Digital Image & Video Processing

Lecture 6

Image Pre-processing

(Image Local Pre-processing: Edge Detection)

Lecturer: Associate Prof. Lý Quốc Ngọc

3. Image Pre-processing

3.1. Color Transformations

3.2. Geometric Transformations

3.3. Local Pre-processing

3.3. Local Pre-processing

3.3.1. Image Smoothing

3.3.2. Edge Detection

3.3.2. Edge Detection

3.3.2.1. Gradient operator

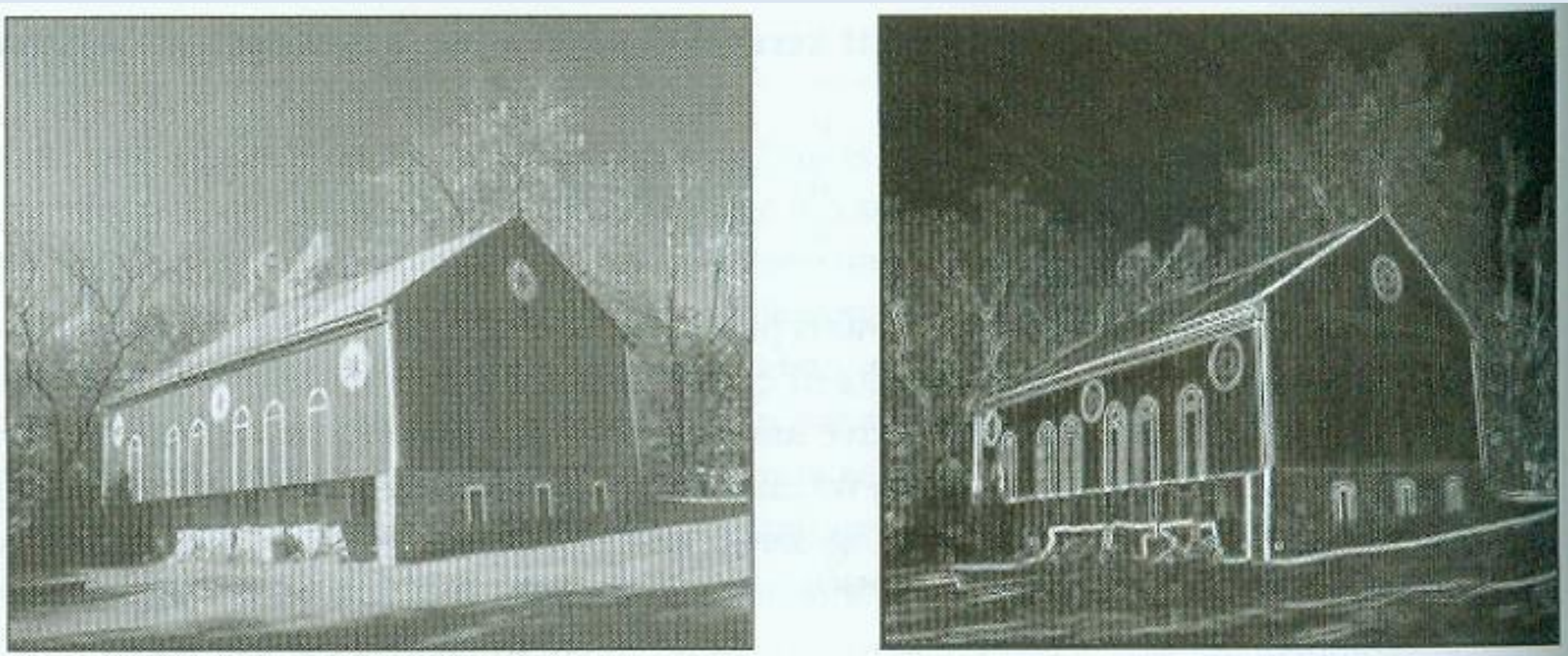
3.3.1.2. Laplace operator

3.3.1.3. Laplace of Gaussian

3.3.1.4. Canny method

3.3.2. Edge Detection

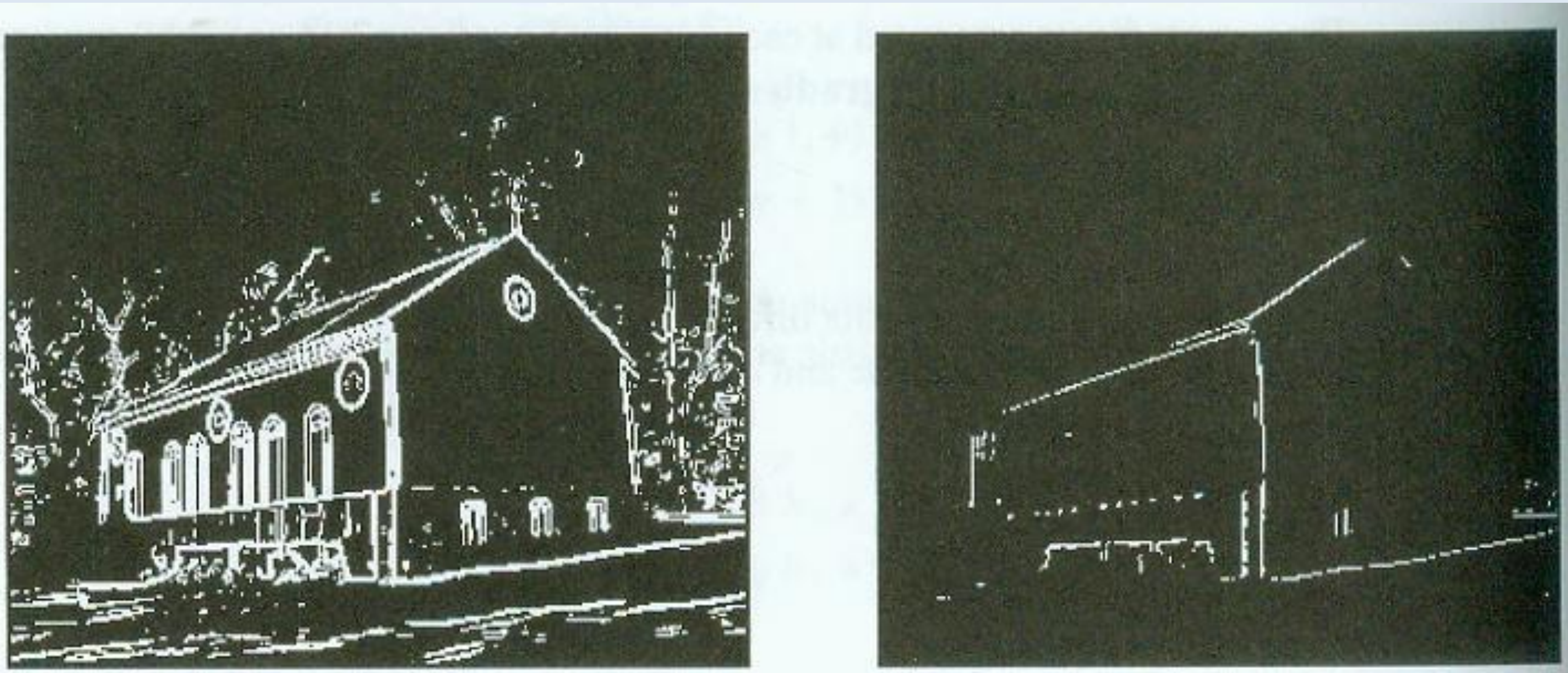
3.3.2.1. Gradient operator



(a) (b)

3.3.2. Edge Detection

3.3.2.1. Gradient operator

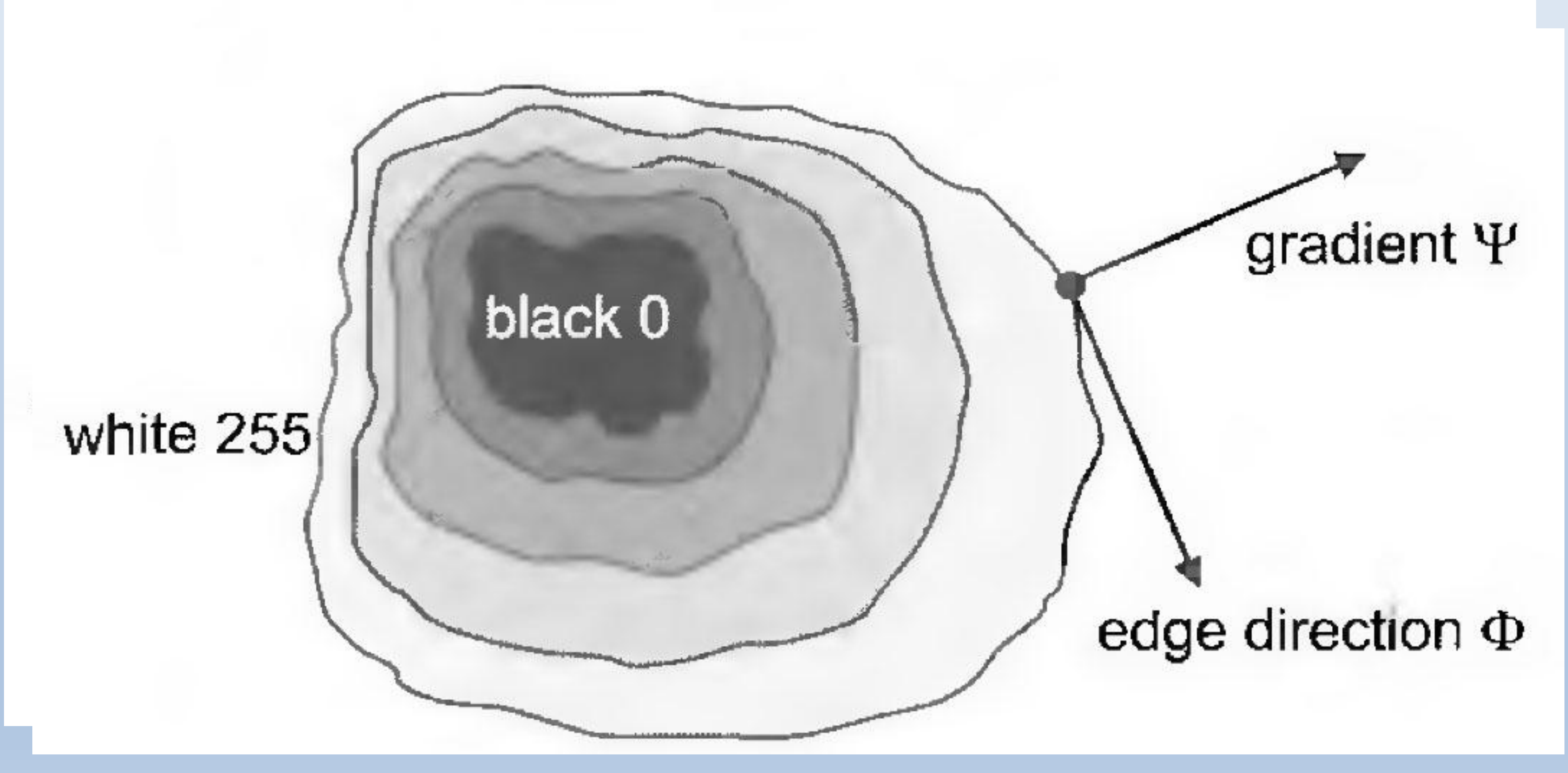


(a)

(b)

