Digital Image Processing

Image Segmentation:
Point, Line and Edge Detection

By: Dr. Hafeez

Contents

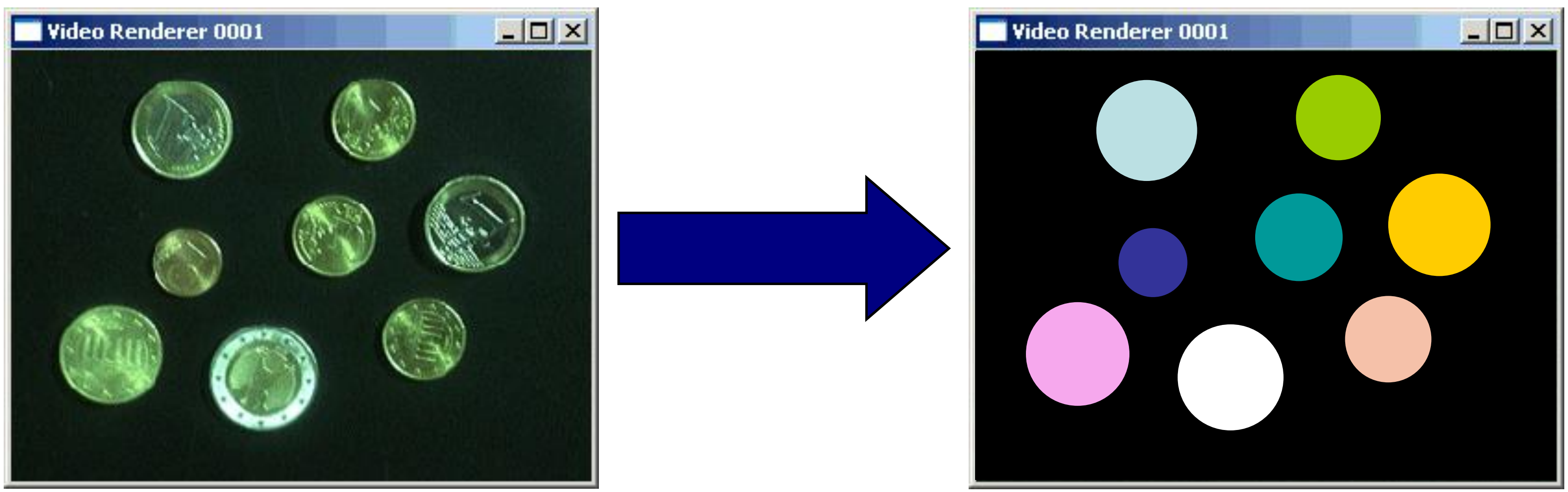
So far we have been considering image processing techniques used to transform images for **human interpretation**

Today we will begin looking at automated image analysis by examining the thorny issue of **image segmentation**:

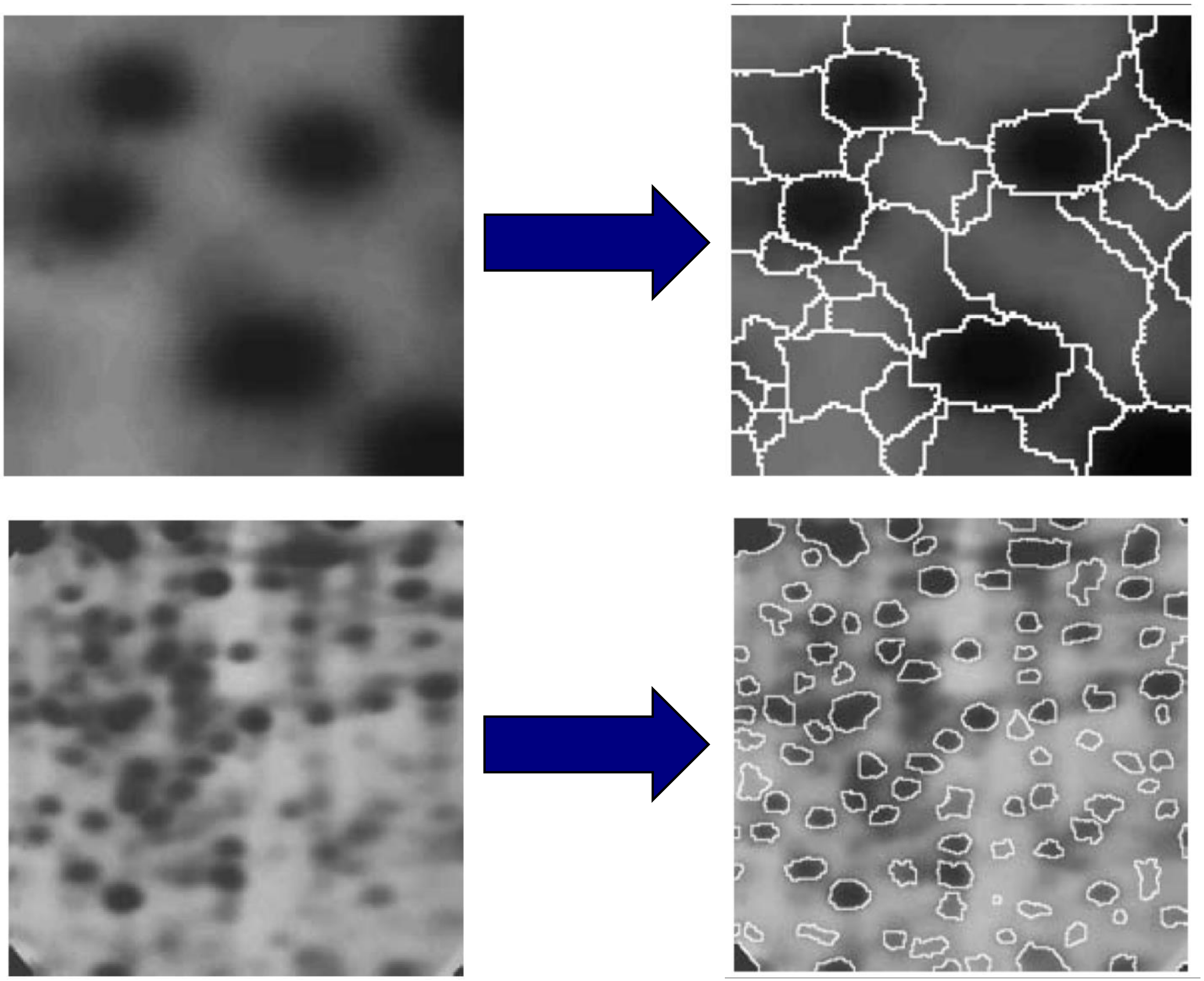
- The segmentation problem
- Finding **points**, **lines** and **edges**

The Segmentation Problem

- Segmentation attempts to partition the pixels of an image into groups that strongly correlate with the objects in an image
- Typically the first step in any automated computer vision application



Segmentation Examples



Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Detection Of Discontinuities

There are three basic types of grey level discontinuities that we tend to look for in digital images:

- Points
- Lines
- Edges

We typically find discontinuities using masks and correlation

Point Detection

Point detection can be achieved simply using the mask below:

