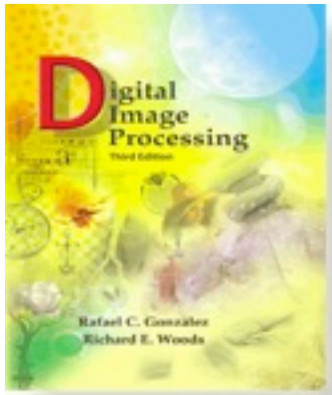
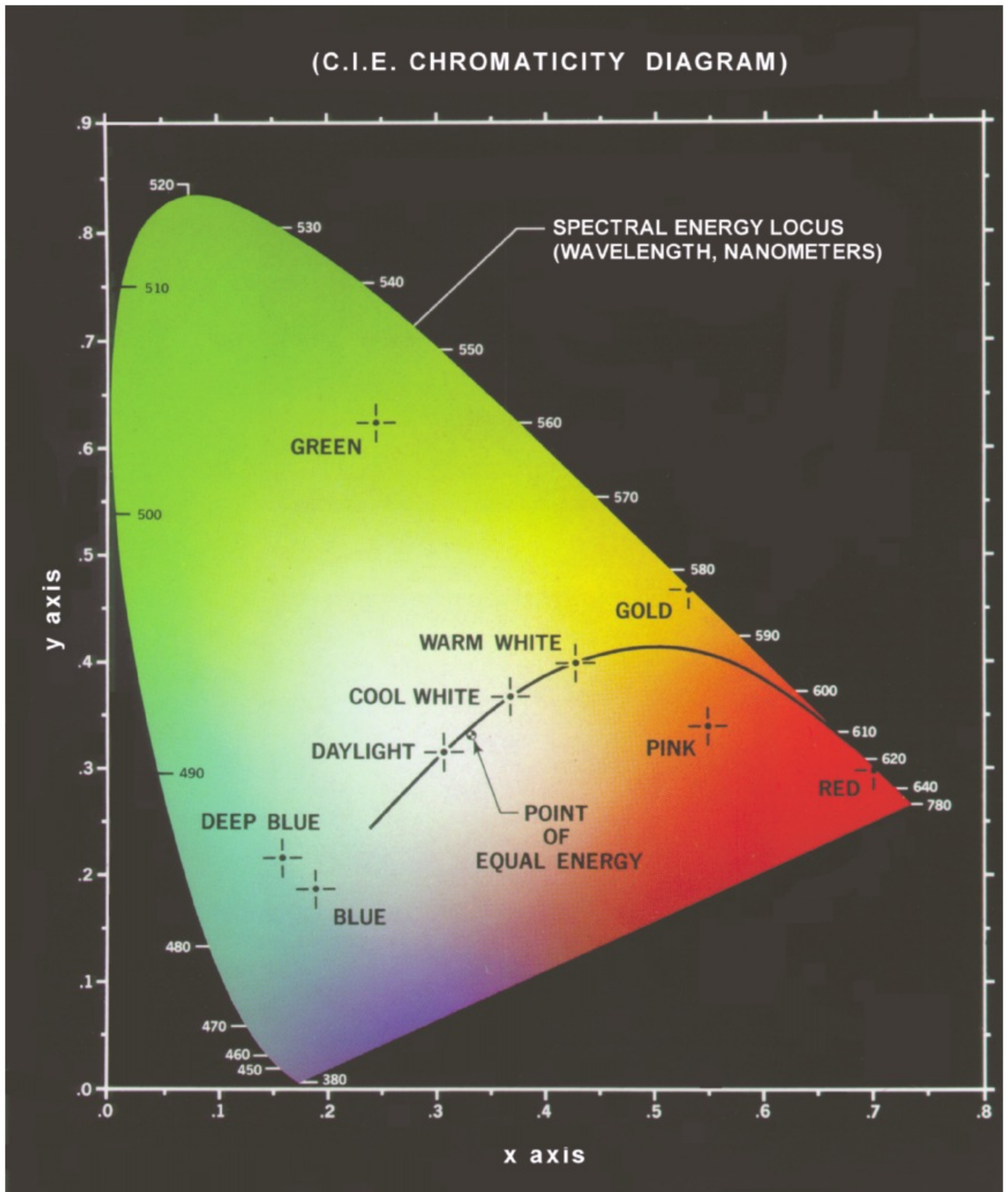
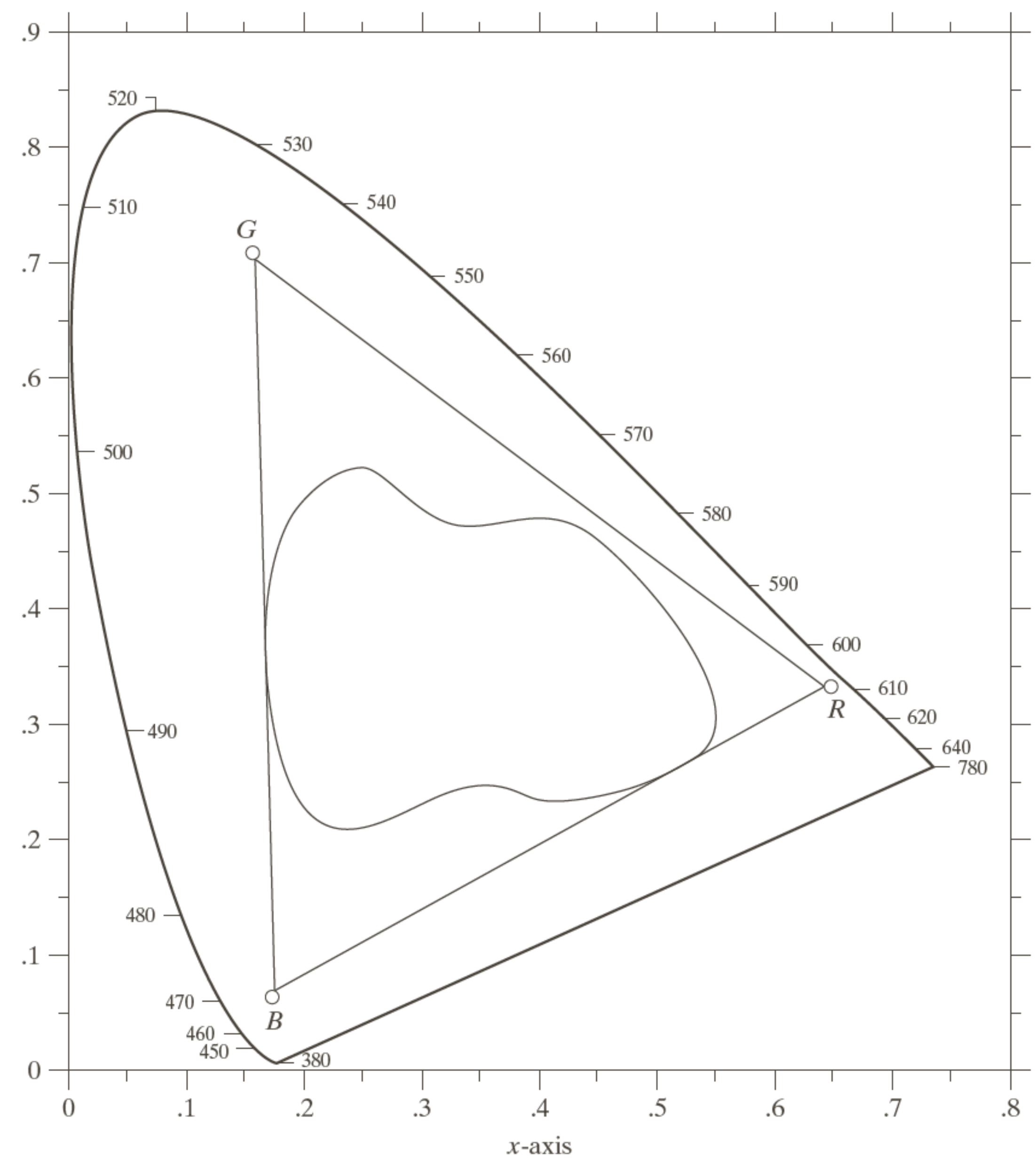


