Image Segmentation

Anas Toma

(Slides are based on Rafael C. Gonzalez and Richard E. Woods, Sufyan Samara, and Samer Arandi)

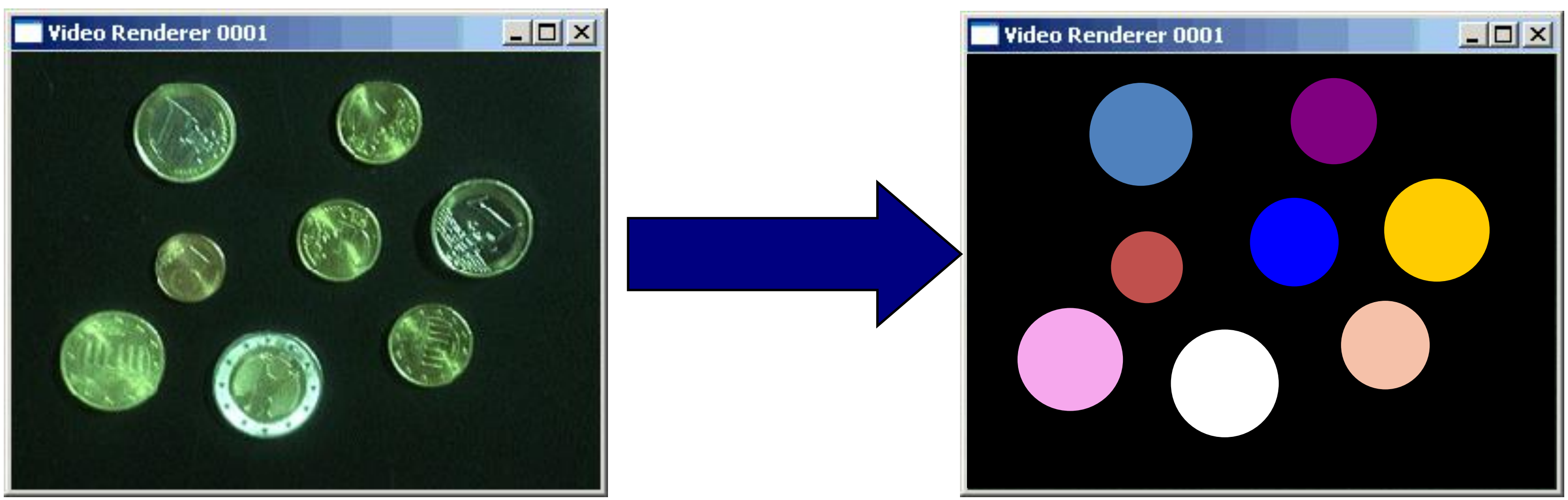
Images are from the Digital Image Processing textbook, see slide 7

Introduction

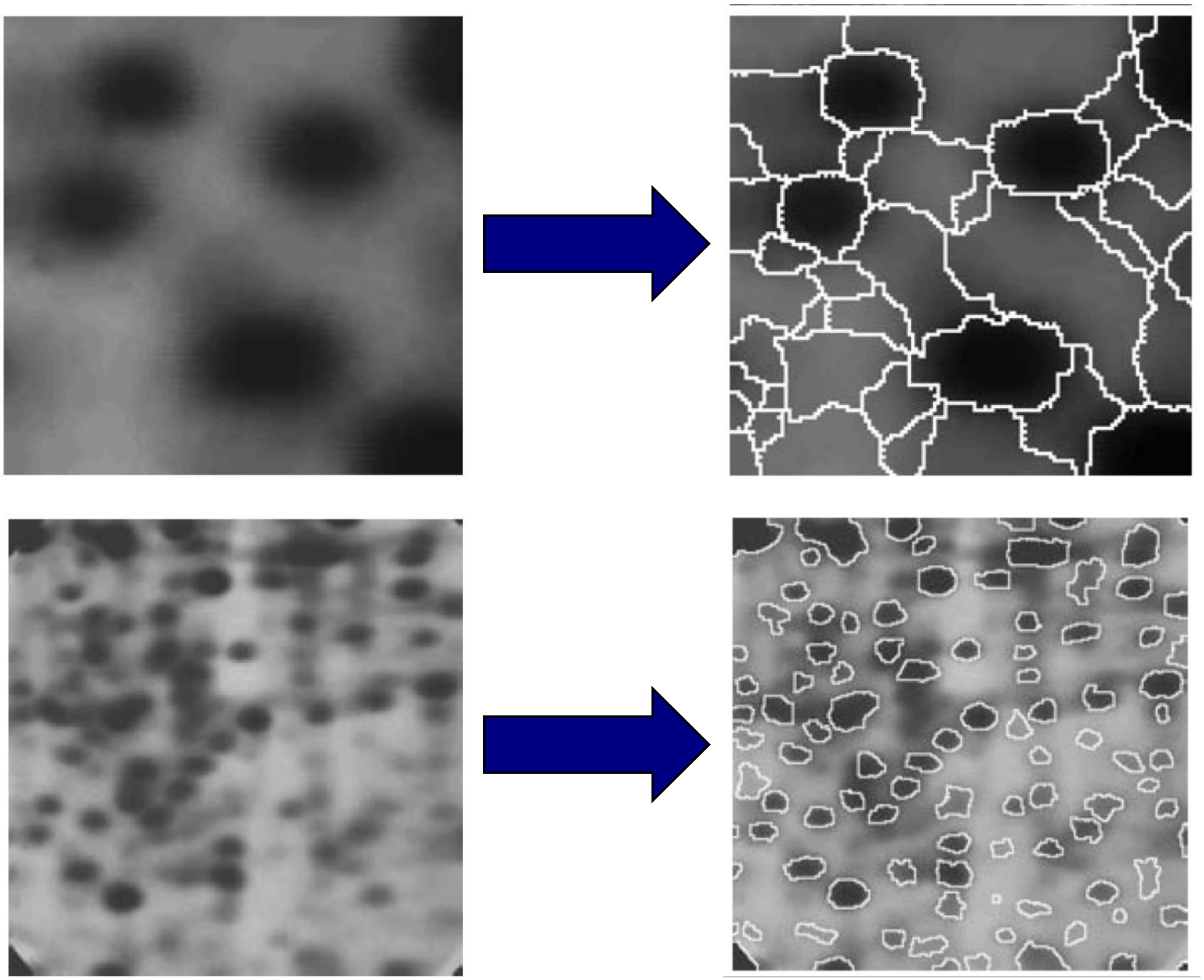
- So far we have been considering image processing techniques used to **transform images for human interpretation**
- We will begin looking at **automated image analysis** by examining the image segmentation:
  - The segmentation problem
  - Finding points, lines and edges

Image Segmentation

- Segmentation is to **subdivide an image into its component regions or objects**.
- It should stop when the **objects of interest** in an application have been **isolated**.
- Typically the **first step** in any automated **computer vision** application



Segmentation Examples



Principal approaches

- Segmentation algorithms generally are based on one of **2 basis properties of intensity values**
  - **Discontinuity** : to partition an image based on **sharp changes** in intensity (such as edges)
  - **Similarity** : to partition an image into **regions that are similar** according to a set of **predefined criteria**.

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:

−1	−1	−1
−1	8	−1
−1	−1	−1

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

Point Detection

−1	−1	−1
−1	8	−1
−1	−1	−1

- A point can be detected at the location on which the mask is centered if

$$|R| \geq T$$

- where
  - T is a **nonnegative threshold**
  - R is the **sum of products** of the coefficients with the gray levels contained in the region encompassed by the mark.

Point Detection

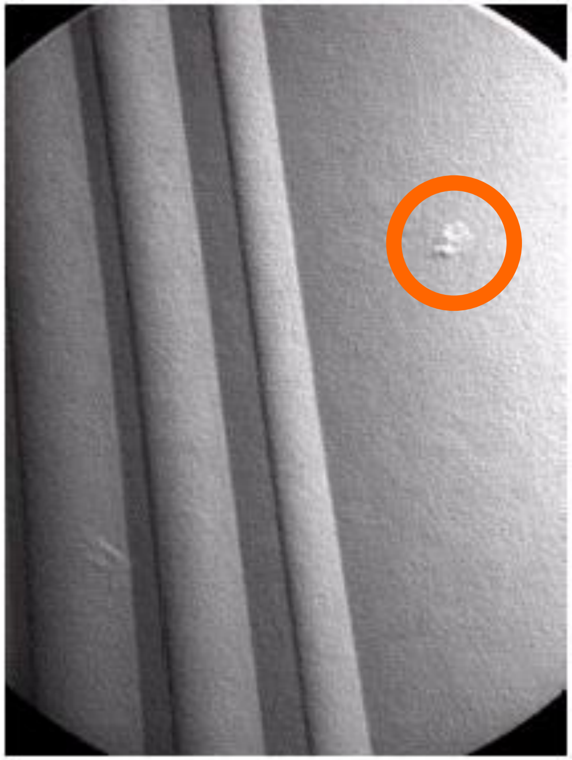
- Note that the mask is the same as the mask of **Laplacian Operation** (in chapter 3)

- The only differences that are considered of interest are those **large enough** (as determined by T) to be considered **isolated points**.

