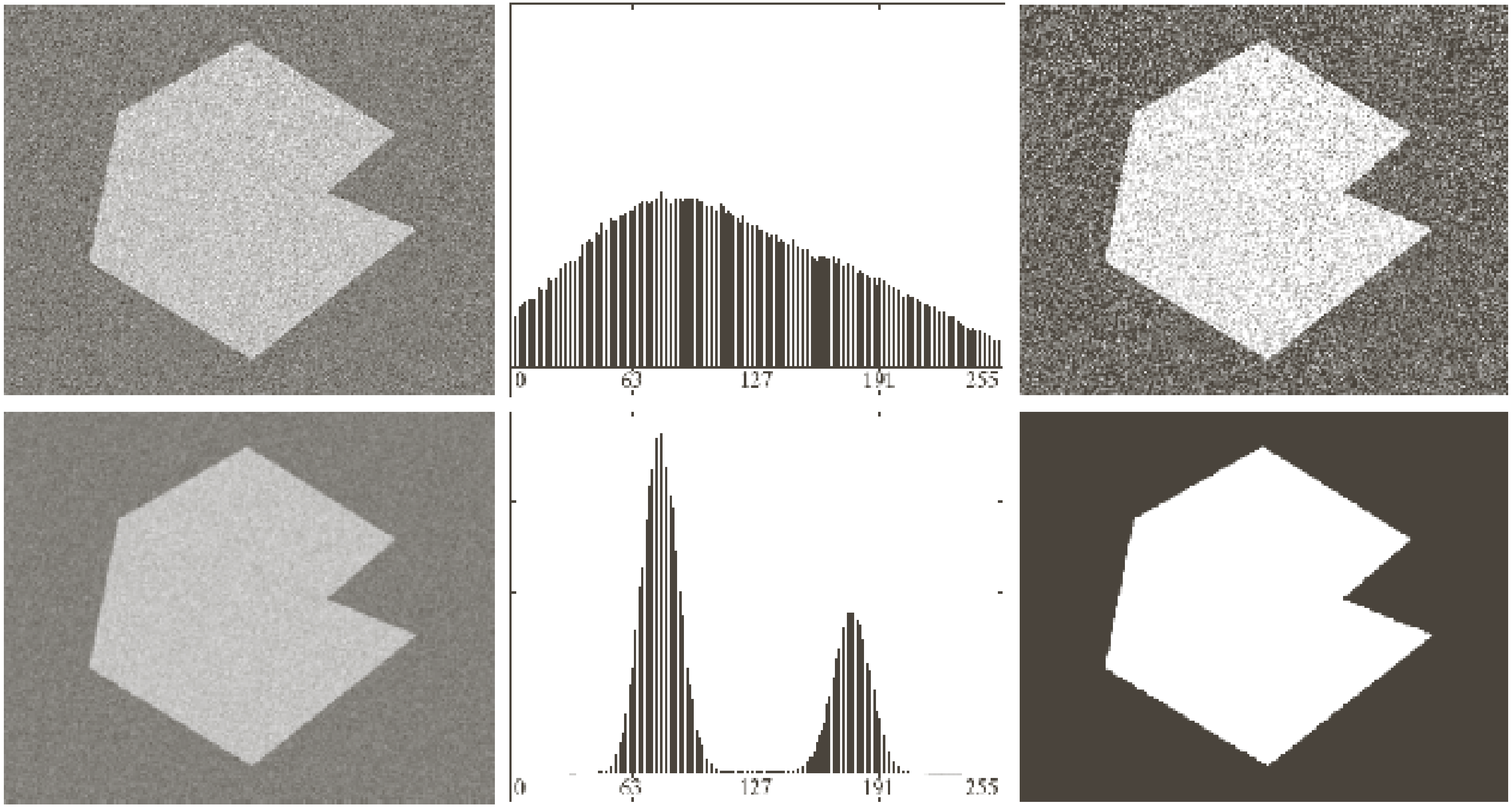
T is selected by computing $\sigma_B^2(T)$ for all possible thresholds (there are 256 of them for an 8-bit intensity resolution) and determining the maximizer.