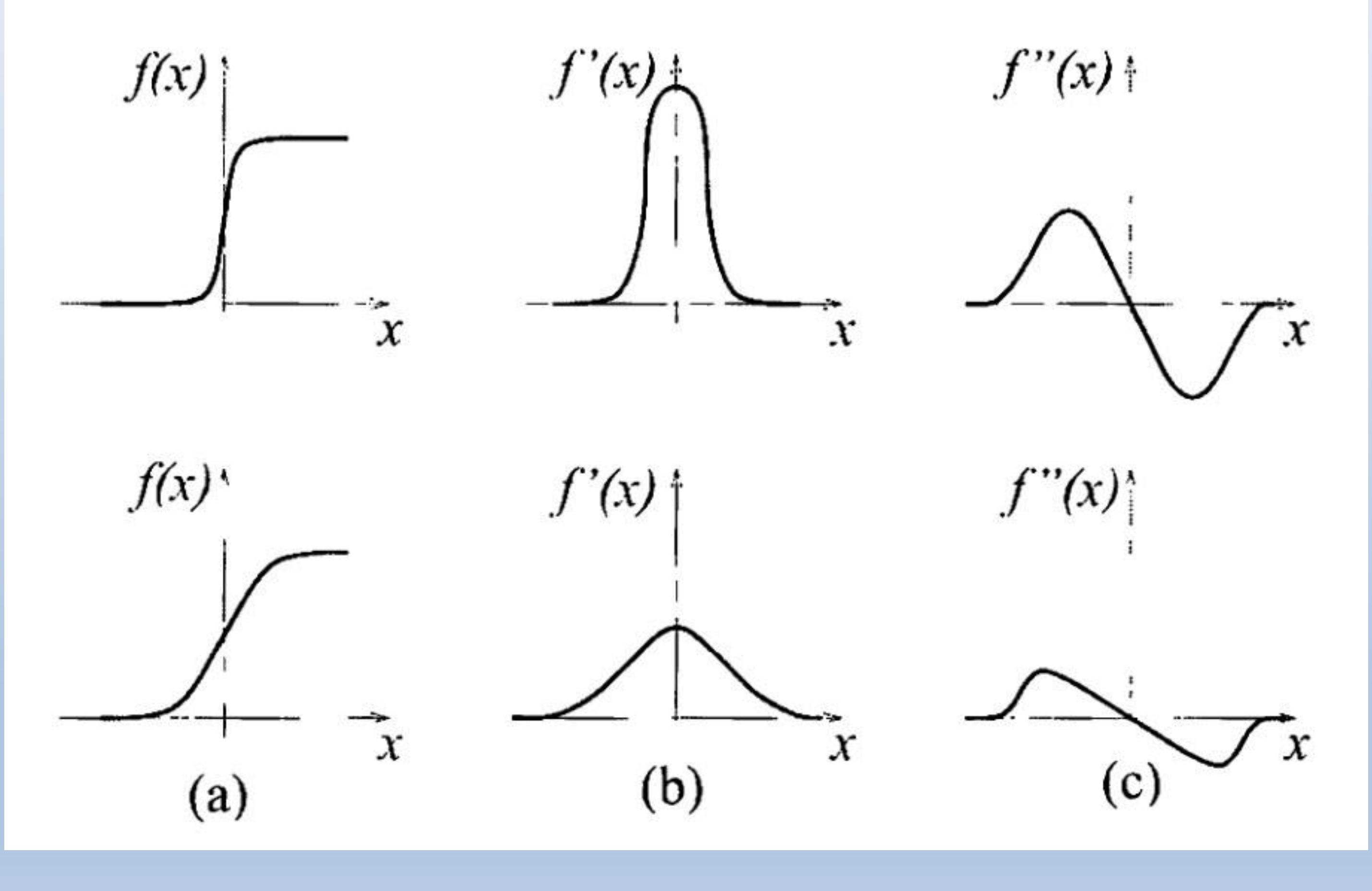
3.3.2. Edge Detection

3.3.2.1. Gradient operator



3.3.2. Edge Detection

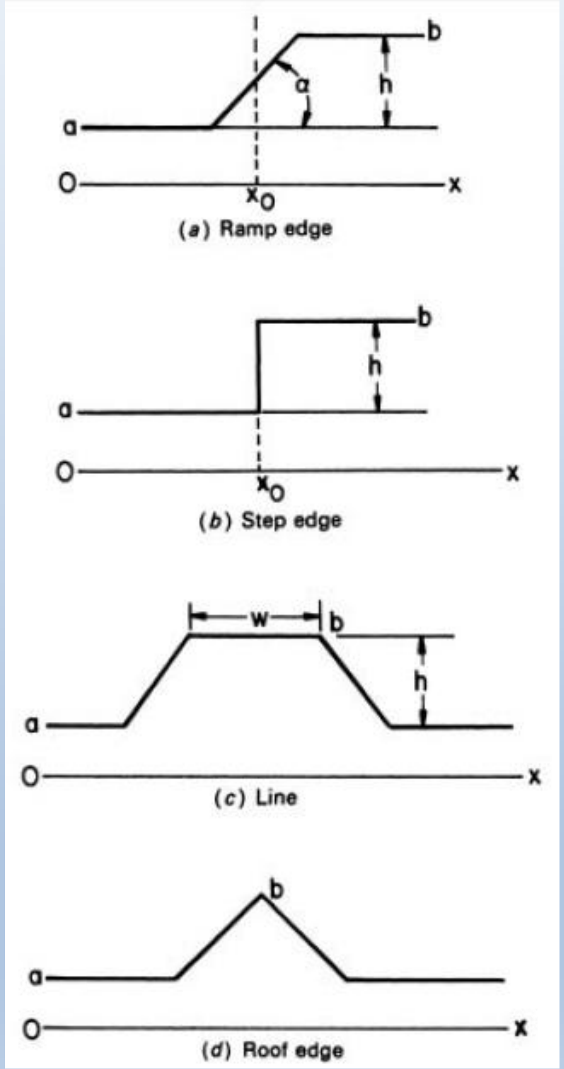
3.3.2.1. Gradient operator



Associate Prof. Lý Quốc Ngọc

3.3.2. Edge Detection

3.3.2.1. Gradient operator



3.3.2. Edge Detection

3.3.2.1. Gradient operator

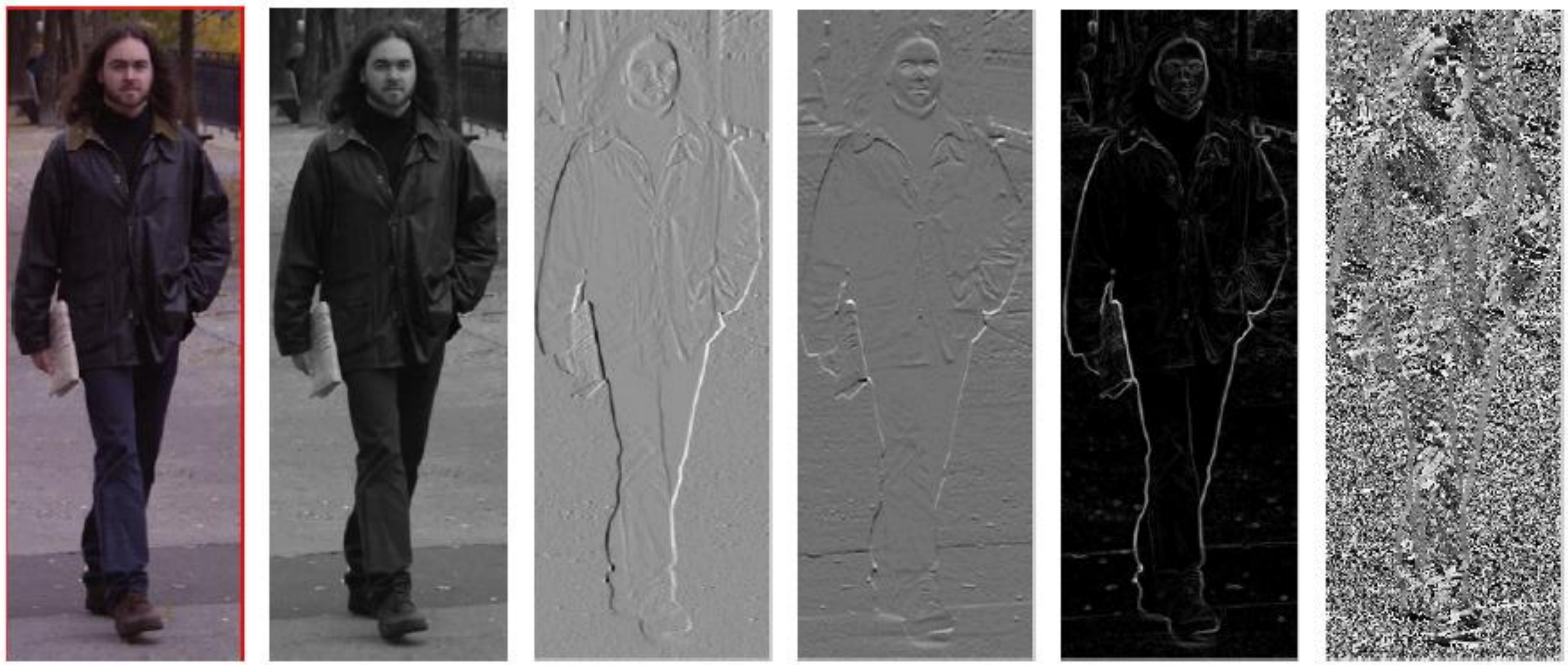
$$(\nabla f)(x, y) = [\partial f / \partial x \quad \partial f / \partial y]^T = [\, f_x \quad f_y \,]^T$$

$$e(x, y) = (\, f_x{}^2 (x, y) + f_y{}^2 (x, y))^{1/2}$$

$$\phi(x, y) = \arctan(\, f_x \quad f_y \,)$$

3.3.2. Edge Detection

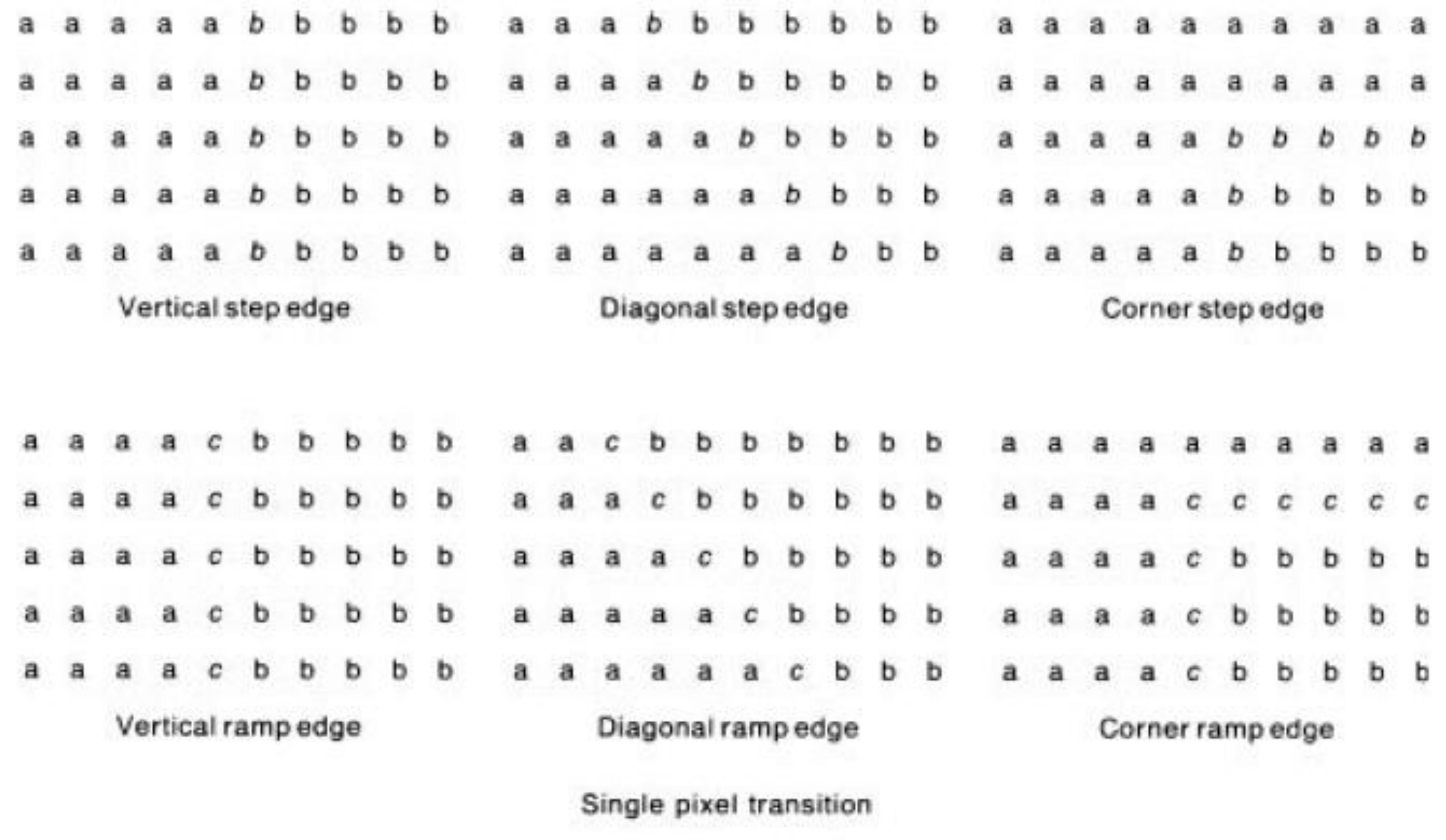
3.3.2.1. Gradient operator



(a) Initial Image (b) I (c) I_x (d) I_y (e) $\sqrt{I_x^2 + I_y^2}$ (f) $\arctan\left(I_y/I_x\right)$

3.3.2. Edge Detection

3.3.2.1. Gradient operator



a	a	a	a	a	b	b	b	b	b
a	a	a	a	a	b	b	b	b	b
a	a	a	a	a	b	b	b	b	b
a	a	a	a	a	b	b	b	b	b
a	a	a	a	a	b	b	b	b	b

Vertical step edge

a	a	a	b	b	b	b	b	b	b
a	a	a	a	b	b	b	b	b	b
a	a	a	a	a	b	b	b	b	b
a	a	a	a	a	a	b	b	b	b
a	a	a	a	a	a	a	b	b	b

Diagonal step edge

a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a	a	a
a	a	a	a	a	b	b	b	b	b
a	a	a	a	a	b	b	b	b	b
a	a	a	a	a	b	b	b	b	b

Corner step edge

a	a	a	a	c	b	b	b	b	b
a	a	a	a	c	b	b	b	b	b
a	a	a	a	c	b	b	b	b	b
a	a	a	a	c	b	b	b	b	b
a	a	a	a	c	b	b	b	b	b

Vertical ramp edge

a	a	c	b	b	b	b	b	b	b
a	a	a	c	b	b	b	b	b	b
a	a	a	a	c	b	b	b	b	b
a	a	a	a	a	c	b	b	b	b
a	a	a	a	a	a	c	b	b	b