Output 1 if:

$$R = \sum_{i=1}^{9} w_i z_i \qquad |R| \geq T$$

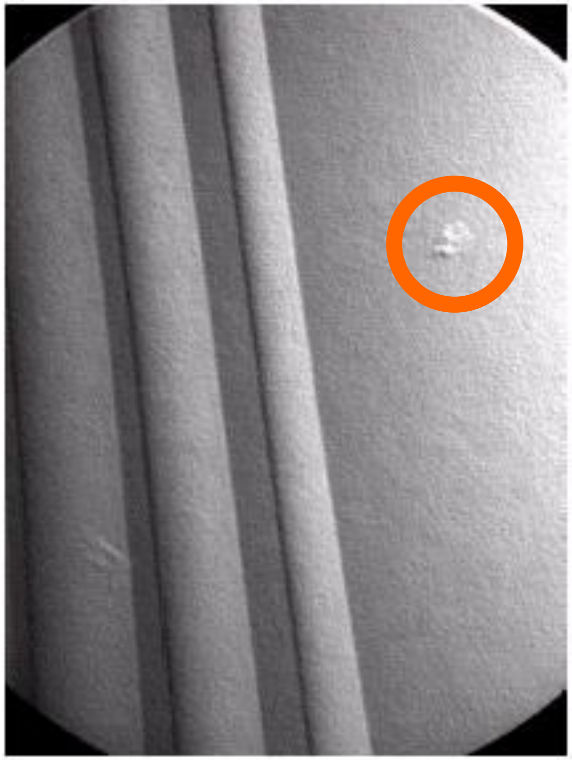
Else output 0.

-1	-1	-1
-1	8	-1
-1	-1	-1

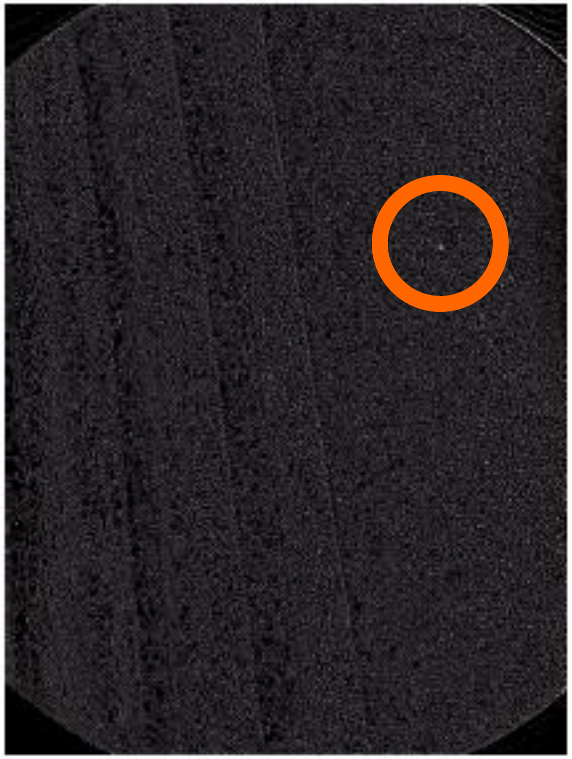
R is called Response of the filter.

Points are detected at those pixels in the subsequent filtered image that are above a set threshold

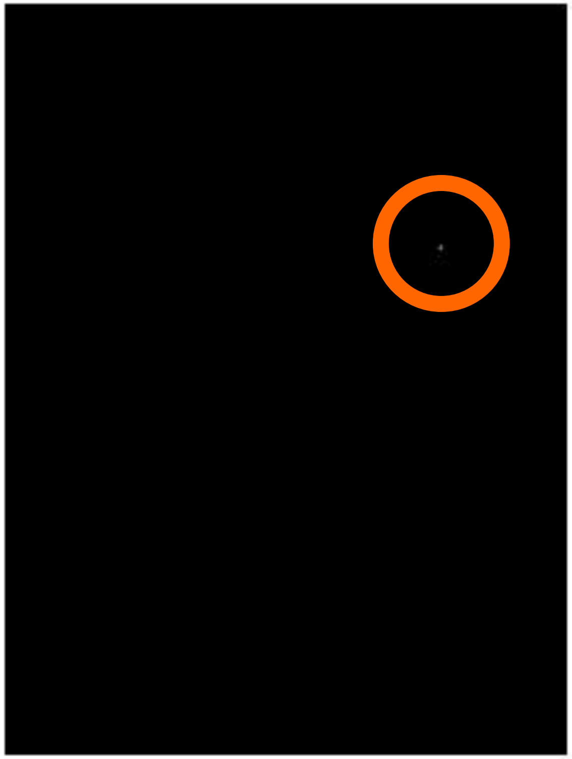
Point Detection (cont…)



X-ray image of a turbine blade



Result of point detection



Result of thresholding

Line Detection

The next level of complexity is to try to detect lines

The masks below will extract lines that are one pixel thick and running in a particular direction

−1	−1	−1
2	2	2
−1	−1	−1

Horizontal

−1	−1	2
−1	2	−1
2	−1	−1

+45°

−1	2	−1
−1	2	−1
−1	2	−1

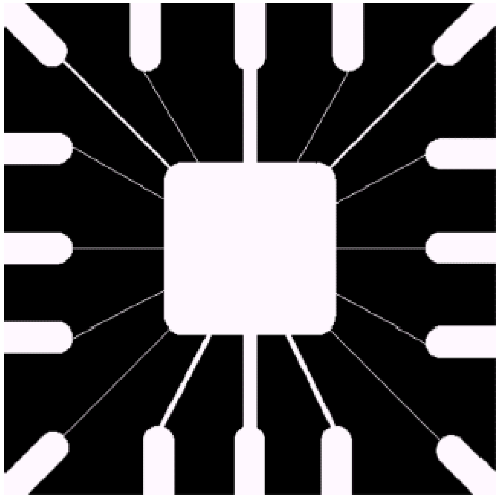
Vertical

2	−1	−1
−1	2	−1
−1	−1	2

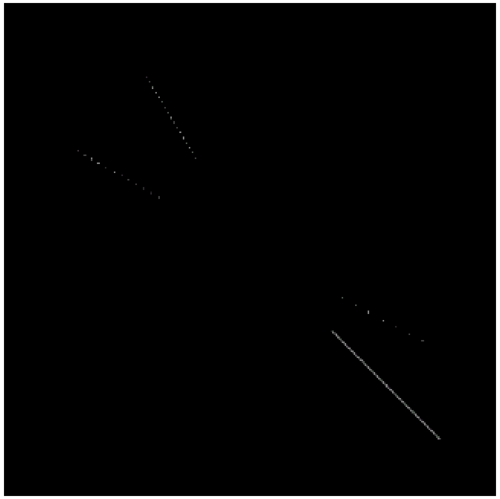
−45°

Line Detection (cont…)

