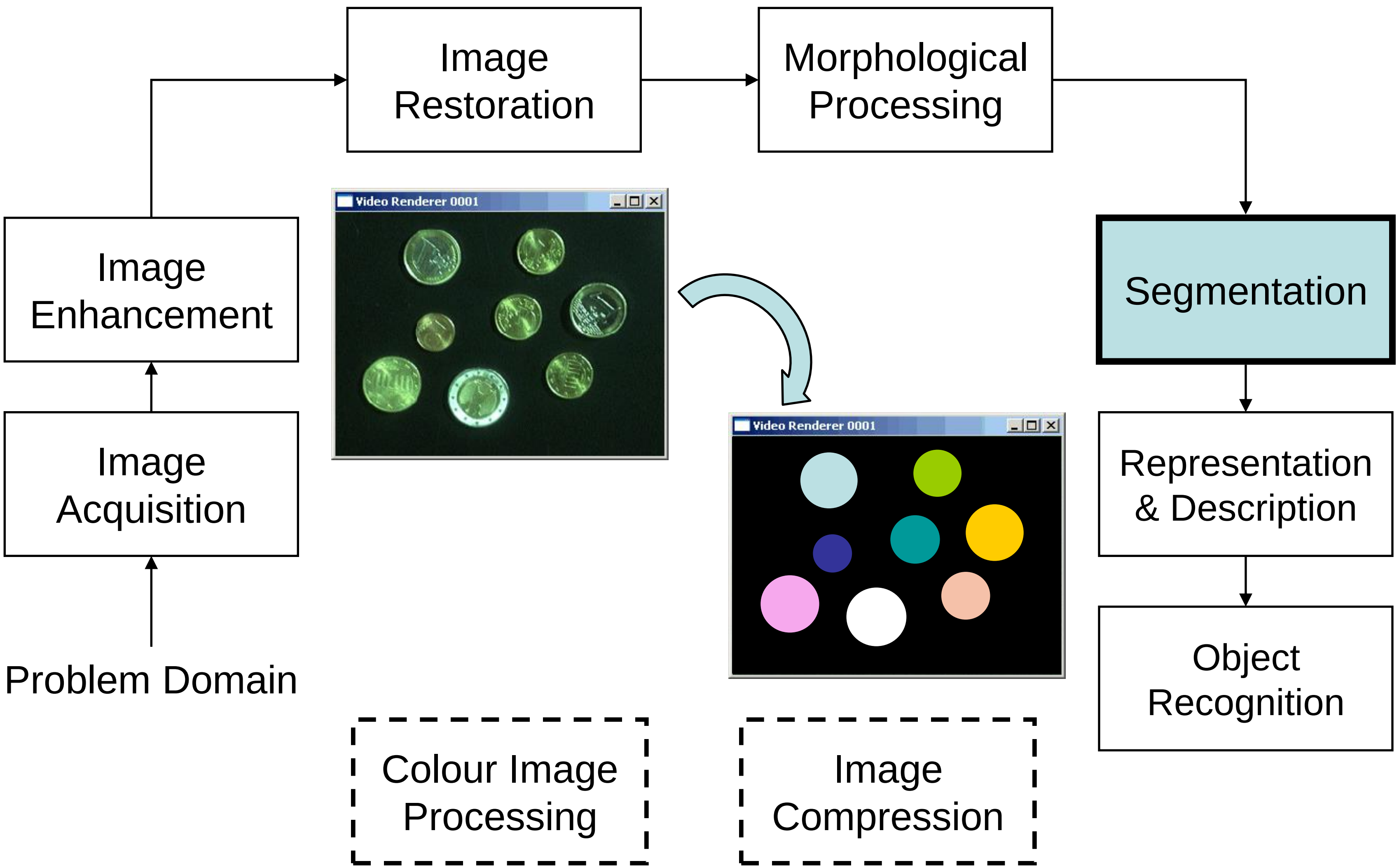
Image and Video Processing

Segmentation

Course Outline



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

What is segmentation?

- Segmentation is the process of dividing an image into separate/non-overlapping regions.
- All of the pixels in an image must belong to some region or another.
- A region might be an object, a part of an object or the background
- A region can be specified by either defining the pixels that constitute it, or the pixels that bound it.

An Example

Identify the regions that you would segment this image into?

Region definition

Basic Formulation:

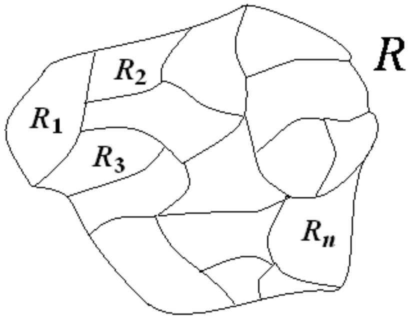
$$(a)\quad \bigcup_{i=1}^{n} R_i = R$$

$$(b)\quad R_i \text{ is a connected region, } i = 1, 2, \ldots, n$$

$$(c)\quad R_i \cap R_j = \phi \text{ for all i and j, } i \neq j$$

$$(d)\quad P(R_i) = TRUE \text{ for } i = 1, 2, \ldots, n$$

$$(e)\quad P(R_i \cup R_j) = FALSE \;\; for\; i \neq j$$



P(Ri) is a homogeniety predicate property defined over the points in set Ri

Ex. P(Ri) = TRUE, if all pixels in Ri have the ‘same’ gray level.

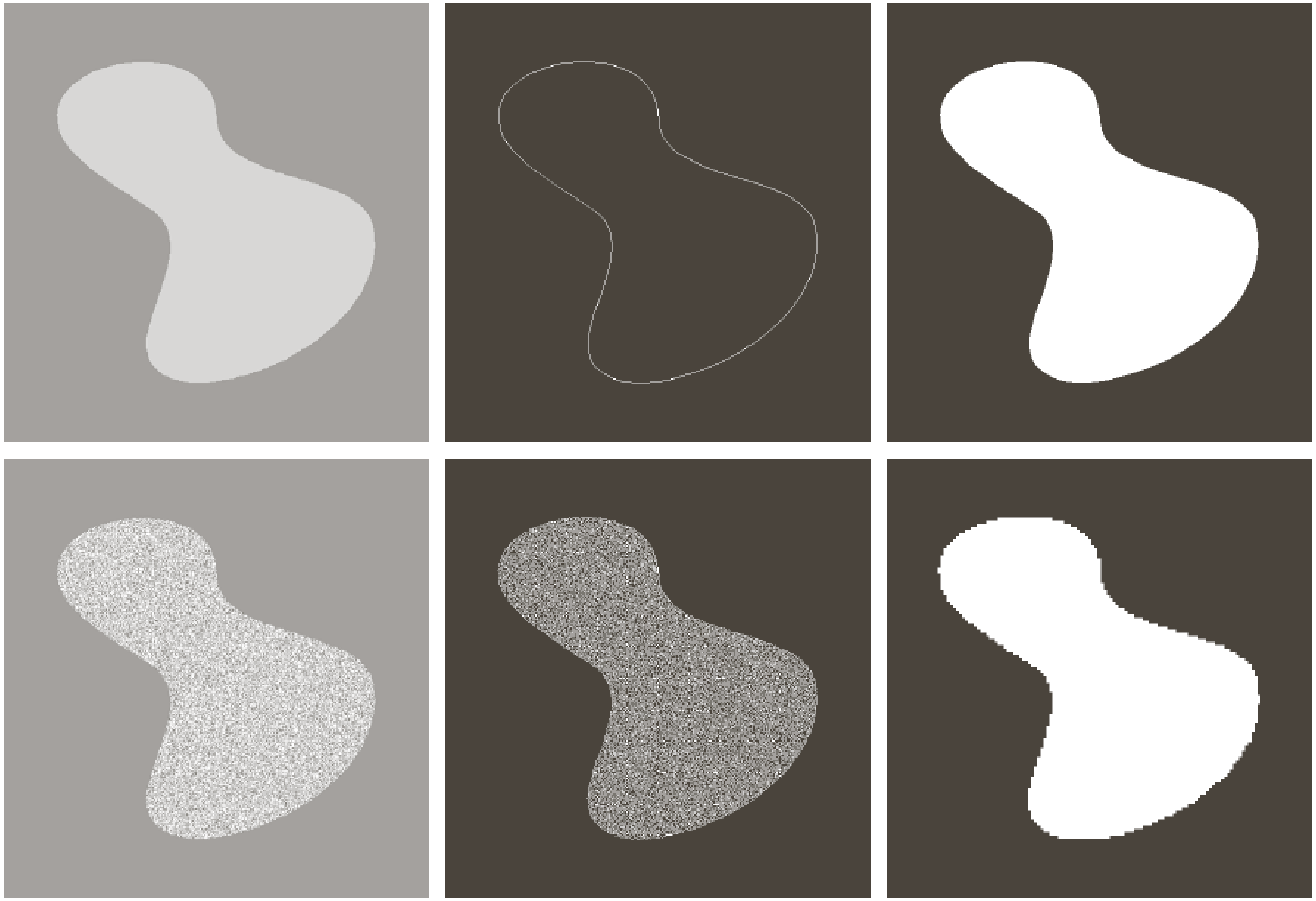
Basic Principle

- Segmentation algorithms are often based on one of the following two basic properties of intensity values:
- **Similarity**: Partitioning an image into regions that are similar according to a set of predefined criteria.
- **Discontinuity**: Detecting boundaries of regions based on local discontinuity in intensity.