Chapter 6

Color Image Processing





Pseudocolor processing (vs. truecolor)

Pseudocolor = "false color"

Is the process of assigning color to a grayscale (or a set of grayscale) image.

This is to enhance/ emphasize some properties of the image.

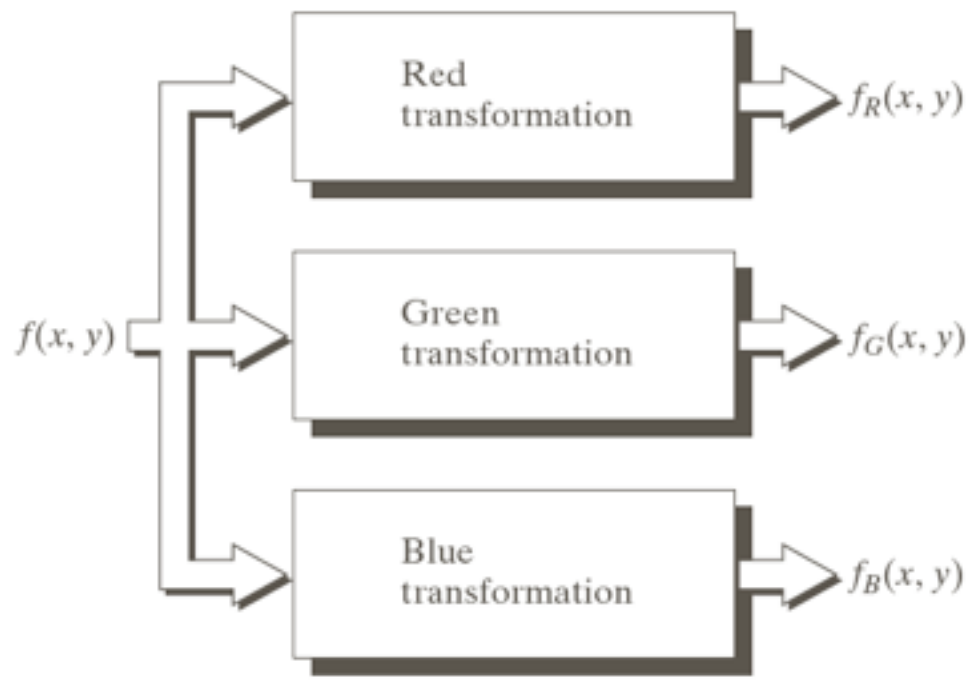


FIGURE 6.23 Functional block diagram for pseudocolor image processing. f_R, f_G, and f_B are fed into the corresponding red, green, and blue inputs of an RGB color monitor.

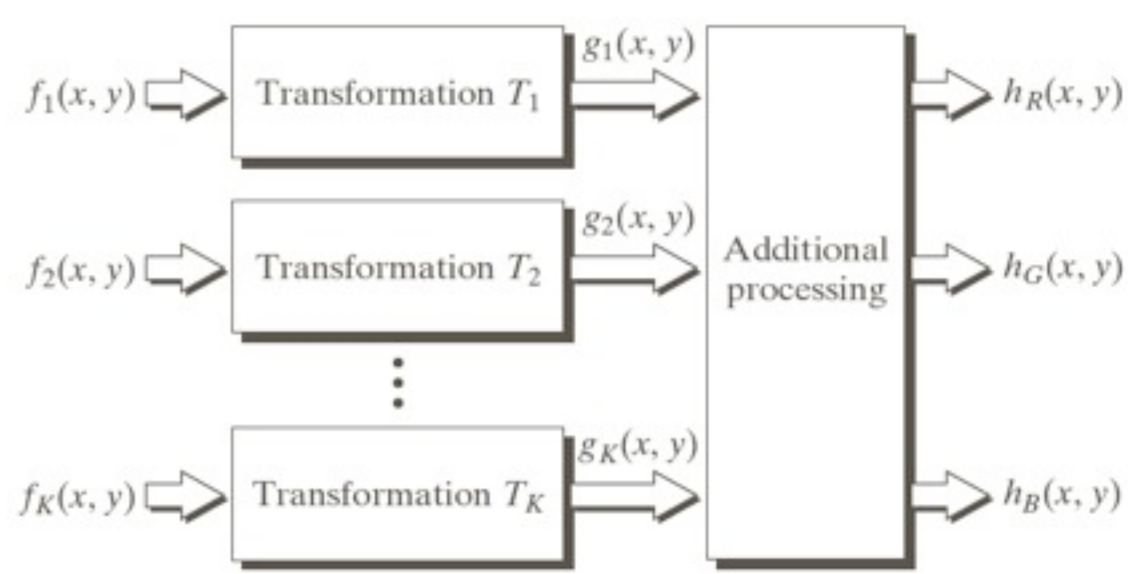


FIGURE 6.26 A pseudocolor coding approach used when several monochrome images are available.

Intensity slicing:

HSI representation thresholds on the intensity value.

Assign a color *c1* for the points having intensity less than a prescribed value *l*, and a color *c2* for those between *l* and *L-1*.

