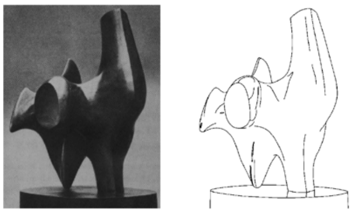
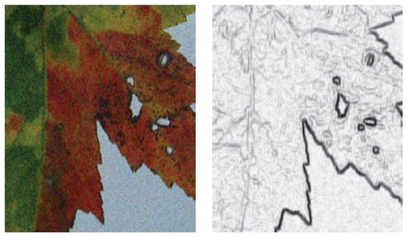
Edge Detection





Yeni Herdiyeni
Departemen Ilmu Komputer FMIPA IPB

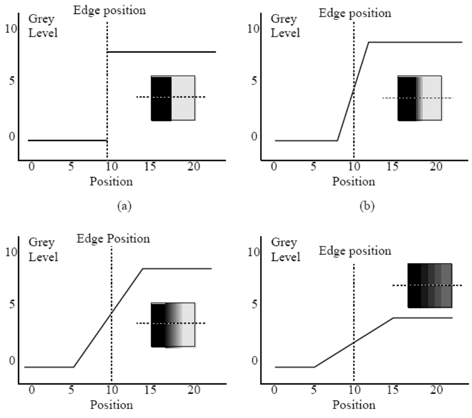
Outline

- Introduction
- Image Segmentation
- Edge detection method
  - Edge-based
  - Region-based

Teknik Segmentasi

- ☐ Discontinuity → Edge-based
- ☐ Similarity (kemiripan) → Region-based : histogram-based thresholding, region growing, region splitting dan merging, clustering/ classification, dan pendekatan teori graf.

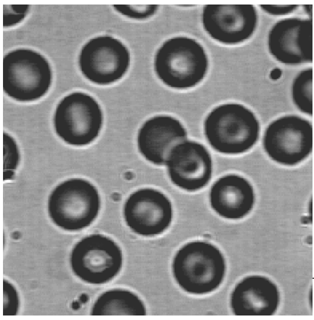
Segmentasi – Edge-based



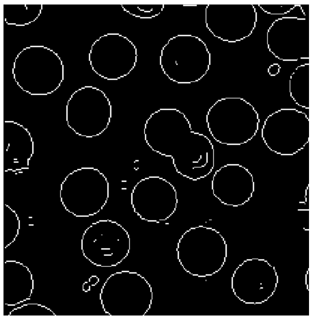
(a) (b) (c) (d)

What are edges?

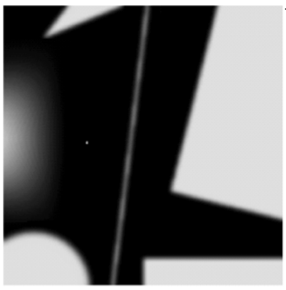
- Local intensity change
- Strong edge = the steep areas in a 3D plot
  (show: blobs-for-edge, surface plot)

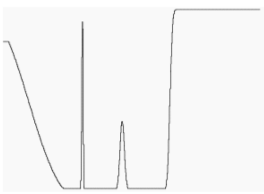


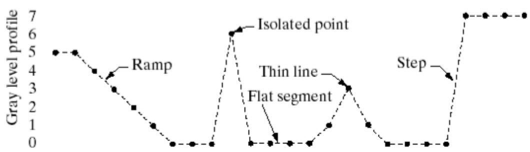
Edge detection



What are edges?







Deteksi Tepi

- Deteksi tepi *(Edge detection)* adalah operasi yang dijalankan untuk mendeteksi garis tepi (*edges*) atau boundary untuk segmentasi, registrasi, dan identifikasi objek.

- Edge adalah beberapa bagian dari citra di mana intensitas kecerahan **berubah** secara drastis.