Similarity vs Discontinuity

Features

- intensity
- texture
- edge sharpness
- any other relevant feature(s)



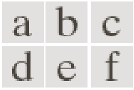
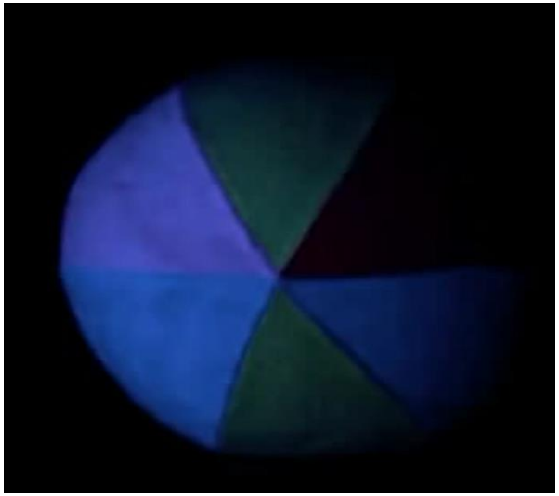


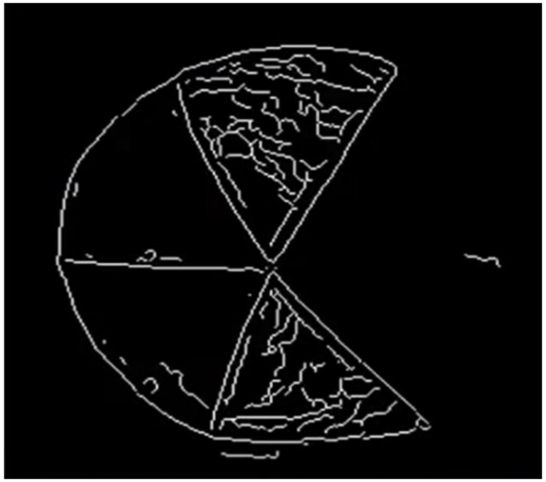
FIGURE 10.1 (a) Image containing a region of constant intensity. (b) Image showing the boundary of the inner region, obtained from intensity discontinuities. (c) Result of segmenting the image into two regions. (d) Image containing a textured region. (e) Result of edge computations. Note the large number of small edges that are connected to the original boundary, making it difficult to find a unique boundary using only edge information. (f) Result of segmentation based on region properties.

Image Segmentation Methods

- **Edge based segmentation [*Discontinuity]***
  - Finding boundary between adjacent regions i.e. Detecting edges that separate regions from each other.
- **Threshold based segmentation [*Similarity]***
  - Finding regions by grouping pixels with similar intensities i.e. Based on pixel intensities (shape of histogram is often used for automation).
- **Region based segmentation [*Similarity]***
  - Finding regions directly using growing or splitting i.e. Grouping similar pixels (with e.g. region growing or merge & split).
- **Motion based segmentation [*Similarity]***
  - Finding regions by comparing successive frames of a video sequence to identify regions that correspond to moving objects

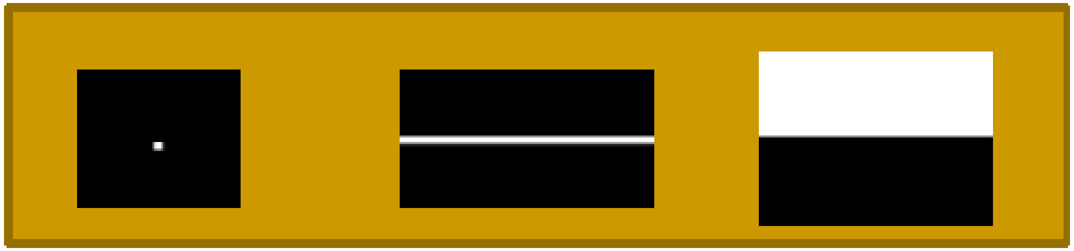
Segmentation Using Discontinuities





There are three basic types of grey level discontinuities in digital images:

- Points
- Lines
- Edges



Point, line & edge detection

- Basic discontinuities in images can be detected from the 1st and 2nd order derivatives of the intensity profile
- Sensitive to image noise
- noise filtering needed

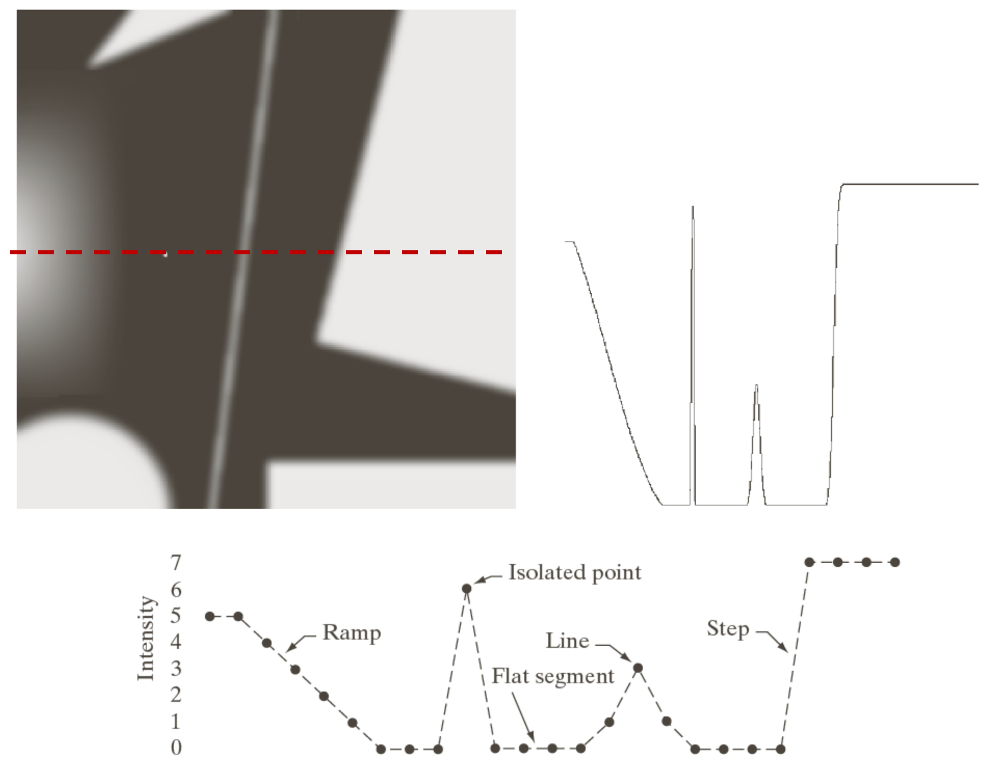


Image strip | 5 | 5 | 4 | 3 | 2 | 1 | 0 | 0 | 0 | 6 | 0 | 0 | 0 | 0 | 1 | 3 | 1 | 0 | 0 | 0 | 0 | 7 | 7 | 7 | 7 | • | •

First derivative −1 −1 −1 −1 −1 0 0 6 −6 0 0 0 1 2 −2 −1 0 0 0 7 0 0 0

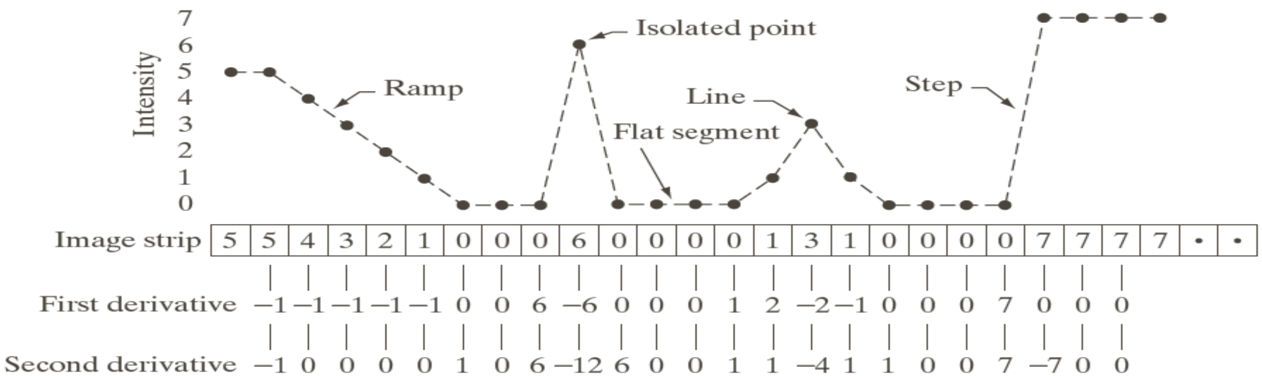
Second derivative −1 0 0 0 0 1 0 6 −12 6 0 0 1 1 −4 1 1 0 0 7 −7 0 0

a b
c

FIGURE 10.2 (a) Image. (b) Horizontal intensity profile through the center of the image, including the isolated noise point. (c) Simplified profile (the points are joined by dashes for clarity). The image strip corresponds to the intensity profile, and the numbers in the boxes are the intensity values of the dots shown in the profile. The derivatives were obtained using Eqs. (10.2-1) and (10.2-2).