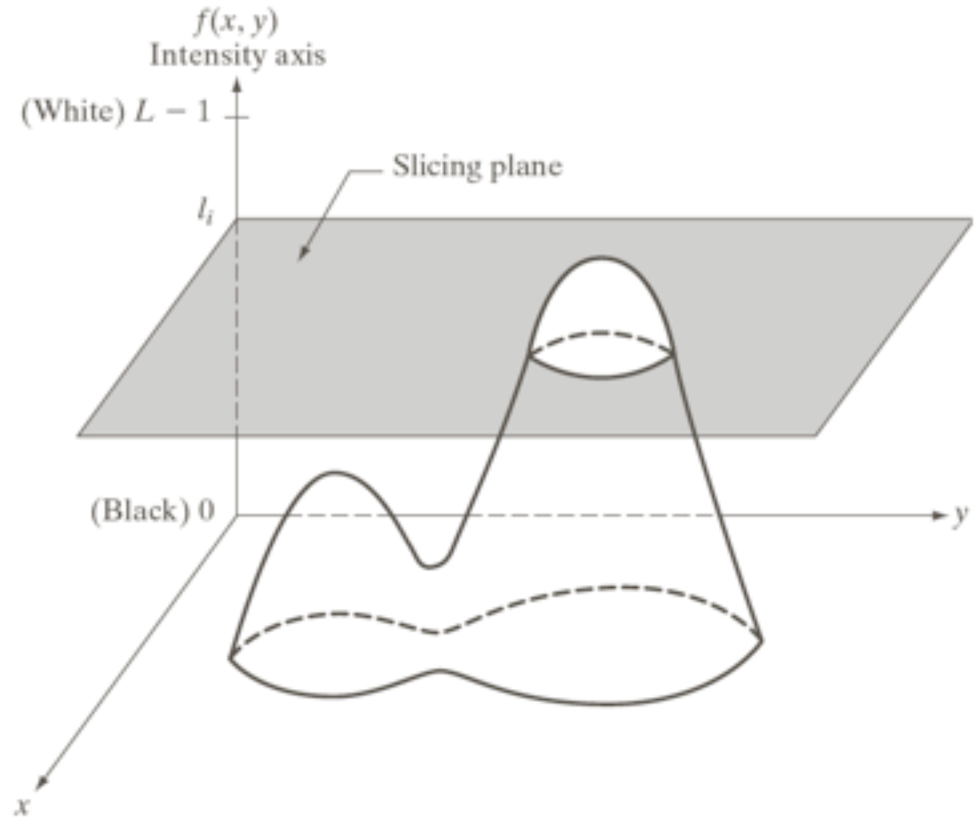


FIGURE 6.18 Geometric interpretation of the intensity-slicing technique.

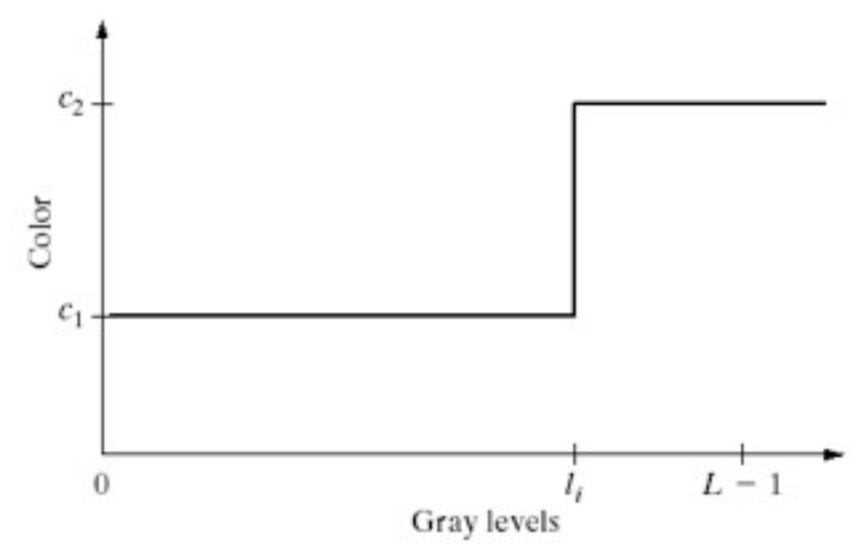
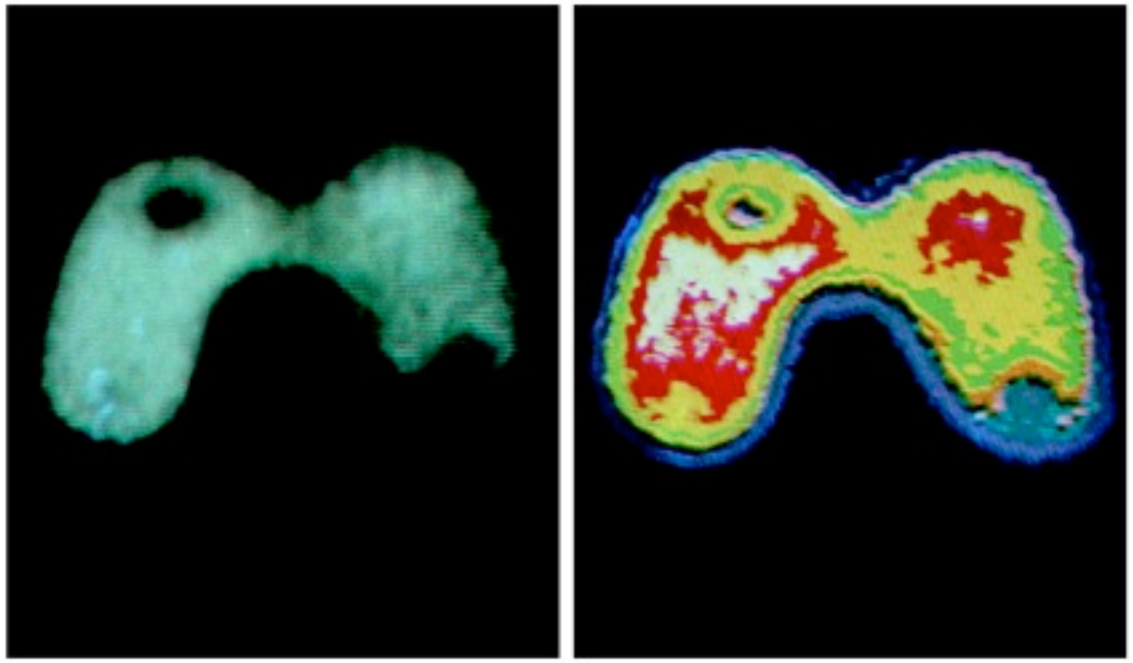