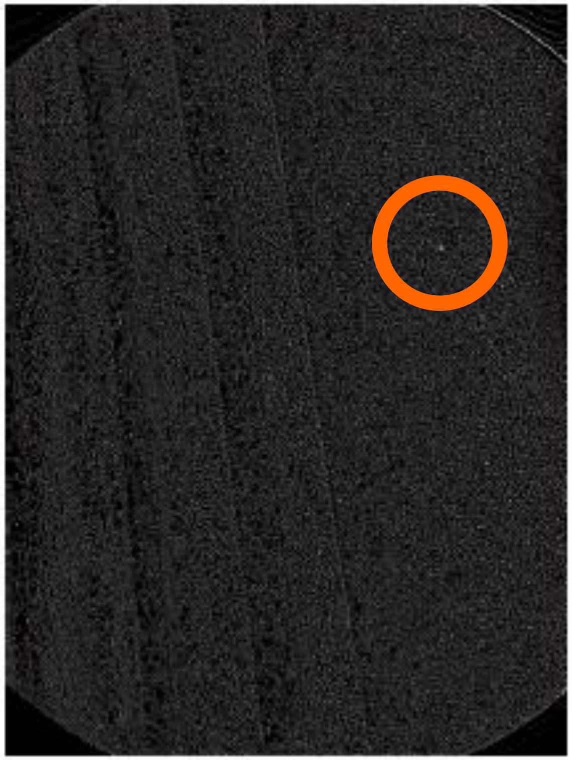
$$|R| \geq T$$

Point Detection (cont…)

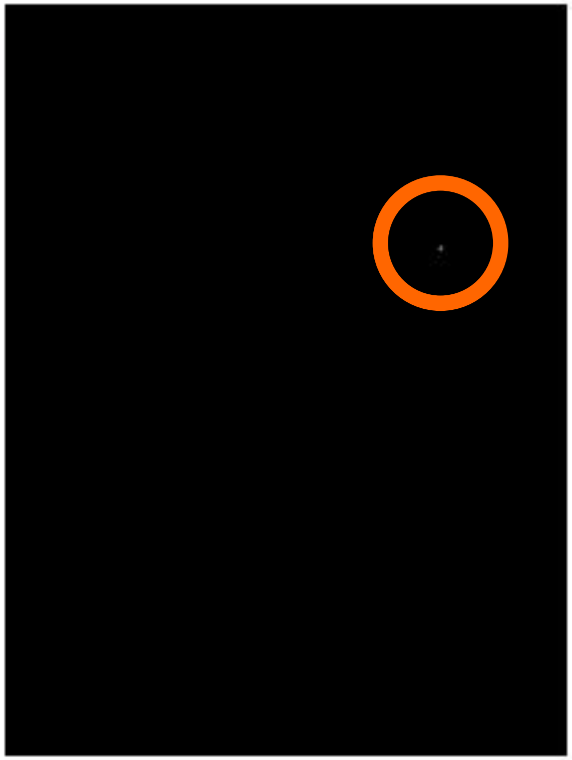
−1	−1	−1
−1	8	−1
−1	−1	−1



X-ray image of a turbine blade



Result of point detection



Result of thresholding

Line Detection

- The masks below will extract lines that are **one pixel thick** and running in a **particular direction**
- E.g. Horizontal mask will result with **max response** when a line passed through the middle row of the mask with a constant background.

−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

- Apply every mask on the image

- let R_1, R_2, R_3, R_4 denotes the **response** of the horizontal, +45 degree, vertical and -45 degree masks, respectively.

- if, at a certain point in the image

$$|R_i| > |R_j|,$$

  – for all $j \neq i$, that point is said to be more likely associated with a line in the direction of mask i.

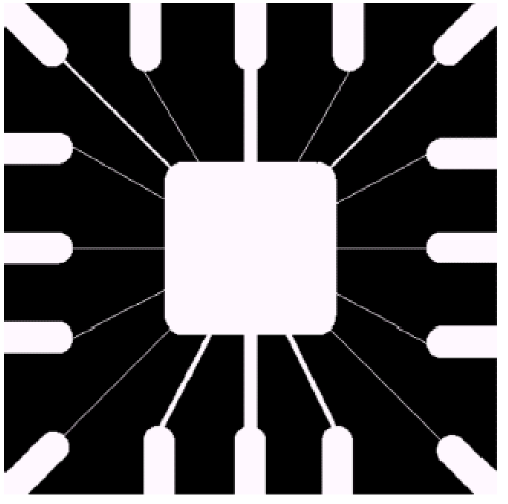
Line Detection

- Alternatively, if we are interested in detecting all lines in an image in the direction defined by a **given mask**, we simply run the mask through the image and **threshold the absolute** value of the result.

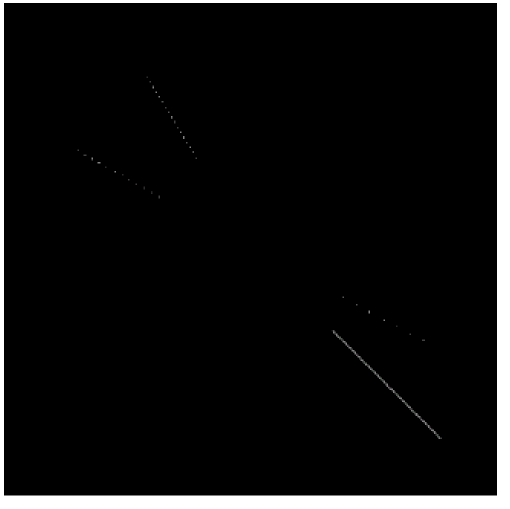
- The points that are left are the **strongest responses**, which, for lines **one pixel thick**, correspond closest to the direction defined by the mask.

Line Detection (cont…)