Characteristics of First and Second Order Derivatives

- 1st order derivatives generally produce thicker edges in image
- 2nd order derivatives have a stronger response to fine detail, such as thin lines, isolated points, and noise
- Second-order derivatives produce a double-edge response at ramp and step transition in intensity
- The sign of the second derivative can be used to determine whether a transition into an edge is from light to dark or dark to light



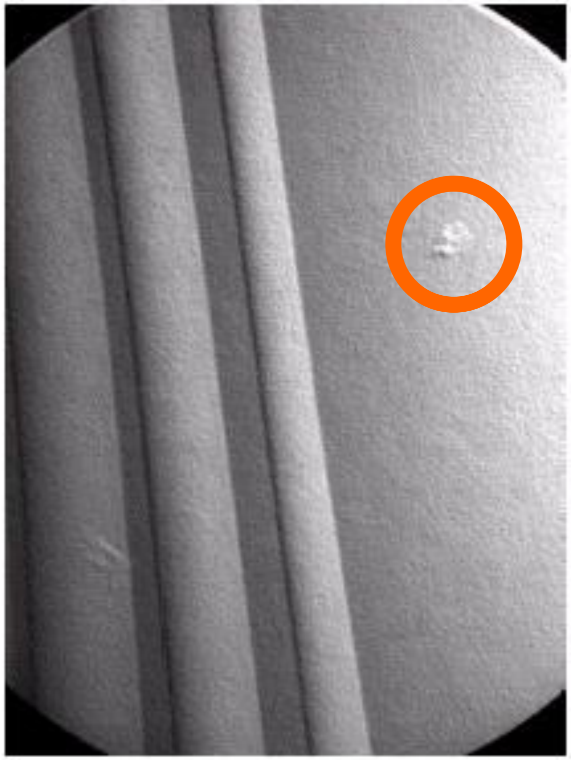
Point Detection

Point detection can be achieved simply using the mask below:

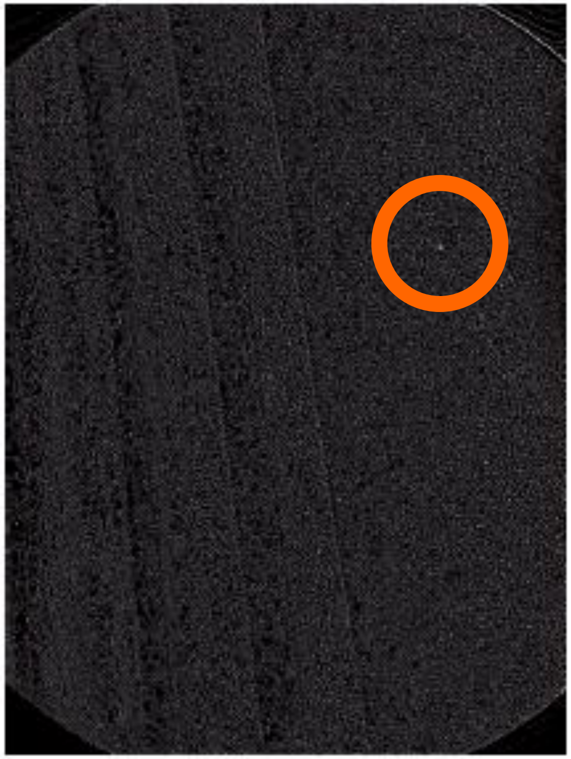
–1	–1	–1
–1	8	–1
–1	–1	–1

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

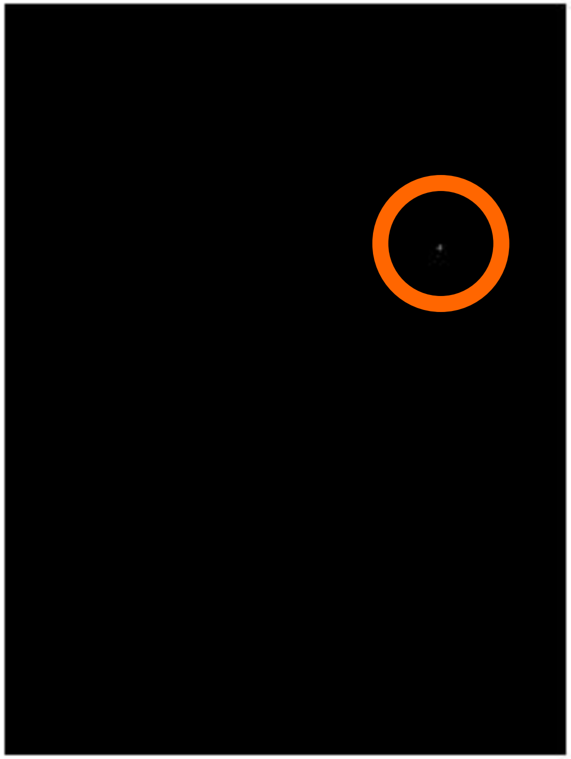
Point Detection (cont…)



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

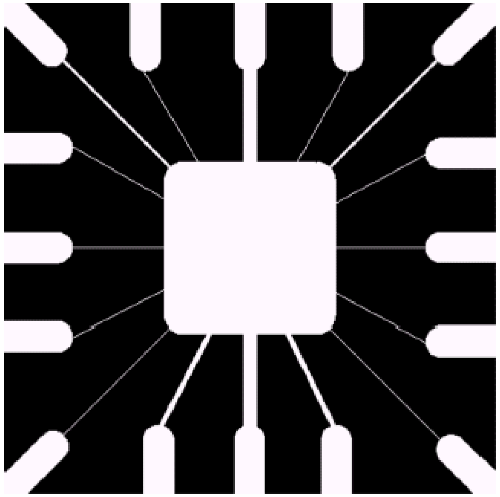
Vertical

2	−1	−1
−1	2	−1
−1	−1	2

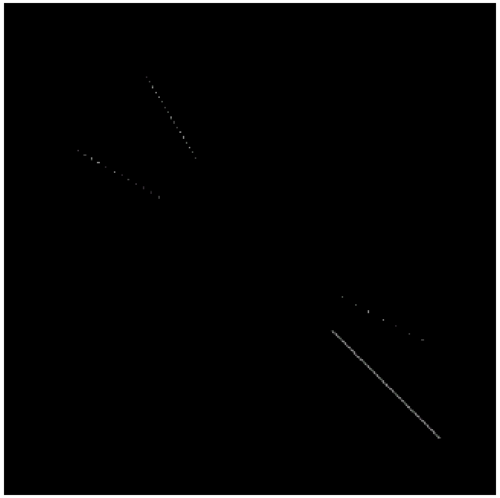
−45°

Line Detection (cont…)

Binary image of a wire bond mask



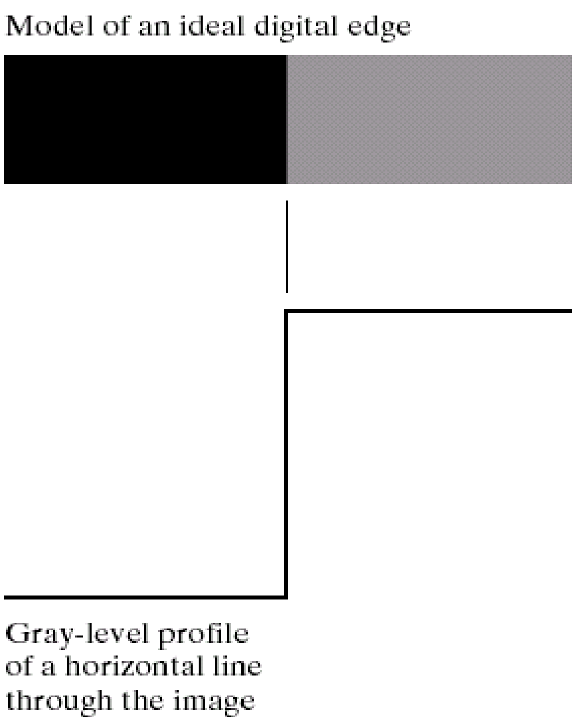
After processing with -45° line detector