7

Deteksi Tepi (edge)

- Edge adalah beberapa bagian dari citra di mana intensitas kecerahan **berubah** secara drastis.
- Dalam objek berdimensi 1, perubahan dapat diukur dengan menggunakan fungsi turunan *(derivative function).*
- Perubahan mencapai maksimum pada saat nilai **turunannya pertamanya mencapai nilai maksimum** atau **nilai turunan kedua (*2nd derivative*) bernilai 0.**

8

Gradient Operators

• Motivasi: mendeteksi perubahan changes

Perubahan nilai piksel yang besar $\longrightarrow$ Gradient besar

image $x(m,n)$ $\longrightarrow$ [Gradient operator] $\xrightarrow{g(m,n)}$ [Thresholding] $\longrightarrow$ edge map $I(m,n)$

$$I(m,n) = \begin{cases} 1 & |g(m,n)| > th \\ 0 & otherwise \end{cases}$$

MATLAB function: > help edge

Introduction

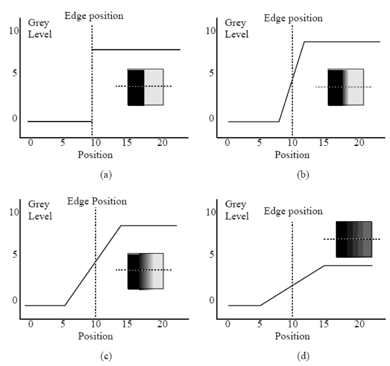


Fig.1. Edges are boundaries between different textures.Edge also can be defined as discontinuities inimage intensity from one pixel to another.. [4]

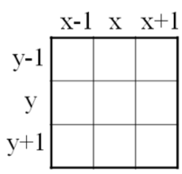
EDGE DETECTION METHOD

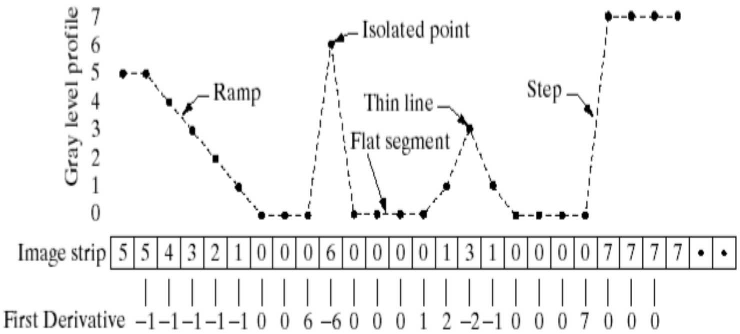
- First-Order Derivative Edge Detection
  - The Roberts operators
  - The Prewitt operators
  - The Sobel operators
  - First-Order of Gausssian (FDOG)
- Second-Order Derivative Edge Detection
  - Laplacian
  - Canny

Edge Detectors

- Mudah diimplementasikan dan cepat
- Berdasarkan **grey-level gradient**
  - Dihitung pada setiap piksel => Gradient image: g(x,y)
  - Menggunakan 16-bit atau 32-bit image untuk merepresentasikan gradient!
- Gradient : the first-order derivative: $f'(x,y) = g(x,y)$
- Bekerja pada arah x dan y:

$$g_x(x,y) \approx f(x+1,y) - f(x-1,y)$$
$$g_y(x,y) \approx f(x,y+1) - f(x,y-1)$$





Correlation with this kernel:

-1	+1

Normally this kernel to avoid bias:

-1	0	+1

Fist-Order Derivative Edge Detection

Definition:

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial x} \end{bmatrix}$$ the gradient vector

$$\nabla f = |\nabla f| = \sqrt{G_x^2 + G_y^2}$$ the magnitude of this vector

$$\textit{angle} \text{ of } \nabla f = \tan^{-1}\left(\frac{G_y}{G_x}\right)$$ the direction of the gradient vector