Binary image of a wire bond mask



After processing with -45° line detector



Result of thresholding filtering result

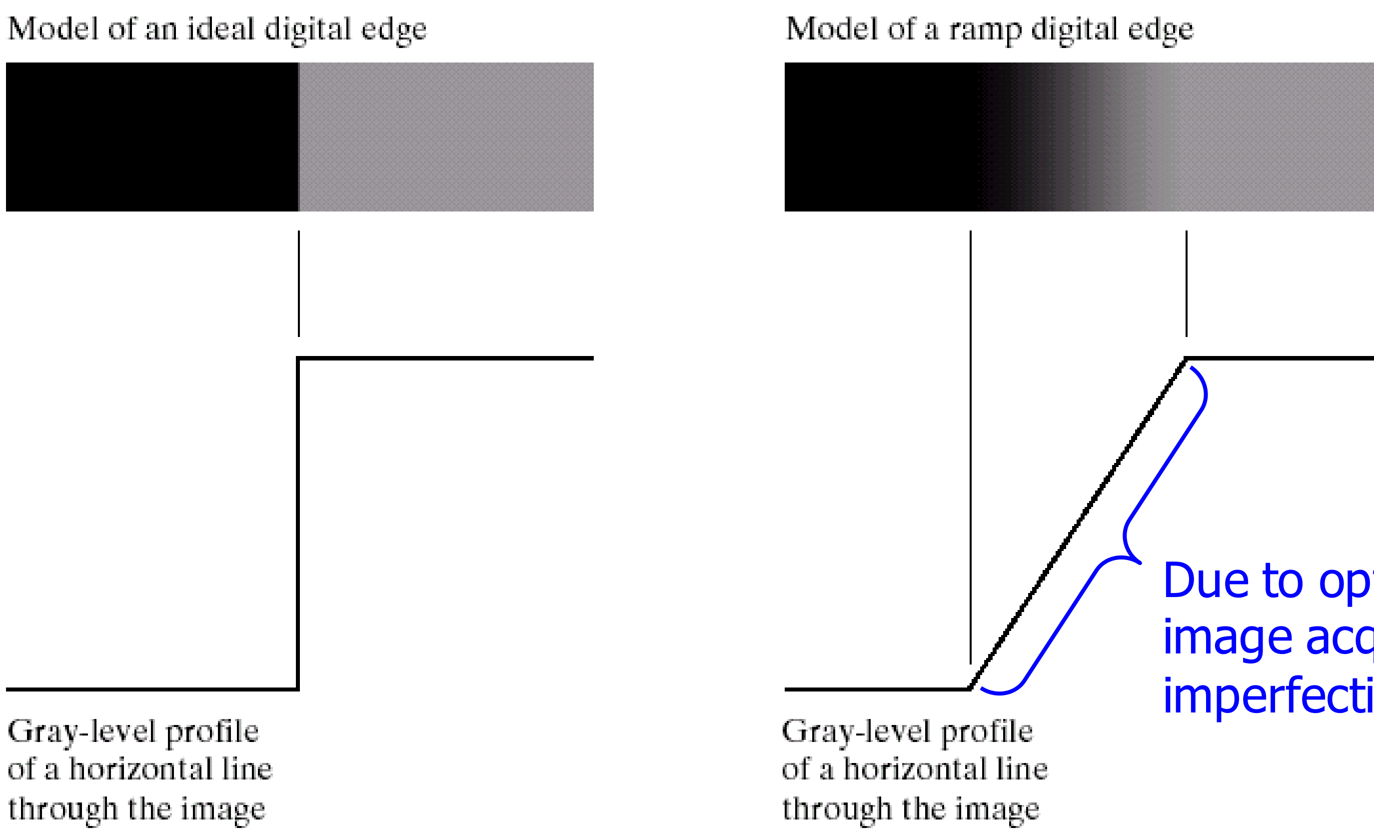
Edge Detection

- we discussed approaches for implementing
  - first-order derivative (Gradient operator)
  - second-order derivative (Laplacian operator)

- Here, we will talk only about their properties for edge detection

- An edge is a set of connected pixels that lie on the boundary between two regions
  - i.e. a set of connected points in the ramp

Ideal and Ramp Edges



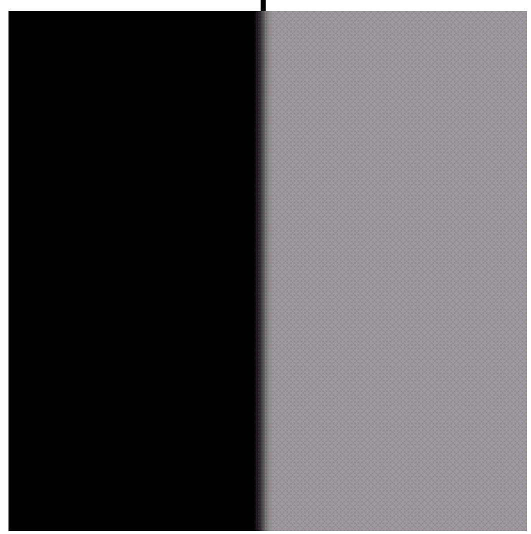
a b

FIGURE 10.5 (a) Model of an ideal digital edge. (b) Model of a ramp edge. The slope of the ramp is proportional to the degree of blurring in the edge.

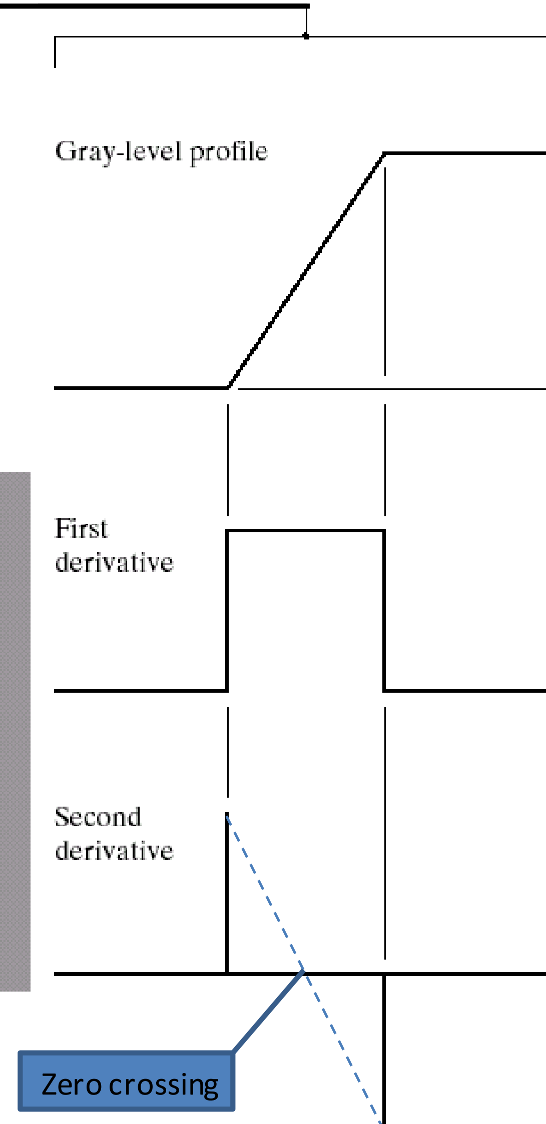
Due to optics, sampling, image acquisition imperfection

Edges & Derivatives

- Derivatives are used to find discontinuities
- 1st derivative tells us where an edge is
- 2nd derivative can be used to show edge direction
- The thickness is determined by the length of the ramp
  - The length is determined by the slope



Two regions separated by a vertical edge



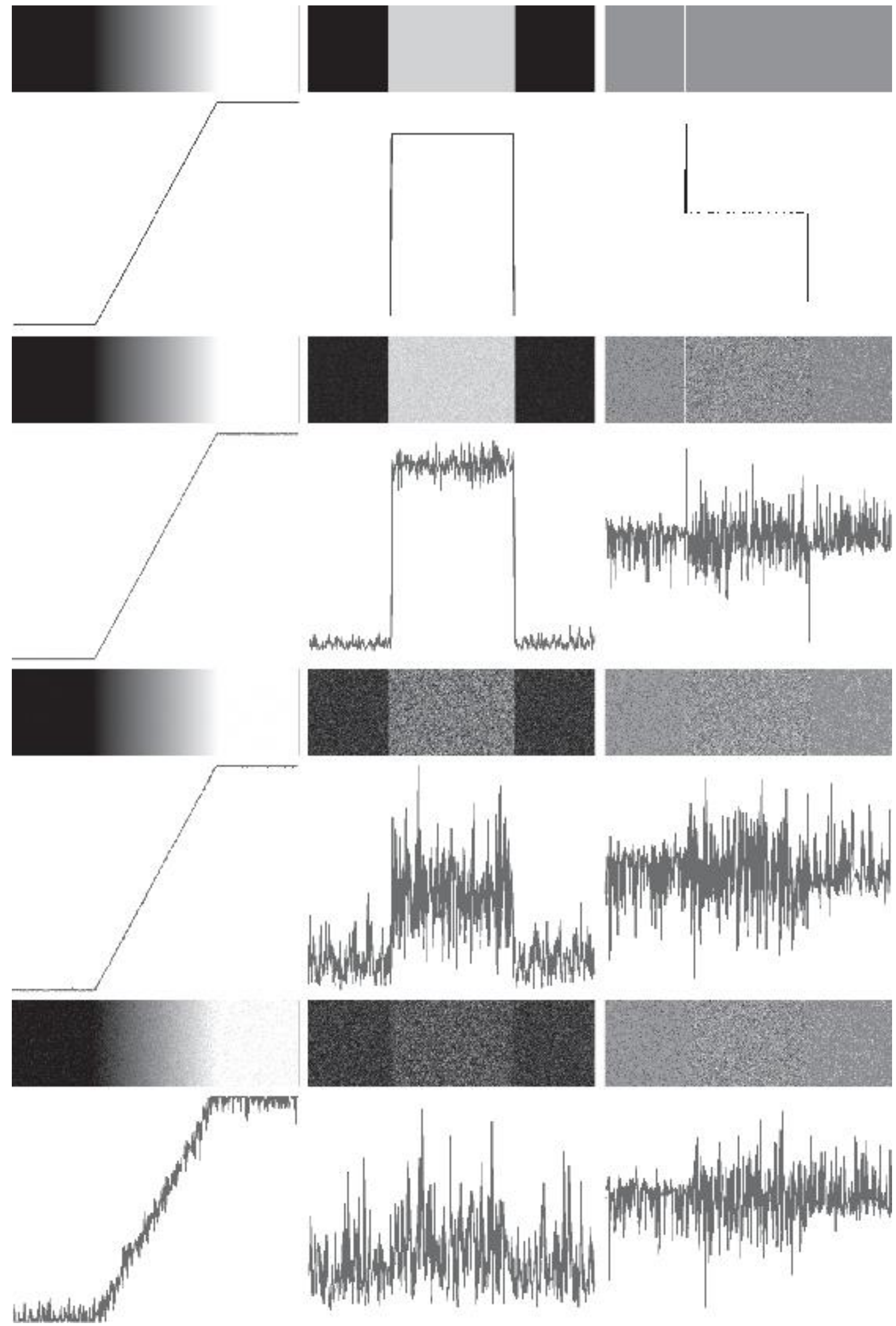
Second derivatives

- Produces 2 values for every edge in an image (an undesirable feature)