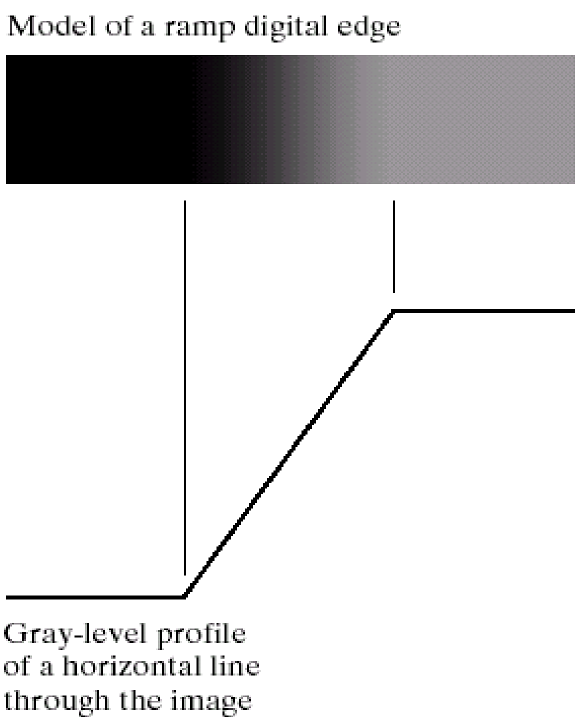


Result of thresholding filtering result

Edge Detection

- **What is an edge?** An edge is a set of connected pixels that lie on the boundary between two regions. Edges are pixels where there is local transition of image intensities





Edge Detection

- **Why Edge detection?**
  - Helps in determining object boundaries and segmentation
  - More compactly represents salient features of the scene than pixels
  - Further features can be extracted from the edges of an image (e.g., corners, lines etc.)
  - These features can be used by higher-level computer vision algorithms (e.g., recognition)

Edges in reality

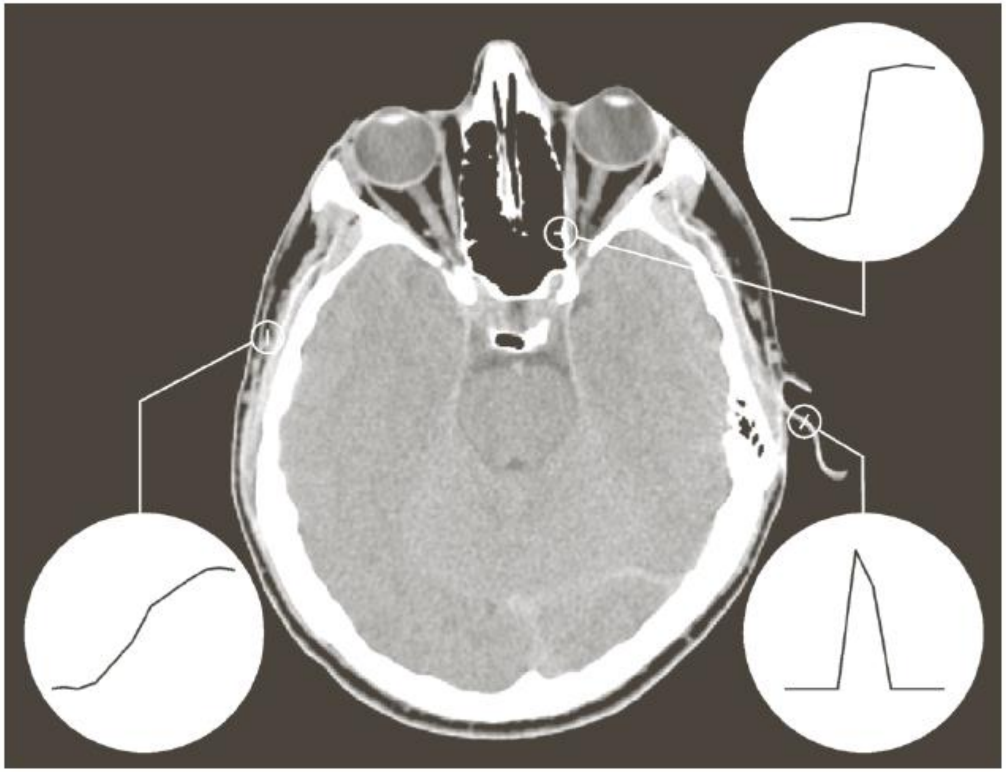
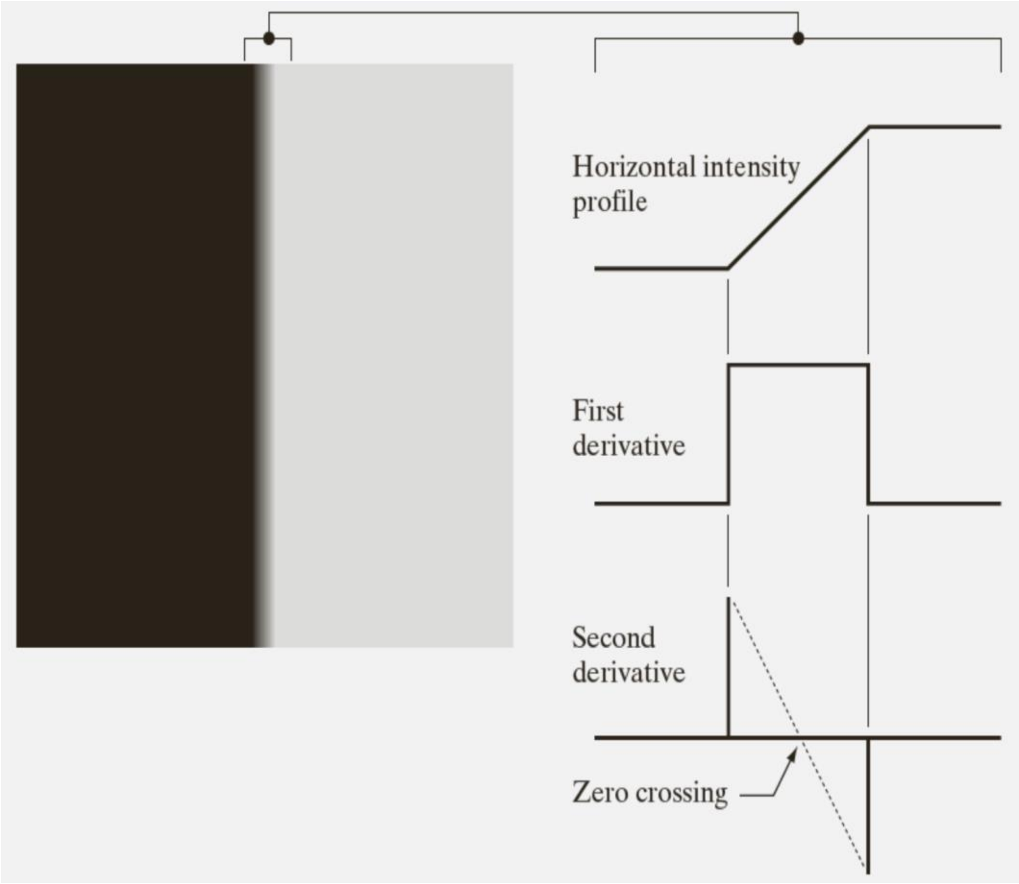


FIGURE 10.9 A 1508 × 1970 image showing (zoomed) actual ramp (bottom, left), step (top, right), and roof edge profiles. The profiles are from dark to light, in the areas indicated by the short line segments shown in the small circles. The ramp and "step" profiles span 9 pixels and 2 pixels, respectively. The base of the roof edge is 3 pixels. (Original image courtesy of Dr. David R. Pickens, Vanderbilt University.)

Edges & Derivatives

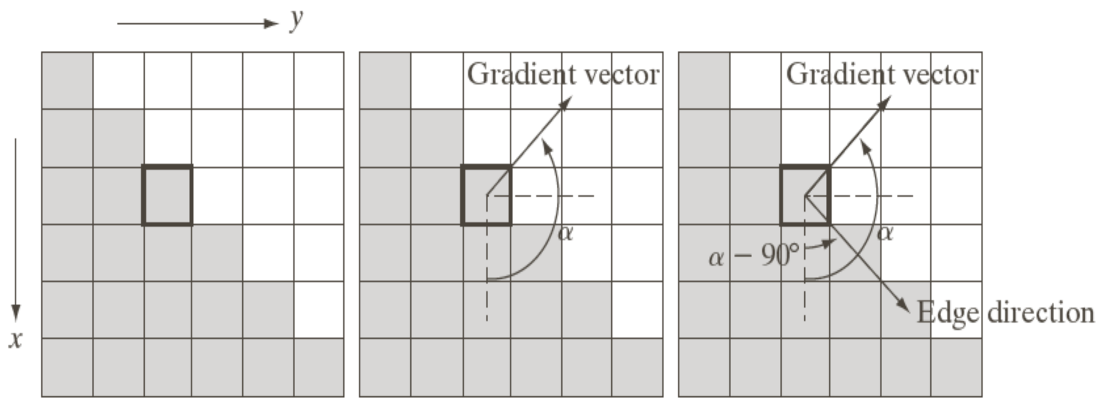
- Recall earlier discussions on how derivatives are used to find discontinuities



Edge Detection

- Finding the edge strength and direction
- Basic Edge Detection by Using First-Order Derivative

$$\nabla f = \mathrm{grad}(f) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix}$$



