COL783: Digital Image Processing

Prem Kalra
pkalra@cse.iitd.ac.in
http://www.cse.iitd.ac.in/~pkalra/col783

Department of Computer Science and Engineering
Indian Institute of Technology Delhi

Image Segmentation

Image segmentation is the process of partitioning a digital image into multiple segments (regions)

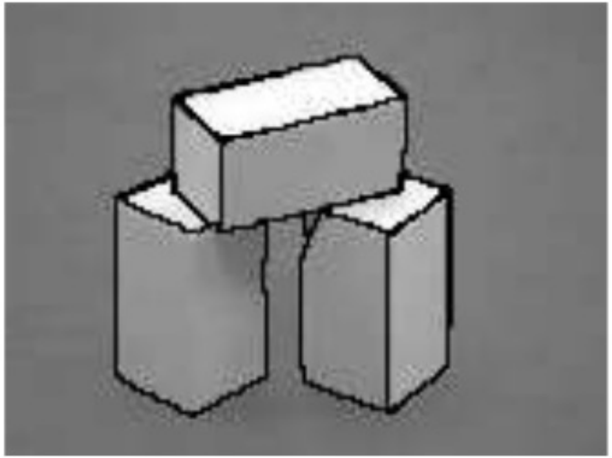


Image Segmentation

Image segmentation is the process of partitioning a digital image into multiple segments (regions)

Image Segmentation

Segmentation: Dividing into the regions/segments of similar properties

- Discontinuity – boundary
- Similarity – region

Discontinuity Detection: Point, Line and Edge

Mask Operation (Review)

FIGURE 10.1 A general 3 × 3 mask.

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

Image Segmentation

Mask Operation

Point

1	1	1
1	−8	1
1	1	1

Line with fixed orientation

−1	−1	−1
2	2	2
−1	−1	−1

Horizontal

2	−1	−1
−1	2	−1
−1	−1	2

+45°

−1	2	−1
−1	2	−1
−1	2	−1

Vertical

−1	−1	2
−1	2	−1
2	−1	−1

−45°

Image Segmentation

Edge: Sharp change in intensity (discontinuity)

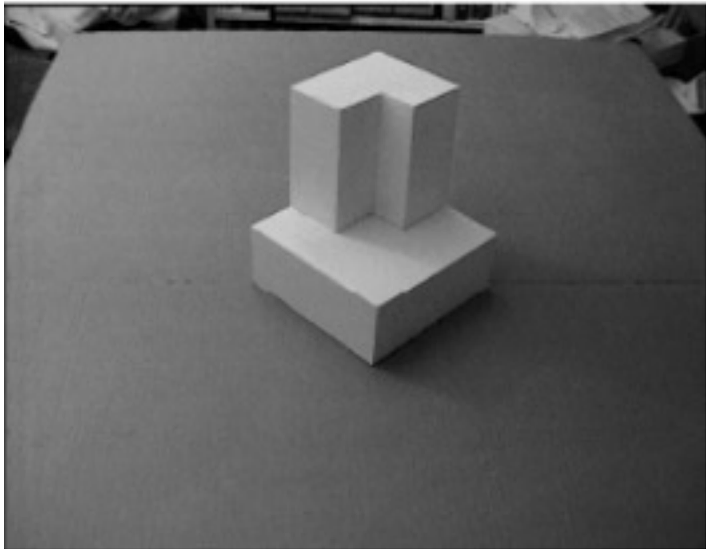
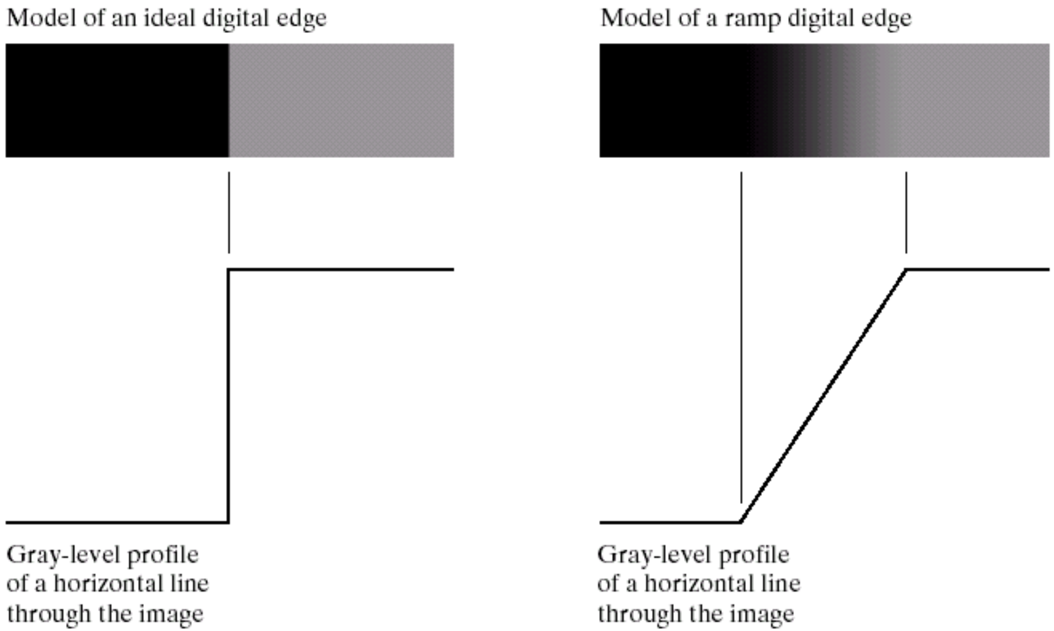