Binary image of a wire bond mask



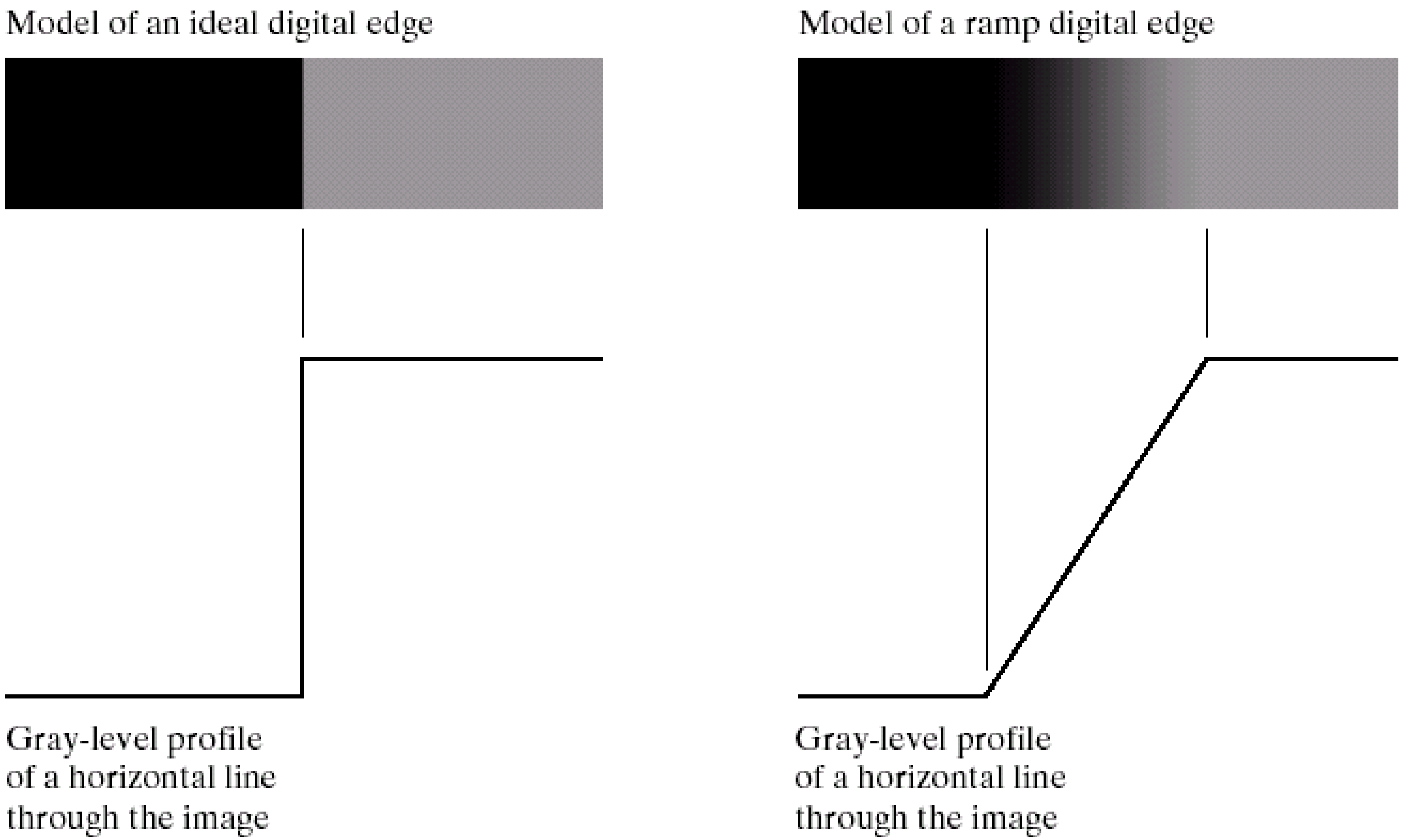
After processing with -45° line detector



Result of thresholding filtering result

Edge Detection

An edge is a set of **connected pixels that lie on the boundary** between two regions



Edge Point

We define a point in an image as being an **edge point** if its 2-D 1st order derivative is greater than a specified threshold.

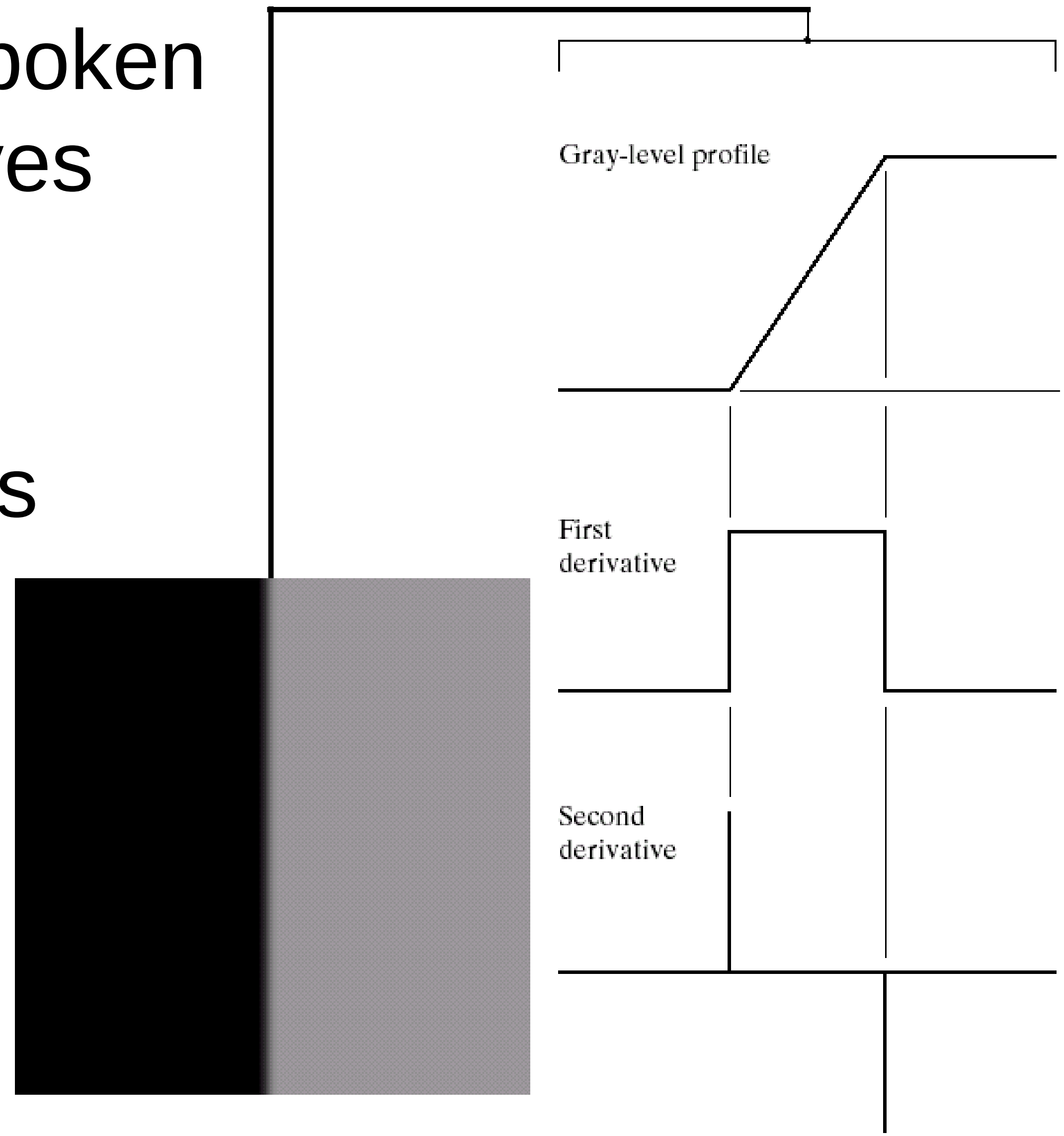
A **set of such points** that are connected according to a predefined criterion of connectedness is by definition an **edge**.