Gradient magnitude (strength): $M(x, y) = \mathrm{mag}(\nabla f) = \sqrt{g_x^2 + g_y^2}$

Gradient direction: $\alpha(x, y) = \tan^{-1}\left[\dfrac{g_y}{g_x}\right]$

- The direction of the edg $\phi = \alpha - 90^\circ$

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

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

Edge Detection Example

Edge Detection Example

Edge Detection Example

Edge Detection Example

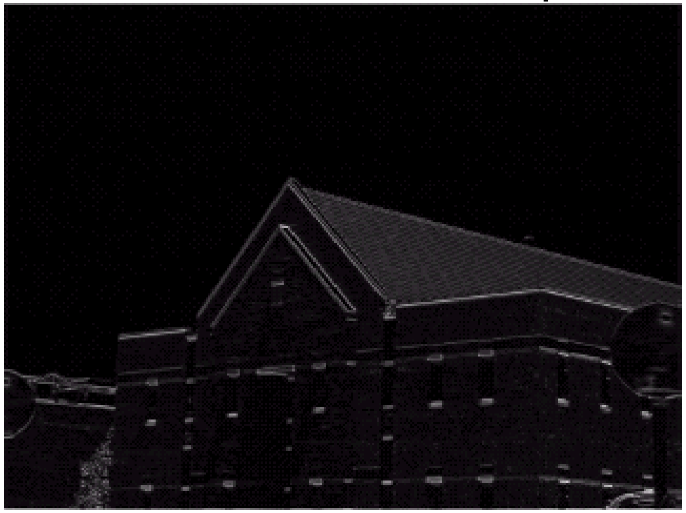
Edge Detection Problems

- 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

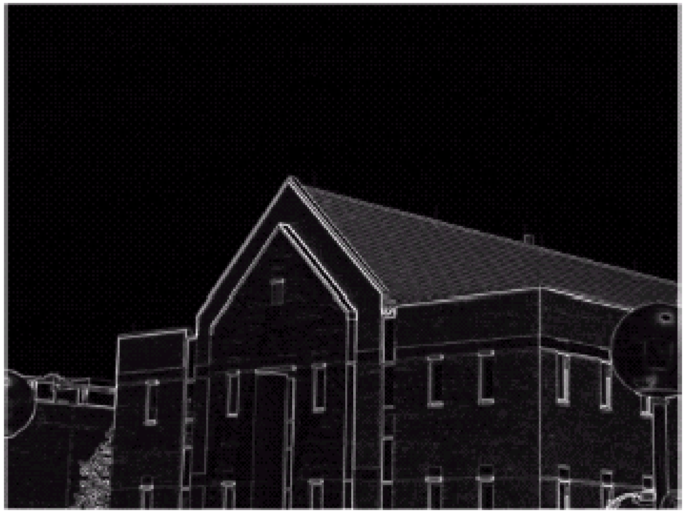
Edge Detection Example With Smoothing

Original Image

Horizontal Gradient Component



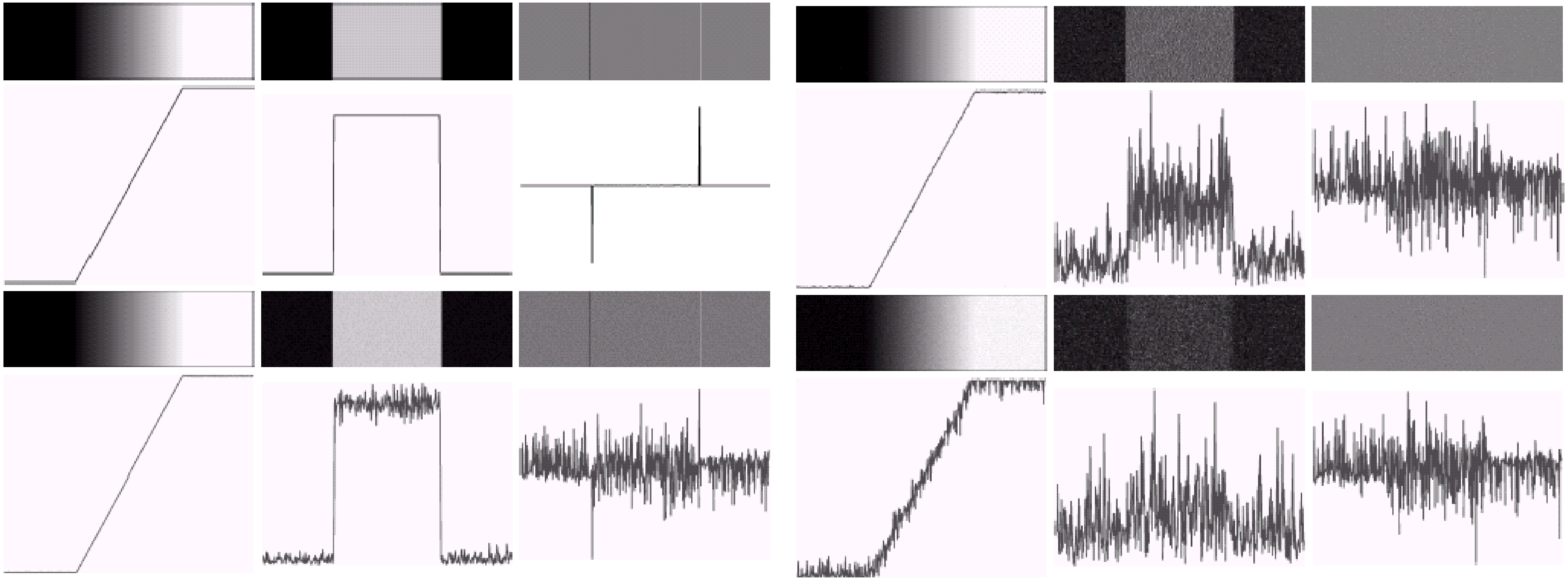
Vertical Gradient Component



Combined Edge Image

Derivatives & Noise

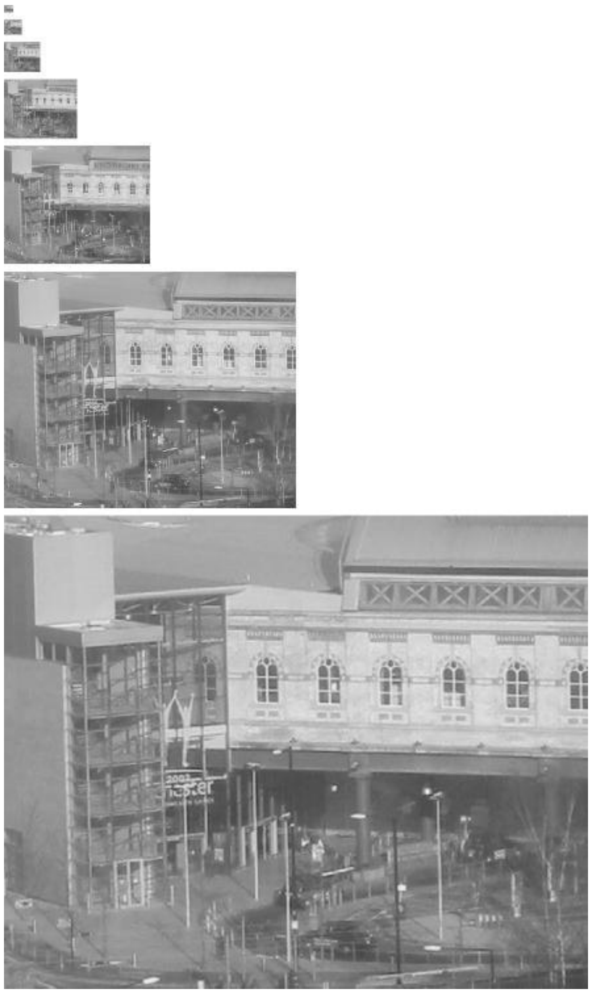
- Derivative based edge detectors are extremely sensitive to noise



