Image segmentation

Image Segmentation

- Segmentation subdivides an image into its constituent regions or objects.
- The level to which the subdivision is carried depends on the problem being solved.
- The goal is usually to find individual objects in an image.
- Eg: Analysing images of the products with the objective of determining the presence or absence of specific anomalies such as missing components or broken connection paths

Segmentation Examples

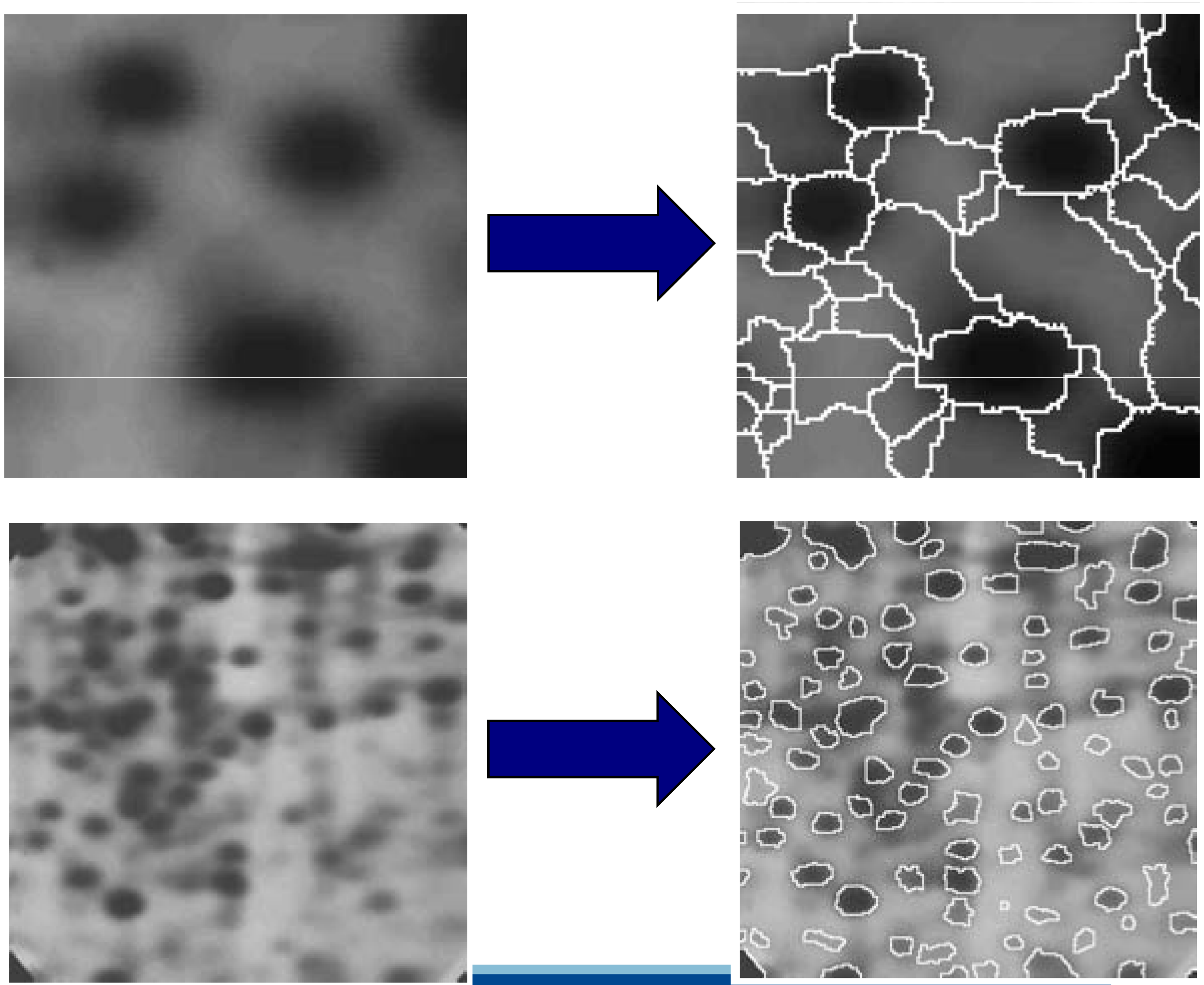


Image Segmentation

- Image segmentation algorithms generally are based on one

of two basic properties of intensity values:

discontinuity and similarity.

- In intensity discontinuity approach, image is partitioned based on abrupt changes in intensity, such as edges in an image.

- The principal approaches in the second category are based

on partitioning an image into regions that are similar according to a set of pre-defined criteria.

Detection of Discontinuities

- There are three kinds of gray-level discontinuities in digital images: points, lines and edges.
- The most common way to look for discontinuities is to scan a small mask over the image.
- The procedure involves computing the sum of products of the coefficients with gray levels contained in the region

$$R = w_1 z_1 + w_2 z_2 + \cdots + w_9 z_9 = \sum_{i=1}^{9} w_i z_i$$

- Where z_i is the gray level of the pixel associated with

Point Detection

- The point is detected at the location on which the mask is centered if

$$|R| \geq T$$

$$\text{where } T\text{:a nonnegative threshold}$$

- The formulation measures the weighted differences betweent the center point and its neighbors.
- The idea is that an isolated point (which is different from background) is different from its surroundings.

$$R = w_1 z_1 + w_2 z_2 + \ldots + w_9 z_9 = \sum_{i=1}^{9} w_i z_i$$

Point Detection

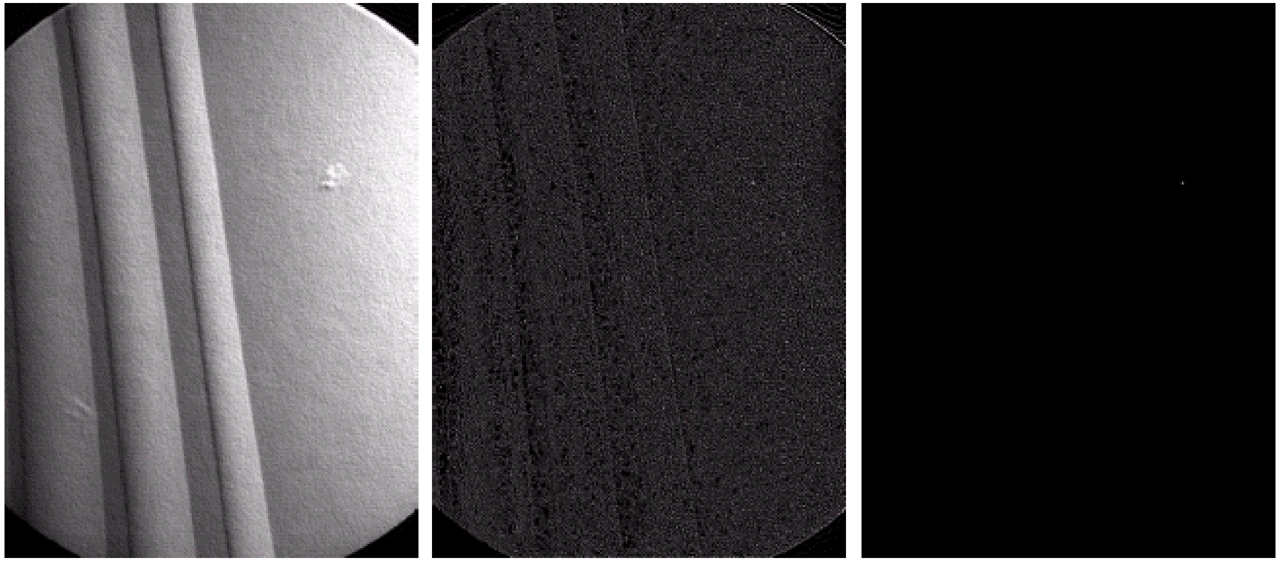
Point detection can be achieved simply using the mask below:

–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
b c d

FIGURE 10.2 (a) Point detection mask. (b) X-ray image of a turbine blade with a porosity. (c) Result of point detection. (d) Result of using Eq. (10.1-2). (Original image courtesy of X-TEK Systems Ltd.)