Image Segmentation

Edge: Sharp change in intensity (discontinuity)



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.

Image Segmentation

Edge Detection

a b

FIGURE 10.6
(a) Two regions separated by a vertical edge.
(b) Detail near the edge, showing a gray-level profile, and the first and second derivatives of the profile.

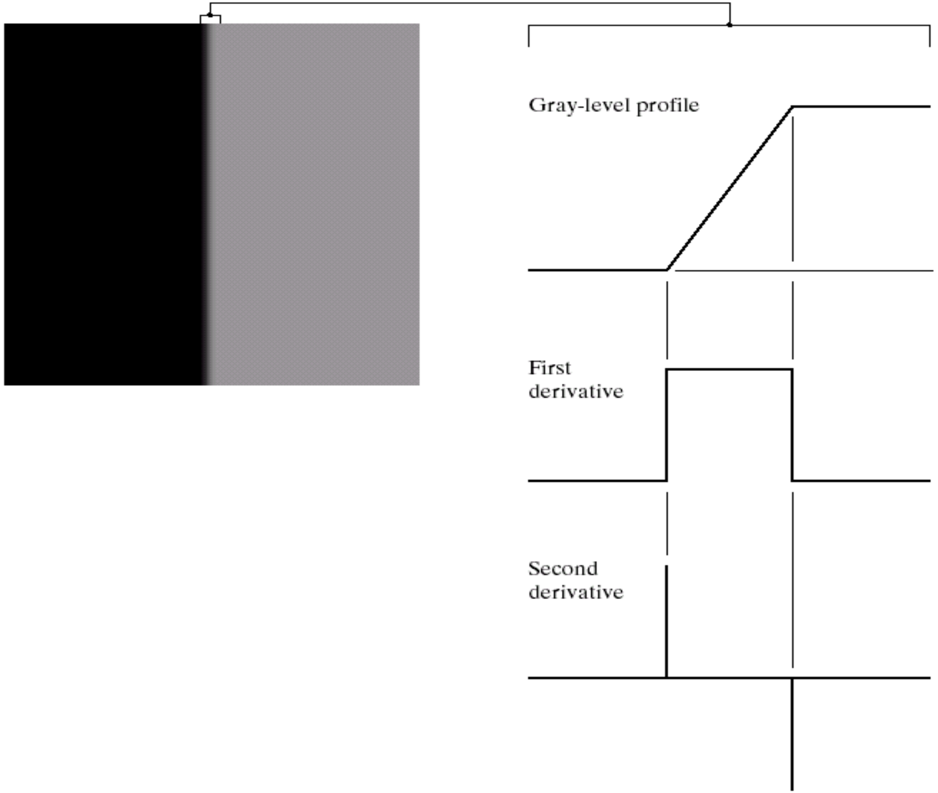


Image Segmentation

Edge Detection

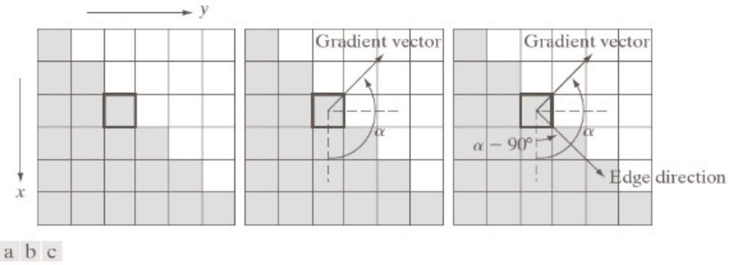


FIGURE 10.12 Using the gradient to determine edge strength and direction at a point. Note that the edge is perpendicular to the direction of the gradient vector at the point where the gradient is computed. Each square in the figure represents one pixel.