Diagonal ramp edge

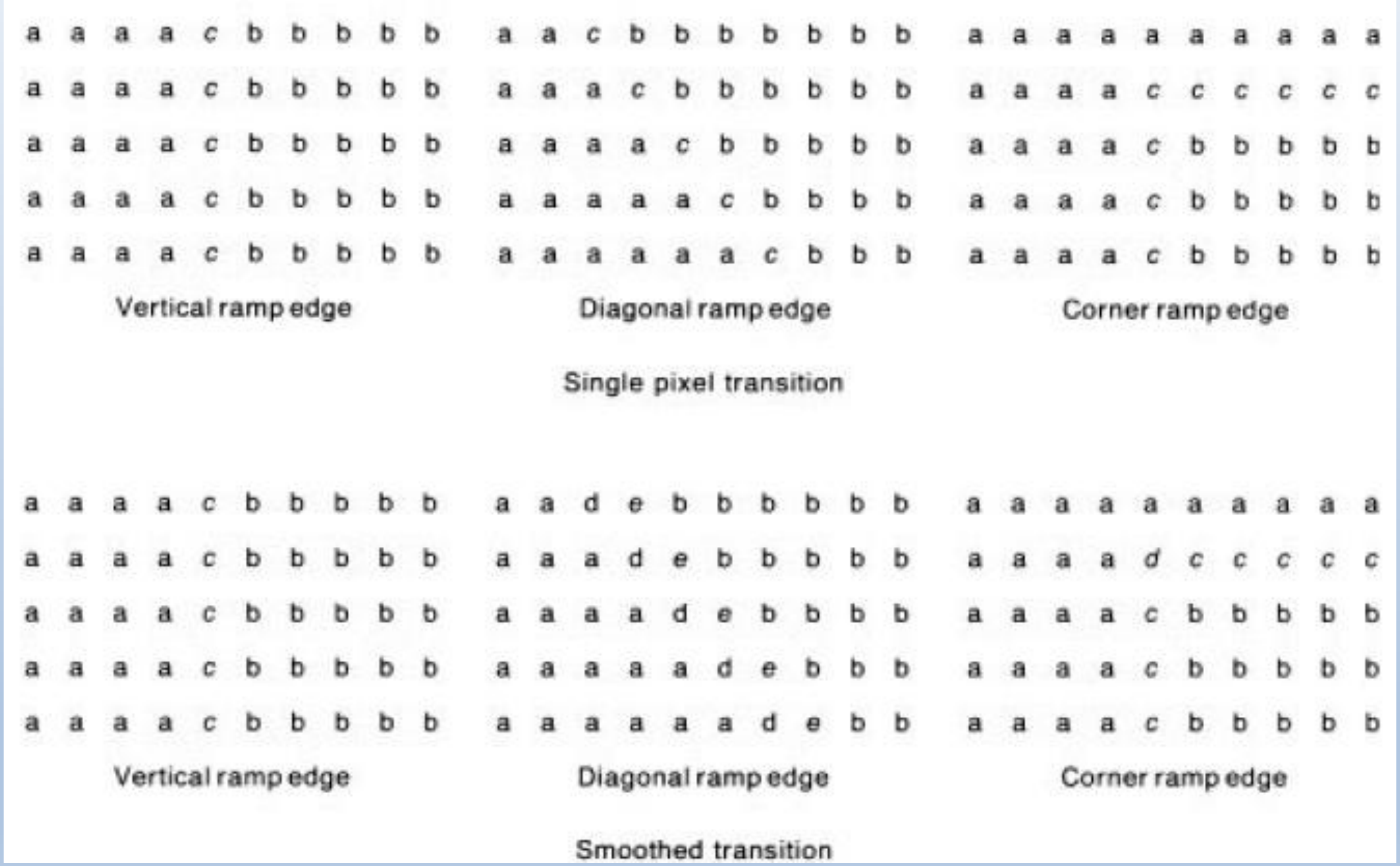
a	a	a	a	a	a	a	a	a	a
a	a	a	a	c	c	c	c	c	c
a	a	a	a	c	b	b	b	b	b
a	a	a	a	c	b	b	b	b	b
a	a	a	a	c	b	b	b	b	b

Corner ramp edge

Single pixel transition

3.3.2. Edge Detection

3.3.2.1. Gradient operator



a	a	a	a	c	b	b	b	b	b
a	a	a	a	c	b	b	b	b	b
a	a	a	a	c	b	b	b	b	b
a	a	a	a	c	b	b	b	b	b
a	a	a	a	c	b	b	b	b	b

Vertical ramp edge

a	a	c	b	b	b	b	b	b	b
a	a	a	c	b	b	b	b	b	b
a	a	a	a	c	b	b	b	b	b
a	a	a	a	a	c	b	b	b	b
a	a	a	a	a	a	c	b	b	b

Diagonal ramp edge

a	a	a	a	a	a	a	a	a	a
a	a	a	a	c	c	c	c	c	c
a	a	a	a	c	b	b	b	b	b
a	a	a	a	c	b	b	b	b	b
a	a	a	a	c	b	b	b	b	b

Corner ramp edge

Single pixel transition

a	a	a	a	c	b	b	b	b	b
a	a	a	a	c	b	b	b	b	b
a	a	a	a	c	b	b	b	b	b
a	a	a	a	c	b	b	b	b	b
a	a	a	a	c	b	b	b	b	b

Vertical ramp edge

a	a	d	e	b	b	b	b	b	b
a	a	a	d	e	b	b	b	b	b
a	a	a	a	d	e	b	b	b	b
a	a	a	a	a	d	e	b	b	b
a	a	a	a	a	a	d	e	b	b

Diagonal ramp edge

a	a	a	a	a	a	a	a	a	a
a	a	a	a	d	c	c	c	c	c
a	a	a	a	c	b	b	b	b	b
a	a	a	a	c	b	b	b	b	b
a	a	a	a	c	b	b	b	b	b

Corner ramp edge

Smoothed transition

3.3.2. Edge Detection

3.3.2.1. Gradient operator

$c=(a+b)/2$,

$d= (3a+b)/4$

$e=(a+3b)/4$

$(a<d<c<e<b)$

3.3.2. Edge Detection

3.3.2.1. Gradient operator

Differencing

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

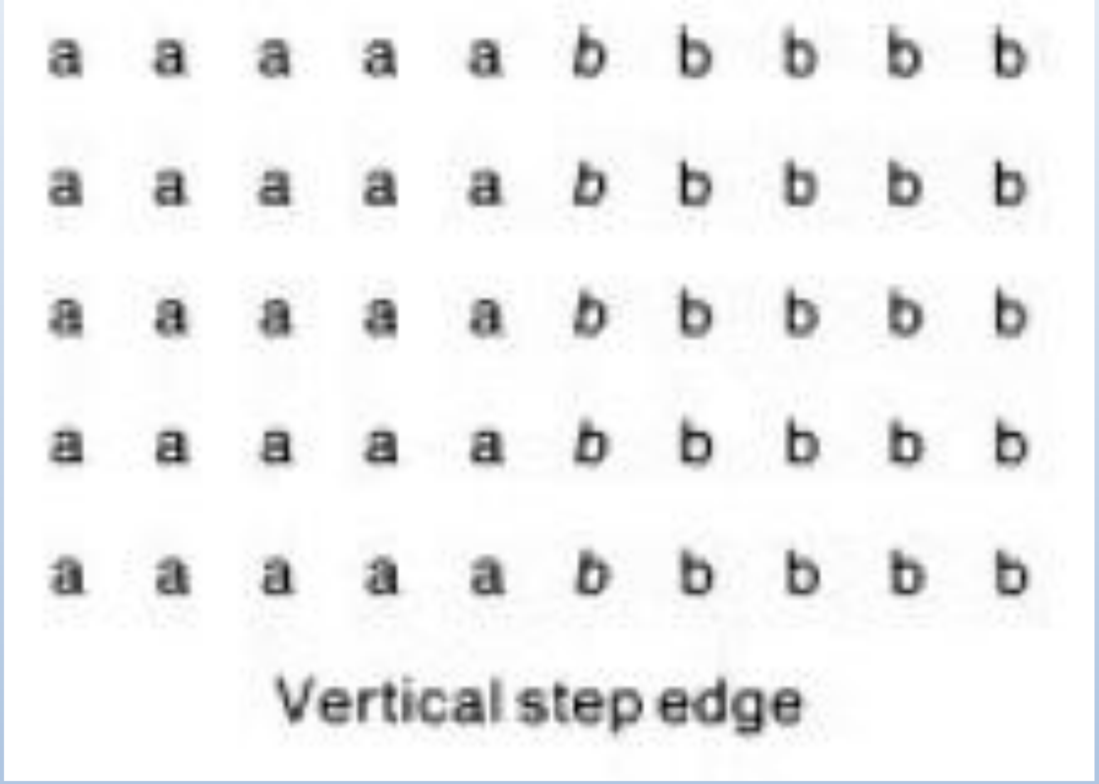
$$f_y(x, y) \approx f(x, y) - f(x, y-1)$$

3.3.2. Edge Detection

3.3.2.1. Gradient operator

Differencing

h=b-a



a	a	a	a	a	b	b	b	b	b
a	a	a	a	a	b	b	b	b	b
a	a	a	a	a	b	b	b	b	b
a	a	a	a	a	b	b	b	b	b
a	a	a	a	a	b	b	b	b	b

Vertical step edge

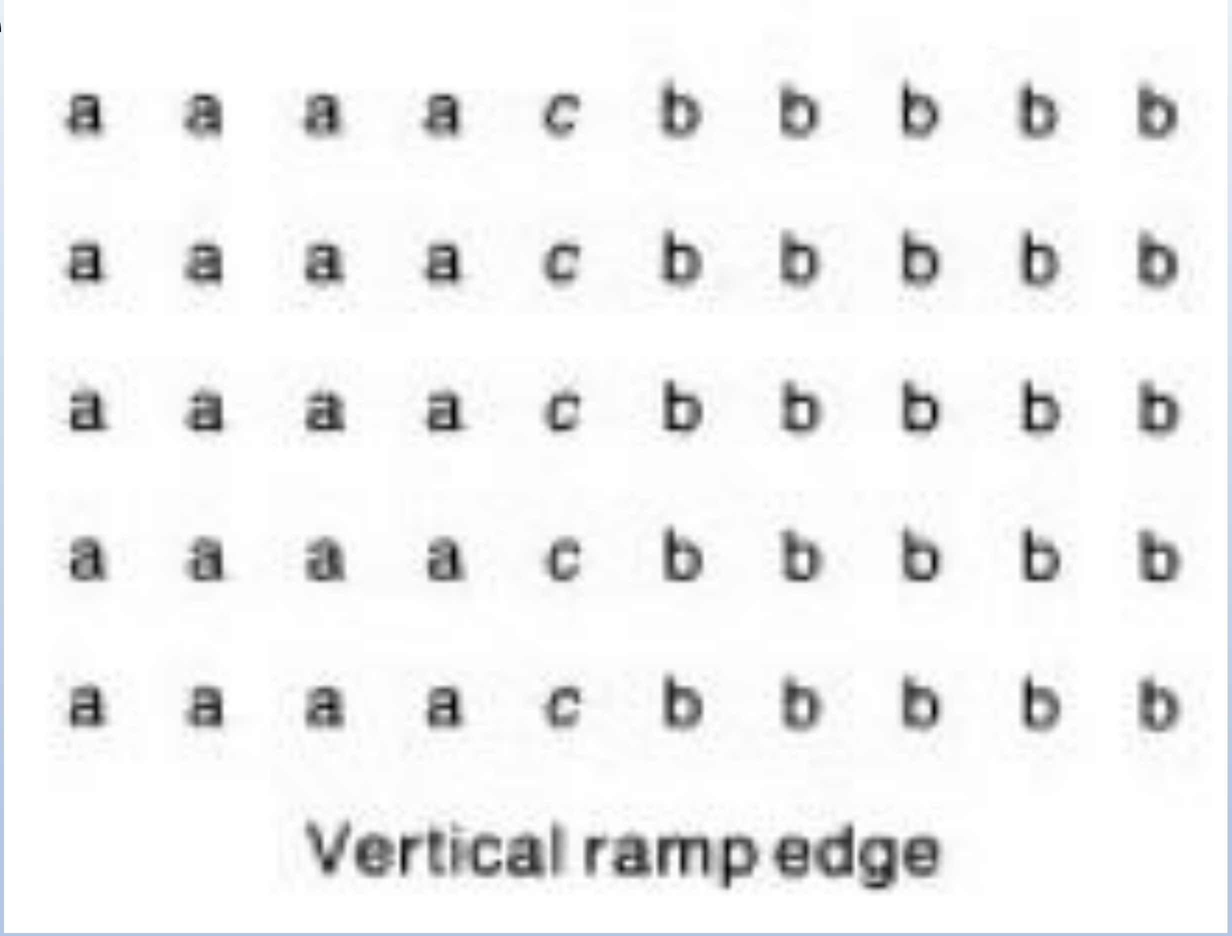
0 0 0 0 h 0 0 0 0

3.3.2. Edge Detection

3.3.2.1. Gradie

Differencing

h=b-a



a	a	a	a	c	b	b	b	b	b
a	a	a	a	c	b	b	b	b	b
a	a	a	a	c	b	b	b	b	b
a	a	a	a	c	b	b	b	b	b
a	a	a	a	c	b	b	b	b	b