Edges & Derivatives

We have already spoken about how derivatives are used to find discontinuities

1^{st} derivative tells us where an edge is

2^{nd} derivative show double response with thin edges



Common Edge Detectors

Given a 3*3 region of an image the following edge detection filters can be used

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

−1	−1	−1
0	0	0
1	1	1

−1	0	1
−1	0	1
−1	0	1

Prewitt

−1	0
0	1

0	−1
1	0

Roberts

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

Sobel

Edge Detection Example

Original Image

Horizontal Gradient Component

Vertical Gradient Component

Combined Edge Image

Edge Detection Example

Edge Detection Example

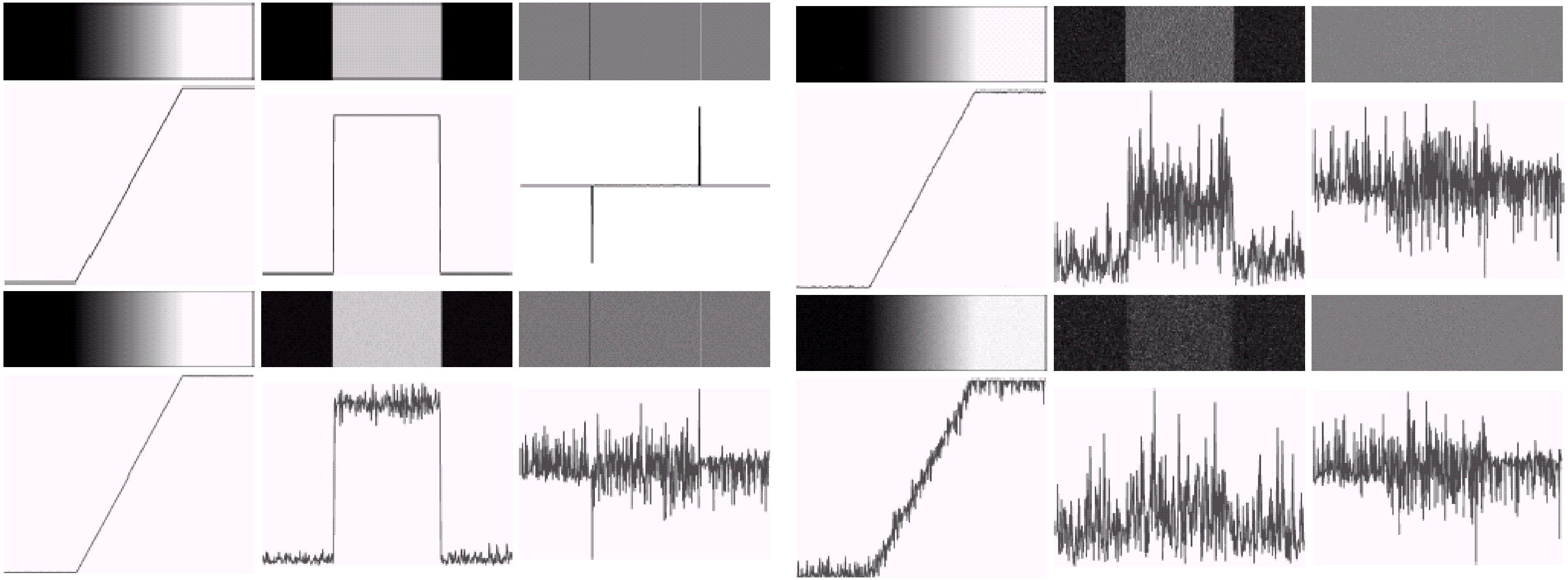
Edge Detection Example

Edge Detection Example

Derivatives & Noise

Derivative based edge detectors are extremely sensitive to noise

We need to keep this in mind



Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Edge Detection Problems

Often, problems arise in **edge detection** in that there is **too much detail**

For example, the brickwork in the previous example

One way to **overcome this** is to **smooth images** prior to edge detection

Laplacian Edge Detection

We encountered the 2nd-order derivative based Laplacian filter already

0	−1	0
−1	4	−1
0	−1	0

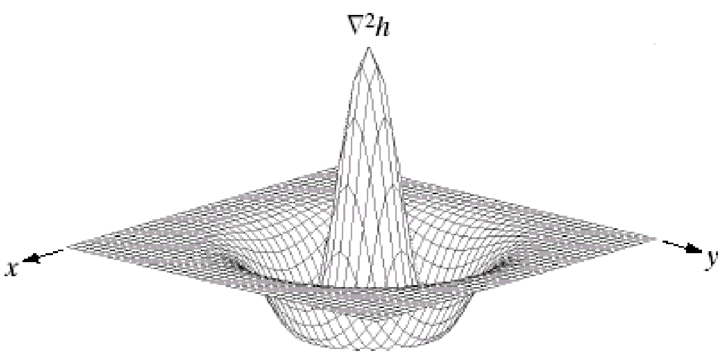
−1	−1	−1
−1	8	−1
−1	−1	−1

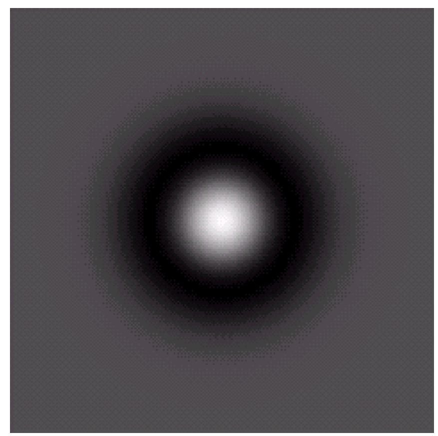
The Laplacian is typically not used by itself as it is too sensitive to noise

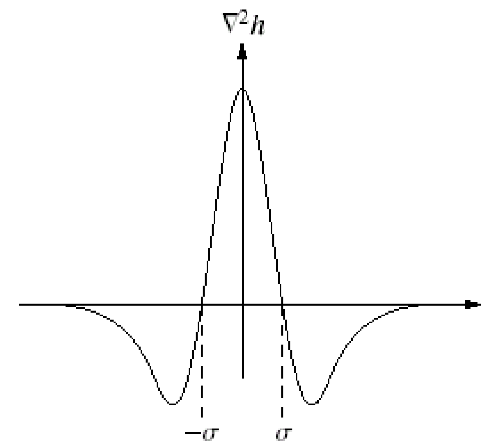
Usually used for edge detection the Laplacian is combined with a smoothing Gaussian filter

Laplacian Of Gaussian

The Laplacian of Gaussian (or Mexican hat) filter uses the Gaussian for noise removal and the Laplacian for edge detection