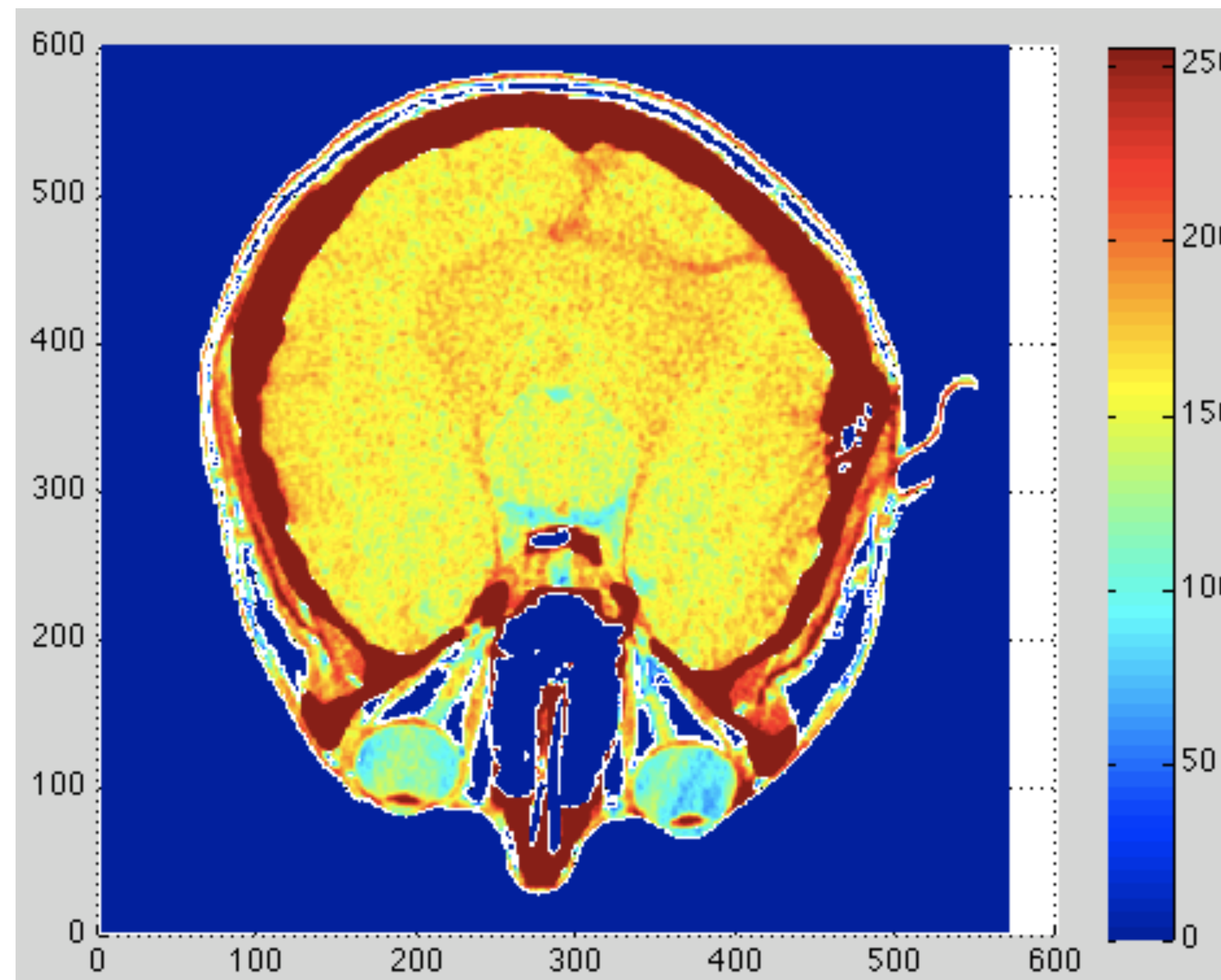
FIGURE 6.19 An alternative representation of the intensity-slicing technique.

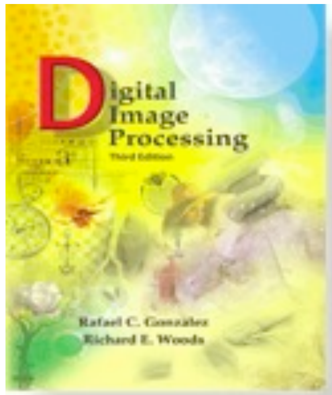


a b

FIGURE 6.20 (a) Monochrome image of the Picker Thyroid Phantom. (b) Result of density slicing into eight colors. (Courtesy of Dr. J. L. Blankenship, Instrumentation and Controls Division, Oak Ridge National Laboratory.)

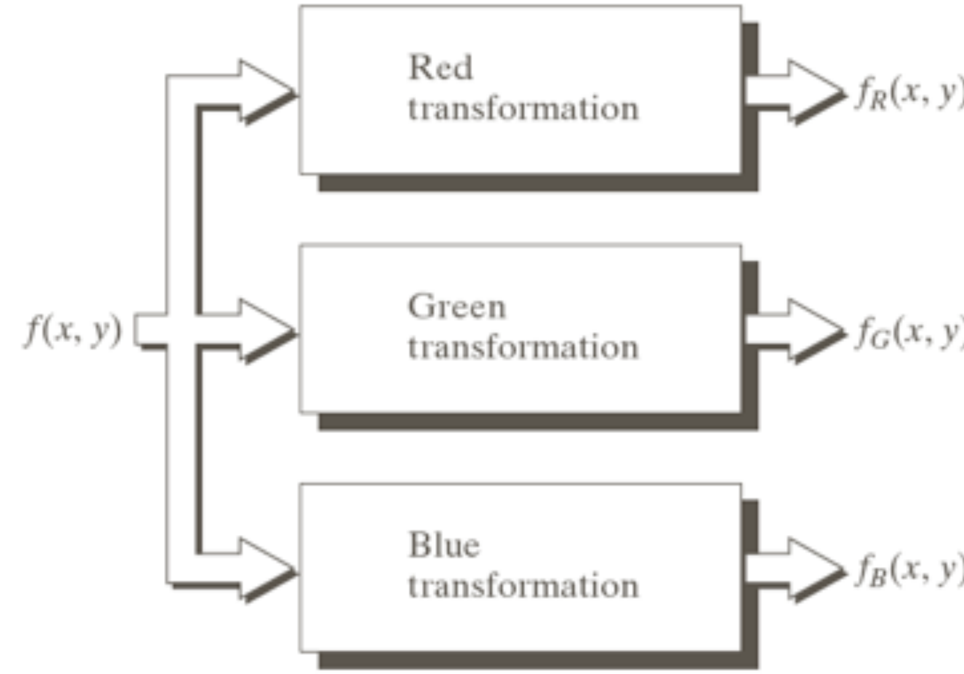
Example: Matlab's surface view is a pseudocolor transformation, assigning a color to height (in this case, grayscale intensity)