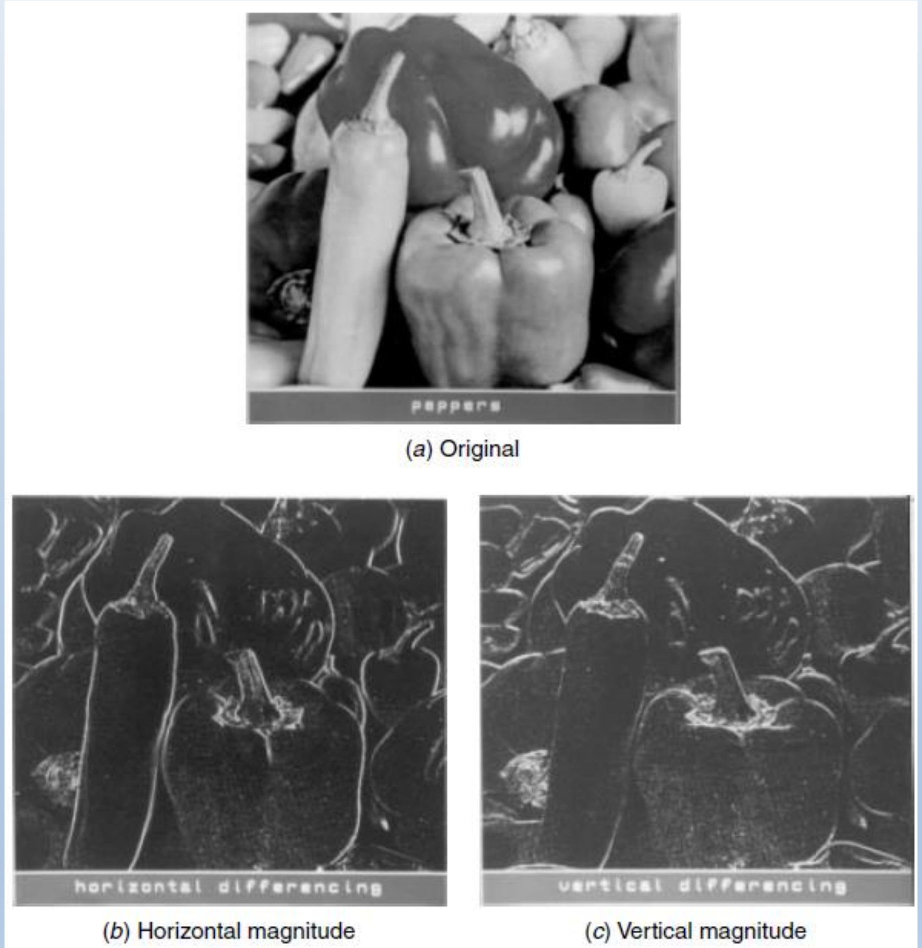
Vertical ramp edge

0 0 0 0 h/2 h/2 0 0 0

3.3.2. Edge Detection

3.3.2.1. Gradient operator

Differencing



(*a*) Original

(*b*) Horizontal magnitude

(*c*) Vertical magnitude

3.3.2. Edge Detection

3.3.2.1. Gradient operator

Robert operator

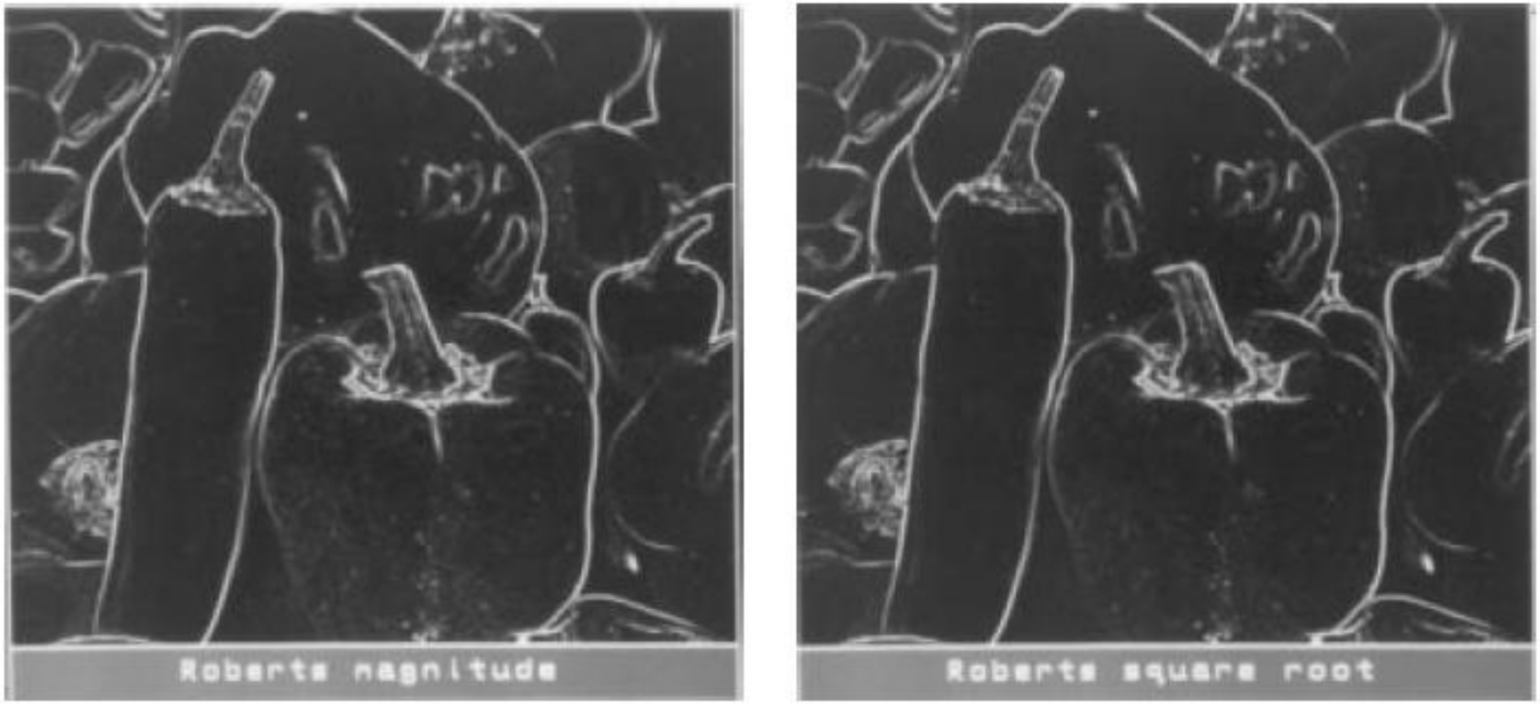
$$f_x(x,y) \approx f(x,y) - f(x+1,y+1)$$

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

3.3.2. Edge Detection

3.3.2.1. Gradient operator

Robert operator



(*a*) Magnitude

(*b*) Square root

3.3.2. Edge Detection

3.3.2.1. Gradient operator

Differencing (localize edge center of ramp edge)

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

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

3.3.2. Edge Detection

3.3.2.1. Gradient operator

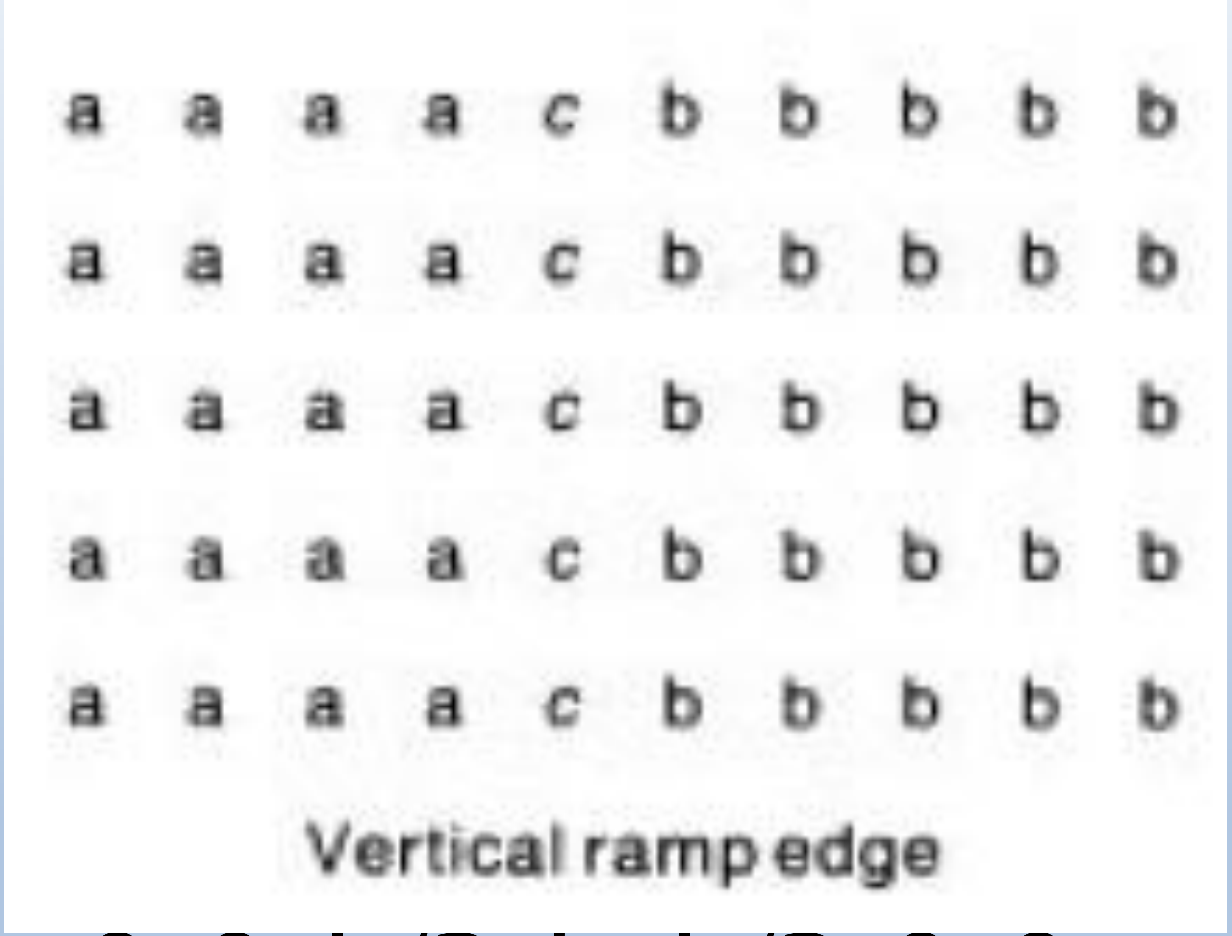
Differencing

(localize edge

center of

ramp edge)

h=b-a



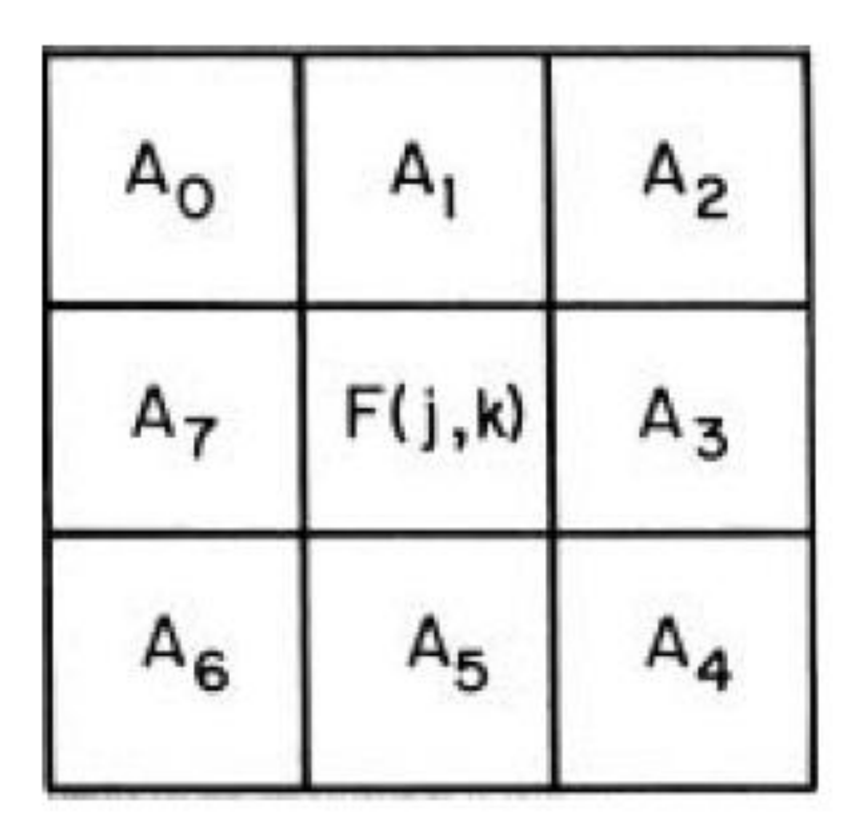
a	a	a	a	c	b	b	b	b	b
a	a	a	a	c	b	b	b	b	b
a	a	a	a	c	b	b	b	b	b
a	a	a	a	c	b	b	b	b	b
a	a	a	a	c	b	b	b	b	b

Vertical ramp edge

0 0 h/2 h h/2 0 0

3.3.2. Edge Detection

3.3.2.1. Gradient operator



A_0	A_1	A_2
A_7	$F(j,k)$	A_3
A_6	A_5	A_4

3.3.2. Edge Detection

3.3.2.1. Gradient operator

Prewitt operator (k=1)

$$f_x \approx \frac{1}{k+2}[(A_2 + kA_3 + A_4) - (A_0 + kA_7 + A_6)]$$

$$f_y \approx \frac{1}{k+2}[(A_0 + kA_1 + A_2) - (A_6 + kA_5 + A_4)]$$

3.3.2. Edge Detection

3.3.2.1. Gradient operator

Sobel operator (k=2)

$$f_x \approx \frac{1}{k+2}[(A_2 + kA_3 + A_4) - (A_0 + kA_7 + A_6)]$$

$$f_y \approx \frac{1}{k+2}[(A_0 + kA_1 + A_2) - (A_6 + kA_5 + A_4)]$$

3.3.2. Edge Detection

3.3.2.1. Gradient operator

Frei-Chen (k=2$^{1/2}$)

$$f_x \approx \frac{1}{k+2}[(A_2 + kA_3 + A_4) - (A_0 + kA_7 + A_6)]$$

$$f_y \approx \frac{1}{k+2}[(A_0 + kA_1 + A_2) - (A_6 + kA_5 + A_4)]$$

3.3.2. Edge Detection

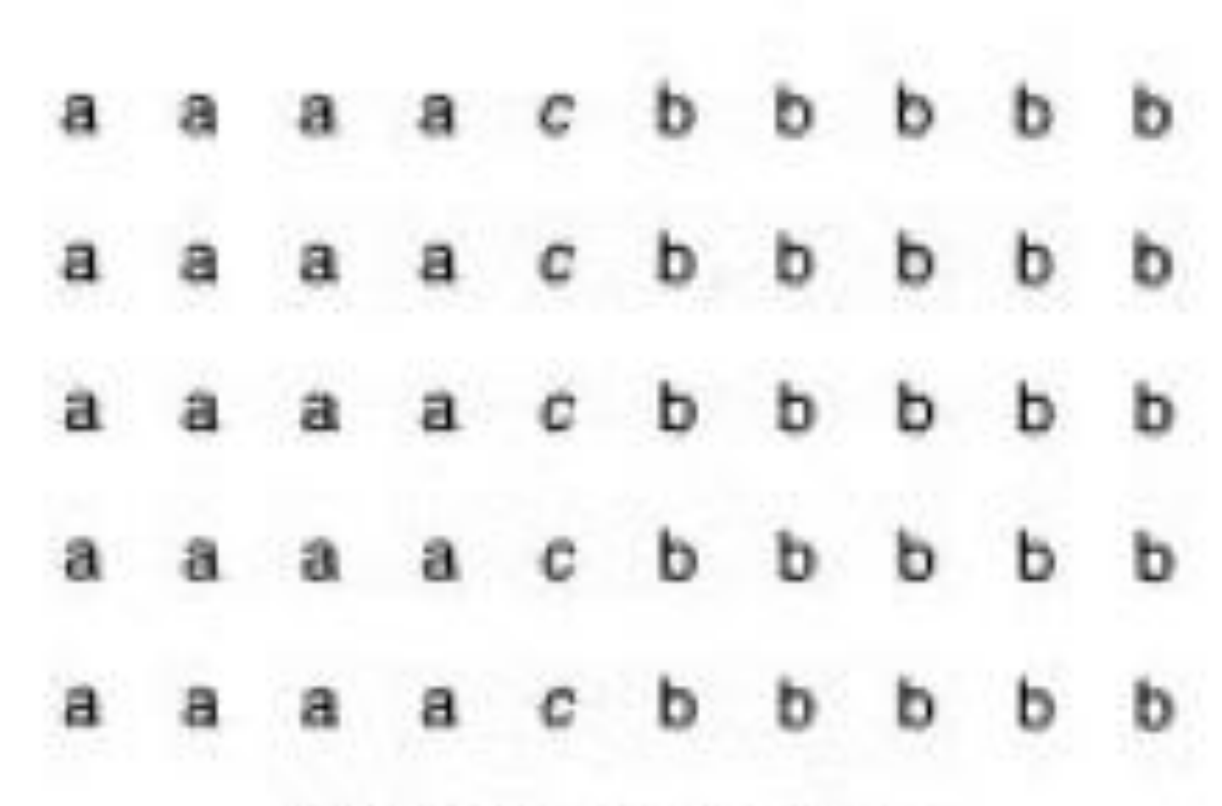
3.3.2.1. Gradient operator

Prewitt

Sobel

Frei-Chen

h=b-a



a	a	a	a	c	b	b	b	b	b
a	a	a	a	c	b	b	b	b	b
a	a	a	a	c	b	b	b	b	b
a	a	a	a	c	b	b	b	b	b
a	a	a	a	c	b	b	b	b	b