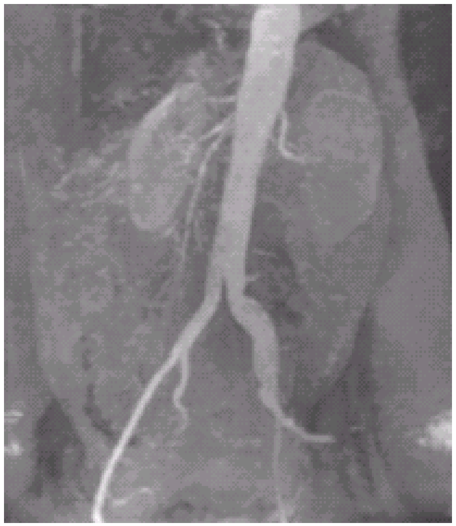


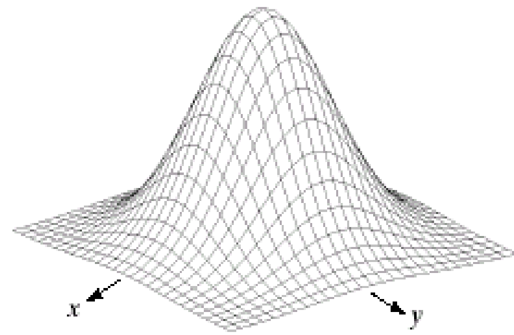
$$h(r) = -\exp(-r^2 / 2\sigma^2)$$

$$\nabla^2 h(r) = -\left[\frac{r^2 - \sigma^2}{\sigma^4}\right]\exp(-r^2 / 2\sigma^2)$$

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

Laplacian Of Gaussian Example





−1	−1	−1
−1	8	−1
−1	−1	−1

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Thresholding

Thresholding is usually the first step in any segmentation approach

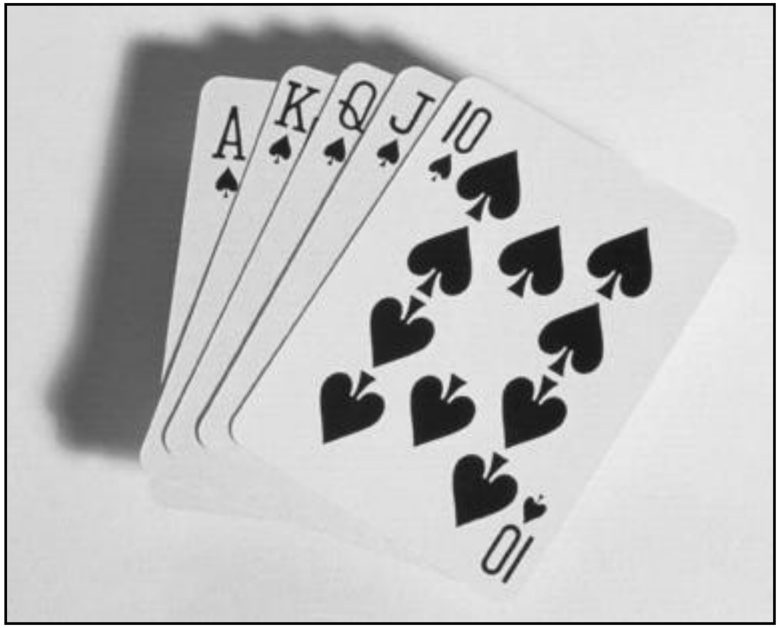
We have talked about simple single value thresholding already

Single value thresholding can be given mathematically as follows:

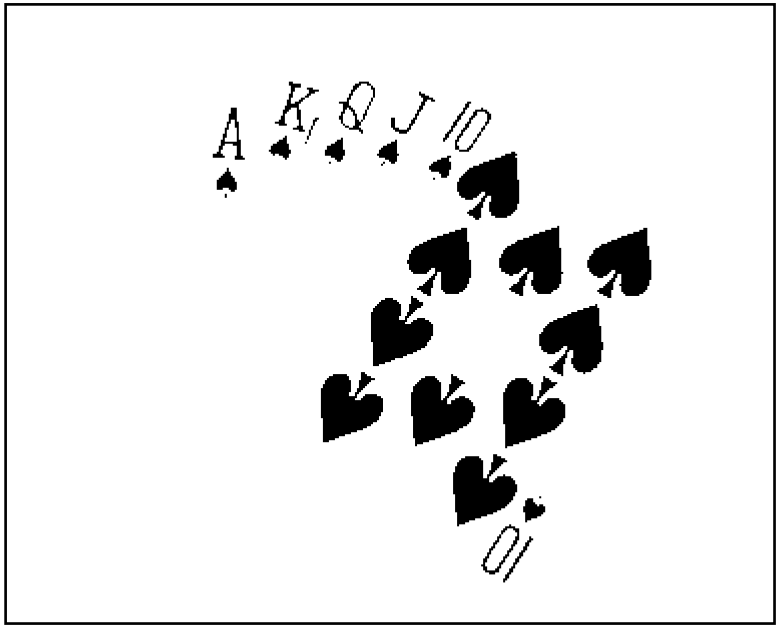
$$g(x, y) = \begin{cases} 1 \; if \;\; f(x, y) > T \\ 0 \; if \;\; f(x, y) \le T \end{cases}$$

Thresholding Example

Imagine a poker playing robot that needs to visually interpret the cards in its hand