Fist-Order Derivative Edge Detection

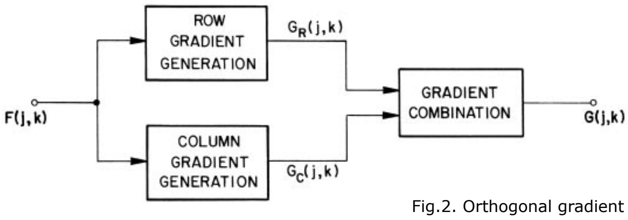


Fig.2. Orthogonal gradient generation.[2]

$$G(x,y) = \frac{\partial F(x,y)}{\partial x}\cos\theta + \frac{\partial F(x,y)}{\partial y}\sin\theta$$
The gradient along the line normal to the edge slope

$$G(j,k) = [[G_R(j,k)]^2 + [G_C(j,k)]^2]^{1/2}$$
The spatial gradient amplitude

$$G(j,k) = |G_R(j,k)| + |G_C(j,k)|$$
The gradient amplitude combination

The Roberts operators

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

0	0	0
0	−1	0
0	0	1

$$G_x = (z_9 - z_5)$$

0	0	0
0	0	−1
0	1	0

$$G_y = (z_8 - z_6)$$

The Prewitt operators

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

$G_x = (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)$ $G_y = (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)$

17

The Sobel operators

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

–1	–2	–1
0	0	0
1	2	1

–1	0	1
–2	0	2
–1	0	1

$G_x = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$ $G_y = (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)$

18

Operators

Operator	Row gradient	Column gradient
Pixel difference	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
Separated pixel difference	$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$
Roberts	$\begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
Prewitt	$\frac{1}{3}\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$	$\frac{1}{3}\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$
Sobel	$\frac{1}{4}\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$	$\frac{1}{4}\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$
Frei–Chen	$\frac{1}{2+\sqrt{2}}\begin{bmatrix} 1 & 0 & -1 \\ \sqrt{2} & 0 & -\sqrt{2} \\ 1 & 0 & -1 \end{bmatrix}$	$\frac{1}{2+\sqrt{2}}\begin{bmatrix} -1 & -\sqrt{2} & -1 \\ 0 & 0 & 0 \\ 1 & \sqrt{2} & 1 \end{bmatrix}$

Fig.3. Impulse response arrays for 3 ×3 orthogonal differential gradient edge operators.[2]

First-Order of Gausssian

$G(x) = e^{-(\frac{x^2}{2\sigma^2})}$ It is hard to find the gradient by using the equation

$G'(x) = (-\frac{x}{\sigma^2})e^{-(\frac{x^2}{2\sigma^2})}$ In order to simplify the computation

$$M_x(x,y) = G_x^{`} * I(x,y)$$
$$M_y(x,y) = G_y^{`} * I(x,y)$$

$M = |M_x| + |M_y|$

First-order derivative edge detection

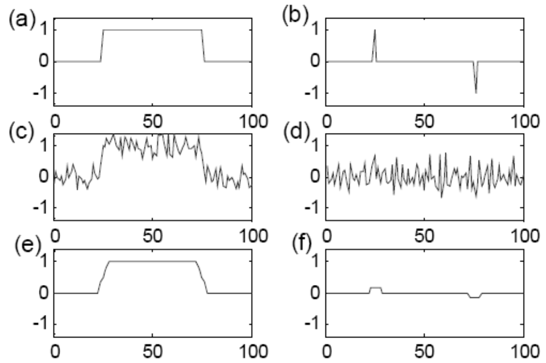


Fig.4. Using 1st-order differentiation to detect (a) the sharp edges, (c) the step edges with noise, and (e) the ramp edges. (b)(d)(e) are the results of differentiation of (a)(c)(e).[3]

Tugas Minggu Depan

- Buat program deteksi Tepi dengan menggunakan operator
  - **Robert**
  - **Prewitt**
  - **Sobel**
- **Program tidak boleh** menggunakan fungsi MATLAB