Importance of Scale

- Structures exist at multiple scales

Scale space



A simple image pyramid

Image scale space generated by varying sigma of a smoothing function

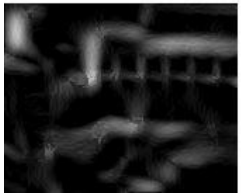
Feature scale space

original

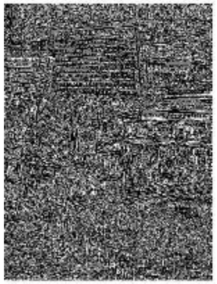
$\sigma = 1.5$

$\sigma = 3.0$

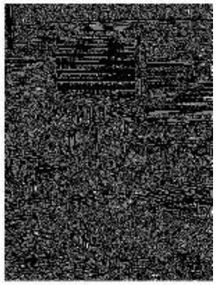
$\sigma = 6.0$



$\sigma = 12.0$



$\sigma = 1.0$



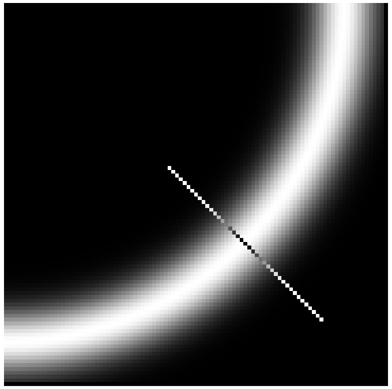
$\sigma = 2.0$

$\sigma = 4.0$

$\sigma = 10.0$

Steps in edge detection

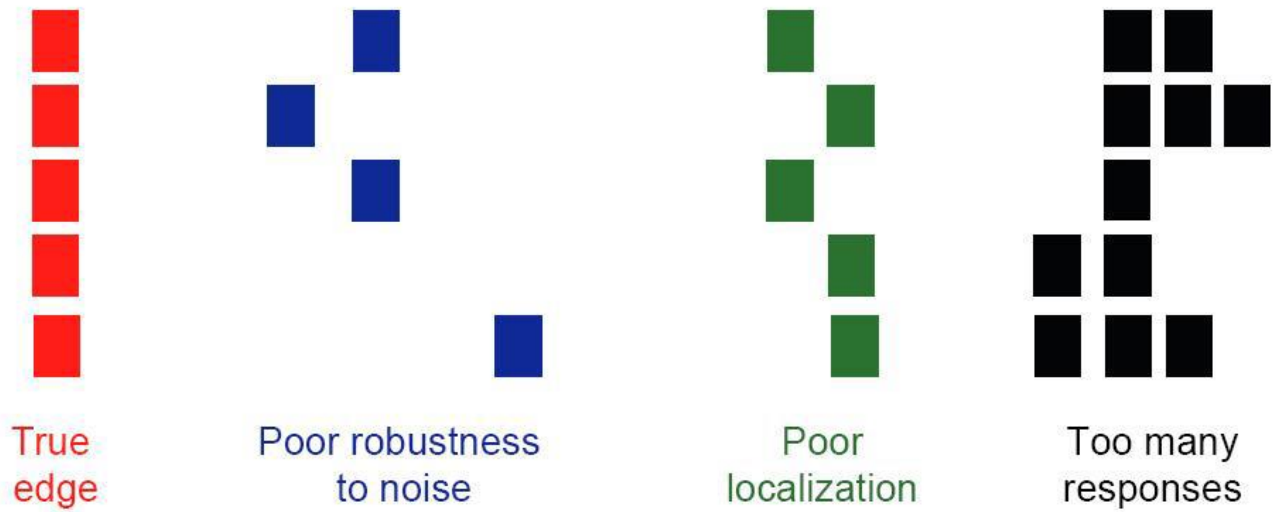
1. Image smoothing for reduction of noise and irrelevant details
2. Detection of edge points
3. Edge localization - Most edges are not sharp dropoffs. Extracting the ideal edge is thus a matter of finding the curve with peak/optimal gradient magnitude.
4. Edge Linking



Optimal edge detector

Criteria for an "optimal" edge detector:

- 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



2nd Derivative Methods

- Finding optimal edges (maxima of gradient magnitude) is equivalent to finding *places where the second derivative is zero*
- The zeroes may not fall exactly on a pixel. We can isolate these zeroes by finding *zero crossings*: places where one pixel is positive and a neighbor is negative (or vice versa)
- zero crossings form closed paths and extremely sensitive to noise

2nd Derivative Methods: Laplacian

We encountered the 2nd-order derivative based Laplacian filter already – isotropic i.e rotationally invariant

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 .It is combined with a smoothing Gaussian filter for purpose of edge detection

Laplacian Of Gaussian (Marr-Hildreth edge detector)

- To reduce the noise effect and also to adjust the scale, the image is first smoothed.
- When the filter chosen is a Gaussian, we call it the LoG edge detector.

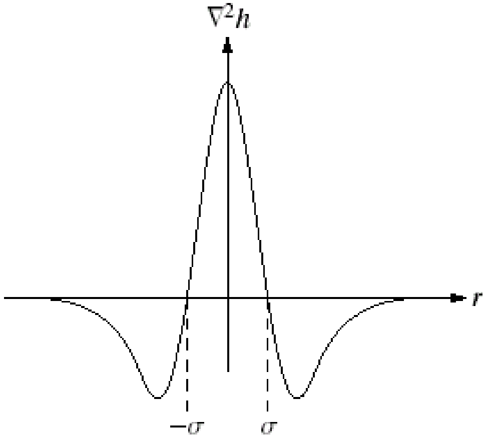
$$G(x, y) = e^{-\frac{x^2+y^2}{2\sigma^2}}$$

σ controls smoothing/adjusts scale

- It can be shown that:

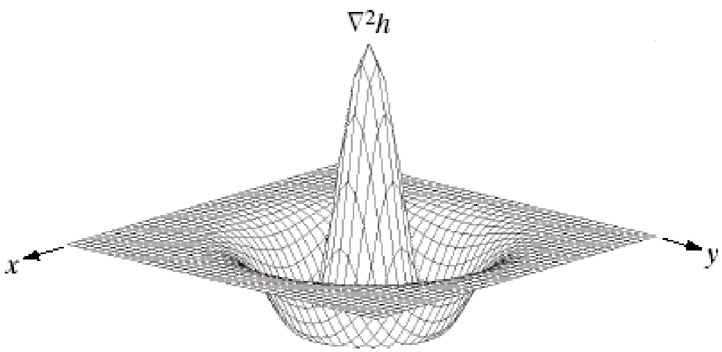
$$\nabla^2[f(x,y) * G(x,y)] = \nabla^2 G(x,y) * f(x,y)$$

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

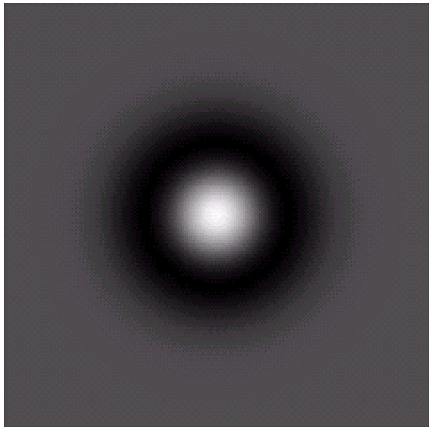


Laplacian Of Gaussian (Marr-Hildreth edge detector)

The Laplacian of Gaussian function can be approximated over a discrete spatial neighborhood to give a convolution kernel which looks like a Mexican hat.