- An imaginary straight line joining the extreme positive and negative values of the second derivative would cross zero near the midpoint of the edge. (**zero-crossing property**)

Noisy Images

- First column: images and gray-level profiles of a ramp edge corrupted by random Gaussian noise of mean 0 and $\sigma = 0.0$, 0.1, 1.0 and 10.0, respectively.
- Second column: first-derivative images and gray-level profiles.
- Third column : second-derivative images and gray-level profiles.
- Use image smoothing carefully
  - Noise is likely to be present



Common Edge Detectors – Gradient Masks

- First derivatives are implemented using the magnitude of the gradient

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

Diagonal edges

0	1	1
−1	0	1
−1	−1	0

−1	−1	0
−1	0	1
0	1	1

Prewitt

0	1	2
−1	0	1
−2	−1	0

−2	−1	0
−1	0	1
0	1	2

Sobel

Edge Detection Example

Original Image | Horizontal Gradient Component

Vertical Gradient Component | Combined Edge Image

a b
c d

FIGURE 10.10 (a) Original image. (b) $|G_x|$, component of the gradient in the x-direction. (c) $|G_y|$, component in the y-direction. (d) Gradient image, $|G_x| + |G_y|$.

Edge Detection Example

Edge Detection Example

Edge Detection Example

Edge Detection Example

Example

a b

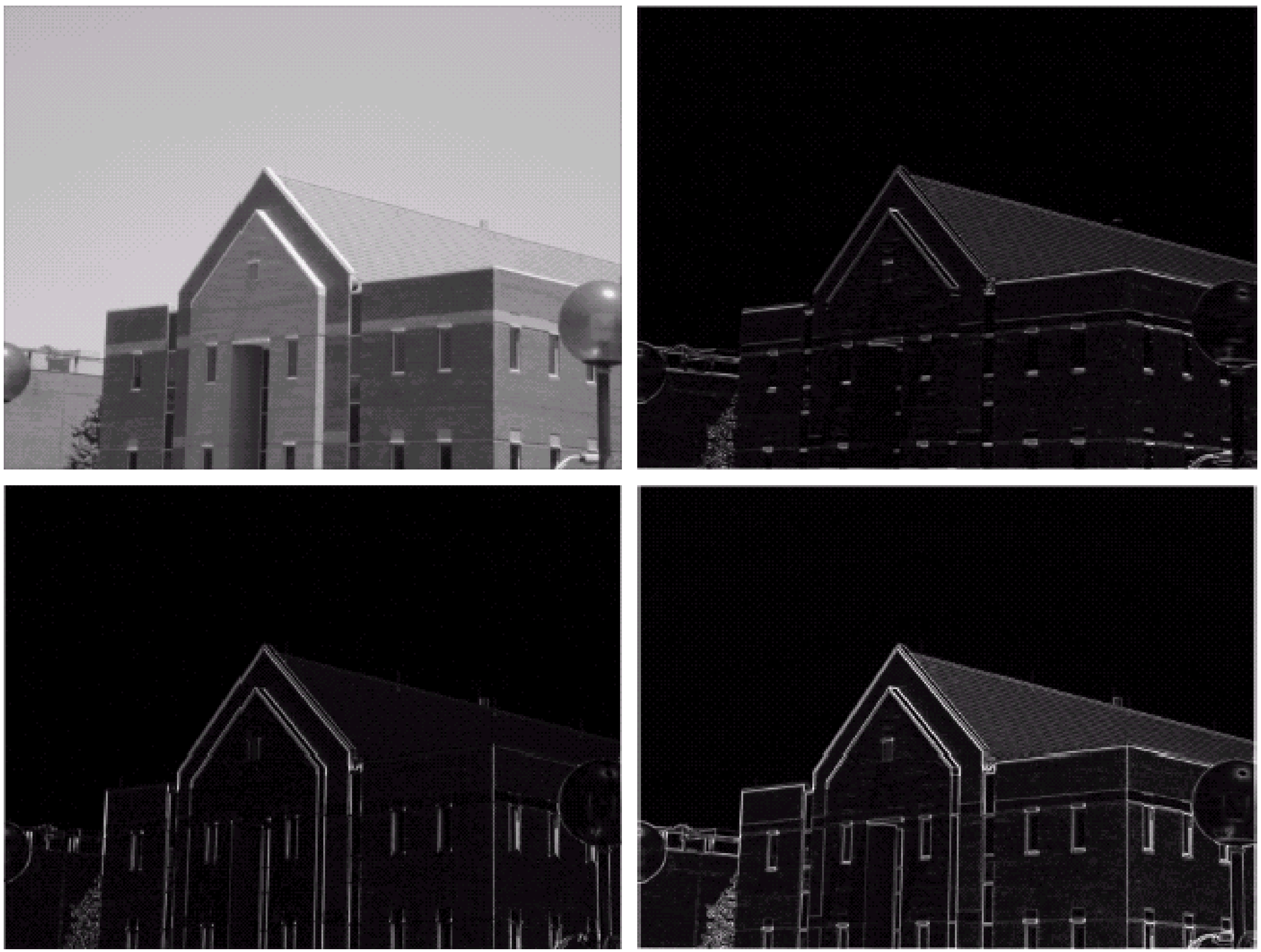
FIGURE 10.12 Diagonal edge detection. (a) Result of using the mask in Fig. 10.9(c). (b) Result of using the mask in Fig. 10.9(d). The input in both cases was Fig. 10.11(a).

Edge Detection Problems

- Often, problems arise in edge detection in that there are 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

Edge Detection Example With Smoothing

Original Image Horizontal Gradient Component



Vertical Gradient Component Combined Edge Image

a	b
c	d

FIGURE 10.11 Same sequence as in Fig. 10.10, but with the original image smoothed with a 5 × 5 averaging filter.

Laplacian Edge Detection

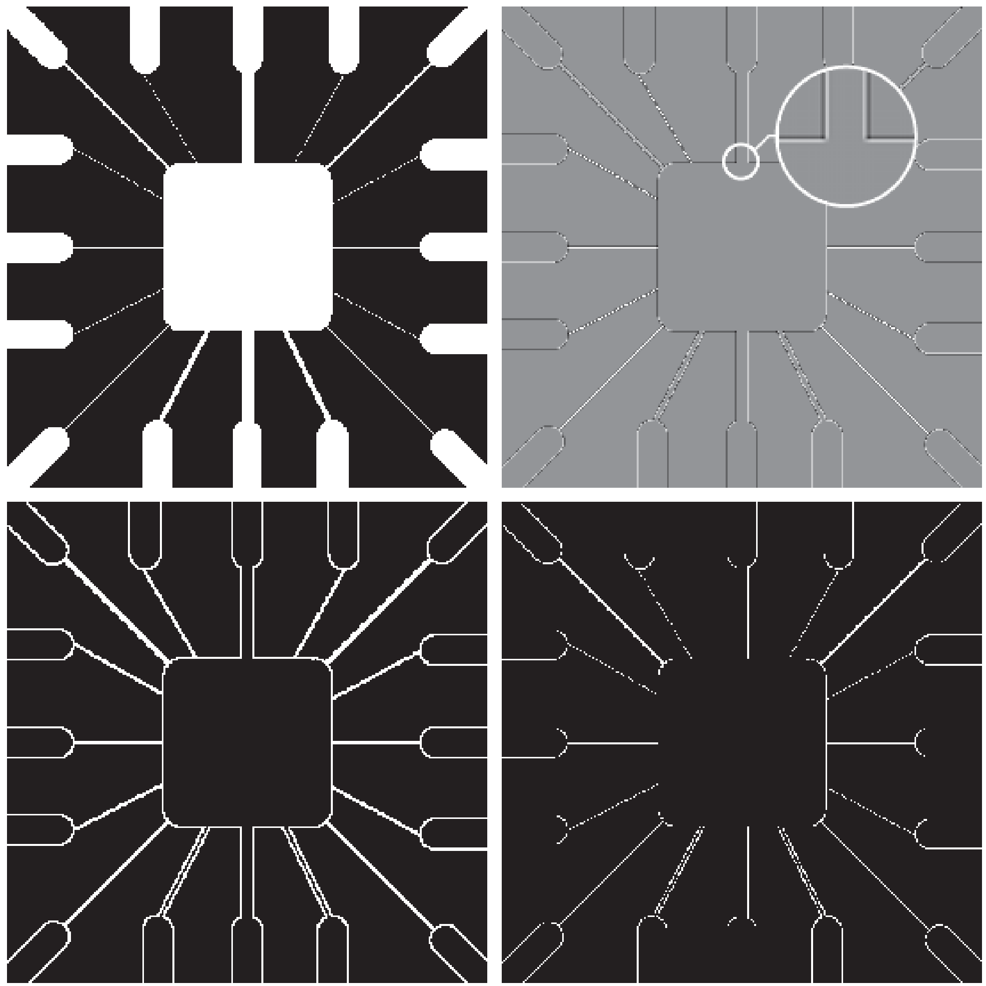
- We encountered the 2nd-order derivative based Laplacian filter already