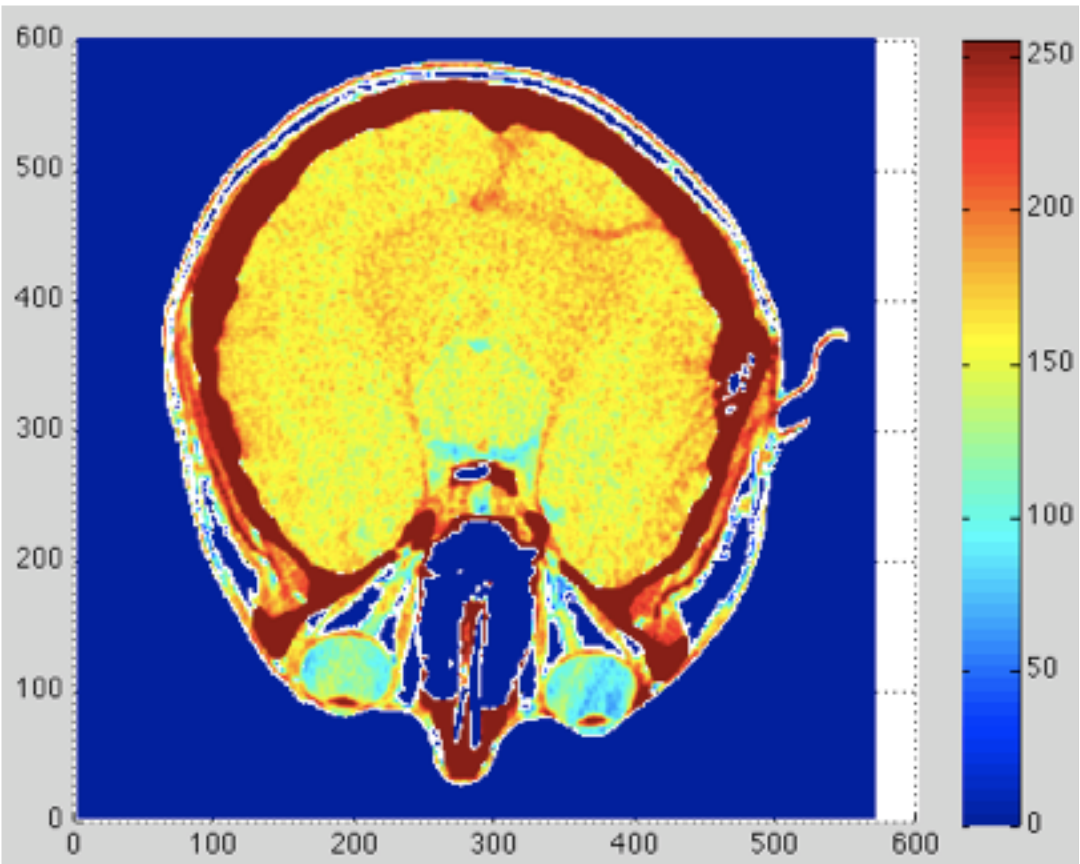


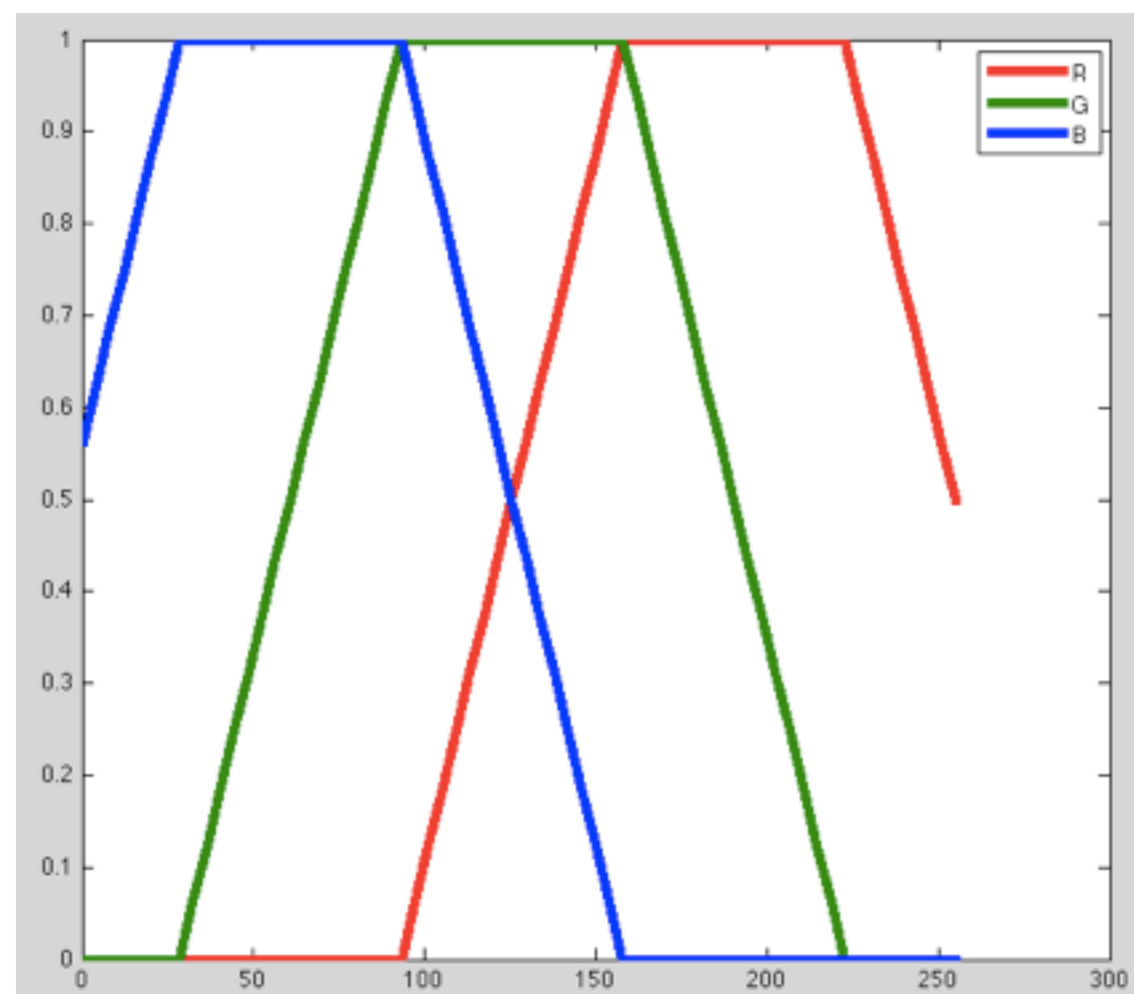


Chapter 6

Color Image Pro



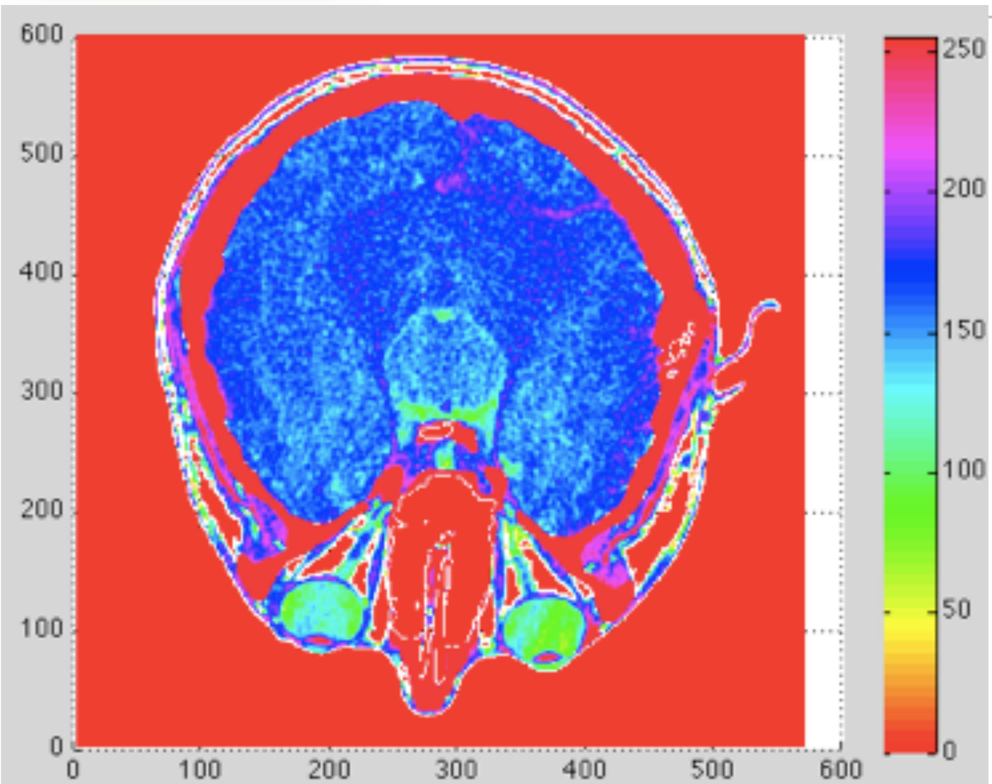