Global Thresholding



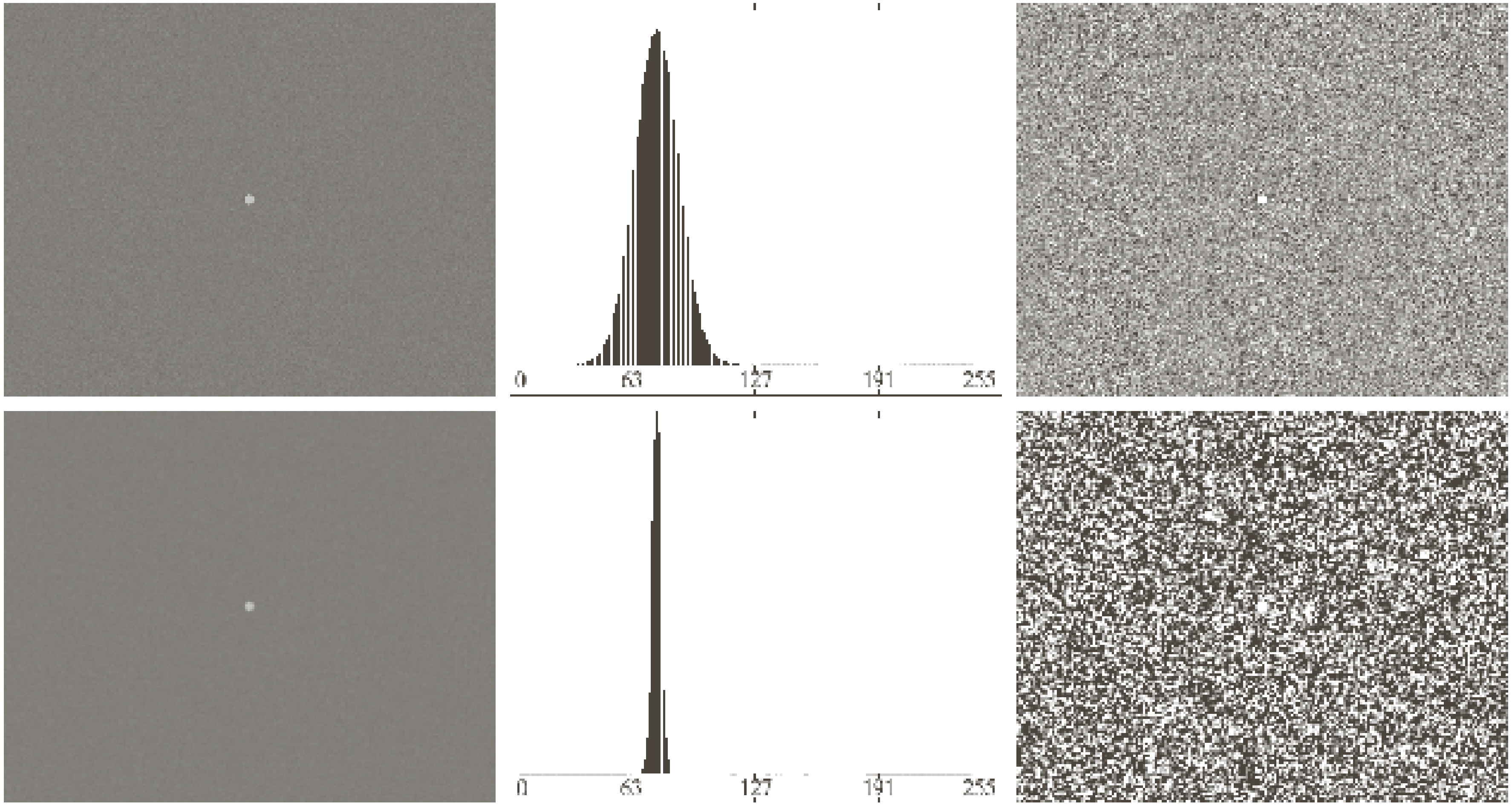
Result of thresholding with the two different threshold selection algorithms (right : Otsu's method).