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

Laplacian Edge Detection



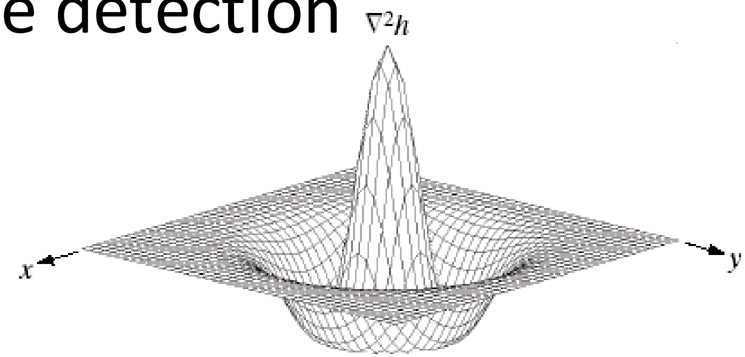
a b
c d

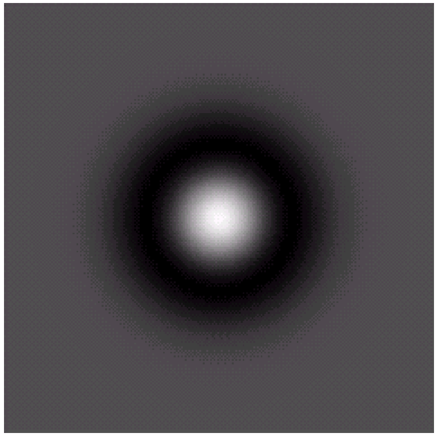
FIGURE 10.5
(a) Original image.
(b) Laplacian image; the magnified section shows the positive/negative double-line effect characteristic of the Laplacian.
(c) Absolute value of the Laplacian.
(d) Positive values of the Laplacian.

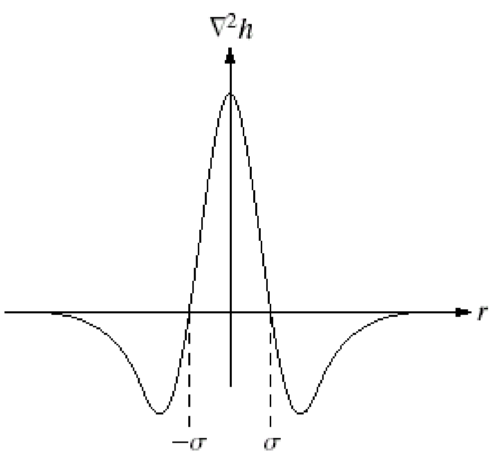
Images are from the Digital Image Processing textbook, see slide 7

Laplacian Of Gaussian (LoG)

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







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

a b
c d

FIGURE 10.14 Laplacian of a Gaussian (LoG). (a) 3-D plot. (b) Image (black is negative, gray is the zero plane, and white is positive). (c) Cross section showing zero crossings. (d) 5×5 mask approximation to the shape of (a).

The coefficients must be sum to zero

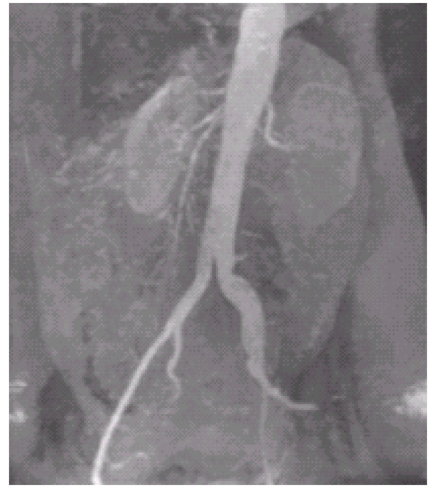
Zero crossing & LoG

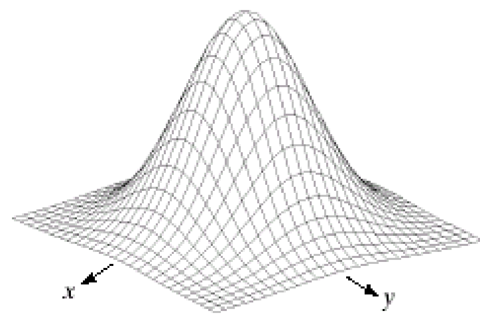
To find edges via zero crossing

- Approximate the zero crossing from LoG image
- To threshold the LoG image by setting all its positive values to white and all negative values to black.
- The zero crossing occurs between positive and negative values of the thresholded LoG.

Example

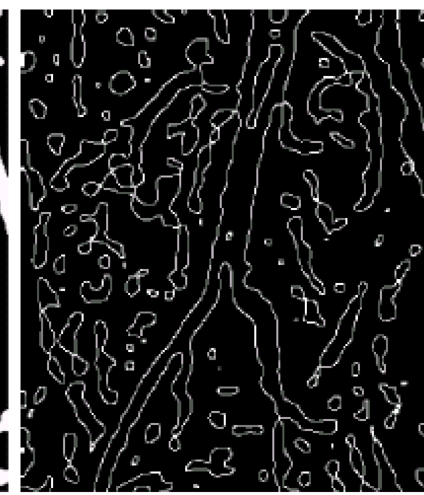
a) Original image
b) Sobel Gradient
c) Spatial Gaussian smoothing function
d) Laplacian mask
e) LoG
f) Threshold LoG
g) Zero crossing