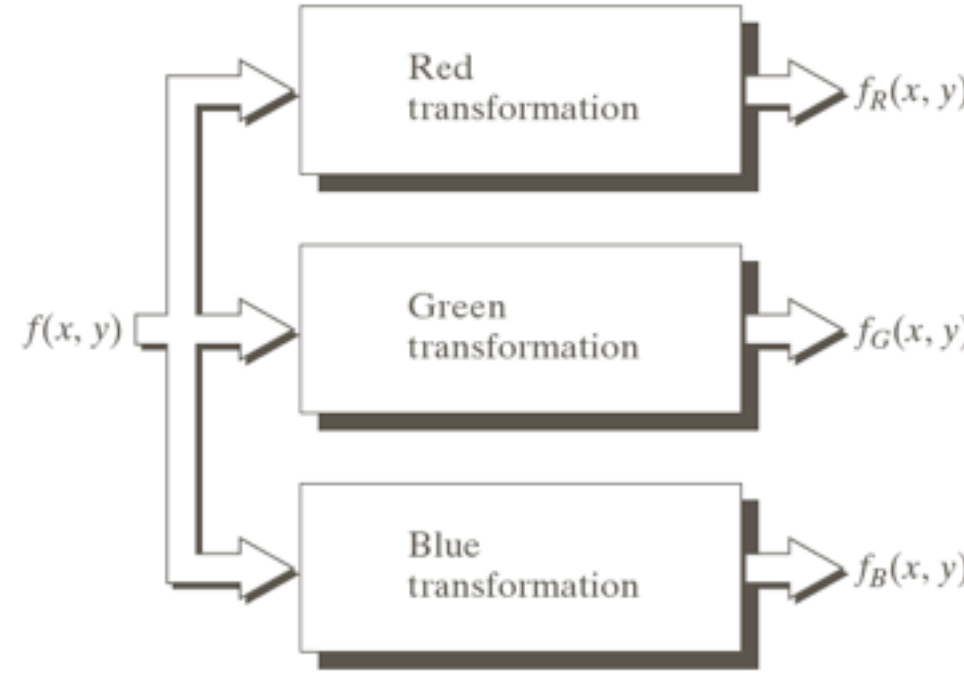


‘jet’ colormap

Chapter 6

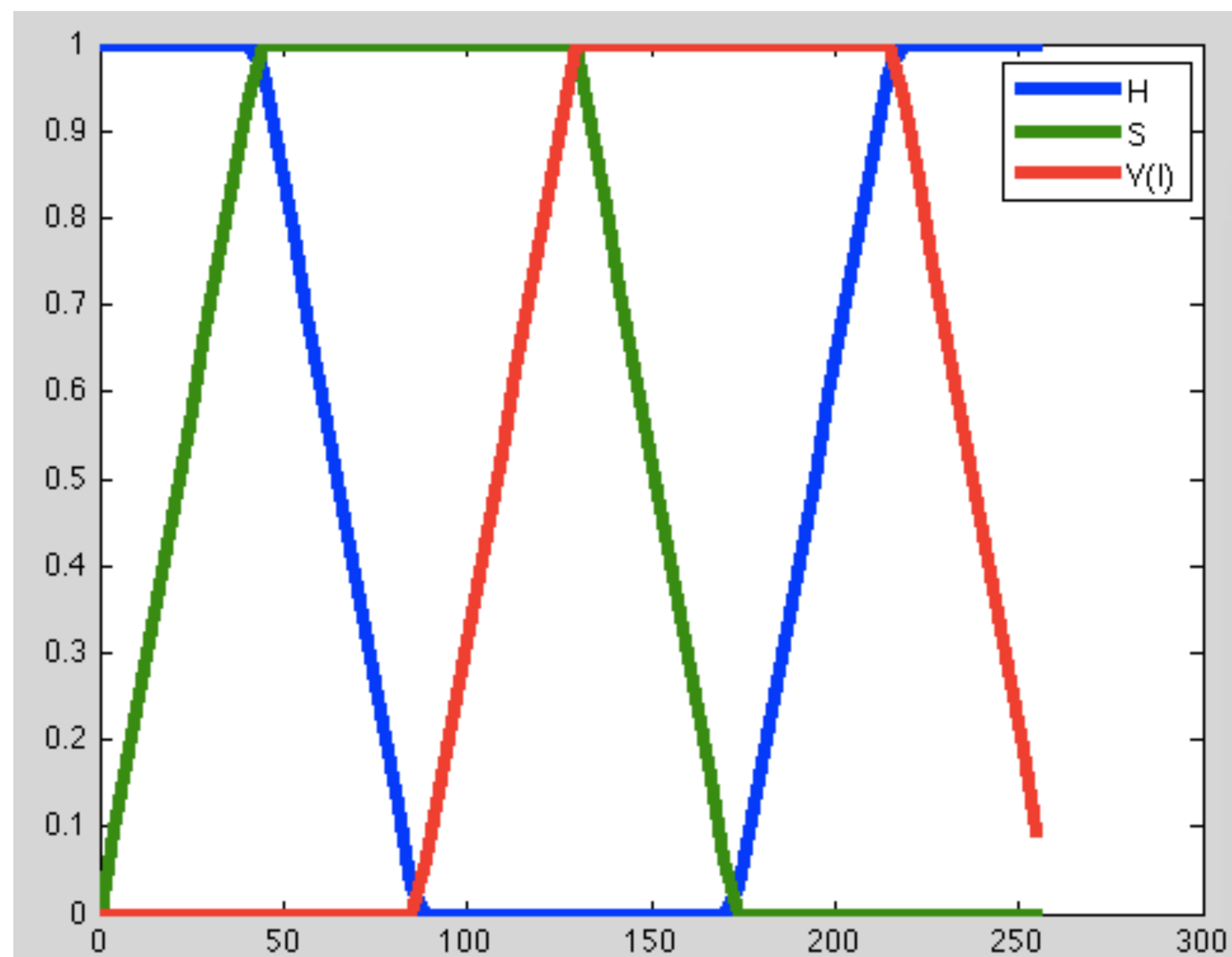
Color Image Pro

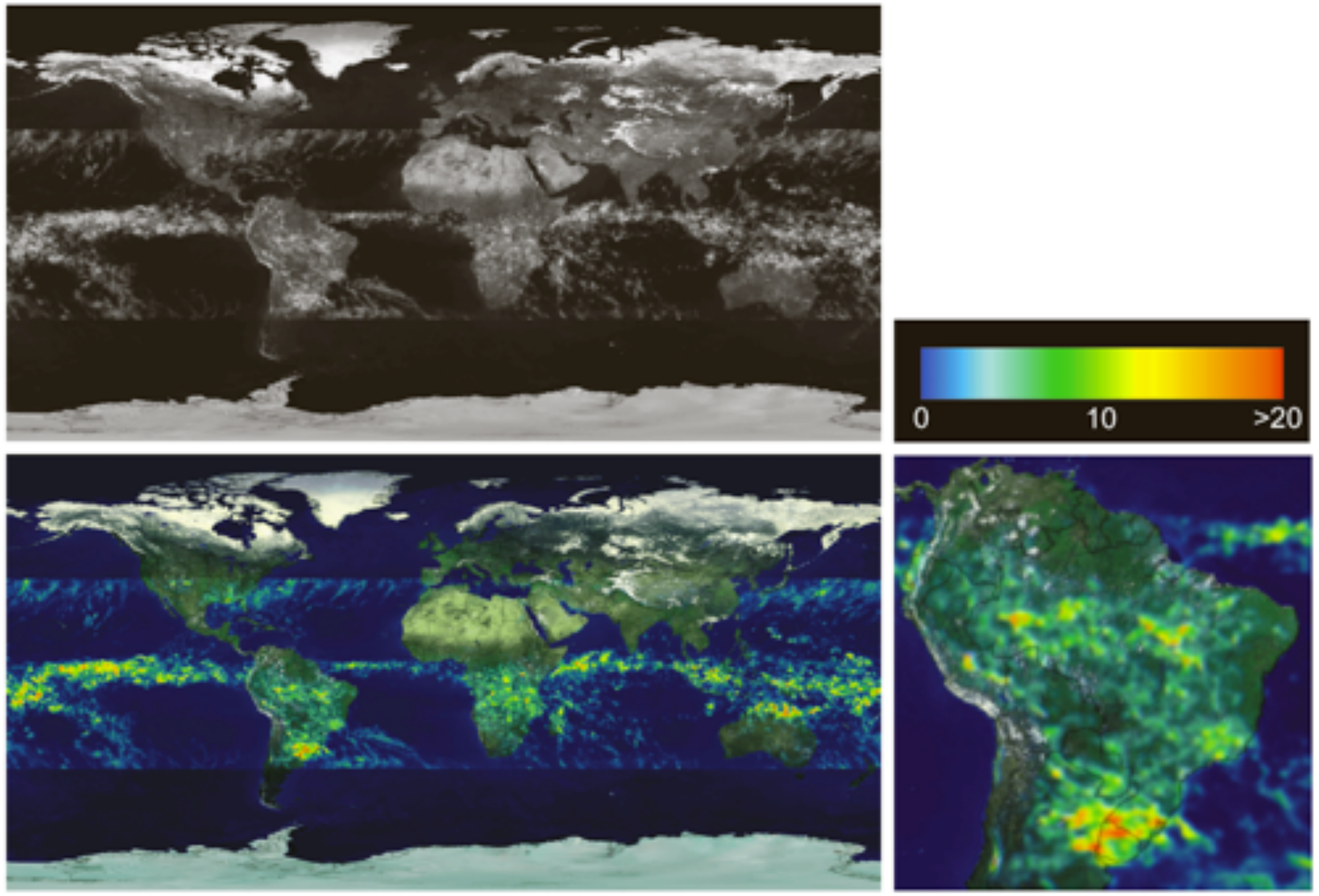