Dealing with Noise – Smoothing



Threshold, selected with Otsu's method, applied with and without lowpass filtering the noisy image.

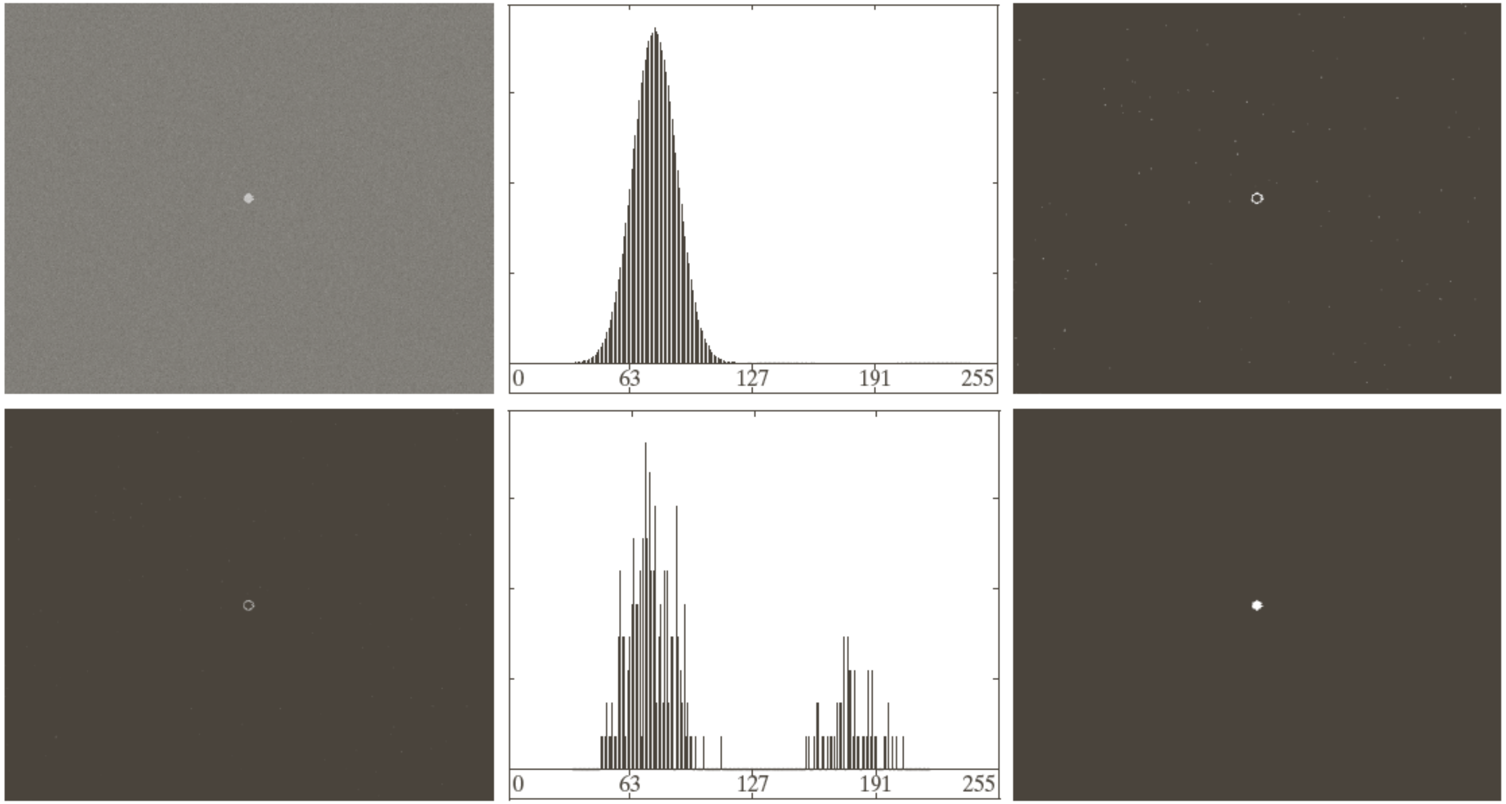
Dealing with Noise



Threshold selection based on the histogram of the whole image (lowpass filtered or not) is doomed to failure in this example. Why?