Vertical ramp edge

0 0 h/2 h h/2 0 0

3.3.2. Edge Detection

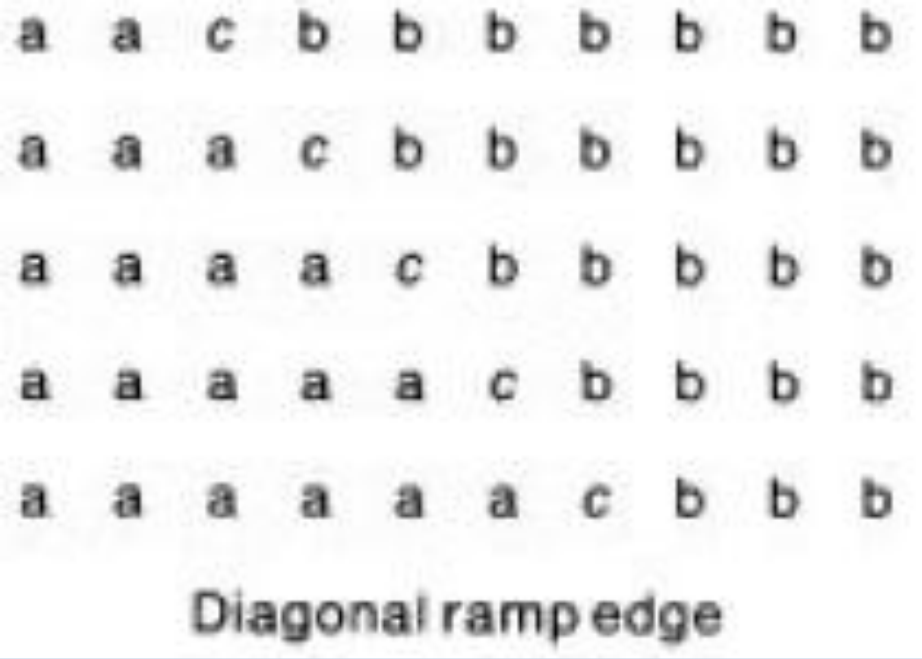
3.3.2.1. Gradient operator

Prewitt

Sobel

Frei-Chen

h=b-a



a	a	c	b	b	b	b	b	b	b
a	a	a	c	b	b	b	b	b	b
a	a	a	a	c	b	b	b	b	b
a	a	a	a	a	c	b	b	b	b
a	a	a	a	a	a	c	b	b	b

Diagonal ramp edge

$$0 \quad \frac{h}{\sqrt{2}(2+k)} \quad \frac{h}{\sqrt{2}} \quad \frac{\sqrt{2}(1+k)h}{(2+k)} \quad \frac{h}{\sqrt{2}} \quad \frac{h}{\sqrt{2}(2+k)} \quad 0$$

3.3.2. Edge Detection

3.3.2.1. Gradient operator

$$\mathrm{Frei\text{-}Chen}: \mathrm{k} = \sqrt{2} \Rightarrow \frac{\sqrt{2}(1+\mathrm{k})\mathrm{h}}{(2+k)} = \mathrm{h}$$

$$\mathrm{Prewitt}: \qquad \mathrm{k} = 1 \Rightarrow \frac{\sqrt{2}(2)\mathrm{h}}{(3)} = 0.94 \times \mathrm{h}$$

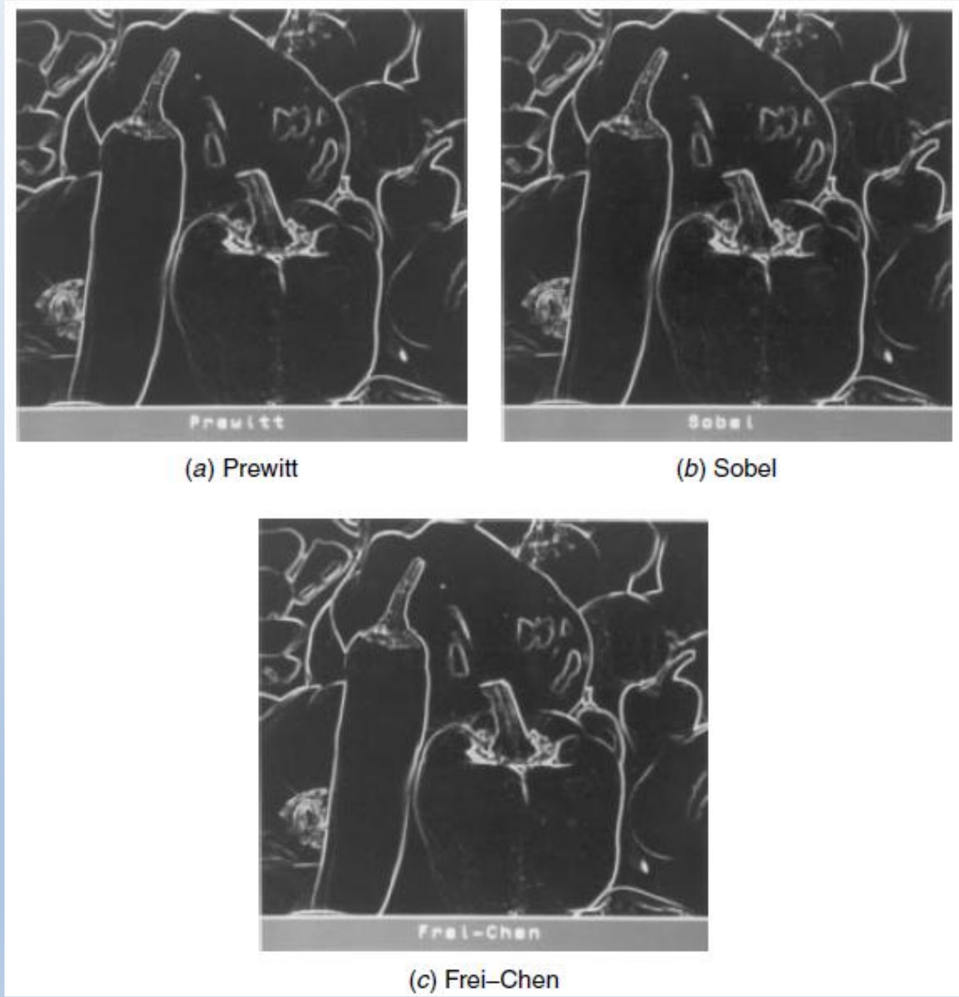
$$\mathrm{Sobel}: \qquad \mathrm{k} = 2 \Rightarrow \frac{\sqrt{2}(3)\mathrm{h}}{(4)} = 1.06 \times \mathrm{h}$$

$$\mathrm{h} = \mathrm{b} - \mathrm{a}$$

3.3.2. Edge Detection

3.3.2.1. Gradient operator

Prewitt-Sobel-FreiChen



(*a*) Prewitt

(*b*) Sobel

(*c*) Frei–Chen

3.3.2. Edge Detection

3.3.2.1. Gradient operator

Convolution operator

$$g = f * h$$

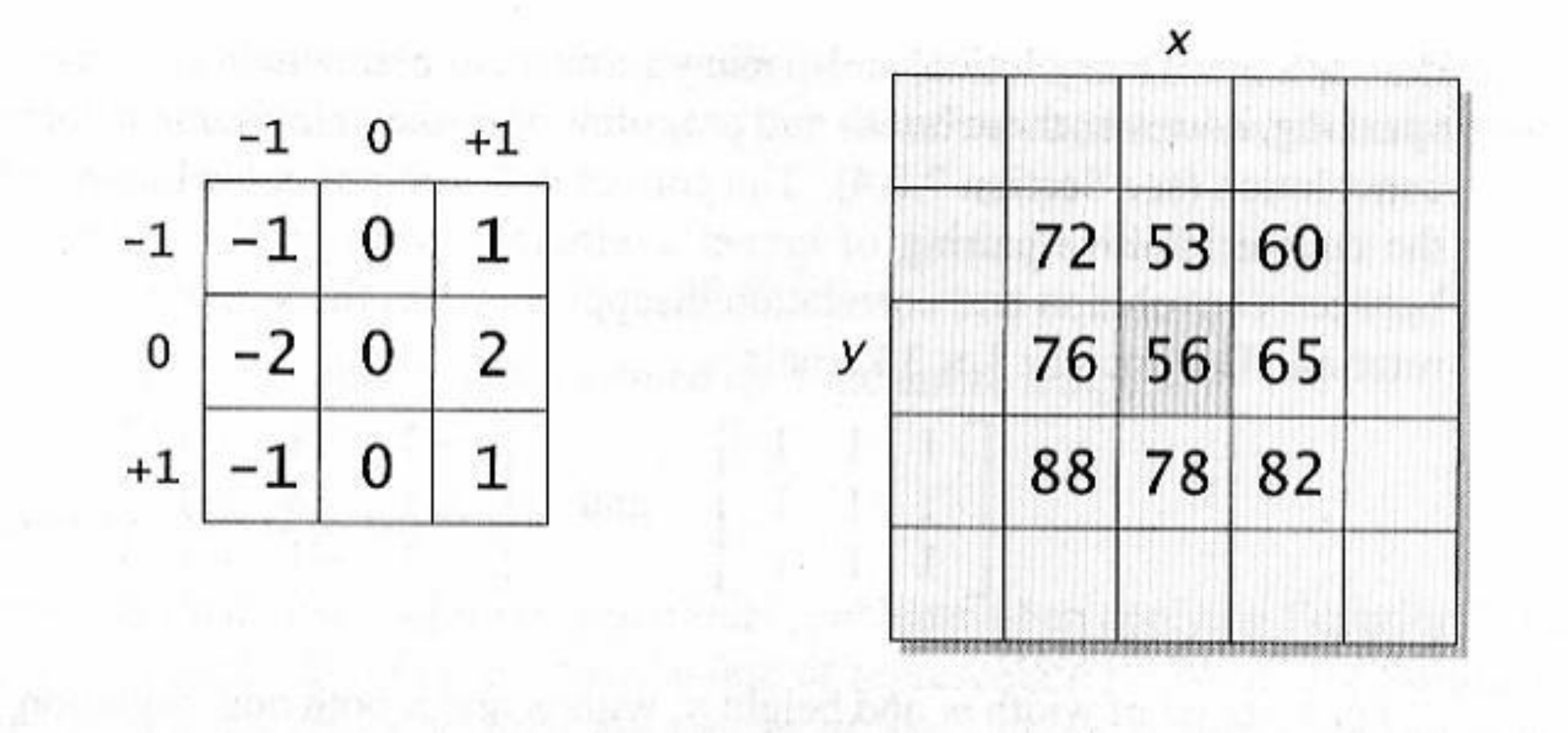
$$g(x,y) = \sum_{i}\sum_{j} f(x-i, y-j).h(i,j),$$

$$(i,j) \in O$$

3.3.2. Edge Detection

3.3.2.1. Gradient operator

Comsider image f and filter h as follows:



	−1	0	+1
-1	−1	0	1
0	−2	0	2
+1	−1	0	1

x

y

	72	53	60	
	76	56	65	
	88	78	82	

3.3.2. Edge Detection

3.3.2.1. Gradient operator

Ouput image g computed at (x,y):

$$
\begin{aligned}
g(x,y) = {} & h(-1,-1)\,f(x+1,y+1) + h(0,-1)\,f(x,y+1) + \\
& h(1,-1)\,f(x-1,y+1) + h(-1,0)\,f(x+1,y) + \\
& h(0,0)\,f(x,y) + h(1,0)\,f(x-1,y) + \\
& h(-1,1)\,f(x+1,y-1) + h(0,1)\,f(x,y-1) + \\
& h(1,1)\,f(x-1,y-1)
\end{aligned}
$$

$$g(x,y) = 40$$

3.3.2. Edge Detection

3.3.2.1. Gradient operator

Pixel difference

$$f_x \approx \hat{f}_x = f * W_x$$

$$f_y \approx \hat{f}_y = f * W_y$$

$$W_x = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 0 \end{bmatrix}, \quad W_y = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

3.3.2. Edge Detection

3.3.2.1. Gradient operator

Separated Pixel difference

$$f_x \approx \hat{f}_x = f * W_x$$

$$f_y \approx \hat{f}_y = f * W_y$$

$$W_x = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix}, \; W_y = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

3.3.2. Edge Detection

3.3.2.1. Gradient operator

Robert operator

$$f_x \approx \hat{f}_x = f * W_x$$

$$f_y \approx \hat{f}_y = f * W_y$$

$$W_x = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \ W_y = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