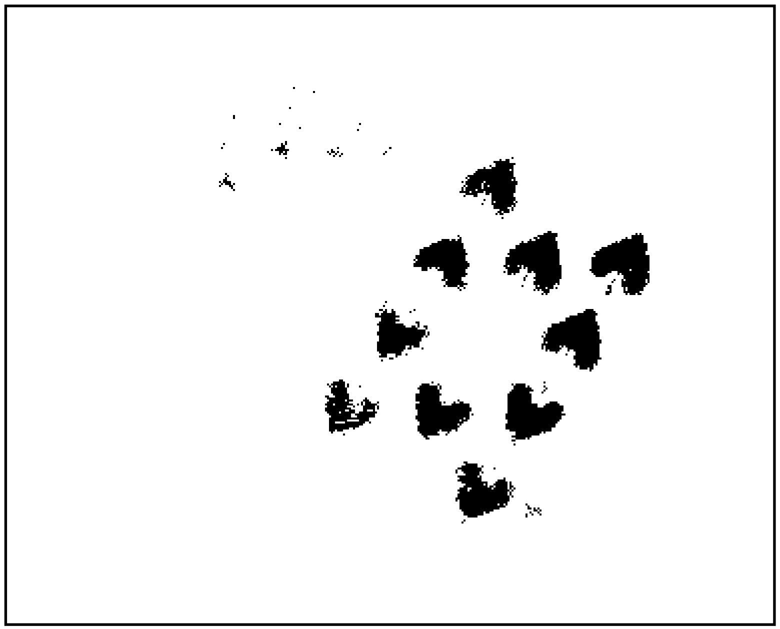
Original Image



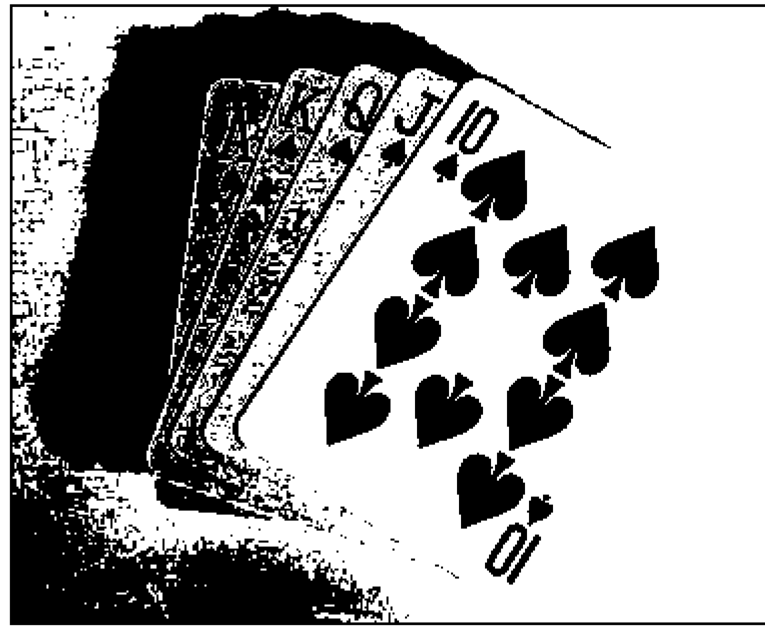
Thresholded Image

But Be Careful

If you get the threshold wrong the results can be disastrous



Threshold Too Low



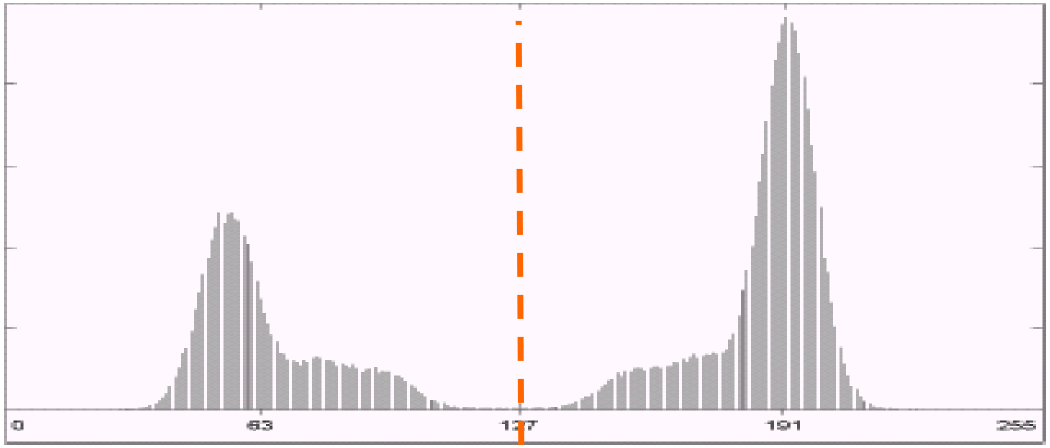
Threshold Too High

Basic Global Thresholding

Based on the histogram of an image

Partition the image histogram using a single **global threshold**

The success of this technique very strongly depends on how well the histogram can be partitioned



Basic Global Thresholding Algorithm

The basic global threshold, T, is calculated as follows:

1. Select an initial estimate for T (typically the average grey level in the image)
2. Segment the image using T to produce two groups of pixels: G_1 consisting of pixels with grey levels >T and G_2 consisting pixels with grey levels ≤ T
3. Compute the average grey levels of pixels in G_1 to give μ_1 and G_2 to give μ_2

Basic Global Thresholding Algorithm

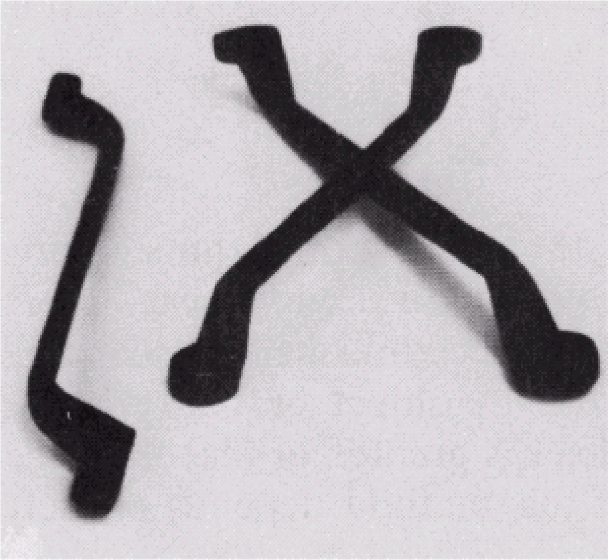
4. Compute a new threshold value:

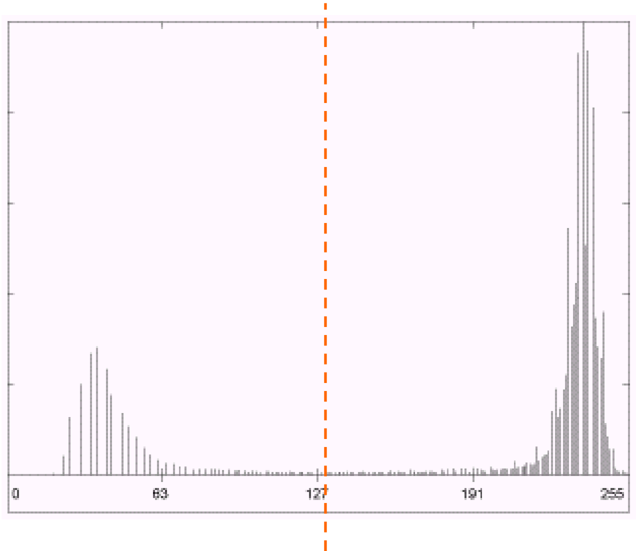
$$T = \frac{\mu_1 + \mu_2}{2}$$

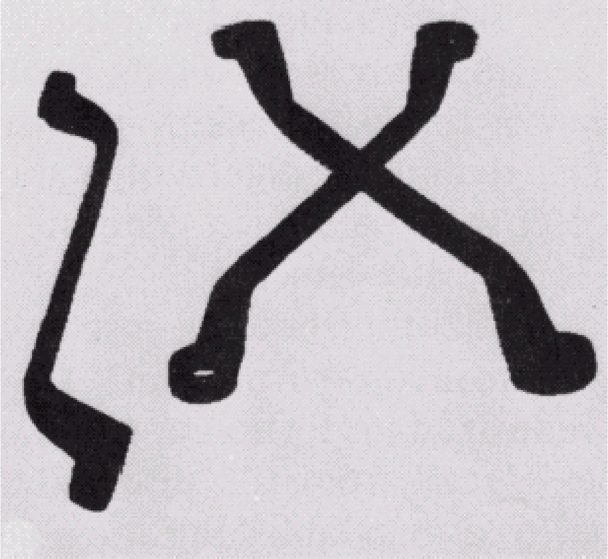
5. Repeat steps 2 – 4 until the difference in T in successive iterations is less than a predefined limit ΔT

This algorithm works very well for finding thresholds when the histogram is suitable