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

- With constant background, the maximum response would result when line passed through the middle row of the mask.
- First mask: Horizontal lines
- Seond mask reponds: lines oriented with +45 degree
- Thrid mask: vertical lines
- Fourth mask: lines -45 degree
- Directions can be established based on weights with

larger coefficients.

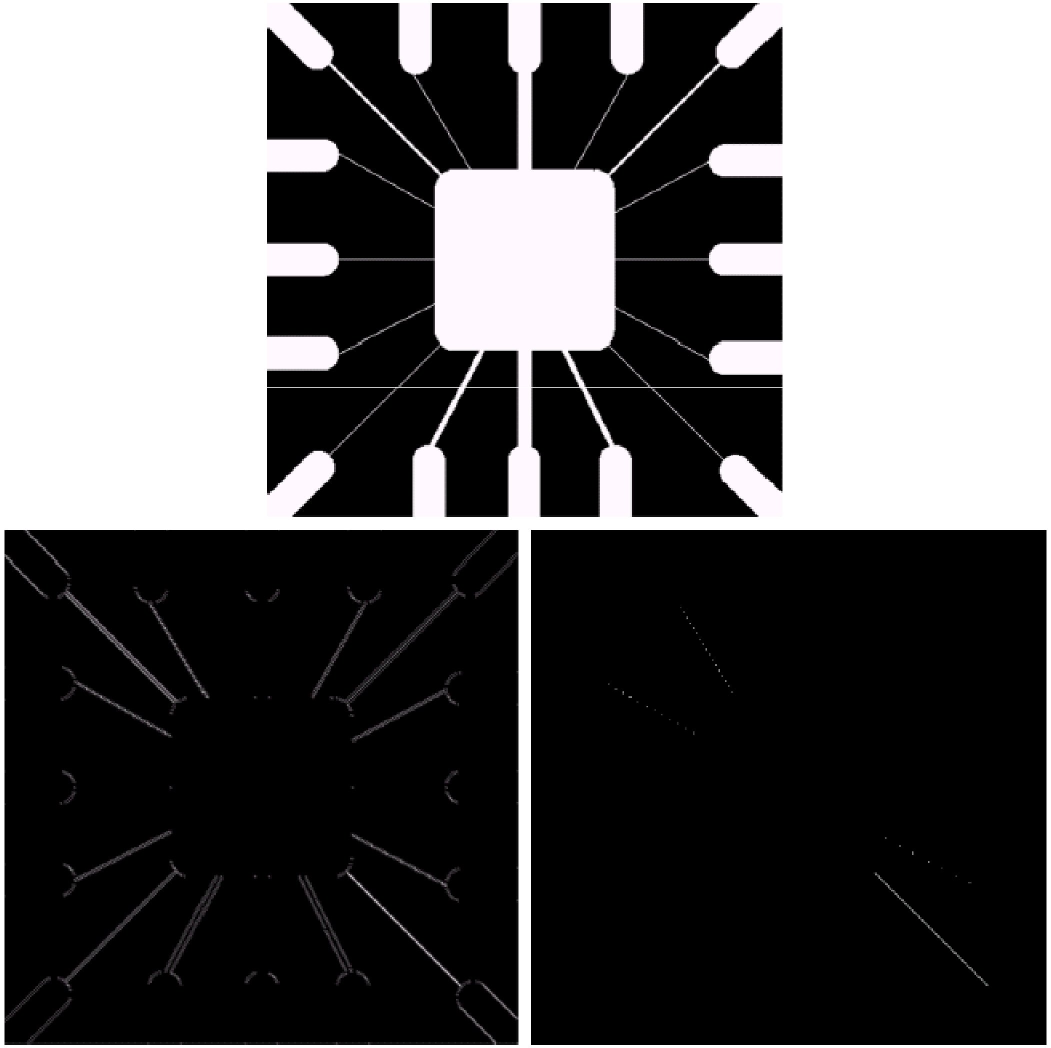
- Coefficients of each mask sum to zero indicates a zero response in areas of constant gray level.

Line Detection

- If we are interested in detecting all the lines in the image in the direction defined by a given mask
  - Run the mask through the image
  - Threshold the absolute value of the result

- The points that are left are the strongest responses where the lines are one pixel thick corresponds to closest direction defined by the mask