Second Order Derivation

Laplacian

23

Second-order derivative edge detection

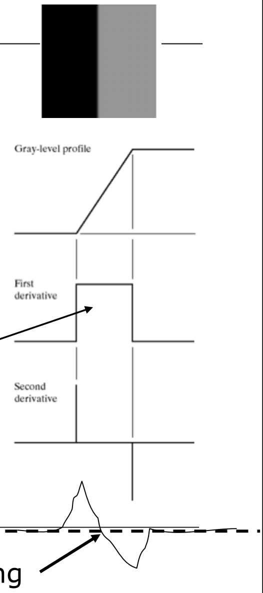
The Laplacian of a 2-D function *f* (*x*, *y*) is a second-order derivative defined as:

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

24

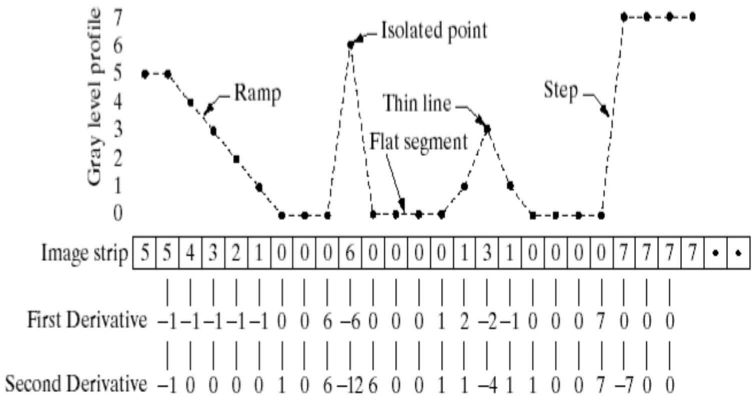
Edge Detector: Laplacian

- Where exactly is the edge (width)?
- Second-order derivative: $f''(x,y)$
  - *The variation of the variation of the gray-level value!*
  - How fast does the gradient change?
  - Find the zero-crossing
    - Center of edge



$f''(x,y) \approx$

0	-1	0
-1	4	-1
0	-1	0



Second-order derivative edge detection

The 2-D Gaussian function:

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

The Laplacian of Gaussian (LOG):

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

27

Second-order derivative edge detection

0	0	0	-1	-1	-2	-1	-1	0	0	0
0	0	-2	-4	-8	-9	-8	-4	-2	0	0
0	-2	-7	-15	-22	-23	-22	-15	-7	-2	0
-1	-4	-15	-24	-14	-1	-14	-24	-15	-4	-1
-1	-8	-22	-14	52	103	52	-14	-22	-8	-1
-2	-9	-23	-1	103	178	103	-1	-23	-9	-2
-1	-8	-22	-14	52	103	52	-14	-22	-8	-1
-1	-4	-15	-24	-14	-1	-14	-24	-15	-4	-1
0	-2	-7	-15	-22	-23	-22	-15	-7	-2	0
0	0	-2	-4	-8	-9	-8	-4	-2	0	0
0	0	0	-1	-1	-2	-1	-1	0	0	0

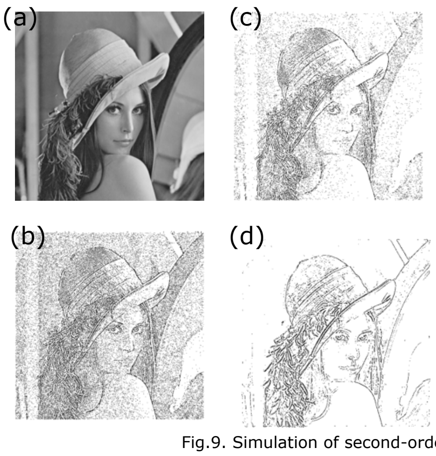
Fig.7. An 11×11 mask approximation to Laplacian of Gaussian (LOG).[3]

28

Second-order derivative edge detection

- ☐ Sensitive to noise
- ☐ Gaussian function
- ☐ Low-pass filter

Second-order derivative edge detection



(a) Original image.