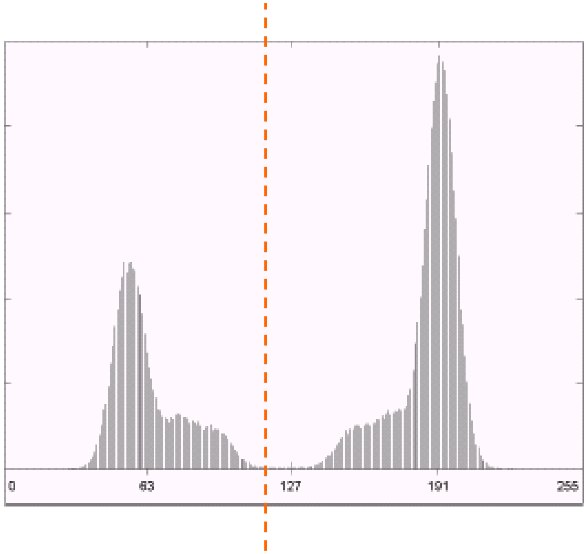
Thresholding Example 1





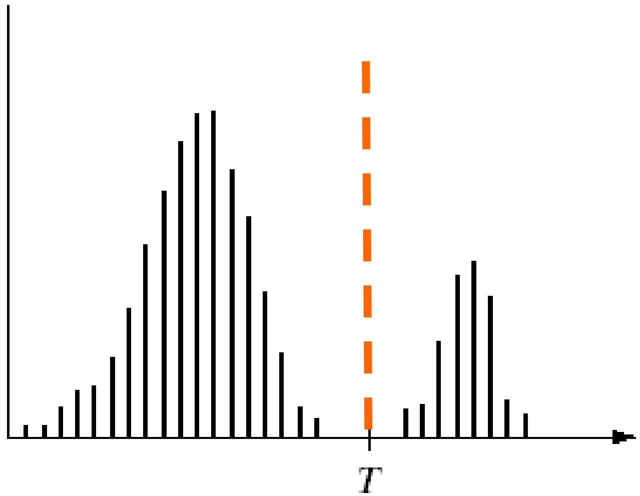


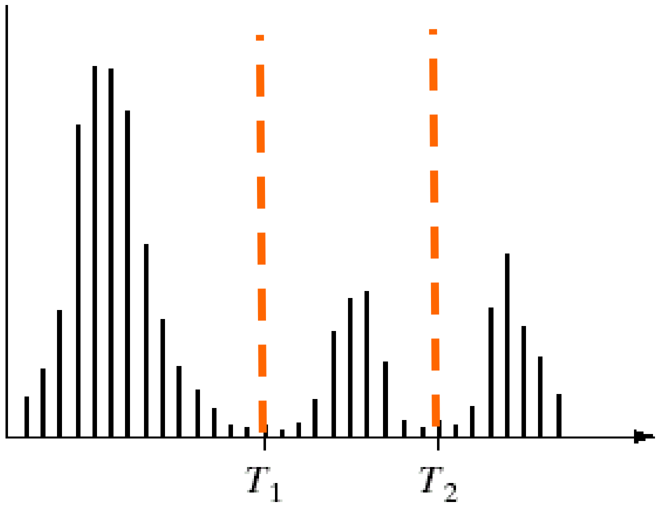
Thresholding Example 2



Problems With Single Value Thresholding

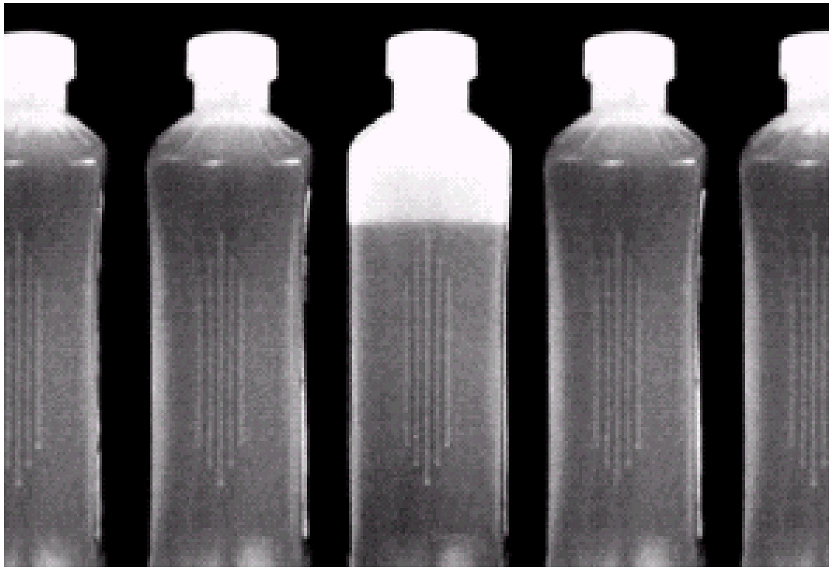
- Single value thresholding only works for bimodal histograms
- Images with other kinds of histograms need more than a single threshold





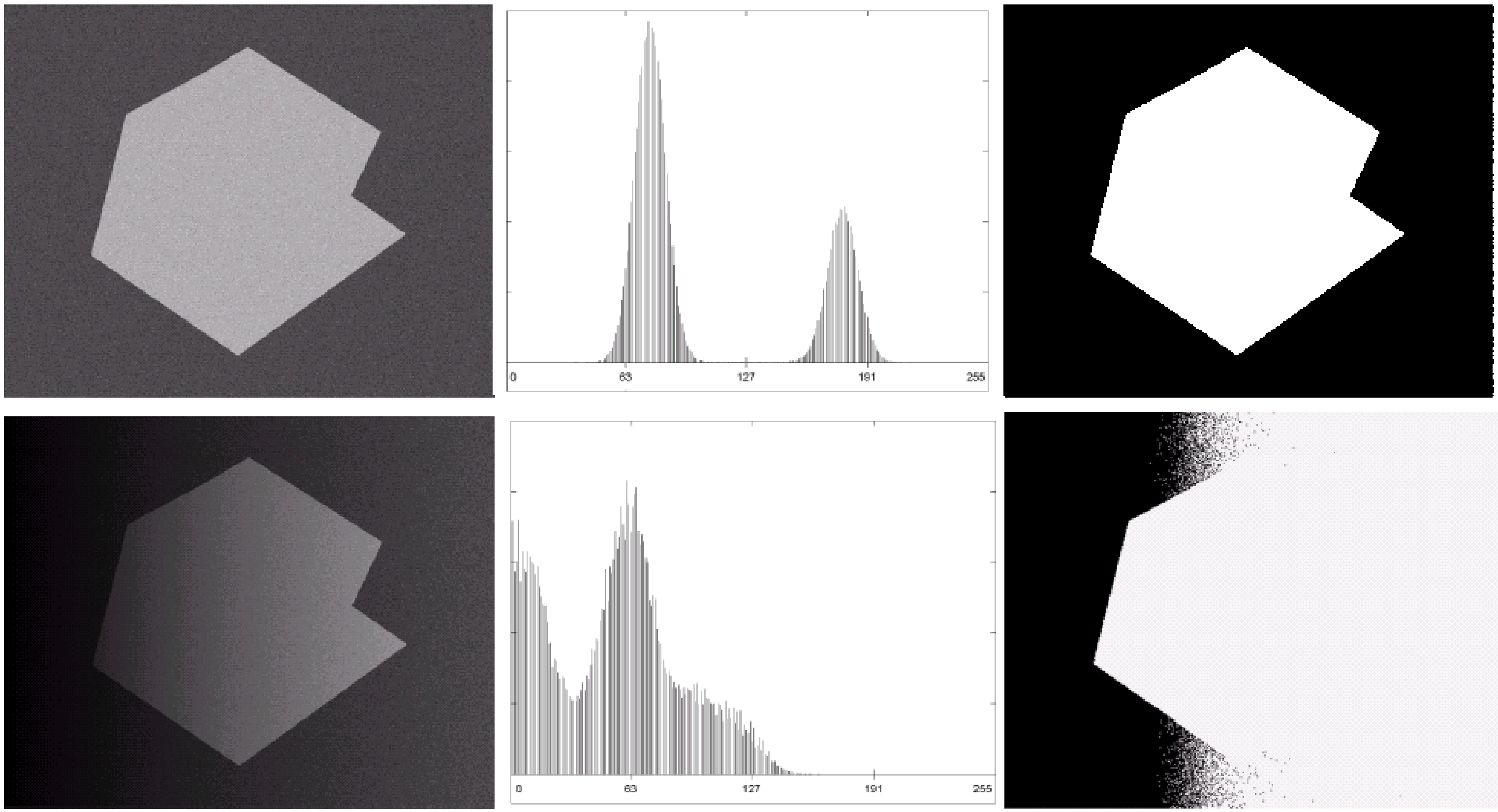
Problems With Single Value Thresholding (cont…)