3.3.2. Edge Detection

3.3.2.1. Gradient operator

Prewitt operator

$$f_x \approx \hat{f}_x = f * W_x$$

$$f_y \approx \hat{f}_y = f * W_y$$

$$W_x = \frac{1}{3}\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}, \quad W_y = \frac{1}{3}\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

3.3.2. Edge Detection

3.3.2.1. Gradient operator

Sobel operator

$$f_x \approx \hat{f}_x = f * W_x$$

$$f_y \approx \hat{f}_y = f * W_y$$

$$W_x = \frac{1}{4}\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}, \quad W_y = \frac{1}{4}\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

3.3.2. Edge Detection

3.3.2.1. Gradient operator

Frei-chen operator

$$f_x \approx \hat{f}_x = f * W_x \text{ , } f_y \approx \hat{f}_y = f * W_y \text{ ,}$$

$$W_x = \frac{1}{2+\sqrt{2}}\begin{bmatrix} 1 & 0 & -1 \\ \sqrt{2} & 0 & -\sqrt{2} \\ 1 & 0 & -1 \end{bmatrix},$$

$$W_y = \frac{1}{2+\sqrt{2}}\begin{bmatrix} -1 & -\sqrt{2} & -1 \\ 0 & 0 & 0 \\ 1 & \sqrt{2} & 1 \end{bmatrix}$$

3.3.2. Edge Detection

3.3.2.2. Laplace operator

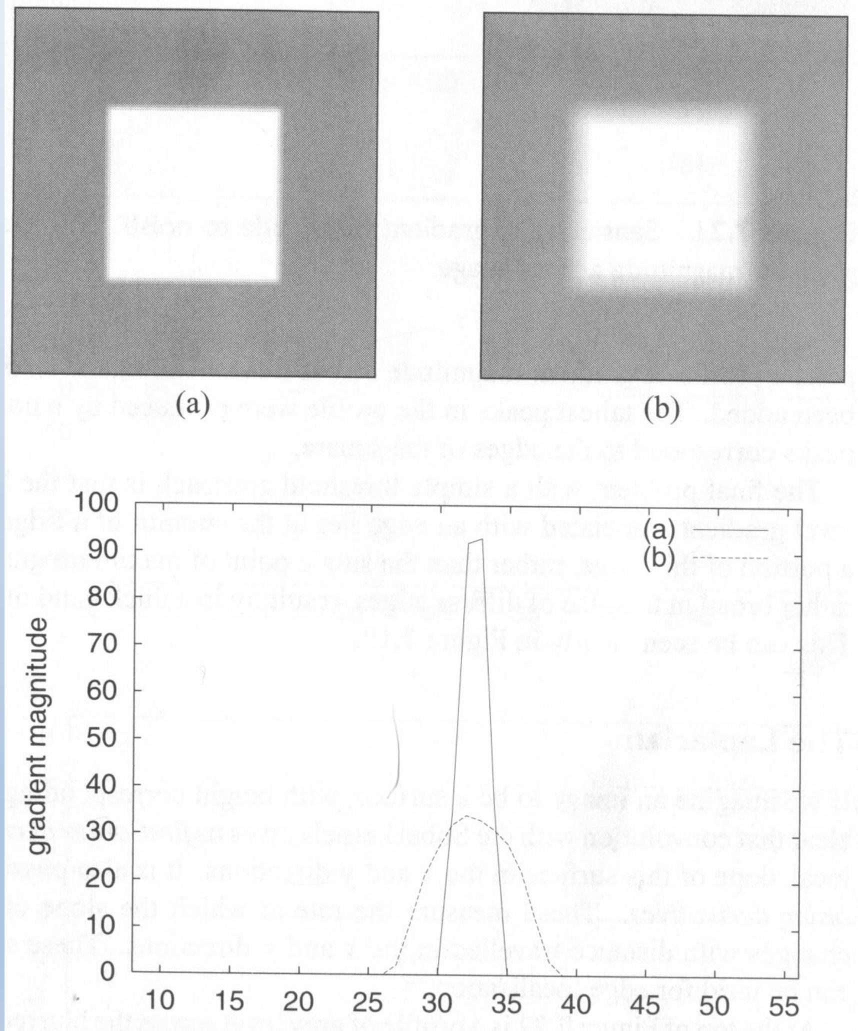
$$(\nabla^2 f)(x,y) = \partial^2 f/\partial x^2 + \partial^2 f/\partial y^2$$

$$(\nabla^2 f)(x,y) \approx f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y)$$

Associate Prof. Lý Quốc Ngọc

3.3.2. Edge Detection

3.3.2.2. Laplace operator



3.3.2. Edge Detection

3.3.2.2. Laplace operator

$$\nabla^2 f \approx f * Laplace$$

$$Laplace = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

3.3.2. Edge Detection

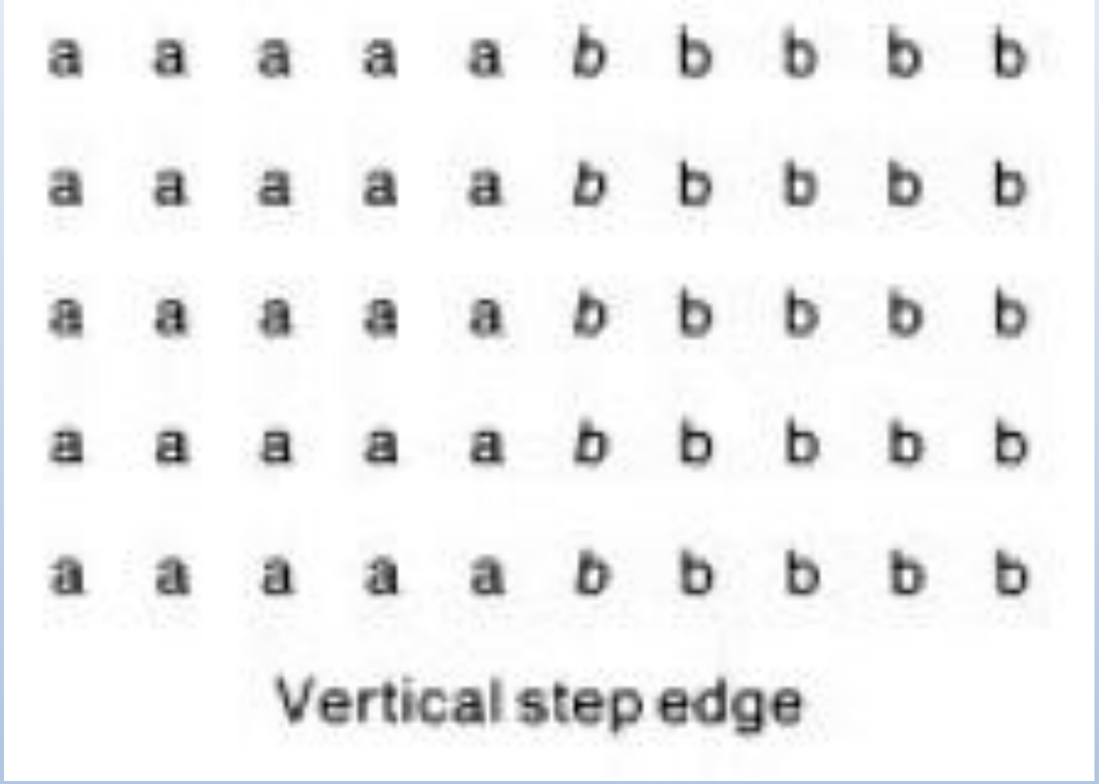
3.3.2.2. Laplace operator

$$\nabla^2 f \approx f * Laplace$$

$$Laplace = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

3.3.2. Edge Detection

3.3.2.2. Laplace operator



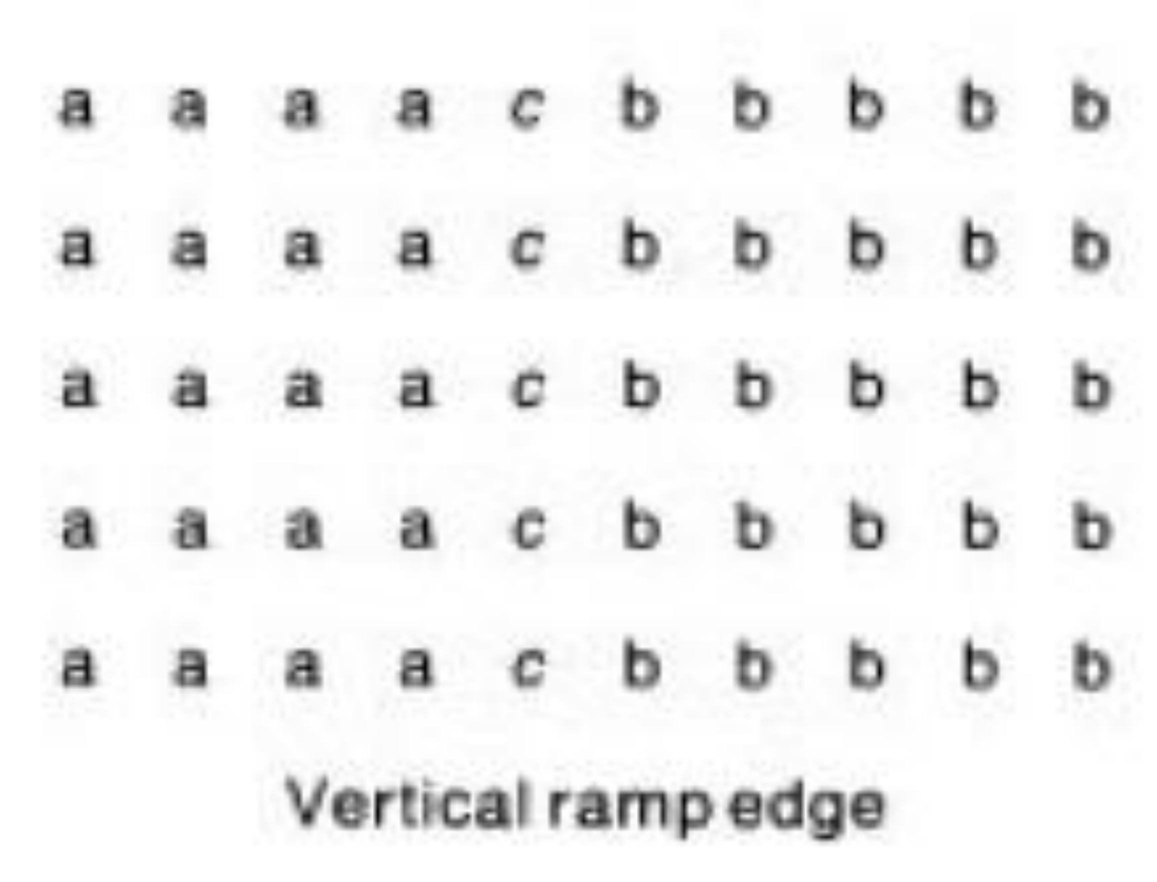
a	a	a	a	a	b	b	b	b	b
a	a	a	a	a	b	b	b	b	b
a	a	a	a	a	b	b	b	b	b
a	a	a	a	a	b	b	b	b	b
a	a	a	a	a	b	b	b	b	b

Vertical step edge

0 -3h/8 3h/8 0

3.3.2. Edge Detection

3.3.2.2. Laplace operator



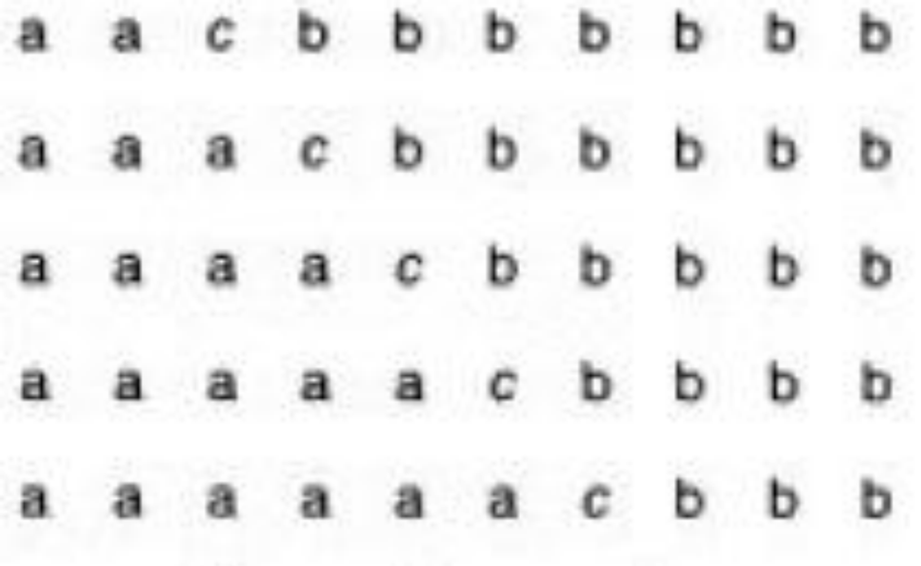
a	a	a	a	c	b	b	b	b	b
a	a	a	a	c	b	b	b	b	b
a	a	a	a	c	b	b	b	b	b
a	a	a	a	c	b	b	b	b	b
a	a	a	a	c	b	b	b	b	b

Vertical ramp edge

0 -3h/16 0 3h/16 0

3.3.2. Edge Detection

3.3.2.2. Laplace operator



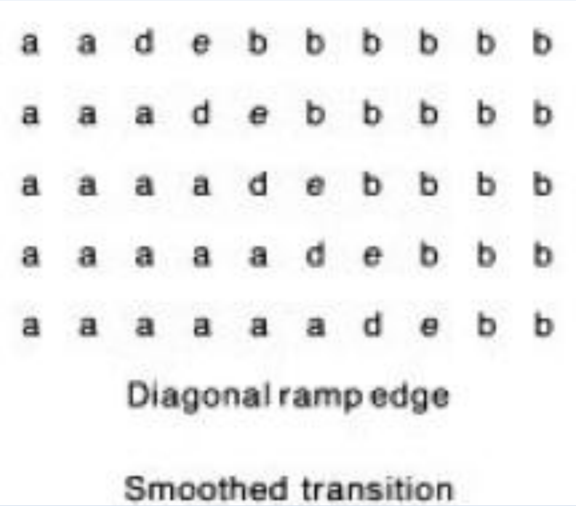
a	a	c	b	b	b	b	b	b	b
a	a	a	c	b	b	b	b	b	b
a	a	a	a	c	b	b	b	b	b
a	a	a	a	a	c	b	b	b	b
a	a	a	a	a	a	c	b	b	b