(b) Laplacian mask of **Fig**.(a)

(c) Laplacian mask of **Fig**.(b)

(d) LOG mask of **Fig**.(b)

Fig.9. Simulation of second-order of derivative edge detection[3]

Second-order derivative edge detection

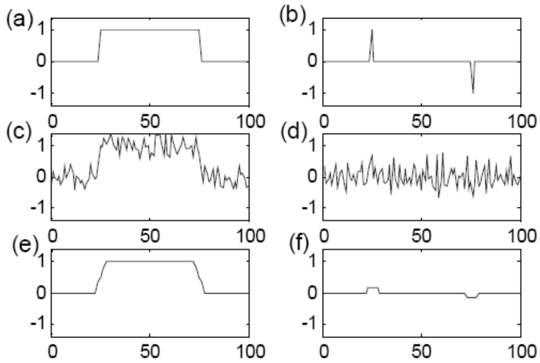


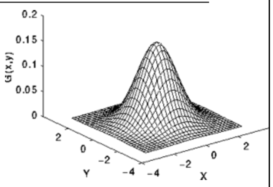
Fig.10. Using 1st-order differentiation to detect (a) the sharp edges, (c) the step edges with noise, and (e) the ramp edges. (b)(d)(e) are the results of differentiation of (a)(c)(e).[3]

Second-order derivative edge detection

The Drawbacks of the Differentiation Method for Edge Detection :

- Sensitivity to noise
- Not good for ramp edges
- Make no difference between the significant edge and the detailed edge

Edge Detector: Laplacian

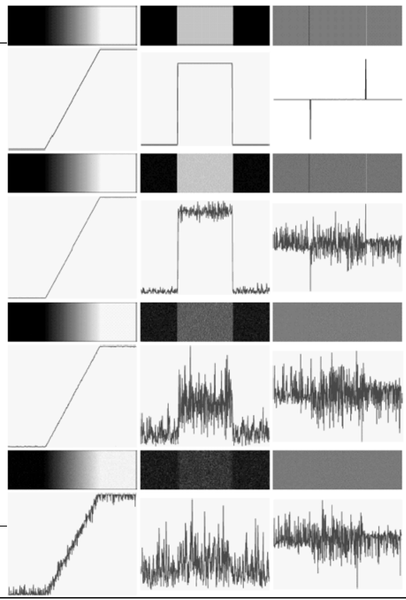


- Noise reduction
  - 2D Gaussian used for smoothing
- Edge enhancement
  - Second-order derivative (second-order gradient)
- Edge localisation
  - Zero-crossings

33

Edge Detector: Laplacian

- The second-order derivative is very sensitive to noise!



34

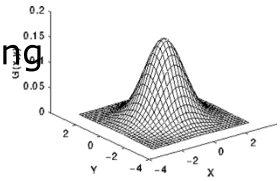
Second Order Derivation

Canny

35

Canny Edge Detector

- Noise reduction
  - 2D Gaussian used for smoothing
- Edge enhancement
  - Magnitude of gradient vector:



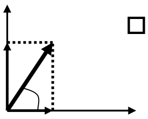
$$g = \sqrt{g_x^2 + g_y^2} \approx |g_x| + |g_y|$$

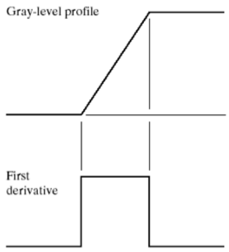
- Edge localisation
  - Complicated…

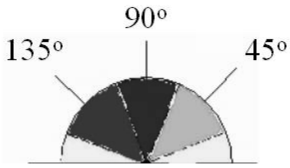
36

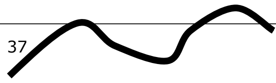
Canny Edge Detector

- Edge localisation
  - Wide edges:
    - Edges give rise to ridges in the gradient image
    - Thin edges using the principle of **non-maximal suppression**
      - Find gradients' directions
      - Each pixel is compared with its two neighbors in the gradient direction. The two smallest are suppressed => set to zero
      - The result is pixel-thin edges