−1	−1	−1
−1	8	−1
−1	−1	−1

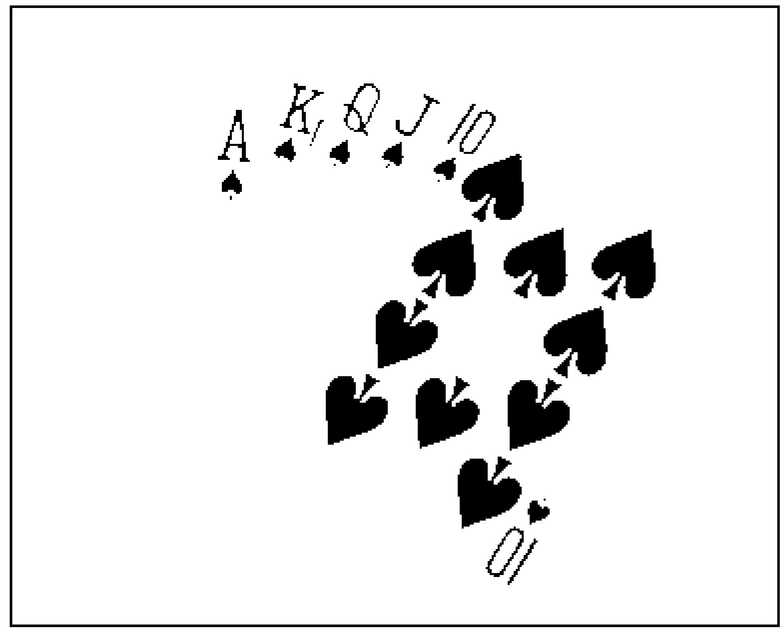




Thresholding Example

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

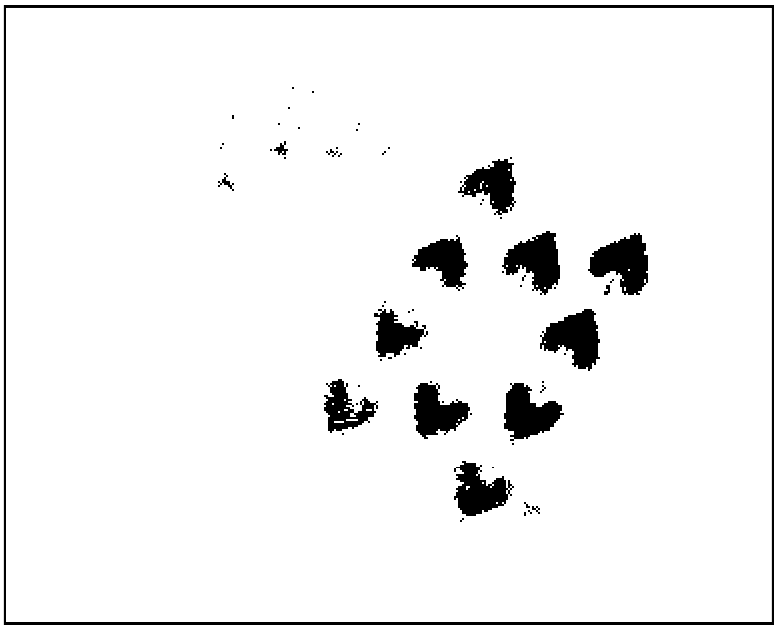
Original Image



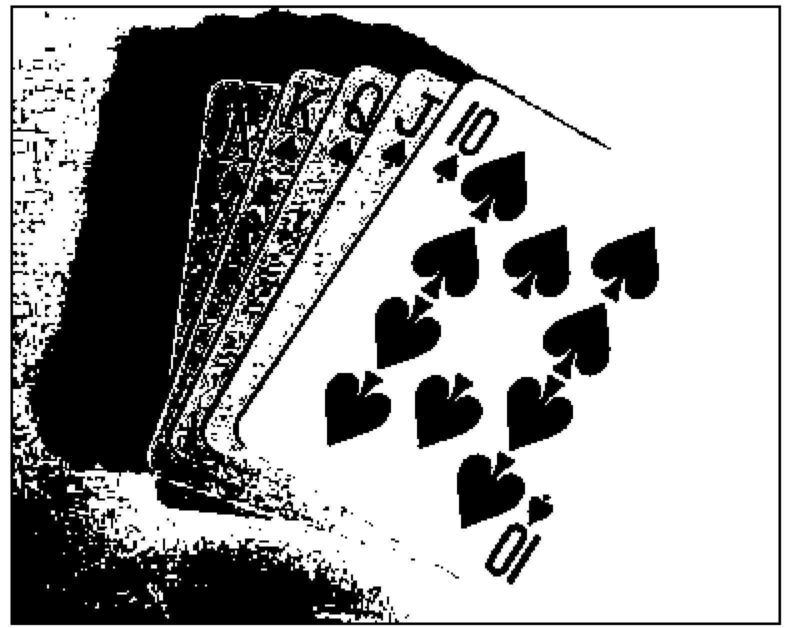
Thresholded Image

But Be Careful

- If you get the threshold wrong the results can be disastrous



Threshold Too Low

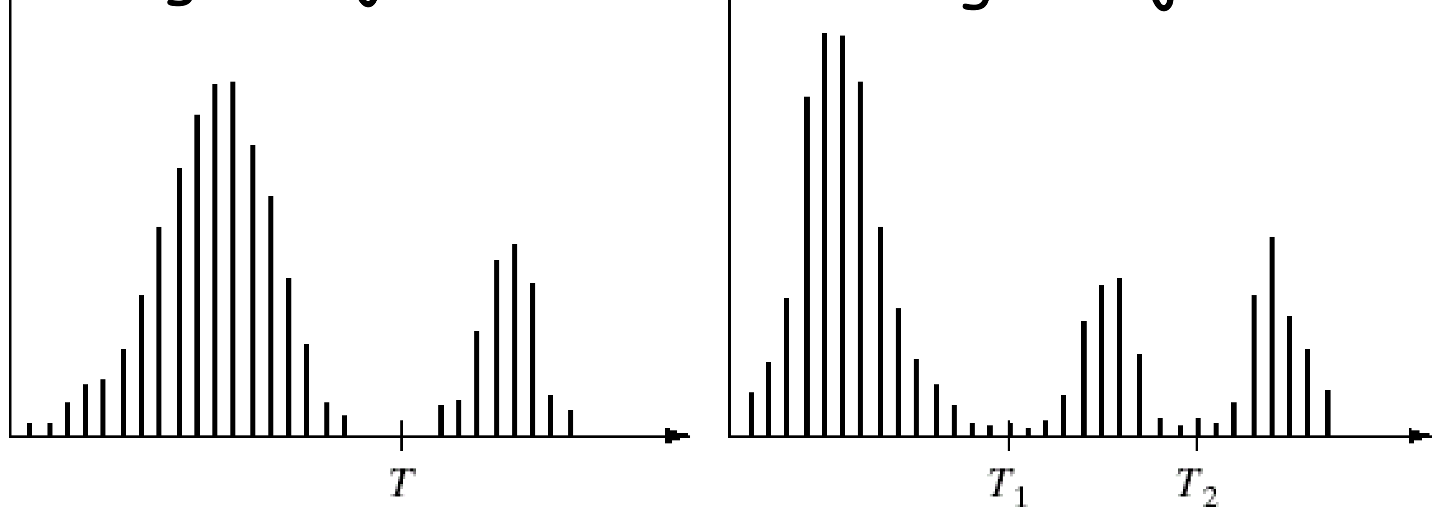


Threshold Too High

Thresholding

Image with dark background and a light object

Image with dark background and two light objects



a b

FIGURE 10.26 (a) Gray-level histograms that can be partitioned by (a) a single threshold, and (b) multiple thresholds.

Thresholding

- We have talked about simple single value thresholding already
- Single value thresholding can be given mathematically as follows:

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

Multilevel thresholding

- A point (x,y) belongs to
  - to an object class if $T_1 < f(x,y) \leq T_2$
  - to another object class if $f(x,y) > T_2$
  - to background if $f(x,y) \leq T_1$

- T depends on
  - only f(x,y) : only on gray-level values ⇨ Global threshold
  - both f(x,y) and p(x,y) : on gray-level values and its neighbors ⇨ Local threshold

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

- Based on visual inspection of histogram

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 $\leq$ 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

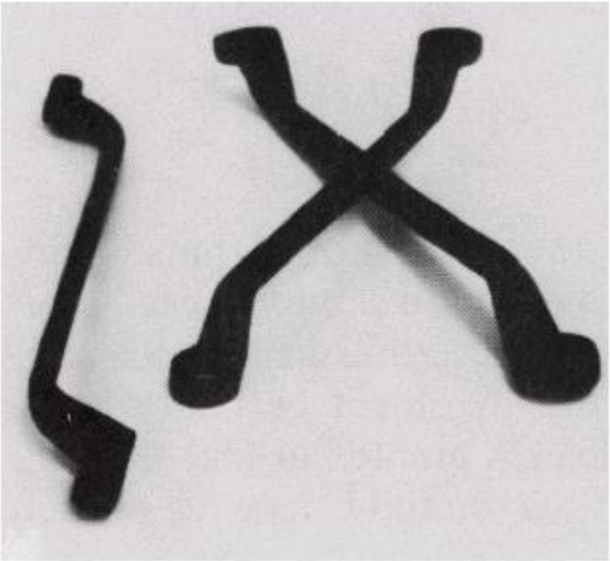
4. Compute a new threshold value:

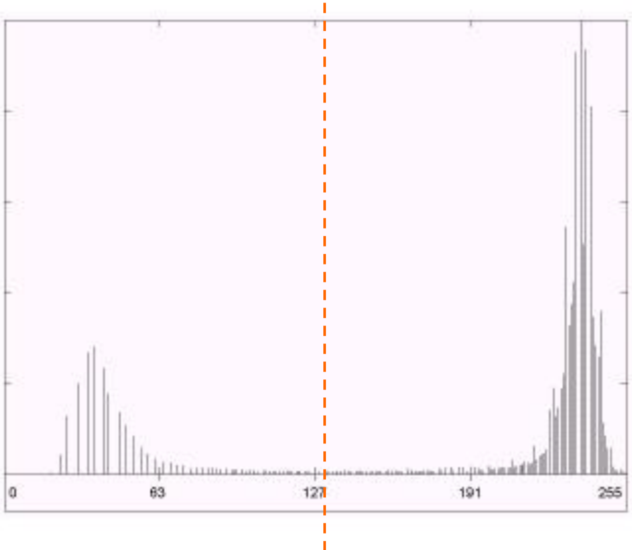
$$T = \frac{\mu_1 + \mu_2}{2}$$

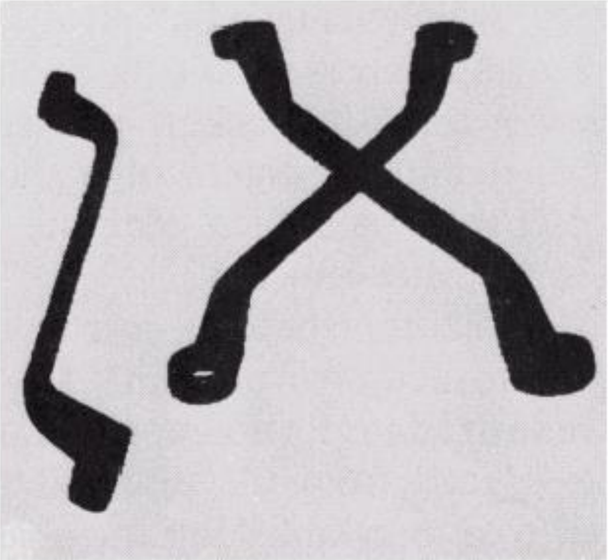
5. Repeat steps 2 – 4 until the difference in T in successive iterations is less than a predefined limit T_∞

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

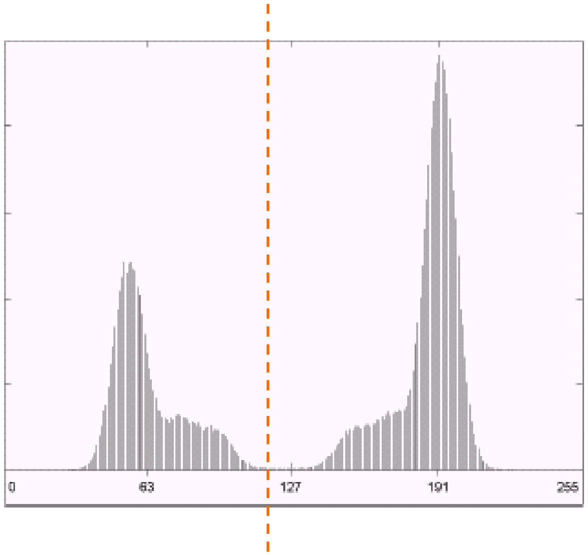
Thresholding Example 1







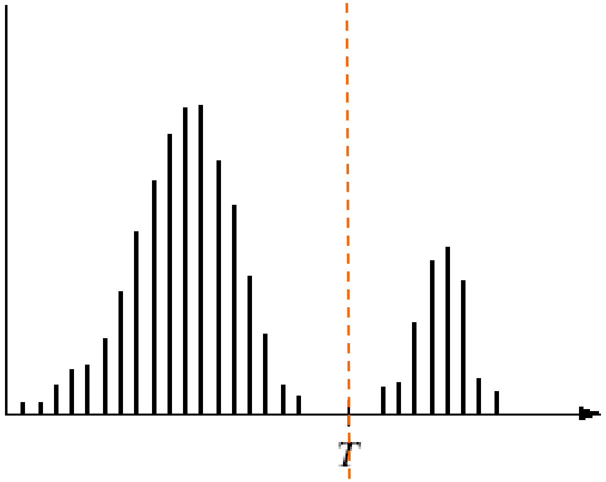
Thresholding Example 2

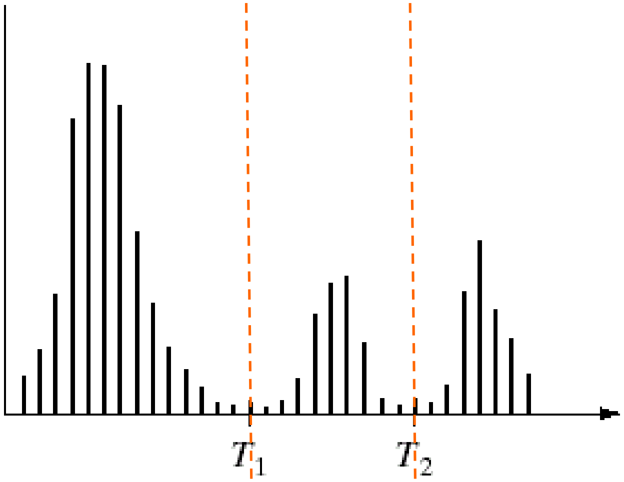


Problems With Single Value Thresholding

- Single value thresholding only works for bimodal histograms

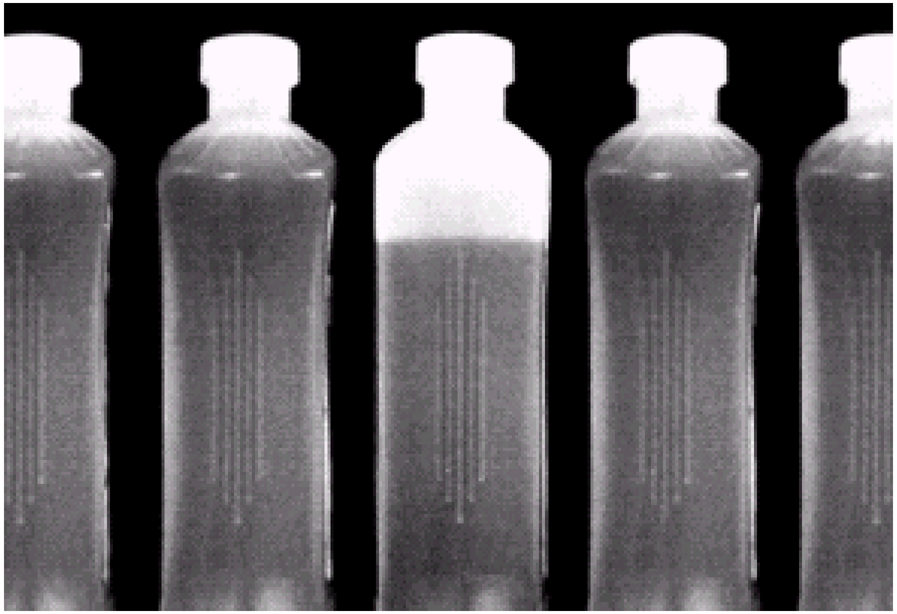
- Images with other kinds of histograms need more than a single threshold





Problems With Single Value Thresholding (cont…)