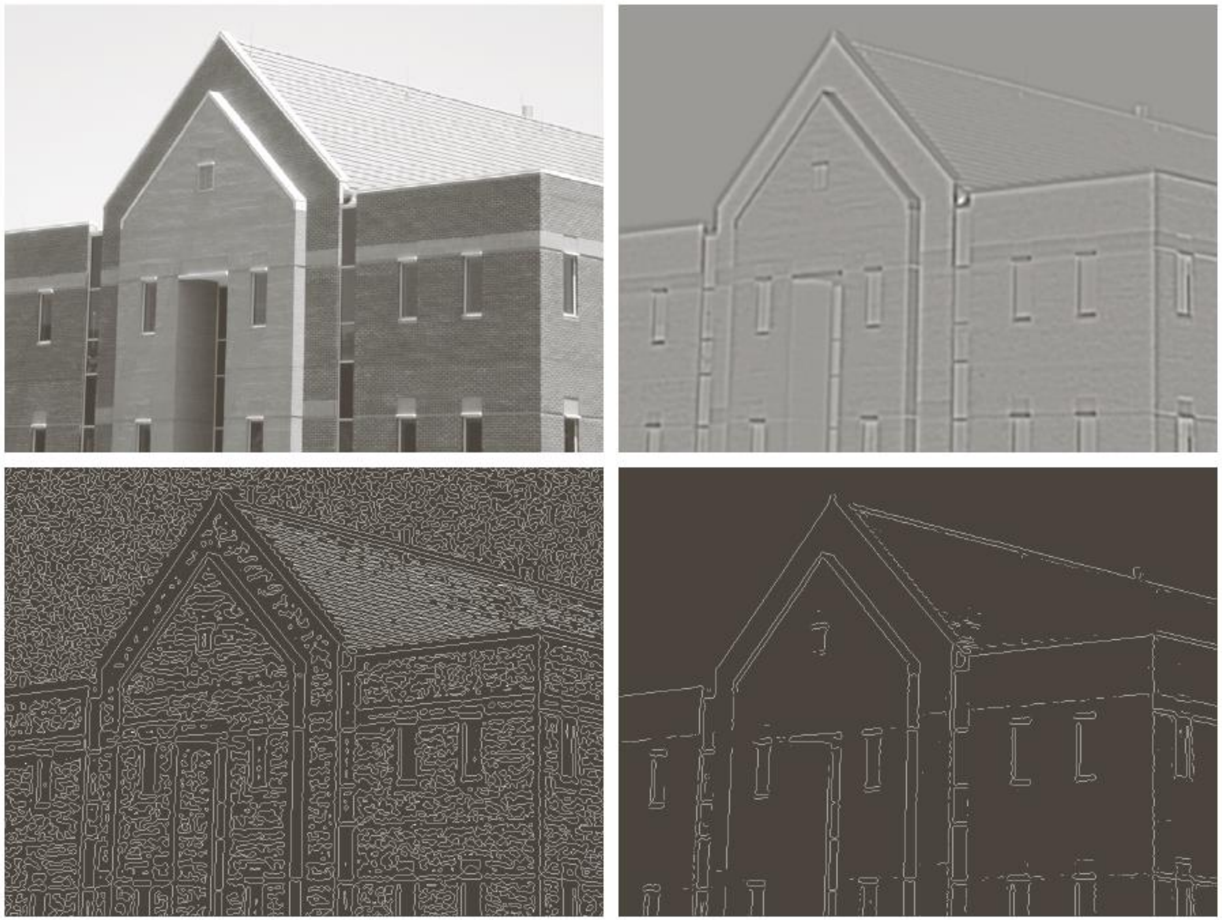


0	0	−1	0	0
0	−1	−2	−1	0
−1	−2	16	−2	−1
0	−1	−2	−1	0
0	0	−1	0	0



n (size of LOG filter) should be smallest odd integer greater or equal to 6σ

Example Result

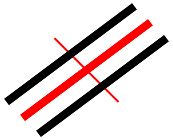


a b
c d

FIGURE 10.22 (a) Original image of size 834 × 1114 pixels, with intensity values scaled to the range [0, 1]. (b) Results of Steps 1 and 2 of the Marr-Hildreth algorithm using $\sigma = 4$ and $n = 25$. (c) Zero crossings of (b) using a threshold of 0 (note the closed-loop edges). (d) Zero crossings found using a threshold equal to 4% of the maximum value of the image in (b). Note the thin edges.

The Canny Edge Detection

- Smooth the image with a Gaussian filter with spread σ.
- Compute gradient magnitude and direction at each pixel of the smoothed image.
- Zero out any pixel response $\leq$ the two neighboring pixels on either side of it, along the direction of the gradient. This is called nonmaximum suppression.



- Use double thresholding to segregate ‘strong’ (high-magnitude) and ‘weak’ edge pixels.
- Visit strong edge pixels and do connectivity analysis to keep only those weak edge pixels which are connected to them

The Canny Edge Detection

a b
c d

FIGURE 10.25 (a) Original image of size 834 × 1114 pixels, with intensity values scaled to the range [0, 1]. (b) Thresholded gradient of smoothed image. (c) Image obtained using the Marr-Hildreth algorithm. (d) Image obtained using the Canny algorithm. Note the significant improvement of the Canny image compared to the other two.

Edge Linking

- Set of pixels from edge detecting algorithms, seldom define a boundary completely because of noise, breaks in the boundary etc.
- Therefore, Edge detecting algorithms are typically followed by linking and other detection procedures, designed to assemble edge pixels into meaningful boundaries.
- 2 types – local and global

Local Processing

- Analyse the characteristics of pixels in a small neighbourhood (3x3, or 5x5) about every point that has undergone edge detection.
- All points that are similar are linked, forming a boundary of pixels that share some common properties.
  - strength of the response of the gradient operator used to produce the edge pixel
  - direction of the gradient.

$$\left|\nabla f(x,y) - \nabla(x_0, y_0)\right| \le E,$$

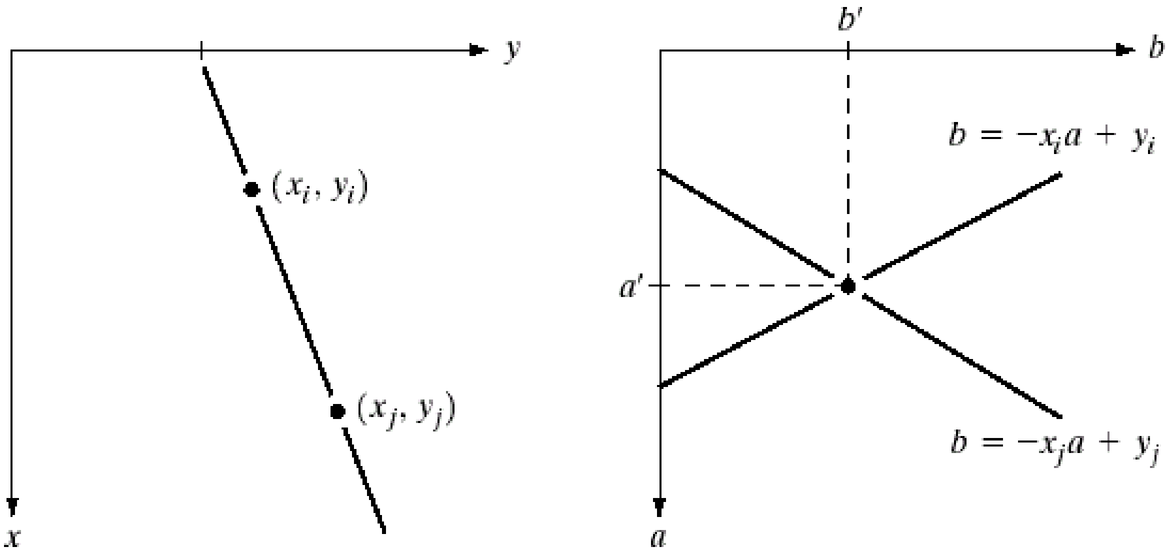
$$\left|\alpha(x,y) - \alpha(x_0, y_0)\right| < A$$

Global Processing via Hough Transform

Hough transform: a way of finding edge points in an image that lie on curves of specified shape.

Example: *xy*-plane v.s. *ab*-plane (parameter space)

$$y_i = ax_i + b$$

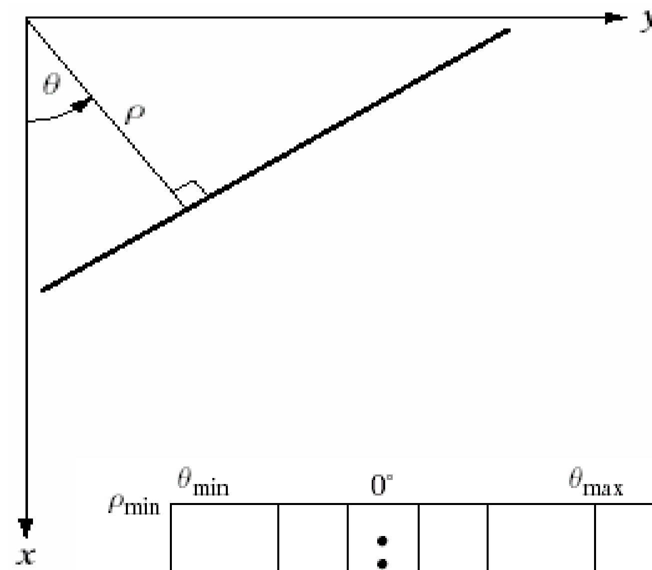


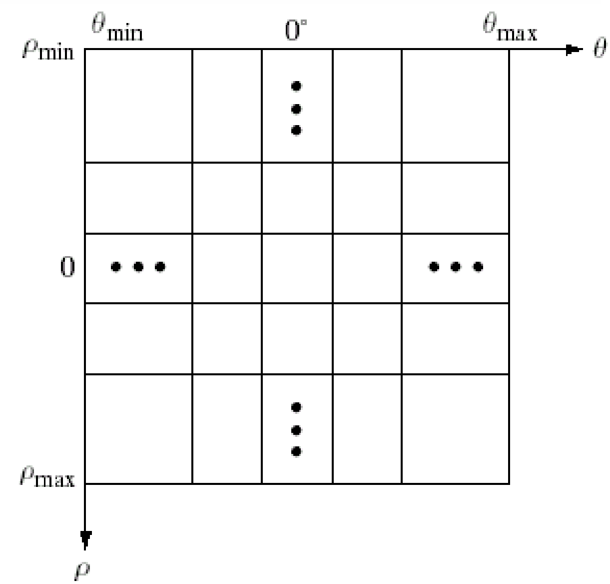
a b

FIGURE 10.17
(a) *xy*-plane.
(b) Parameter space.

Hough Transform

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





The Hough transform consists of finding all pairs of values of θ and ρ which satisfy the equations that pass through (x,y).

These are accumulated in what is basically a 2-dimensional histogram.

When plotted these pairs of θ and ρ will look like a sine wave. The process is repeated for all appropriate (x,y) locations.