Canny Edge Detection

- Good detection
- Good localization
- Single response
  - Many edge candidate
  - The accurate edge

Simulation

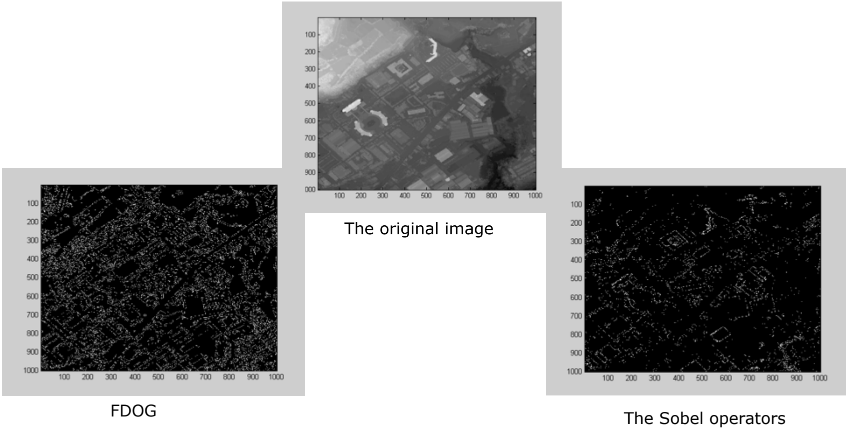
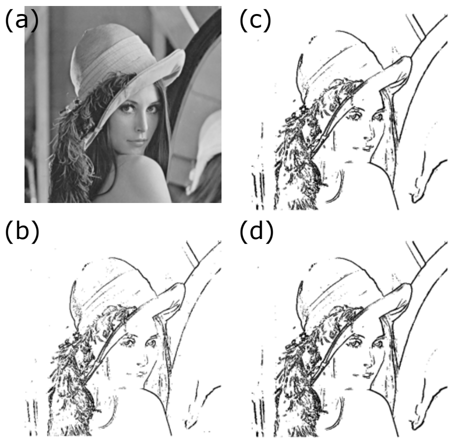


Fig.5. Comparison with FDOG and the Sobel operators

39

Simulation



(a) Original image.

(b) Roberts operator.

(c) Prewitt operator.

(d) Sobel operator.

Fig.6. Simulation of first-order of derivative edge detection[3]

40

Second-order derivative edge detection

0	−1	0
−1	4	−1
0	−1	0

(a)

−1	−1	−1
−1	8	−1
−1	−1	−1

(b)

$$\nabla^2 f = 4z_5 - (z_2 + z_4 + z_6 + z_8)$$

$$\nabla^2 f = 8z_5 - (z_1 + z_2 + z_3 + z_4 + z_6 + z_7 + z_8 + z_9)$$

41

Conlusion

- Fist-Order Derivative Edge Detection:
  - the simplest method
- Second-Order Derivative Edge Detection:
  - sensitive to noise

42

Reference

[1] Soo-Chang Pei, Jian-Jiun Ding, ” Imporved Harris' Algotithm ForConer And Edge Detections”, vol 1, 2005.

[2] William K. Pratt , “Digital Image Processing_William K. Pratt 3rd ”,chapter 15.

[3] Jiun-De Huang, “Image Compression by Segmentation and Boundary Description”, chapter 2.

[4] J. Canny, “A Computational Approach to Edge Detection,” IEEE Trans. Pattern Analysis and Machine Intelligence, PAMI-8, 6,November 1986, 679–698.

END