Detection of Discontinuities
Line Detection



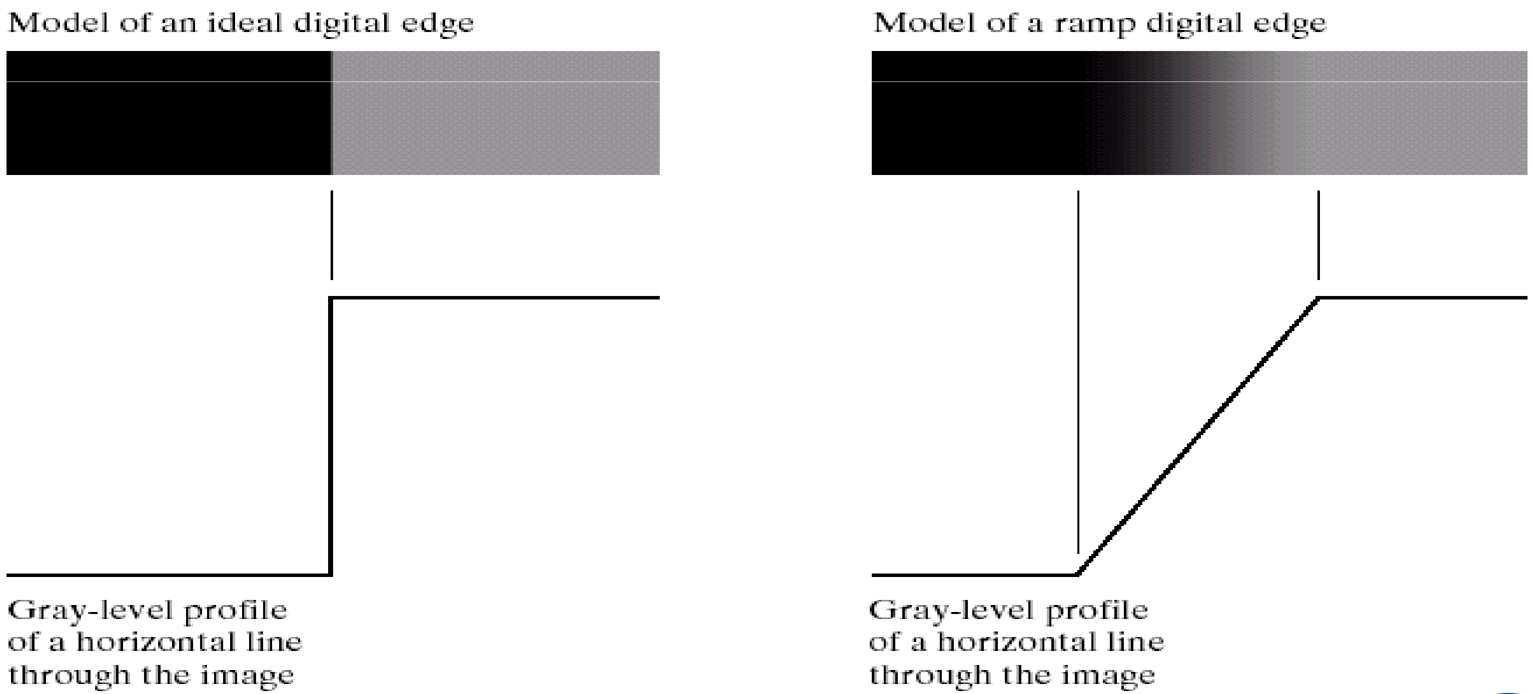
a
b c

FIGURE 10.4 Illustration of line detection. (a) Binary wire-bond mask. (b) Absolute value of result after processing with −45° line detector. (c) Result of thresholding image (b).

Edge Detection

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

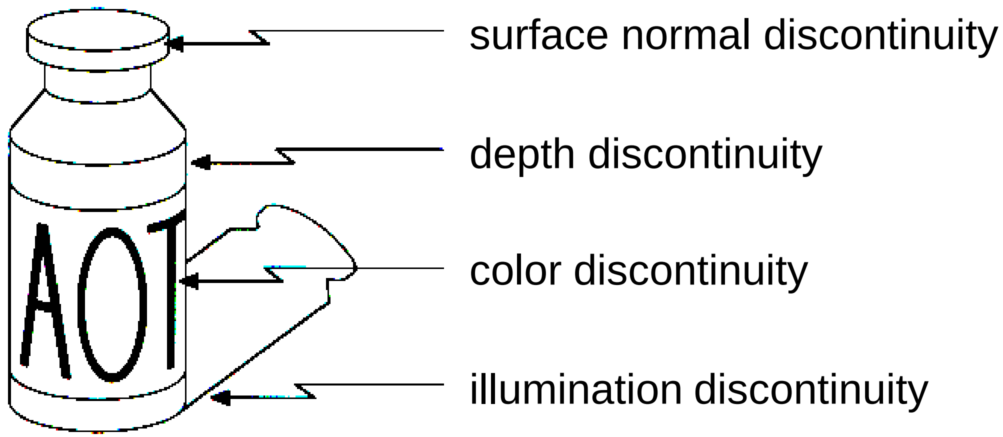
Edge has the ability to measure gray-level transitions.



Edge detection (intensity changes)

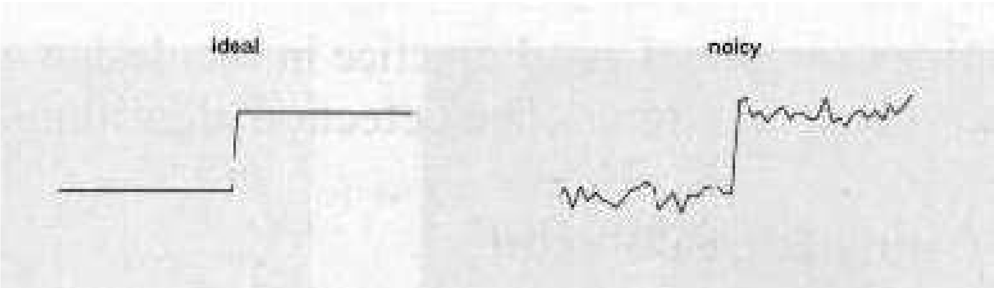
- Edges are significant local changes in intensity in an image
- Geometric events
  - surface orientation (boundary) discontinuities
  - depth discontinuities
  - color and texture discontinuities

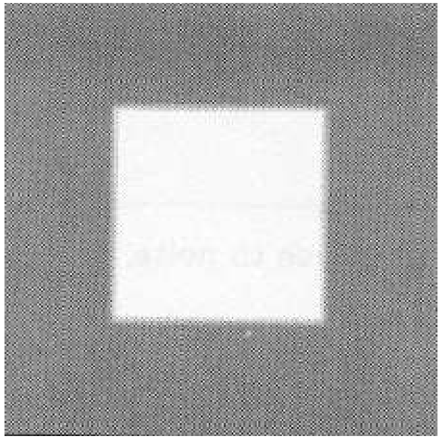
- Non-geometric events
  - illumination changes
  - specularities
  - shadows
  - inter-reflections



Step Edge

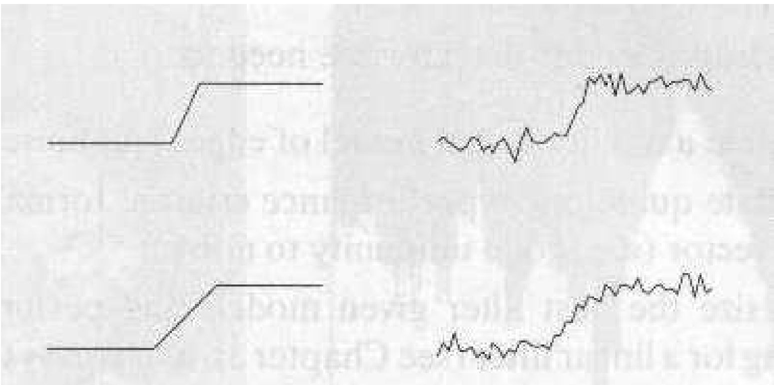
- **Step edge:** the image intensity abruptly changes from one value on one side of the discontinuity to a different value on the opposite side.

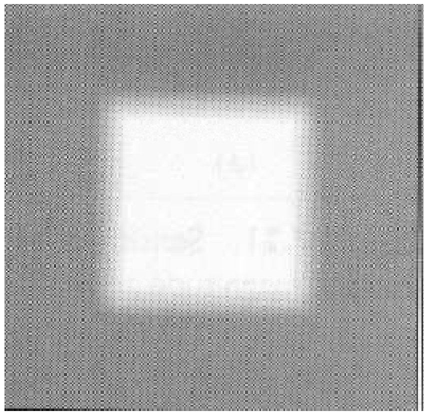




Ramp Edge

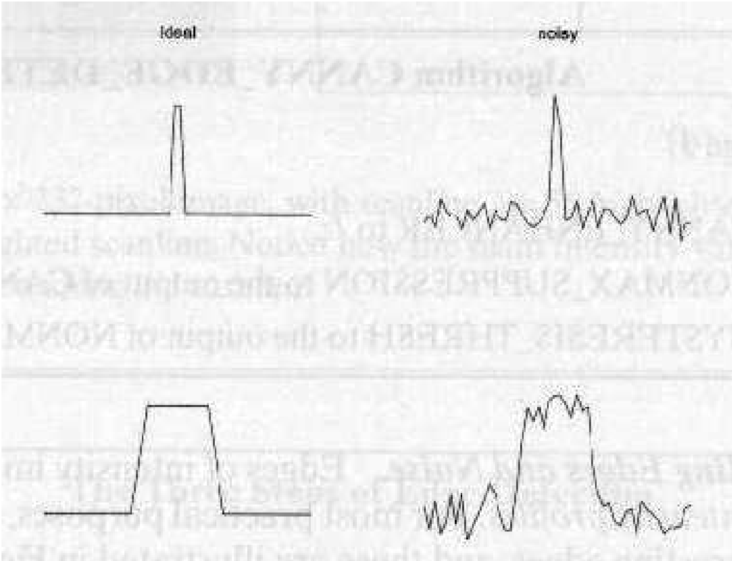
- **Ramp edge:** a step edge where the intensity change is not instantaneous but occur over a finite distance.
- Slope of the ramp inversely proportional to the degree of blurring
- Length is determined by the slope
- Blurred edges-more thick, sharp edges-thin