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} = \begin{bmatrix} g_x \\ g_y \end{bmatrix}$$

Strength and direction of an edge can be determined using the gradient

Strength (magnitude)

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

Direction

$$\alpha(x,y) = Tan^{-1}\left(\frac{g_x}{g_y}\right)$$

Image Segmentation

Edge Detection

- Partial derivatives of images replaced by finite differences

$$\Delta_x f = f(x,y) - f(x-1,y) \qquad \begin{bmatrix} -1 & 1 \end{bmatrix} \qquad \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

$$\Delta_y f = f(x,y) - f(x,y-1)$$

- Alternatives are:

$$\Delta_{2x} f = f(x+1,y) - f(x-1,y) \qquad \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$$

$$\Delta_{2y} f = f(x,y+1) - f(x,y-1)$$

- Robert's gradient

$$\Delta_+ f = f(x+1,y+1) - f(x,y) \rightarrow \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

$$\Delta_- f = f(x,y+1) - f(x+1,y) \rightarrow \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

Prewitt

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Sobel

Image Segmentation

Edge Detection

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, $|G_x| + |G_y|$.

Image Segmentation

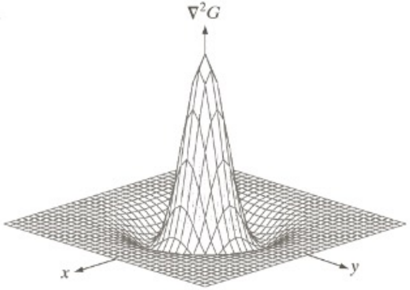
Edge Detection

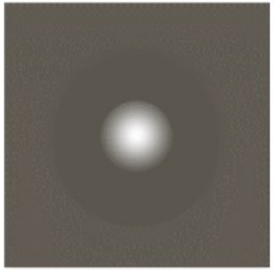
Laplacian of Gaussian (LoG)

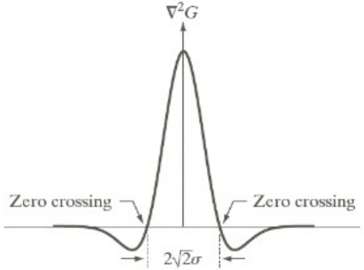
The LoG is sometimes called the Mexican hat operator

The Laplacian is NEVER used directly because of its strong noise sensitivity

Combining the Laplacian with a Gaussian gives the LoG







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 10.21 (a) Three-dimensional plot of the *negative* of the LoG. (b) Negative of the LoG displayed as an image. (c) Cross section of (a) showing zero crossings. (d) 5 × 5 mask approximation to the shape in (a). The negative of this mask would be used in practice.

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

Image Segmentation

Edge Detection

Laplacian of Gaussian (LoG)

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

$$\nabla^2 G(x,y) = [\frac{x^2+y^2-2\sigma^2}{\sigma^4}]e^{-\frac{x^2+y^2}{2\sigma^2}}$$

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

Image Segmentation

Edge Detection

Marr-Hildereth algorithm:

- Filter image with a nxn Gaussian low-pass filter $G(x,y)=e^{-\frac{x^2+y^2}{2\sigma^2}}$
- Compute the Laplacian of the filtered image using an appropriate mask
- Find the zero crossings of this image

This operator is based upon a 2nd derivative operator and can be scaled using the parameter σ to fit a particular image or application, i.e., small operators for sharp detail and large operators for blurry edges

Image Segmentation

Edge Detection

a b

FIGURE 10.23 (a) Negatives of the LoG (solid) and DoG (dotted) profiles using a standard deviation ratio of 1.75:1. (b) Profiles obtained using a ratio of 1.6:1.