Dealing with Noise – Making use of Edges

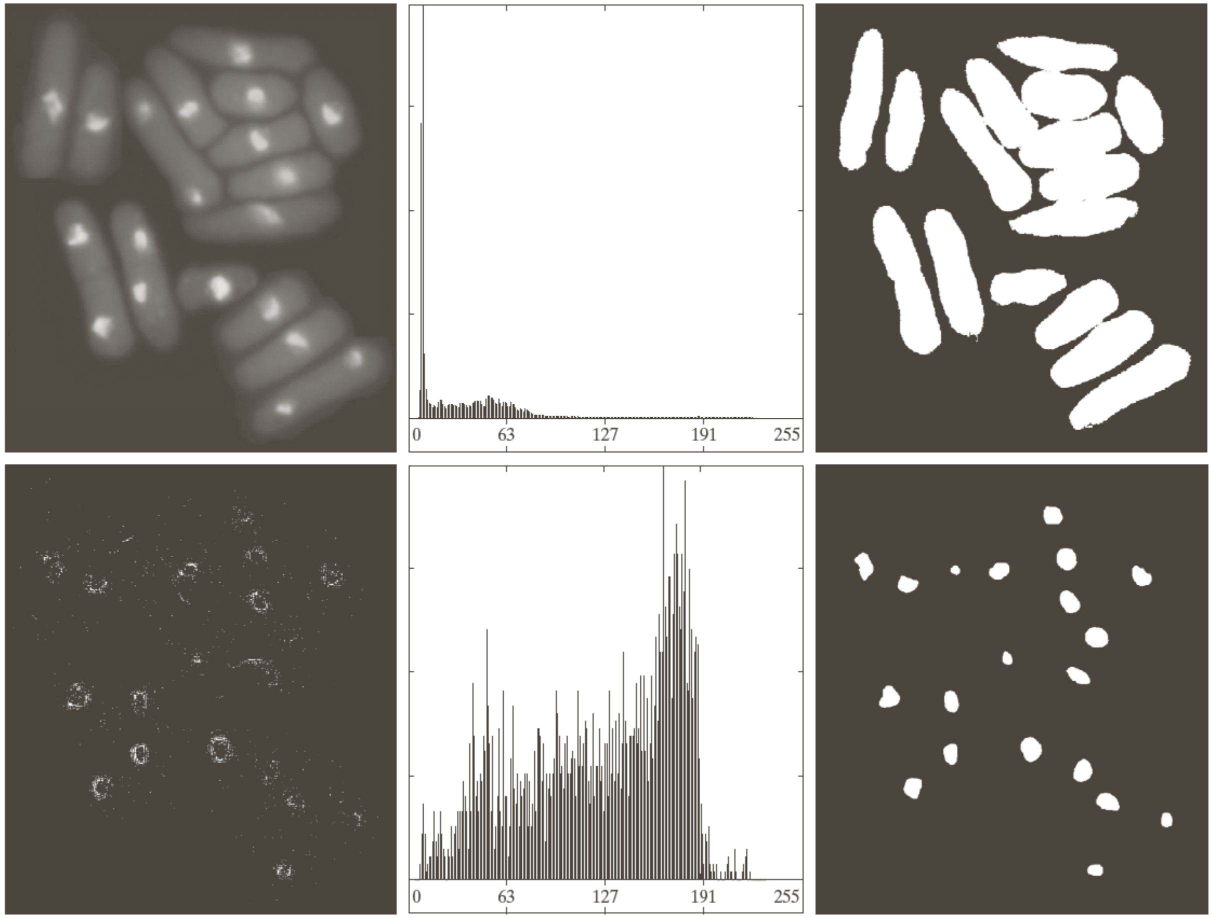
Idea : Form the histogram based on the pixels that lie on a neighborhood of the edges.