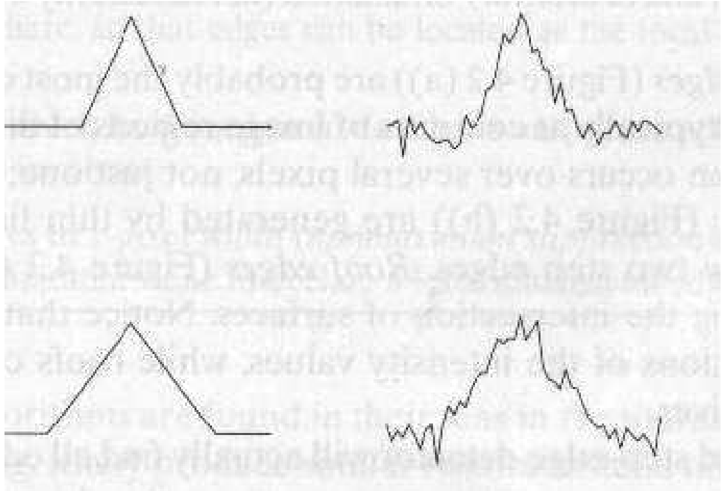
Ridge edge

- **Ridge edge:** the image intensity abruptly changes value but then returns to the starting value within some short distance (i.e., usually generated by lines).



Roof edge

Roof edge: a ridge edge where the intensity change is not instantaneous but occur over a finite distance (i.e., usually generated by the intersection of two surfaces



Main steps

Smoothing: suppress as much noise as possible, without destroying true edges.

Enhancement: apply differentiation to enhance the quality of edges (i.e., sharpening).

Thresholding: determine which edge pixels should be discarded as noise and which should be retained (i.e., threshold edge magnitude).

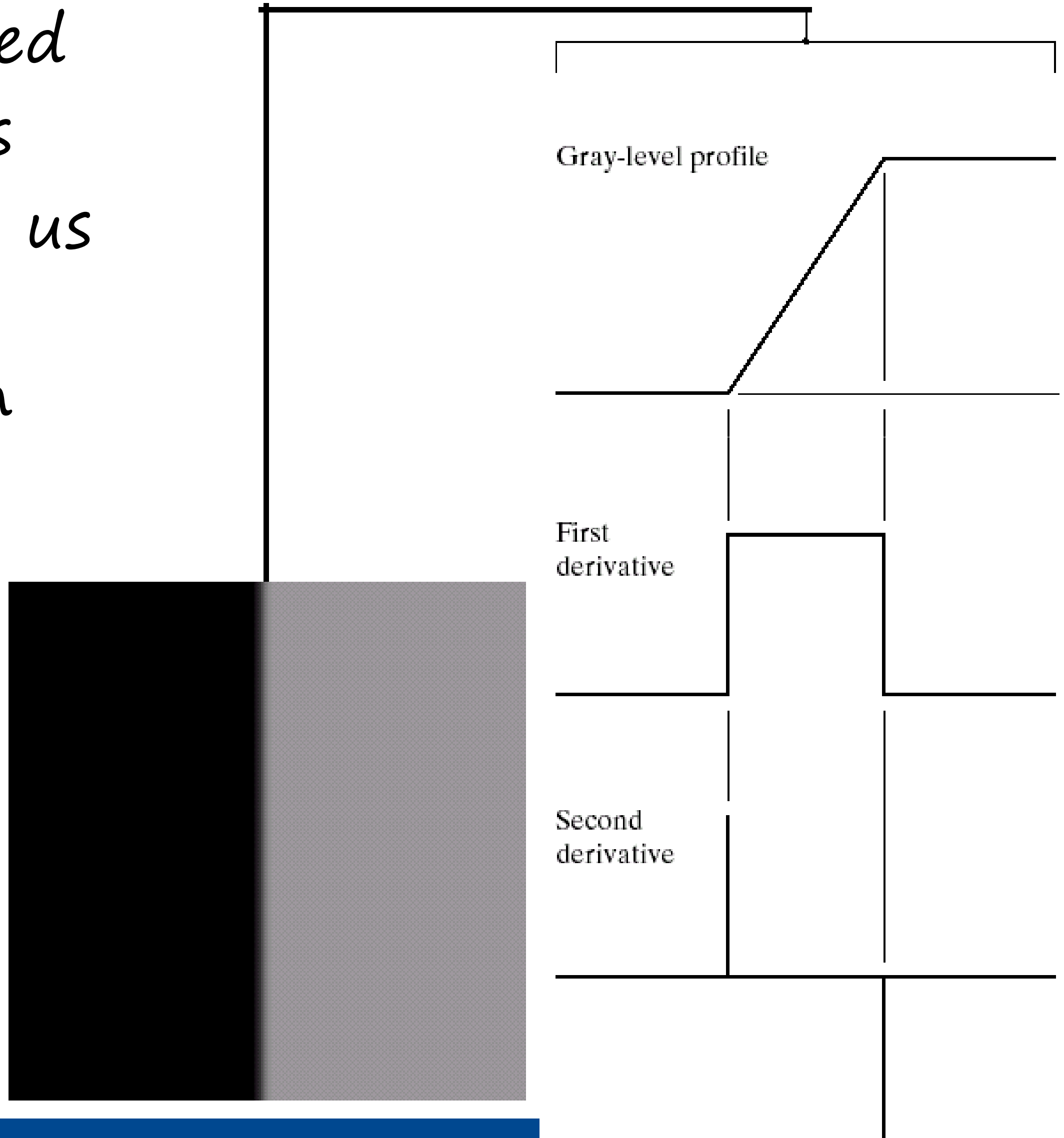
Localization: determine the exact edge location.