‘HSV’ (HSI) colormap

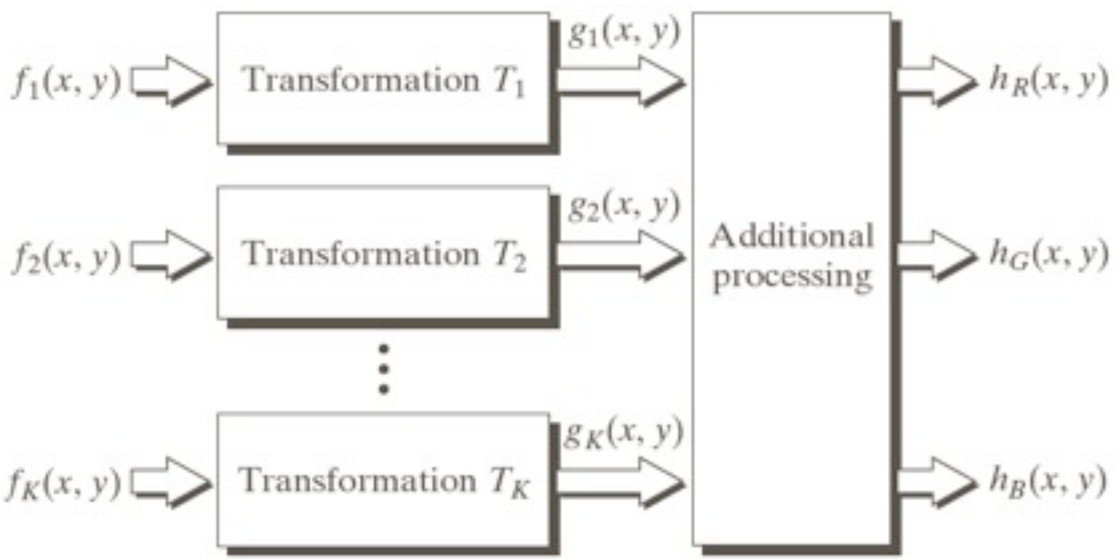
NB. Since the hue is an angle,
this map is very useful for representing
periodic images