- Let's say we want to isolate the contents of the bottles
- Think about what the histogram for this image would look like



- What would happen if we used a single threshold value?

Single Value Thresholding and Illumination



Uneven illumination can really upset a single valued thresholding scheme

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

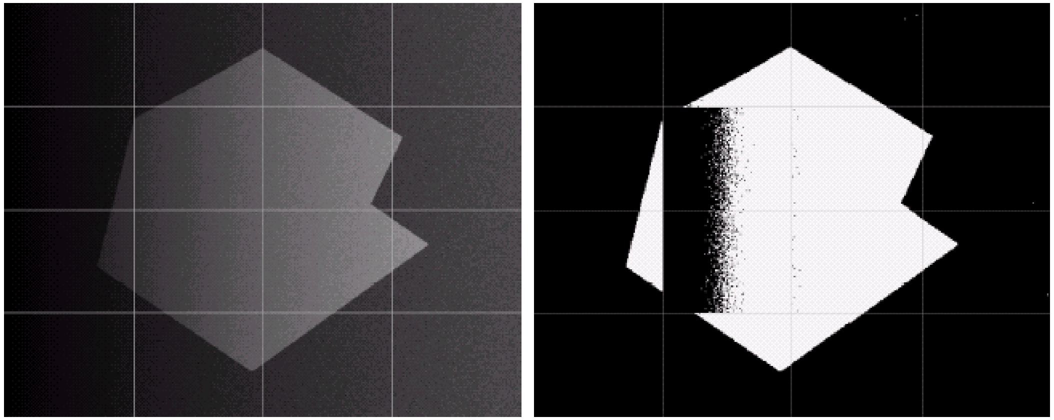
Basic Adaptive Thresholding

An approach to handling situations in which single value thresholding will not work is to **divide an image into sub images and threshold these individually**

Since the threshold for each pixel depends on its location within an image this technique is said to *adaptive*

Basic Adaptive Thresholding Example

The image below shows an example of using adaptive thresholding with the image shown previously



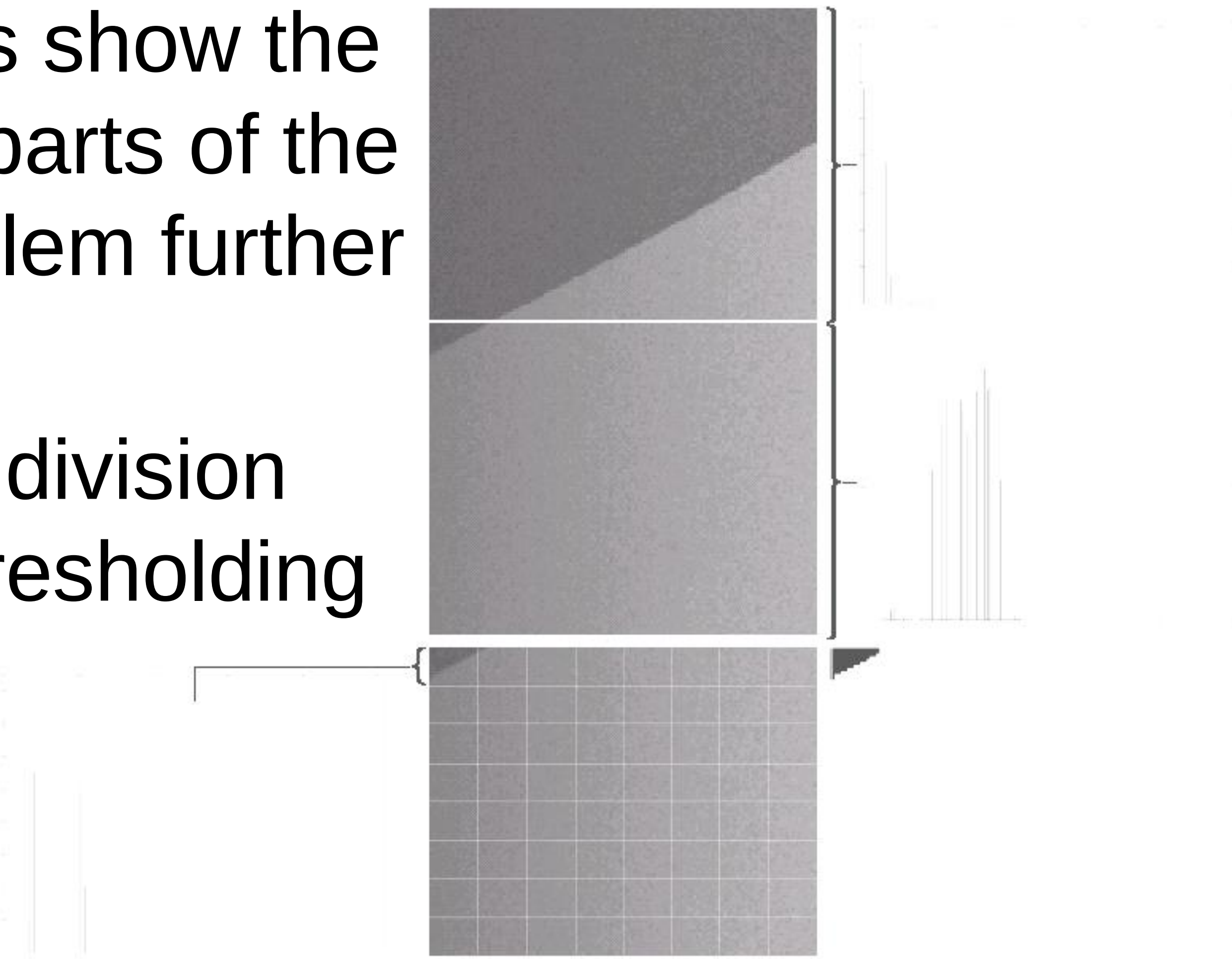
As can be seen success is mixed

But, we can further subdivide the troublesome sub images for more success

Basic Adaptive Thresholding Example (cont…)

These images show the troublesome parts of the previous problem further subdivided

After this sub division successful thresholding can be achieved



Summary

In this lecture we have begun looking at segmentation, and in particular edge detection

Edge detection is massively important as it is in many cases the first step to object recognition

We have also looked at the very basic type of segmentation i.e., Thresholding and what it offers.