Hough Transform

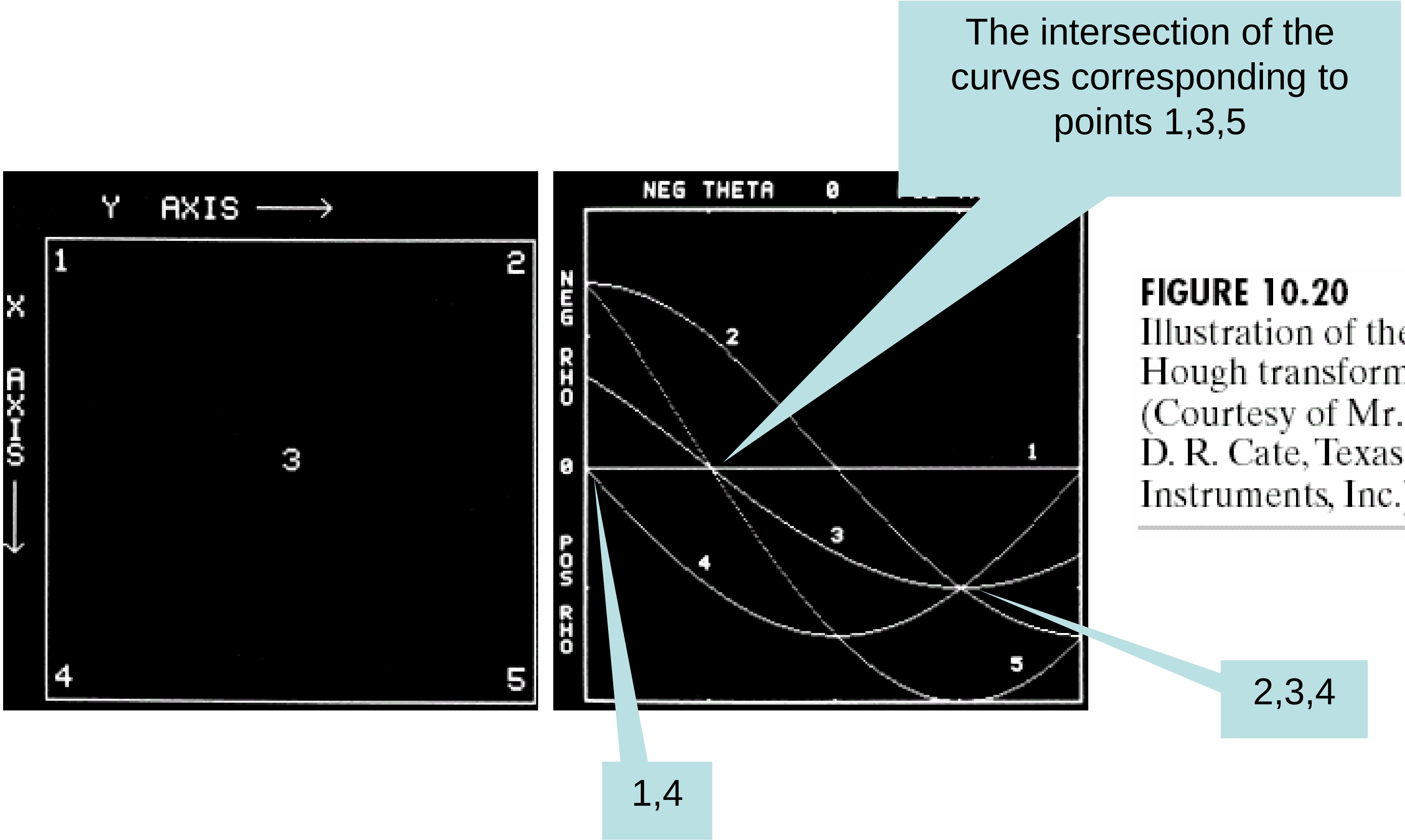


FIGURE 10.20 Illustration of the Hough transform. (Courtesy of Mr. D. R. Cate, Texas Instruments, Inc.)

Hough Transform

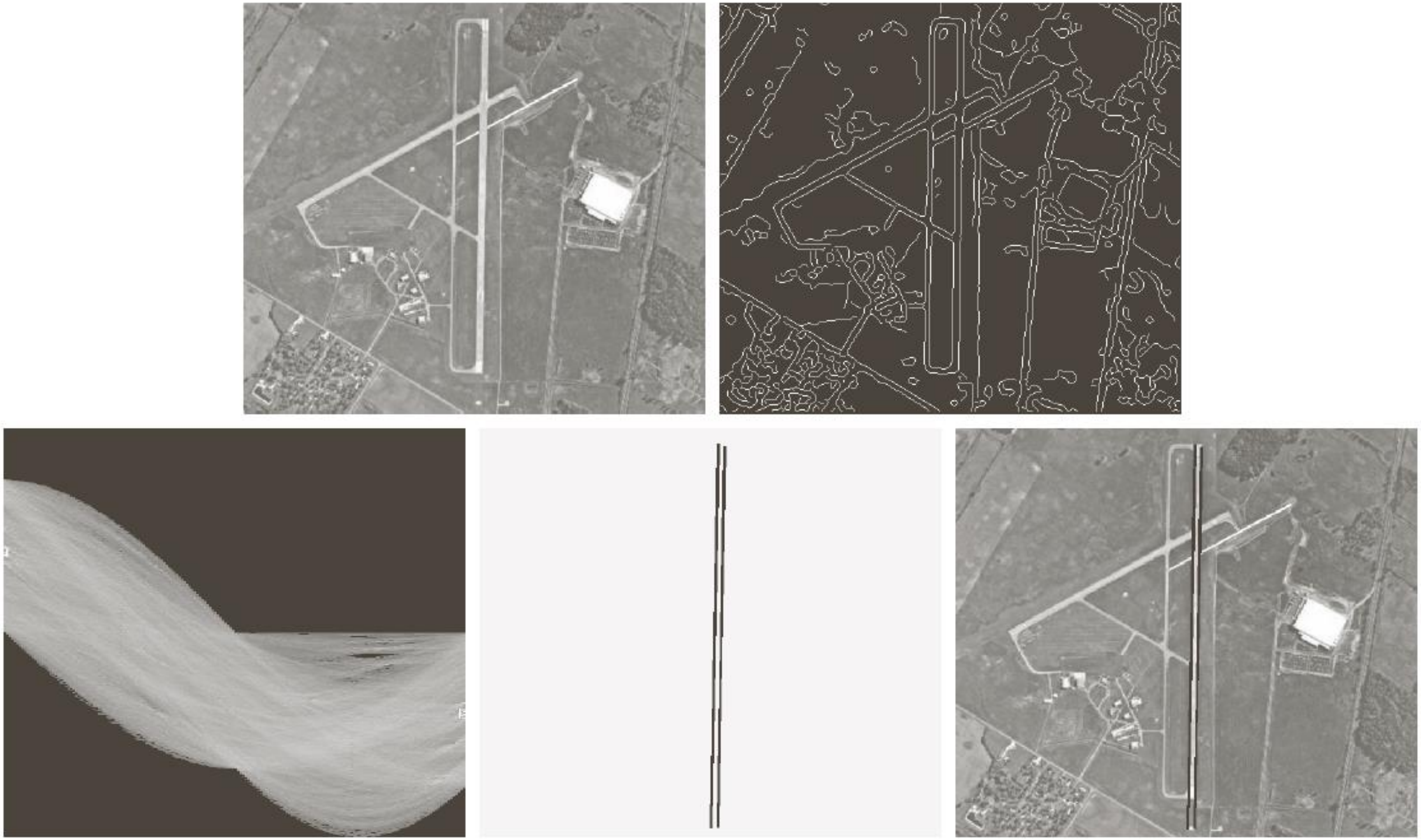
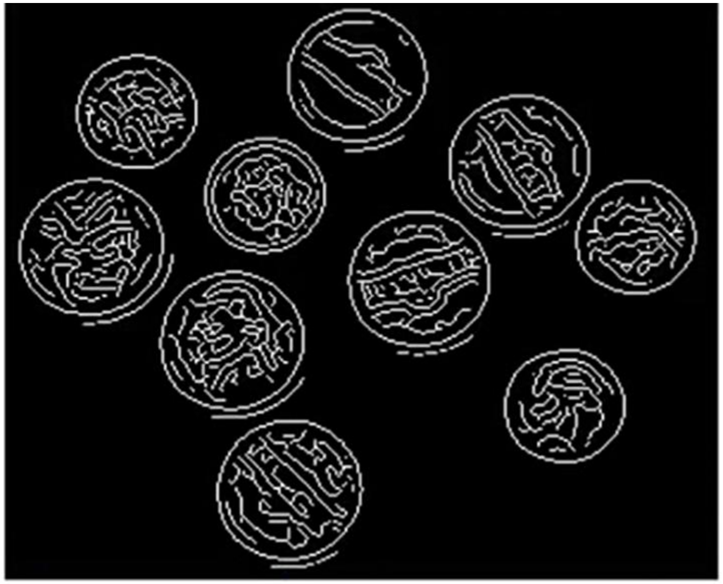


FIGURE 10.34 (a) A 502 × 564 aerial image of an airport. (b) Edge image obtained using Canny's algorithm. (c) Hough parameter space (the boxes highlight the points associated with long vertical lines). (d) Lines in the image plane corresponding to the points highlighted by the boxes). (e) Lines superimposed on the original image.

What about Circles?



Summary

- In this Part, 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 saw methods of edge linking, especially Hough transform