Maxima or minima of first derivative
Zero crossing of the second derivative

Edges & Derivatives

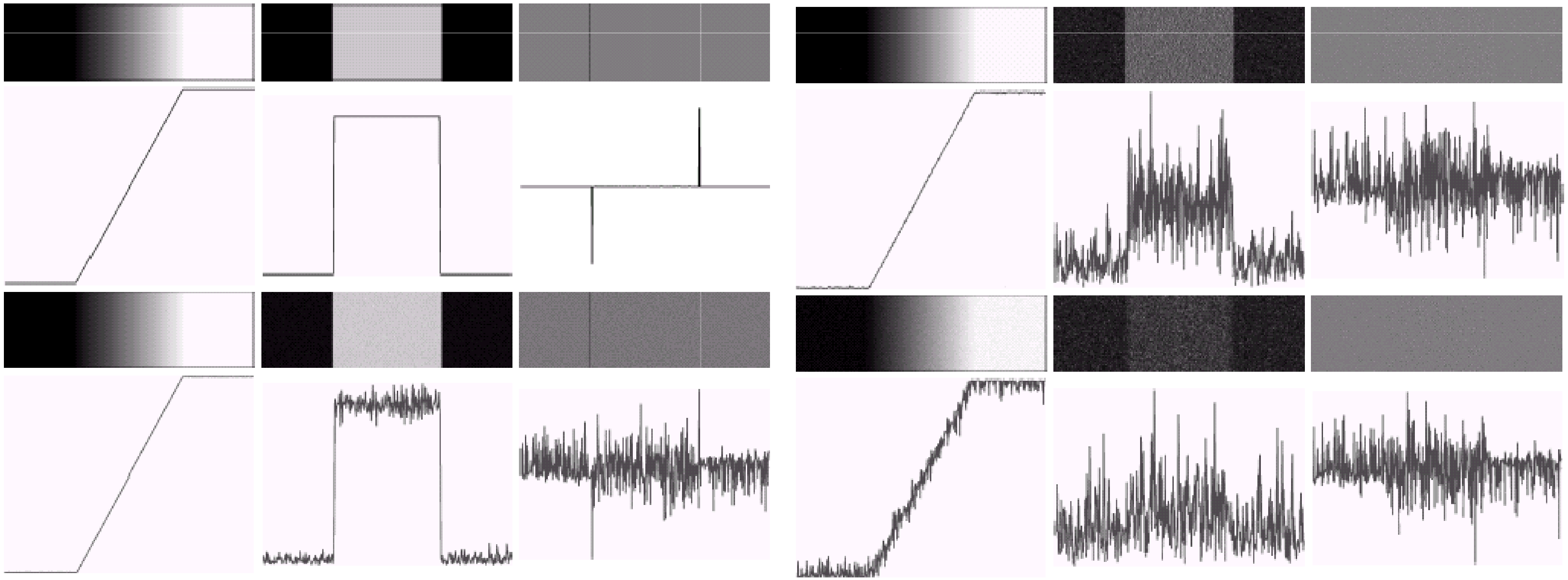
The derivatives are used to find discontinuities

- 1st derivative tells us where an edge is
- 2nd derivative can be used to show edge direction



Derivatives & Noise

- Derivative based edge detectors are extremely sensitive to noise



First order edge detection operator

- Low level features extracted from the image without any shape information (spatial relationship)
- Edge detection highlights the contrast (change in intensity)
- Detecting contrast emphasize the boundaries within the image
- Boundaries (intensity variation occurs)
- Edge detection – first order differentiation

Basic operators

- Differencing adjacent points
- Horizontal edge detection
  - Differencing horizontal adjacent points will detect vertical changes in intensity (vertical edges)

$$Ex_{x,y} = \left|P_{x,y} - P_{x+1,y}\right|$$

- Vertical edge detection
  - Differencing vertical adjacent points will detect horizontal changes in intensity (horizontal edges)
  - $Ex_{x,y} = \left|P_{x,y} - P_{x,y+1}\right|$

Gradient Operators

- First derviatives in image processing implemented using magnitude of gradient
- Let f(x,y) the gradient of f at coordinates (x,y) is defined 2d column vector

$$\nabla f = grad(f) = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix},$$

This vector points to the direction of greatest rate of change of f at location (x,y)

Gradient Operators

- Magnitude of the vector ∇f denoted by M(x,y) where

$$magnitude(grad(f)) = \sqrt{\frac{\partial f}{\partial x}^2 + \frac{\partial f}{\partial y}^2}$$

$$direction(grad(f)) = \tan^{-1}\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x}\right)$$

The square and square root operations by absolute values $M(x,y)=|g_x|+|g_y|$ ----*

Roberts Cross-Gradient Operators

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

−1	0
0	1

0	−1
1	0

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

- Z_5 denotes f(x,y) at an arbirary location (x,y).
- z_1 denotes f(x-1,y-1)
- Roberts use cross differences

$g_x=(z_9-z_5)$ and $g_y=(z_8-z_6)$-----**

- The gradiant of the image is defined as

$M(x,y)=[(z_9-z_5)^2 +(z_8-z_6)^2]^{1/2}$

From * and ** we get

- $M(x,y)=|z_9-z_5| +|z_8-z_6|$ (Roberts cross-gradient operators)

Prewitt edge detection

- Edge detection – sensitive to intensity changes
- Respond to noise as well as step like changes
- Differencing (high pass filter)
- Incorporate averaging within the edge detection process
- Extend the vertical template (Mx) by three rows and horizontal template (My) with three columns

$g_x=(z_7+z_8+Z_9)-(z_1+z_2+Z_3)$ and $g_y=(z_3+z_6+Z_9)-(z_1+z_4+Z_7)$

-1	-1	-1
0	0	0
1	1	1

-1	0	1
-1	0	0
-1	0	1

Sobel operator

- This gives
  - Rate of change of brightness along each axis
  - Edge magnitude – the length of the vector and edge direction

  Angle in degree gives the direction
- Sobel operator
- Weight of the central pixel for the Prewitt template is doubled