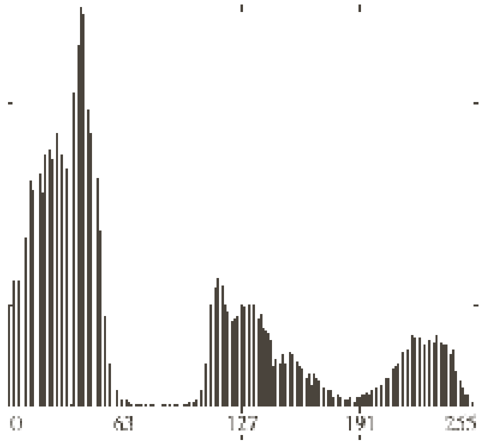
The histogram obtained by restricting attention to a neighborhood of the edge has two clearly distinguishable peaks.

Thresholding – Making use of Edges

Edge information can be useful in other scenarios as well. Here the goal is to segment the bright spots in the yeast cells.



Multiple Objects of Interest

How to segment in this case?

Multiple Thresholds

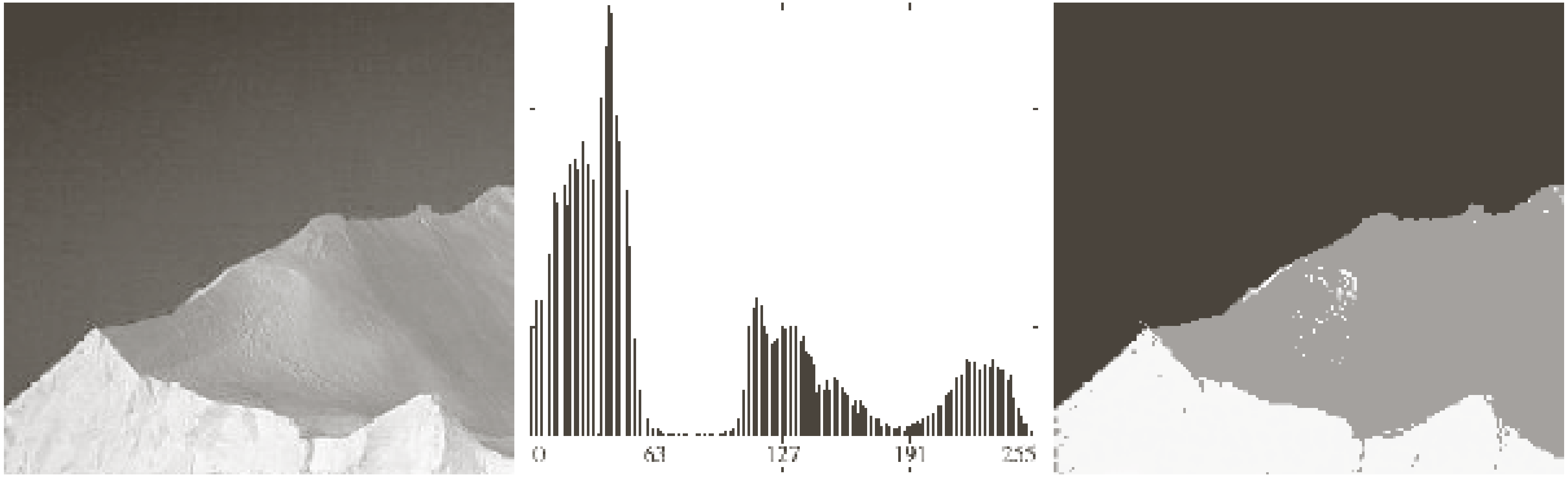
Otsu's threshold selection method can be generalized to K classes.

In this case, the metric to maximize is

$$\sigma_B^2(T_1, T_2, \ldots, T_K) = \sum_{k=1}^{K} P_k \left(m_k - m\right)^2$$