The Laplacian of a Gaussian (LoG) can be approximated by a Difference of Gaussians (DoG) provided the ratio σ_1/σ_2 is picked appropriately. The ratio of 1.6:1 seems to work best in practice.

$$LoG(x,y) = -\left[\frac{x^2+y^2-2\sigma^2}{\sigma^4}\right]e^{-\frac{x^2+y^2}{2\sigma^2}} \qquad DoG(x,y) = \frac{1}{2\pi\sigma_1^2}e^{-\frac{x^2+y^2}{2\sigma_1^2}} - \frac{1}{2\pi\sigma_2^2}e^{-\frac{x^2+y^2}{2\sigma_2^2}}$$

Image Segmentation

Canny Edge Detection

- Low error rate
  All true edges should be found
- Edge points should be well localized
  Edges should be located as close as possible to the true edges
- Single edge point response
  One point for each true edge point
  No of local maxima around true should be minimum

Image Segmentation

Canny Edge Detection

1. Smooth image with a Gaussian filter Low error rate

2. Compute gradient magnitude M[x,y] and direction α[x,y]

3. Apply non-maximal suppression to the gradient magnitude

4. Use double thresholding (and subsequent connectivity analysis) to detect link edges

Image Segmentation

Canny Edge Detection

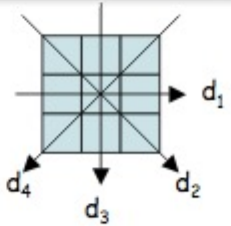
M[i,j] will have large values where the gradient is large. We need to find the local maxima in this array to locate the edges.

Must be thin so only points of the greatest local change remain.

Image Segmentation

Canny Edge Detection

For a 3x3 region quantize α to four directions $\zeta[i,j]=\{d_1, d_2, d_3, d_4\}$



1. Pick the d_i which is closest to $\alpha[x,y]$
2. If $M[x,y]$ is less than one of its two neighbors along $\alpha[x,y]$ then $g_N[x,y]=0$ [suppress a non-maximum] else $g_N[x,y]=M[x,y]$

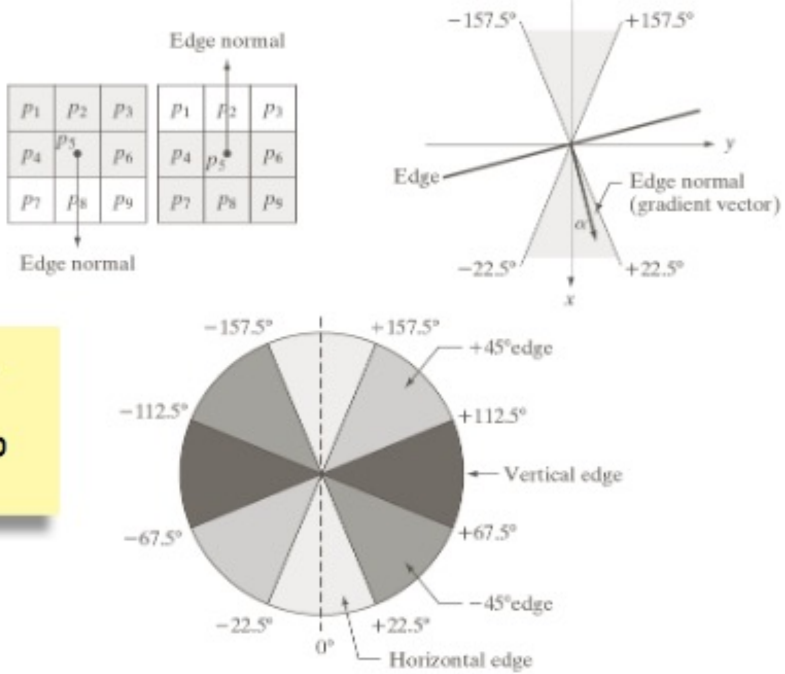
NOTE: Resulting contours may still be be multiple-pixels thick requiring use of a thinning algorithm

Denote the entire process $N[i,j]=Non\_maximal\_suppression\{M[I,j], \zeta[i,j]\}$

Image Segmentation

Canny Edge Detection

Each direction {d_1, d_2,d_3,d_4} actually corresponds to two edge directions



a b
c

FIGURE 10.24 (a) Two possible orientations of a horizontal edge (in gray) in a 3 × 3 neighborhood. (b) Range of values (in gray) of α, the direction angle of the *edge normal*, for a horizontal edge. (c) The angle ranges of the edge normals for the four types of edge directions in a 3 × 3 neighborhood. Each edge direction has two ranges, shown in corresponding shades of gray.

Image Segmentation

Canny Edge Detection

- After non-maximal suppression image contains many false edge fragments caused by noise and fine texture
- You can threshold N[i,j], but good results are difficult to achieve with a single threshold T.
- Use two thresholds T_1 and T_2. Initially link contours using threshold T_1. If a gap is encountered drop to threshold T_2 until you rejoin a T_1 contour.

Image Segmentation

Canny Edge Detection

(a) Original image

(b) Canny, σ=1.0, T_1=255, T_2=1

Image Segmentation

Edge Detection