$g_x=(z_7+2z_8+Z_9)-(z_1+2z_2+Z_3)$ and $g_y=(z_3+2z_6+Z_9)-(z_1+2z_4+Z_7)$

-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

Common Edge Detectors

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

0	1	1
-1	0	1
-1	-1	0

-1	-1	0
-1	0	1
0	1	1

0	1	2
-1	0	1
-2	1	0

-2	-1	0
-1	0	1
0	1	2

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 Problems

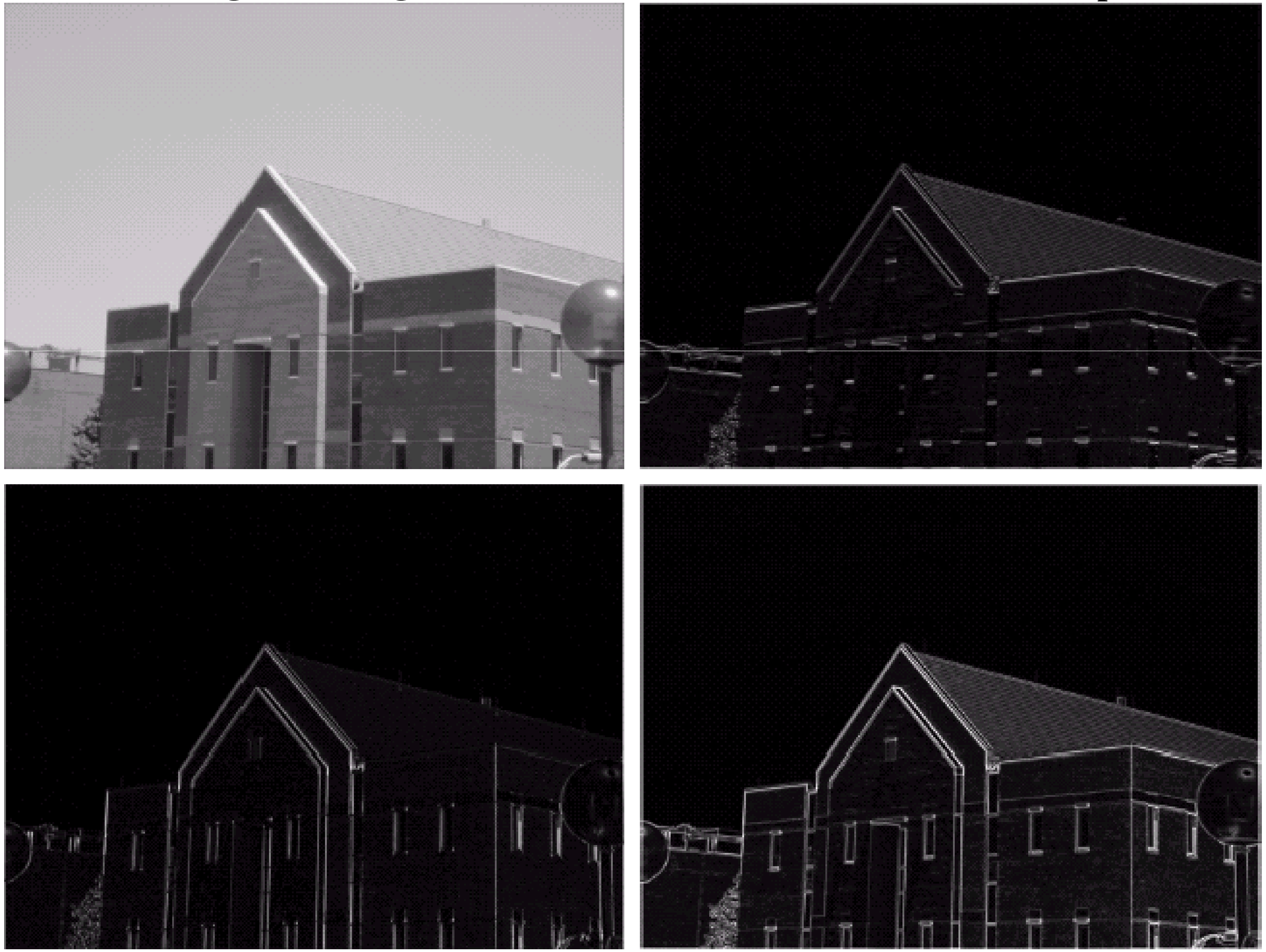
Often, problems arise in edge detection in that there are is too much detail

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

Detection of Discontinuities
Gradient Operators

Second-order derivatives: (The Laplacian)

Cons: Laplacian is very sensible to noise; double edges,unable to detect direction

The Laplacian of an 2D function $f(x,y)$ is defined as

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

- **FIGURE 10.13** Laplacian masks used to implement Eqs. (10.1-14) and (10.1-15), respectively.

0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

Detection of Discontinuities
Gradient Operators

Consider the function:

$$h(r) = -e^{-\frac{r^2}{2\sigma^2}} \quad \text{where} \quad r^2 = x^2 + y^2$$

and σ : the standard deviation

A Gaussian function

The Laplacian of *h* is

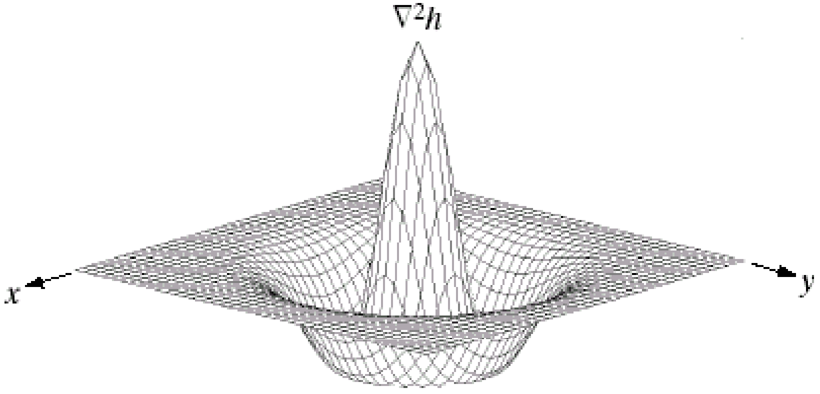
$$\nabla^2 h(r) = -\left[\frac{r^2 - \sigma^2}{\sigma^4}\right] e^{-\frac{r^2}{2\sigma^2}}$$

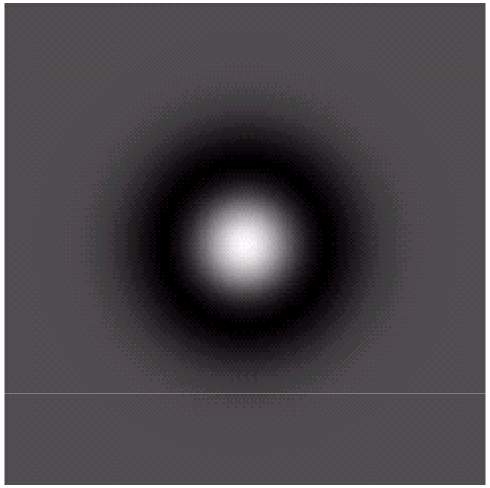
The Laplacian of a Gaussian (LoG)

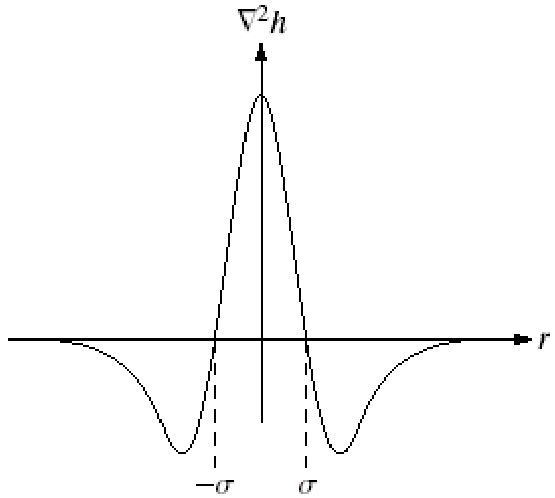
Detection of Discontinuities
Gradient Operators

- The Laplacian combined with smoothing as a precursor in finding edges via zero crossings property
- Those points where the Laplacian changes sign. Such points often occur at `edges' in images
- The Laplacian of a Gaussian sometimes is called the Mexican hat function or commonly referred as Laplacian of a Gaussian
- It also can be computed by smoothing the image with the Gaussian smoothing mask, followed by application of the Laplacian mask.