Diagonal ramp edge

0 -h/8 -h/8 0 h/8 h/8 0

3.3.2. Edge Detection

3.3.2.2. Laplace operator



a	a	d	e	b	b	b	b	b	b
a	a	a	d	e	b	b	b	b	b
a	a	a	a	d	e	b	b	b	b
a	a	a	a	a	d	e	b	b	b
a	a	a	a	a	a	d	e	b	b

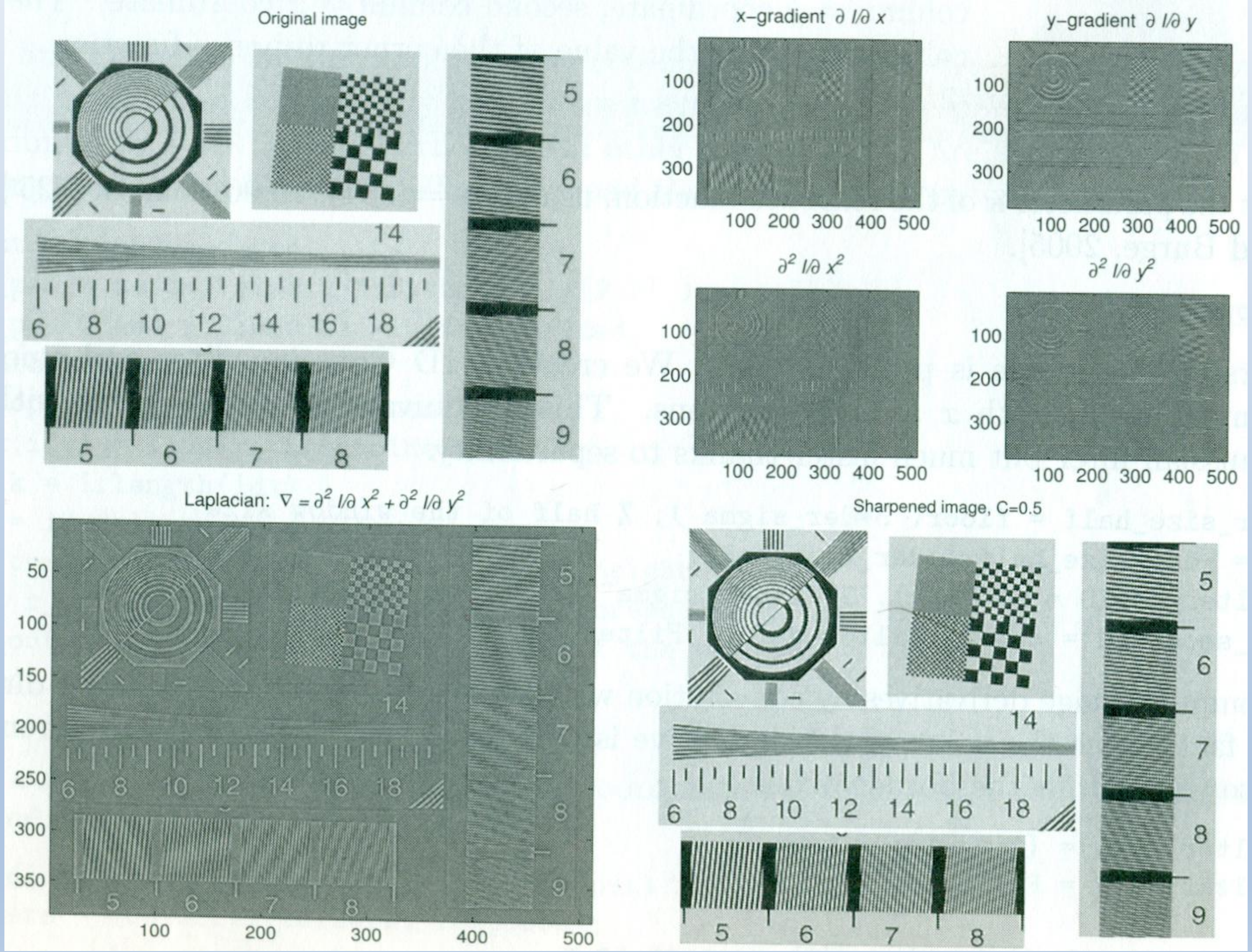
Diagonal ramp edge

Smoothed transition

0 -h/16 -h/8 -h/16 0 h/16 h/8 h/16 0

3.3.2. Edge Detection

3.3.2.2. Laplace operator



3.3.2. Edge Detection

3.3.2.2. Laplace of Gaussian operator

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

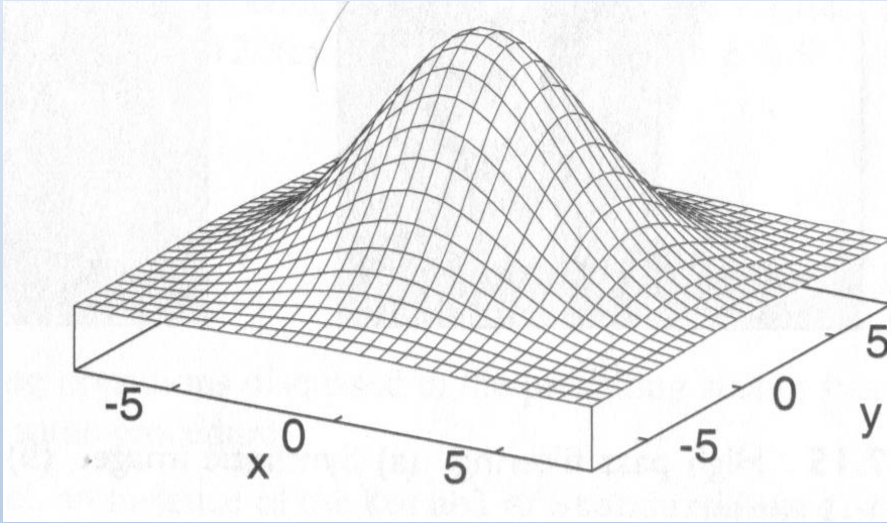
$$G(x,y,\sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$$

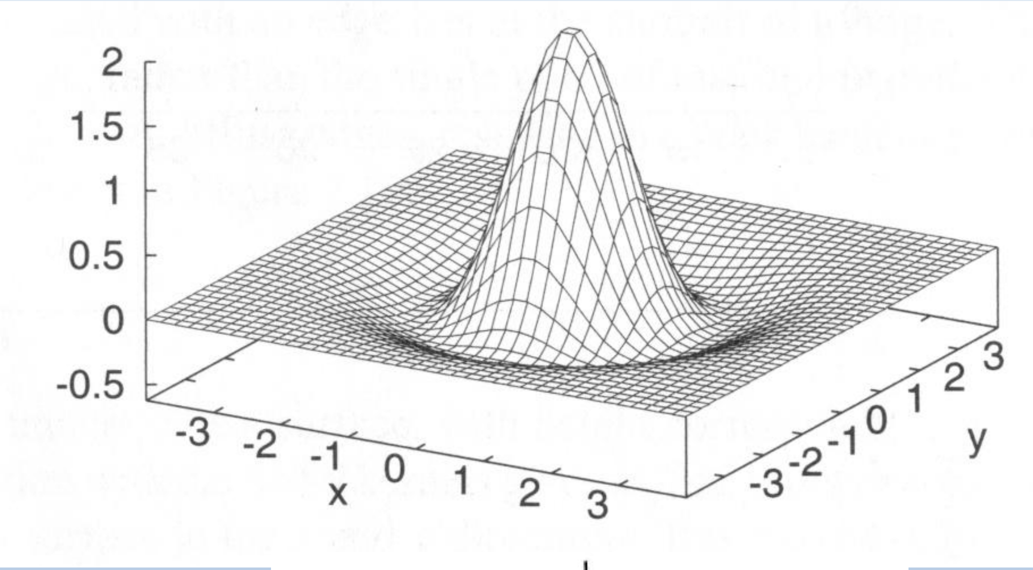
$$\nabla^2 G(x,y,\sigma) = \frac{\partial^2 G}{\partial x^2} + \frac{\partial^2 G}{\partial y^2}$$

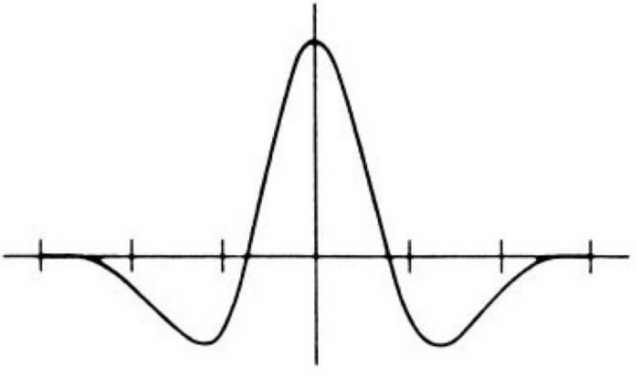
$$= \frac{1}{\pi\sigma^4}\left[\frac{x^2+y^2}{2\sigma^2} - 1\right] e^{-\frac{x^2+y^2}{2\sigma^2}}$$

3.3.2. Edge Detection

3.3.2.2. Laplace of Gaussian operator

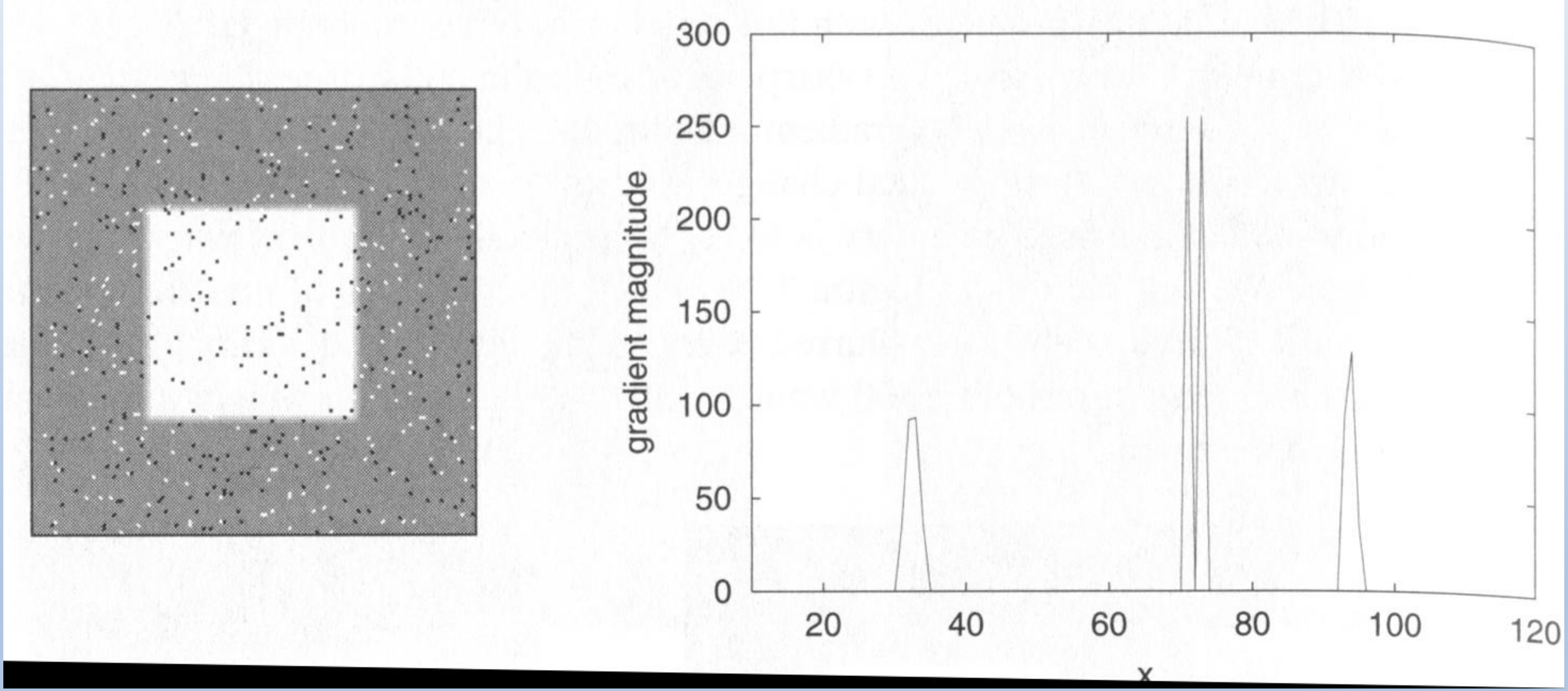






3.3.2. Edge Detection

3.3.2.2. Laplace of Gaussian operator



3.3.2. Edge Detection

3.3.2.2. **Canny edge detection method**

1. $G(x, y, \sigma) * f(x, y)$

2. $n = \nabla(G * f)$

3. Find the location of the edges using non-maximal suppression approach.

4. Compute the magnitude of the edge is measured as

$|\nabla(G * f)|$

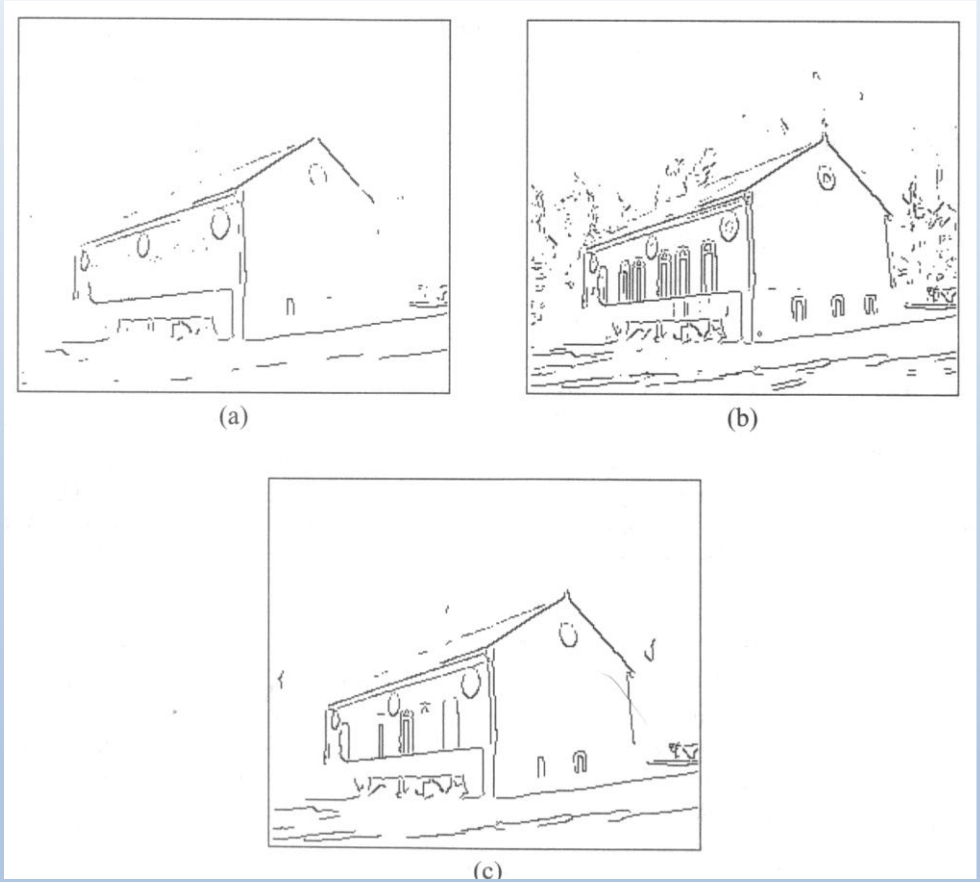
5. Threshold edges in the image with hysteresis to eliminate spurious responses.