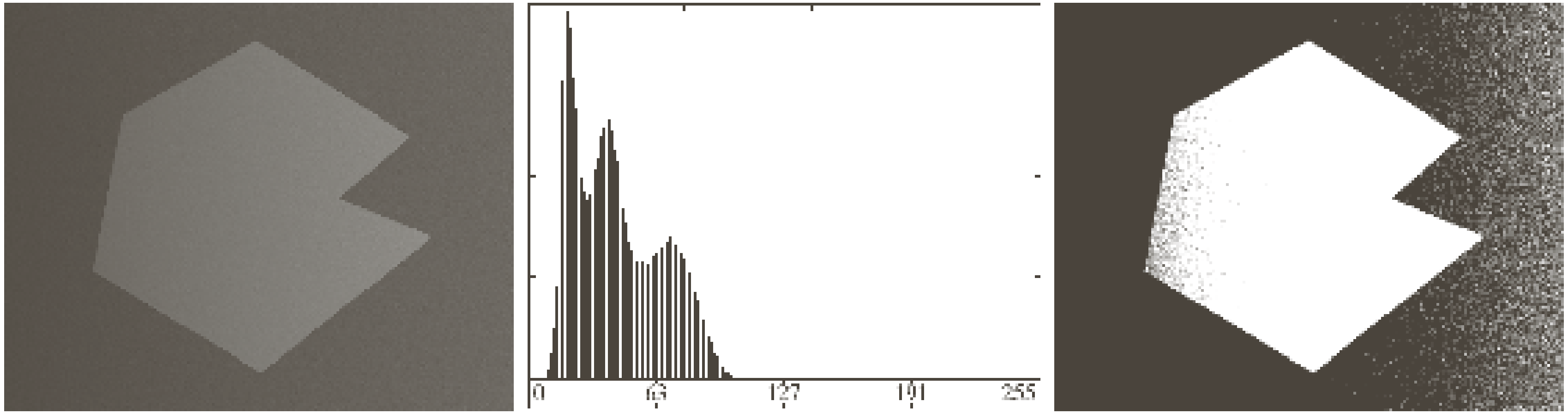
Determine the set $(T_1, \ldots, T_K)$ that maximizes σ_B^2.

Multiple Thresholds



What can go wrong?

Image Partitioning

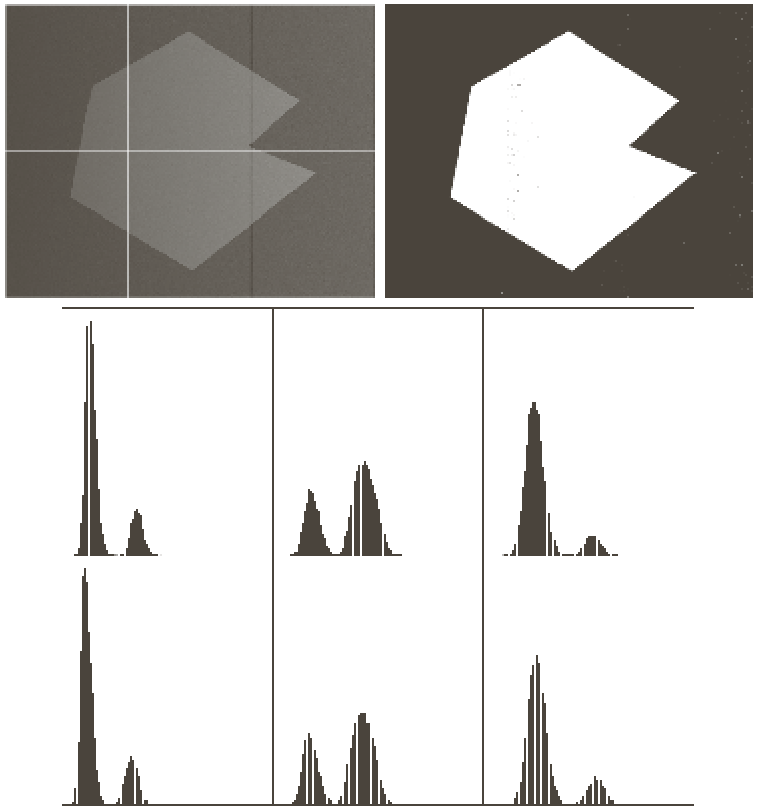


Noisy, shaded images with different characteristics in different regions do not allow the use of a single threshold.

What can we do?

Image Partitioning

Idea : Partition the image and apply a single threshold in each part independently.



Segmenting Images with Changing Characteristics