Detection of Discontinuities
Gradient Operators





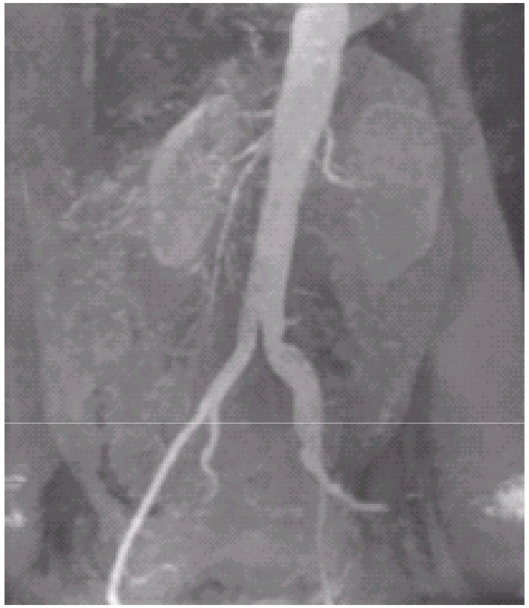


0	0	–1	0	0
0	–1	–2	–1	0
–1	–2	16	–2	–1
0	–1	–2	–1	0
0	0	–1	0	0

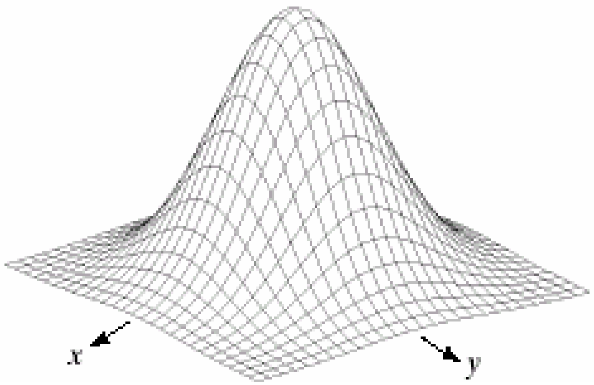
a b
c d

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).

Detection of Discontinuities
Gradient Operators



Sobel gradient

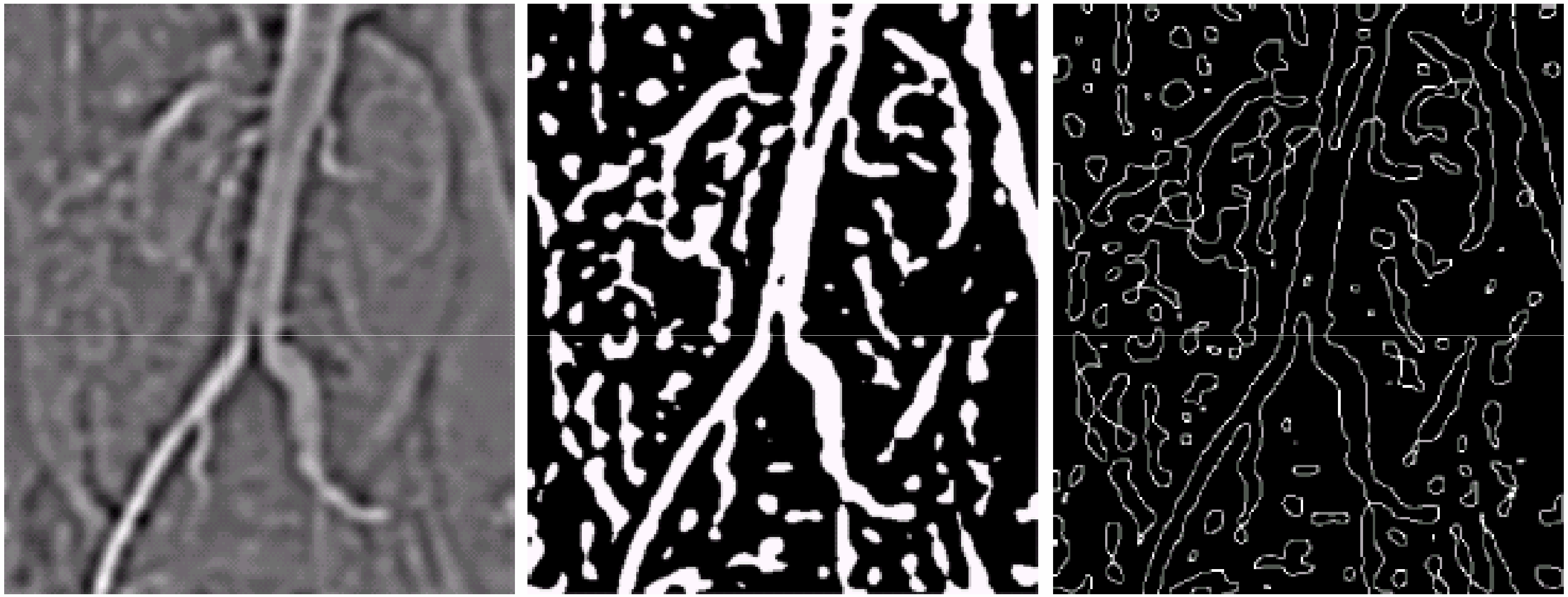


Gaussian smooth function

−1	−1	−1
−1	8	−1
−1	−1	−1

Laplacian mask

Detection of Discontinuities
Gradient Operators



a b
c d
e f g

FIGURE 10.15 (a) Original image. (b) Sobel gradient (shown for comparison). (c) Spatial Gaussian smoothing function. (d) Laplacian mask. (e) LoG. (f) Thresholded LoG. (g) Zero crossings. (Original image courtesy of Dr. David R. Pickens, Department of Radiology and Radiological Sciences, Vanderbilt University Medical Center.)

- Zero crossings edges are thinner
- Form closed contours
- Computation of zero crossings is complex

Canny edge operator

- **Good detection (optimal)**
  - Minimize the probability of false positives (i.e., spurious edges).
  - Minimize the probability of false negatives (i.e., missing real edges).
  - Reduce the response to noise
  - Gaussian filtering
  - Optimal detection with no spurious responses

- **Good localization**
  - Detected edges must be as close as possible to the true edges.
  - Thinning (thin line of edges in the right place)

- **Single response**
  - Minimize the number of local maxima around the true edge

- Steps in Canny Algorithm
- 1. Smooth the input image with the gaussian filter
- 2. Compute the gradient magnitude and angle images.
- 3. Apply non maximum suppression to the gradient magnitude image.
- 4. Use hysteresis (double) thresholding and connectivity analysis to detect and link the edges.

Canny

- Gaussian operator is optimal for image smoothing
- $g(x, y, \sigma) = e^{\frac{-(x^2+y^2)}{2\sigma^2}}$
- $\nabla g(x, y) = \frac{\partial g(x,y,\sigma)}{\partial x} U_x + \frac{\partial g(x,y,\sigma)}{\partial y} U_y$

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

Helps in calculating the coefficients of a derivative of Gaussian template combining the first order differentiation with Gaussian smoothing.