6. Repeat steps 1-5 for ascending values of the standard deviation σ .

7. Aggregate the final information about edges at multiple scale using ‘feature systhesis’ approach

3.3.2. Edge Detection

3.3.2.2. Canny edge detection method



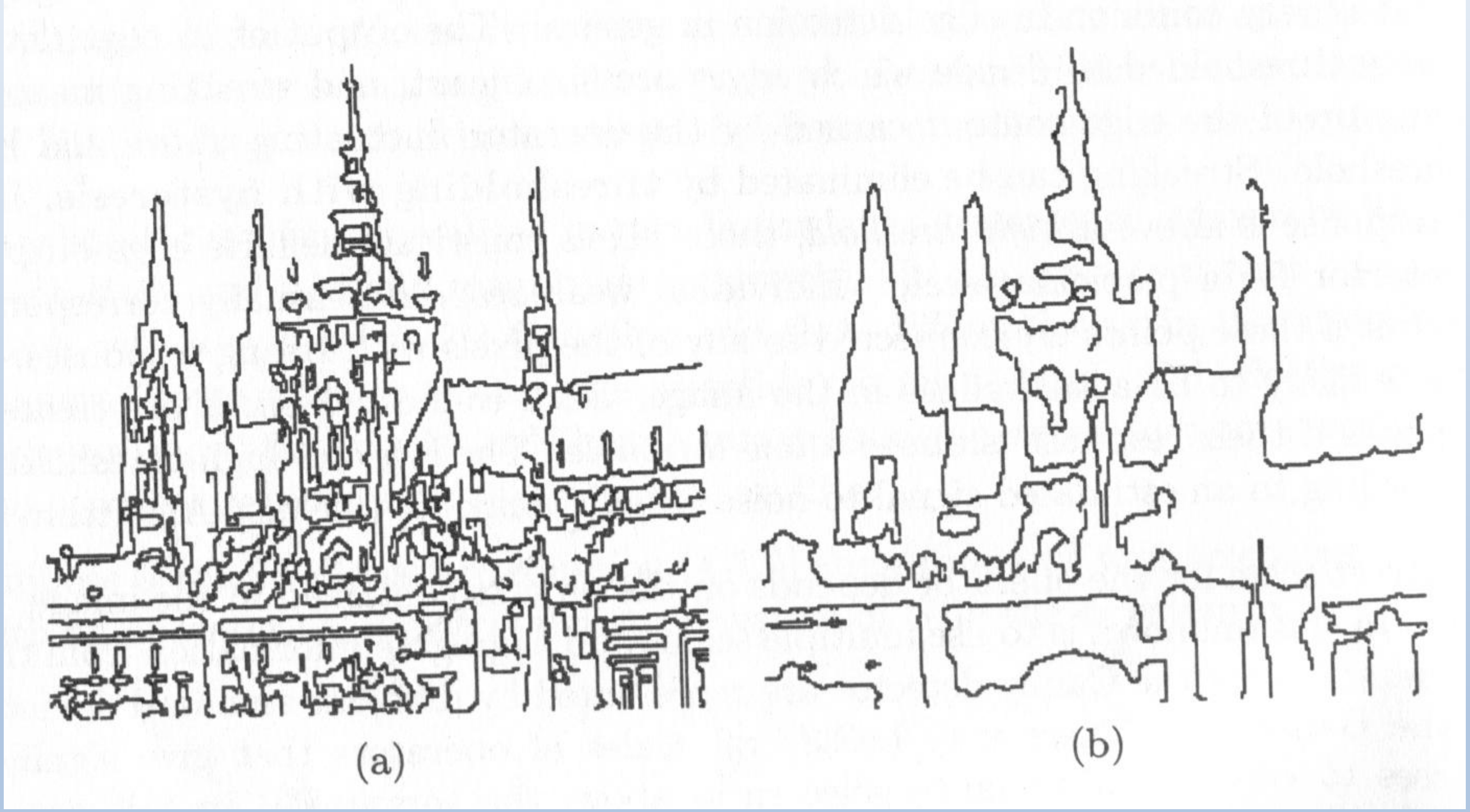
(a)

(b)

(c)

3.3.2. Edge Detection

3.3.2.2. **Canny edge detection method**



3.3.2. Edge Detection

3.3.2.2. Canny edge detection method

Non-maximal suppression of directional edge data

3. Find the location of the edges using non-maximal suppression approach.

3.1. For each pixel with non-zero edge magnitude, inspect two adjacent pixels indicated the direction of its edge.

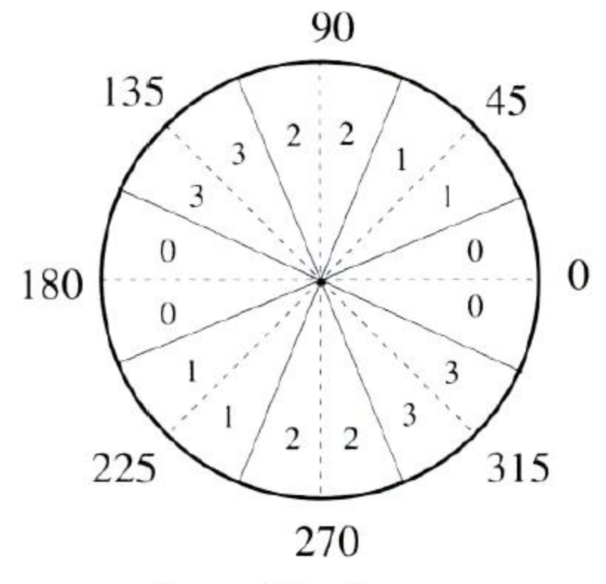
3.2. If the edge magnitude of either of these two exceeds that of the pixel under inspection, mark it for deletion.

3.3. When all pixels have been inspected, re-scan the image and erase to zero all edge data marked for deletion.

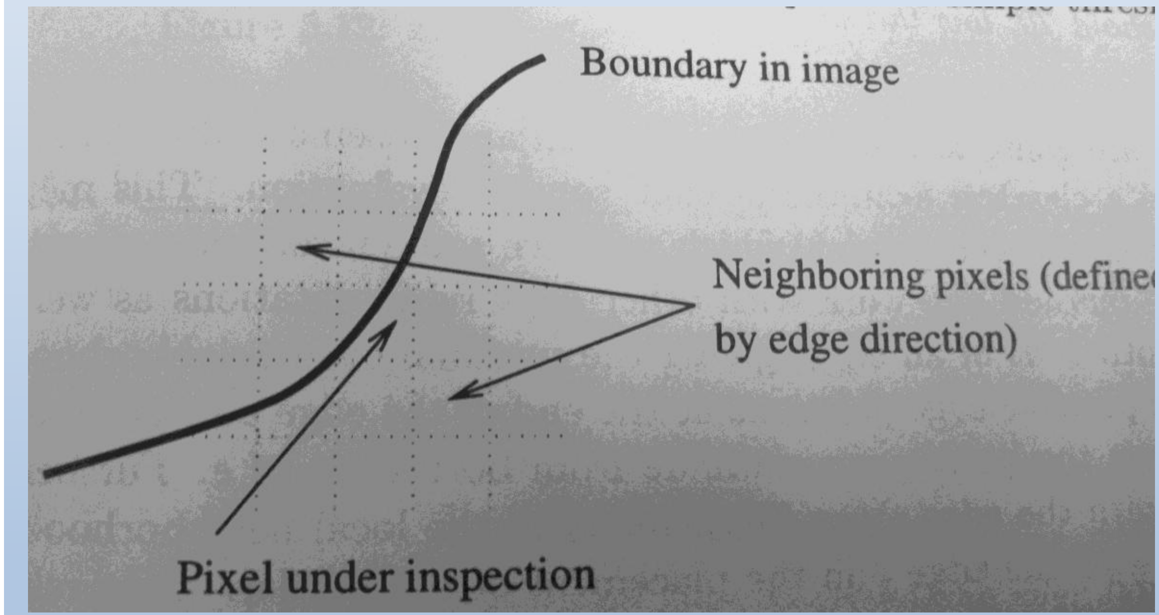
3.3.2. Edge Detection

3.3.2.2. Canny edge detection method

(Non-maximal suppression of directional edge data)



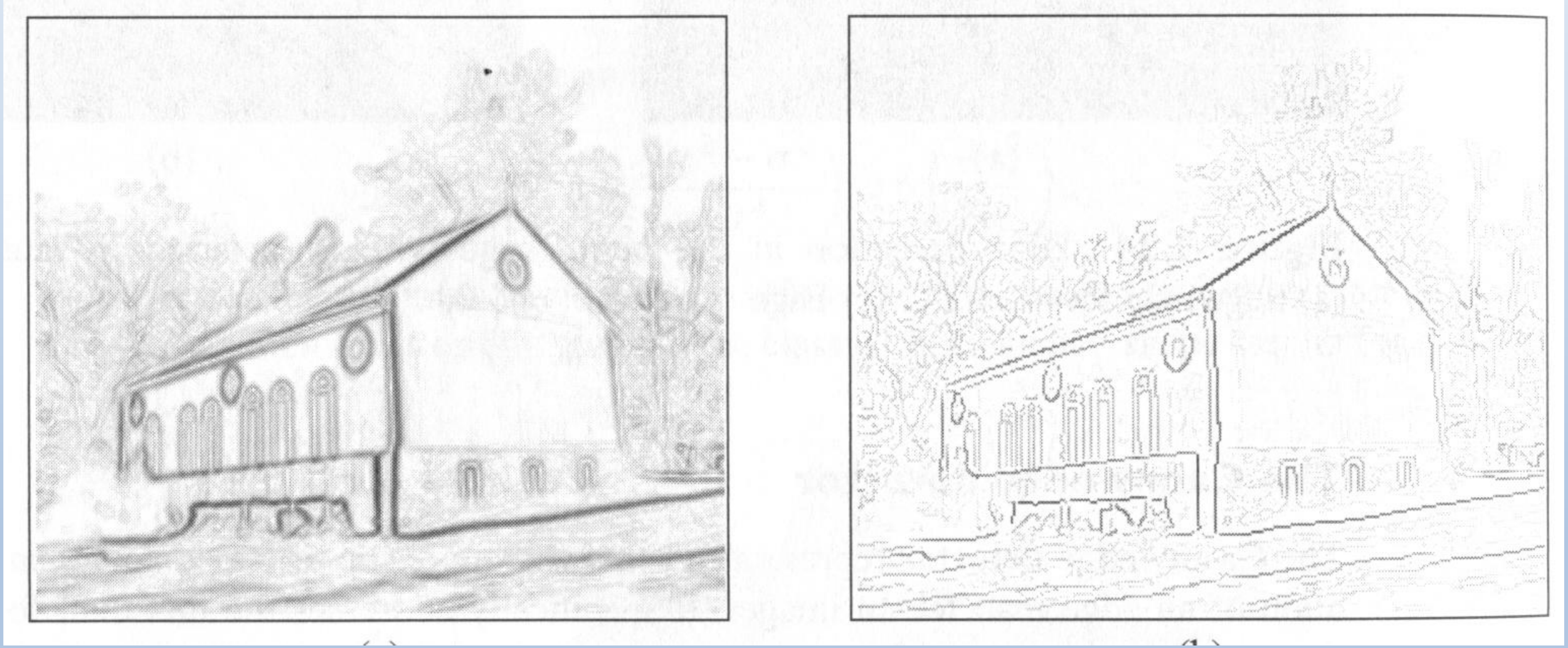
Canny Edge Detector



3.3.2. Edge Detection

3.3.2.2. Canny edge detection method

(Non-maximal suppression of directional edge data)



3.3.2. Edge Detection

3.3.2.2. **Canny edge detection method**

Hysteresis to filter output of an edge detector

5. Threshold edges in the image with hysteresis to eliminate spurious responses.

5.1. Mark all edges with magnitude greater than $\mathbf{t_1}$.

5.2. Scall all pixels with edge magnitude in the range $\mathbf{[t_0\ t_1]}$.

5.3. If such a pixel borders another marked as an edge, the mark it too.

5.4. Repeat from step 5.2 until stability.

3.3.2. Edge Detection

3.3.2.2. Canny edge detection method

Hysteresis to filter output of an edge detector