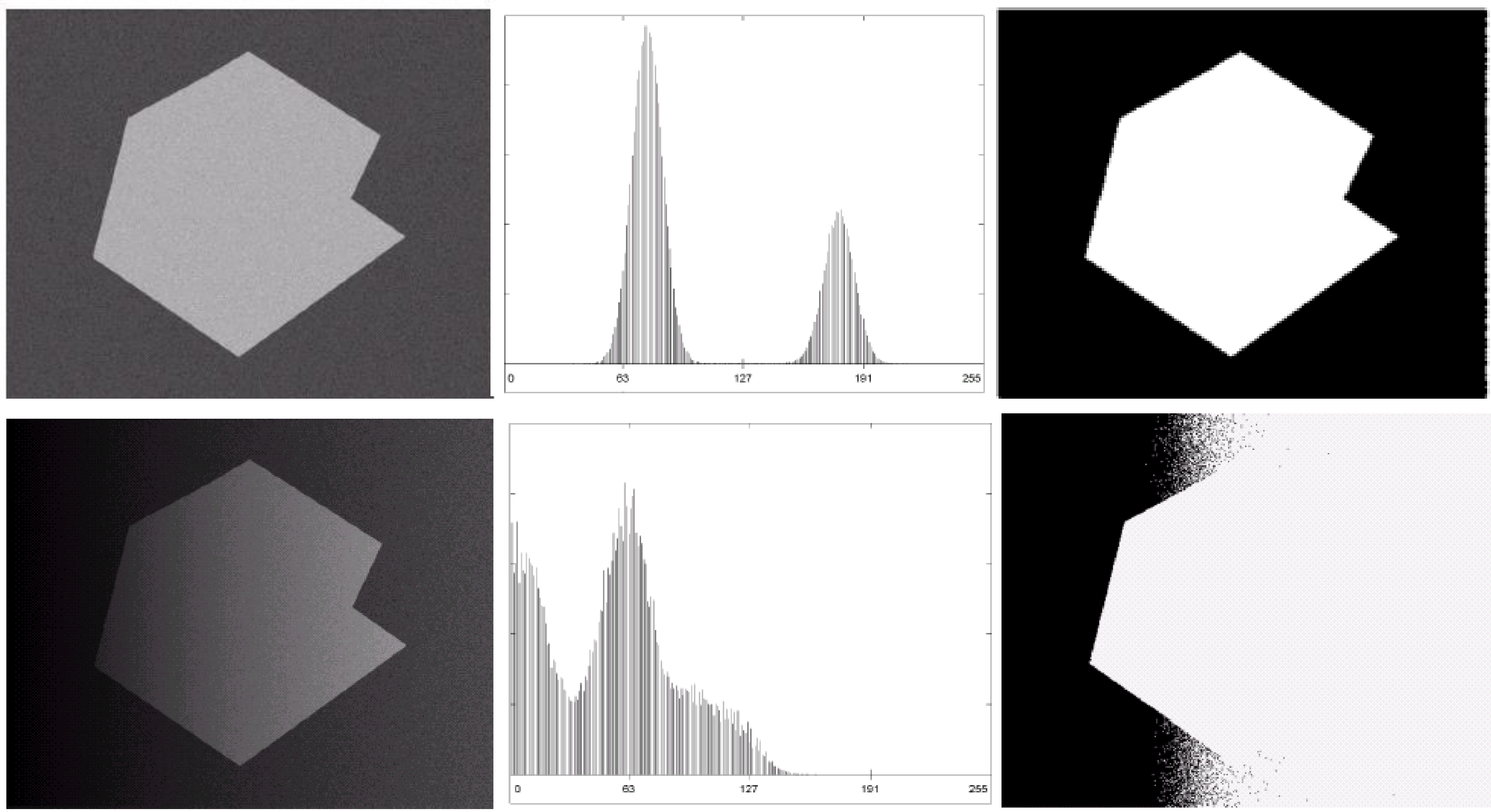
- Let’s say we want to isolate the contents of the bottles



- 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



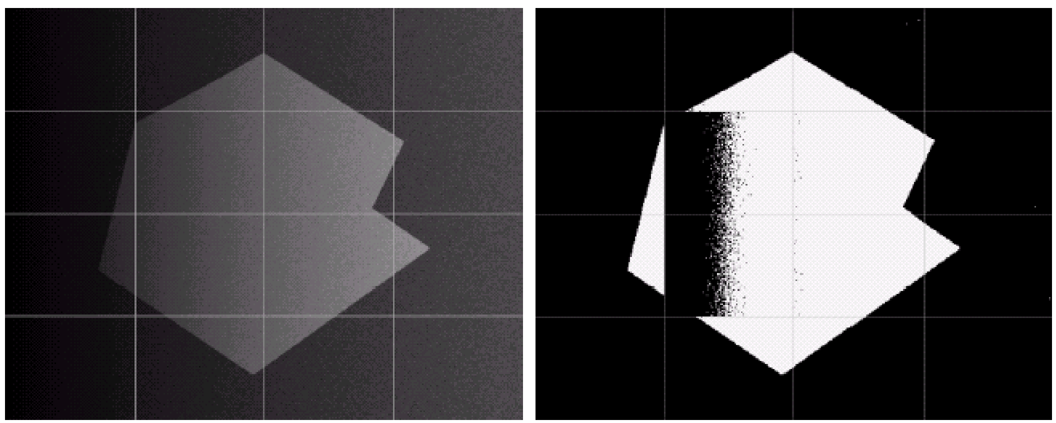
- Uneven illumination can really upset a single valued thresholding scheme

Basic Adaptive Thresholding

- An approach to handle situations in which single value thresholding will not work
- Divide an image into sub images and threshold them 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

Further subdivision

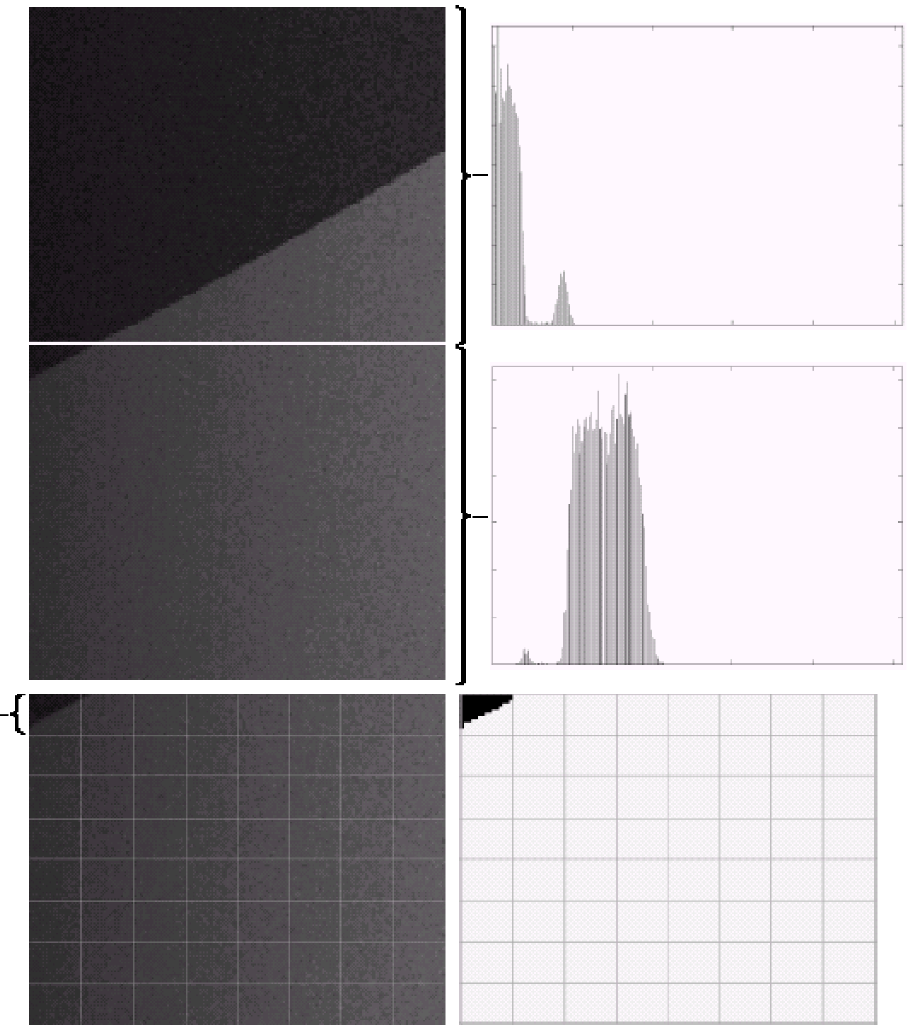
a) Properly and improperly segmented sub-images from previous example.

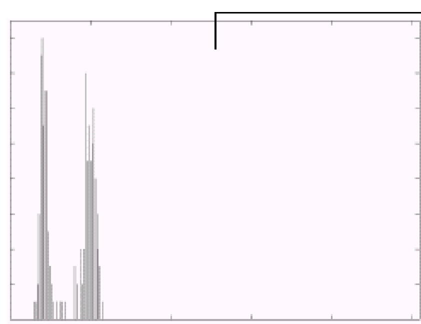
b) and c) Corresponding histograms

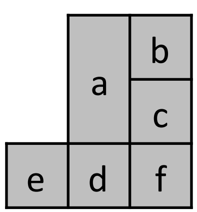
d) Further subdivision of the improperly segmented sub-image.

e) Histogram of the small sub-image at top.

f) Result of adaptively segmenting d).







Region-Based Segmentation (Region Growing)

- Start with a set of **“seed”** points
- Growing by appending to each seed those neighbors that have similar properties such as specific ranges of gray level

Region Growing

Criteria:

1. The absolute gray-level difference between any pixel and the seed has to be less than 65

2. The pixel has to be 8-connected to at least one pixel in that region

Select all seed points with gray level 255

a	b
c	d

FIGURE 10.40 (a) Image showing defective welds. (b) Seed points. (c) Result of region growing. (d) Boundaries of segmented defective welds (in black). (Original image courtesy of X-TEK Systems, Ltd.).