Non-maximum suppression

- Non-maximum suppression locates the highest points in the edge magnitude data
- This is performed by using edge detection information to check that points are at the peak of a ridge.
- In a 3x3 region a point is at a maximum if the gradient on either side of it is less than the gradient at that point.
- If M1 and M2 are the neighbours, M(x,y) is point of interest and if its gradient is greater than the other two it is considered as maximum otherwise set to zero
- Gn(x,y) is the non maxima suppressed image.

canny

- The location of the true edge point is then at the maximum point of Gn convolved with the image.
- This maximum is when the differential is zero

$$\frac{\partial(G_n * P)}{\partial n} = 0$$

$$\frac{\partial^2(G * P)}{\partial n^2} = 0$$

- Edges are detected in the correct place
  - non-maximum suppression
  - Retaining peaks which thins the response of the edge detection operator to give edge points without multiple

Hysteresis thresholding

- To reduce false edge points in Gn(x,y).
- If T is very low it leads to false positive edges and if too high it leads to false negative edges.
- Canny algorithm attempts to improve this by using hysteresis thresholding
- Points are set to white when the magnitude exceeds the upper threshold
- Set ot black when lower threshold is reached
- Upper threshold, lower threshold

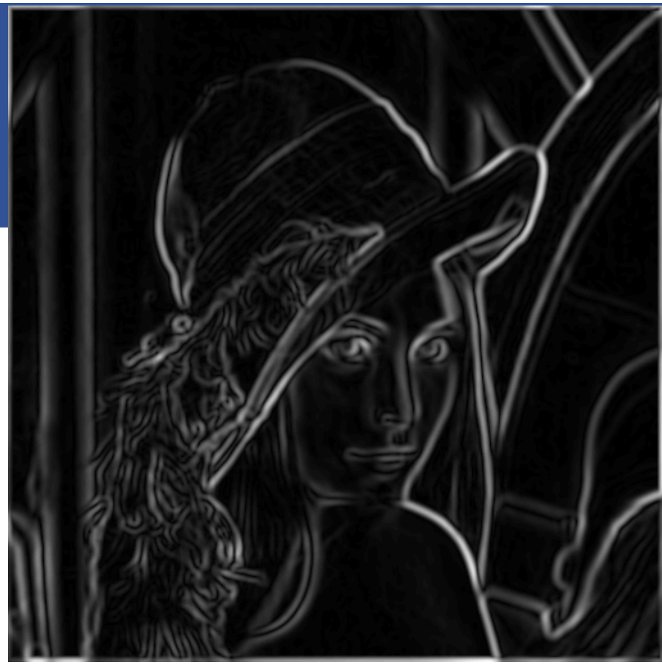
Hysteresis threshold

- The edge point from the non-maximum suppression is found to exceed the upper threshold this is set as edge point (white)
- This form the first point of a line of edge points
- The neighbours of this point is searched to determine whether they exceed the lower threshold
- Any neighbouring point that exceeds the lower threshold is labelled as an edge point and its neighbours are searched and the process is repeated.
- The search is terminated when the branch does not have points exceeding the lower threshold

$\geq$ lower	$\geq$ lower	$\geq$ lower
$\geq$ lower	Seed $\geq$ upper	$\geq$ lower
$\geq$ lower	$\geq$ lower	$\geq$ lower

Input image

Gradient magnitude



Thresholded gradient magnitude

Thinning

Edge Linking and Boundary Detection
Local Processing

- Two properties of edge points are useful for edge linking:
  - the strength (or magnitude) of the detected edge points
  - their directions (determined from gradient directions)
- This is usually done in local neighborhoods.
- Adjacent edge points with similar magnitude and direction are linked.
- For example, an edge pixel with coordinates (x_0,y_0) in a predefined neighborhood of (x,y) is similar to the pixel at (x,y) if

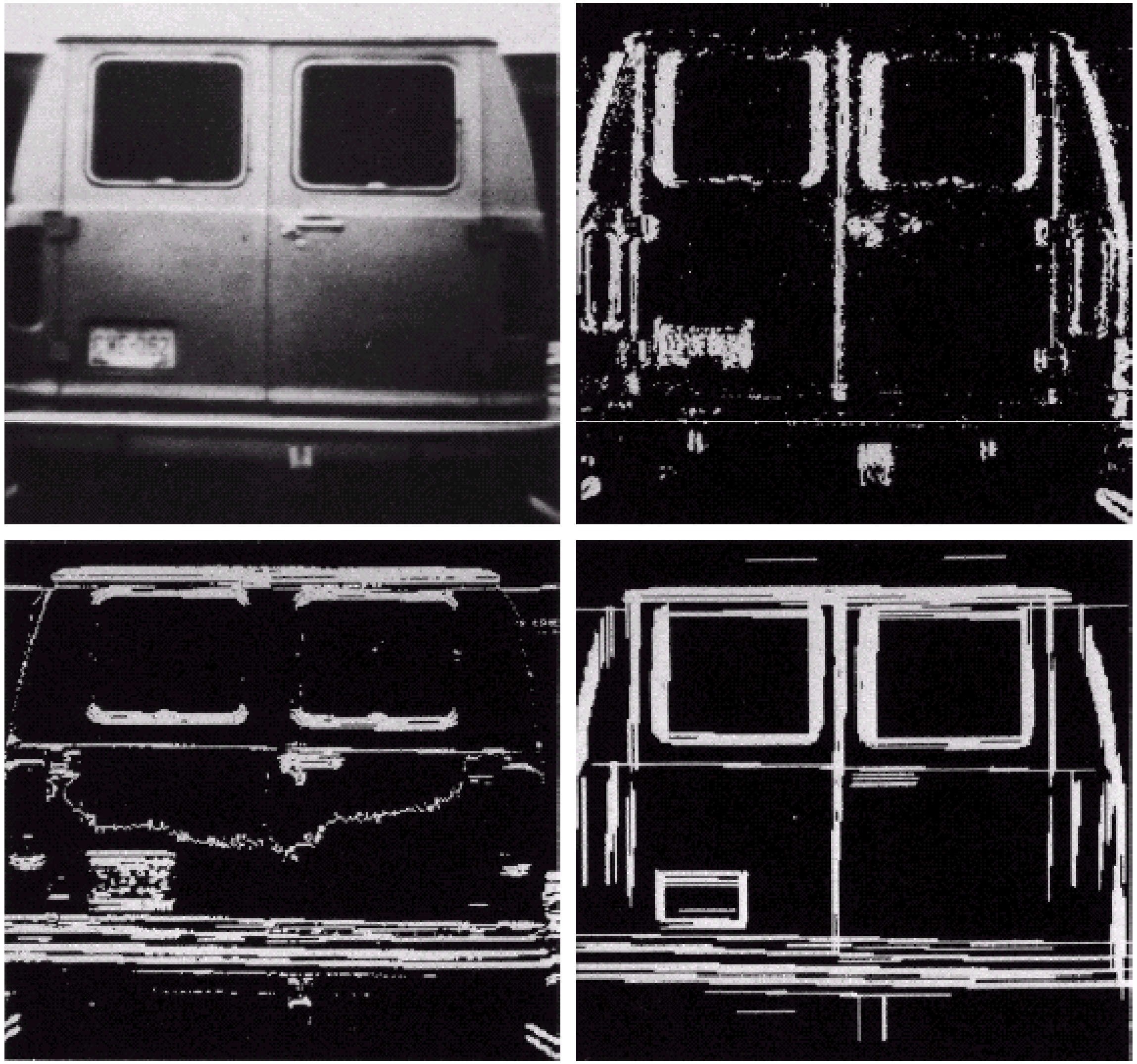
$$\left|\nabla f(x,y) - \nabla(x_0,y_0)\right| \le E, \quad E: \text{a nonnegative threshold}$$

$$\left|\alpha(x,y) - \alpha(x_0,y_0)\right| < A, \quad A: \text{a nonegative angle threshold}$$

Edge Linking and Boundary Detection
Local Processing

a b
c d

FIGURE 10.16
(a) Input image.
(b) G_y component of the gradient.
(c) G_x component of the gradient.
(d) Result of edge linking. (Courtesy of Perceptics Corporation.)



In this example, we can find the license plate candidate after edge linking process.