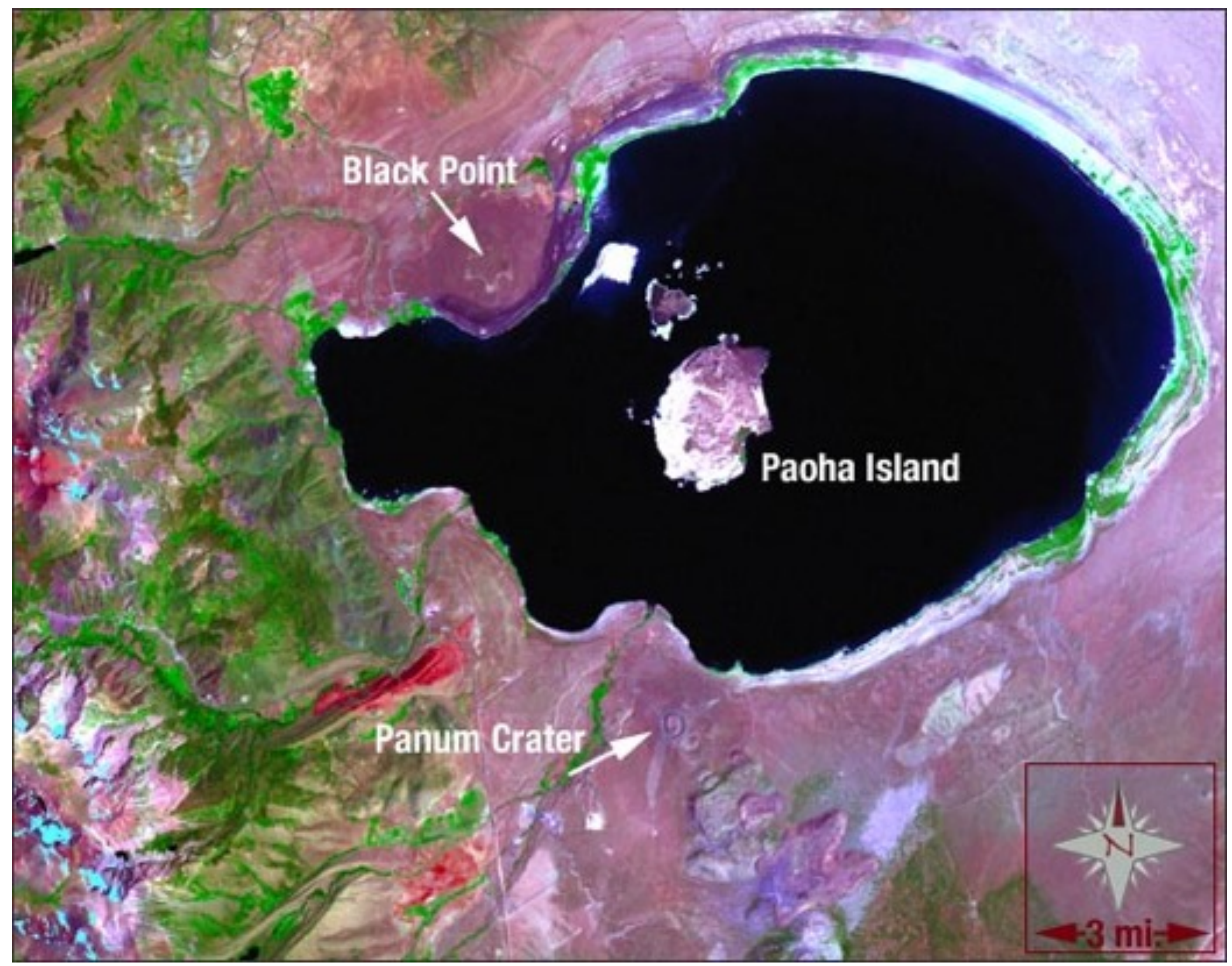


a b
c d

FIGURE 6.22 (a) Gray-scale image in which intensity (in the lighter horizontal band shown) corresponds to average monthly rainfall. (b) Colors assigned to intensity values. (c) Color-coded image. (d) Zoom of the South American region. (Courtesy of NASA.)

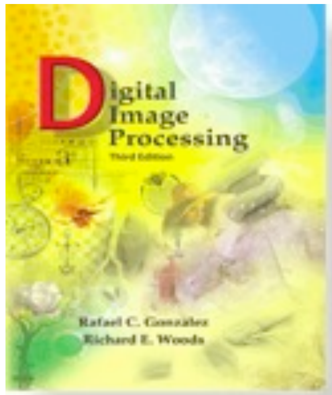


Mono Lake, California



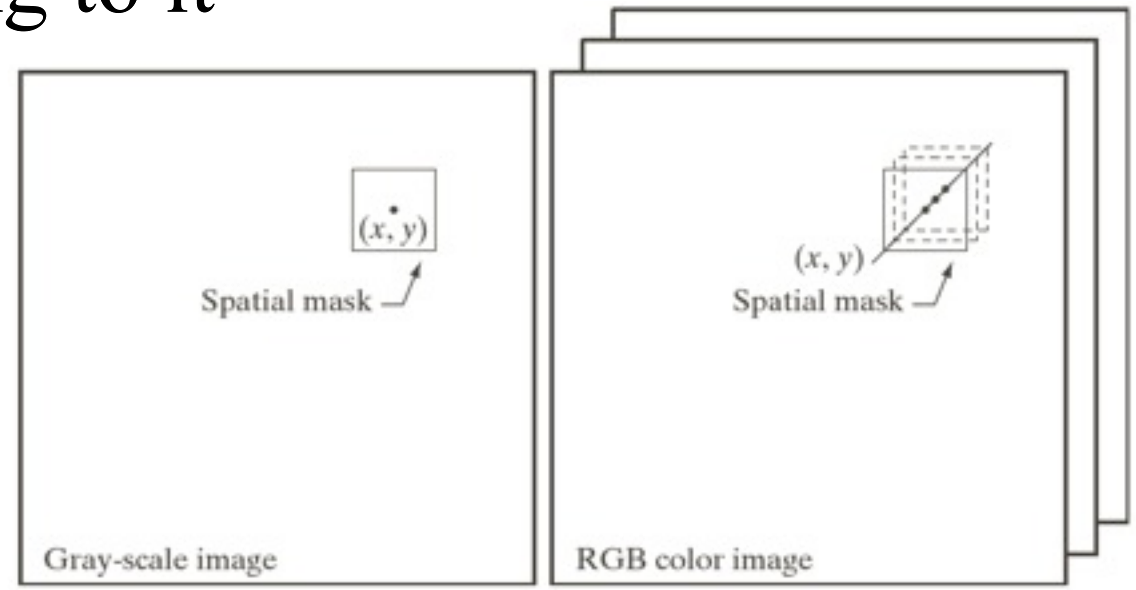
Nasa/Landsat

This Landsat 7 image of Mono Lake was acquired on July 27, 2000. This image is a false-color composite made from the mid-infrared, near-infrared, and green spectral channels of the Landsat 7 ETM+ sensor – it also includes the panchromatic 15-meter band for spatial sharpening purposes. In this image, the waters of Mono Lake appear a bluish-black and vegetation appears bright green. You will notice the vegetation to the west of the lake and following the tributaries that enter the lake.



Full color processing
when we have a full color
image and we apply processing to it

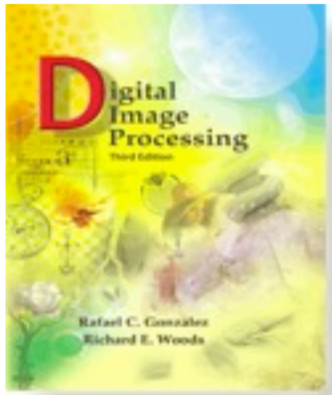
Most of the techniques
studies earlier, can be
applied
to color images.



a b

FIGURE 6.29 Spatial masks for gray-scale and RGB color images.

Consider each of the
color component as a grayscale image:
It is possible to:
- apply the known procedure to the each of the grayscale image
- think about each color point as a vector and apply a method to a
vector image rather than a scalar image.

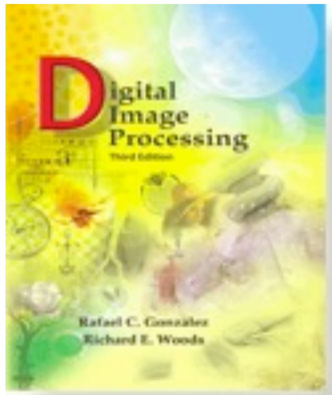


Basic intensity manipulations:

- Color intensity transformations and complement

- Color slicing/segmentation

- Color edge detection

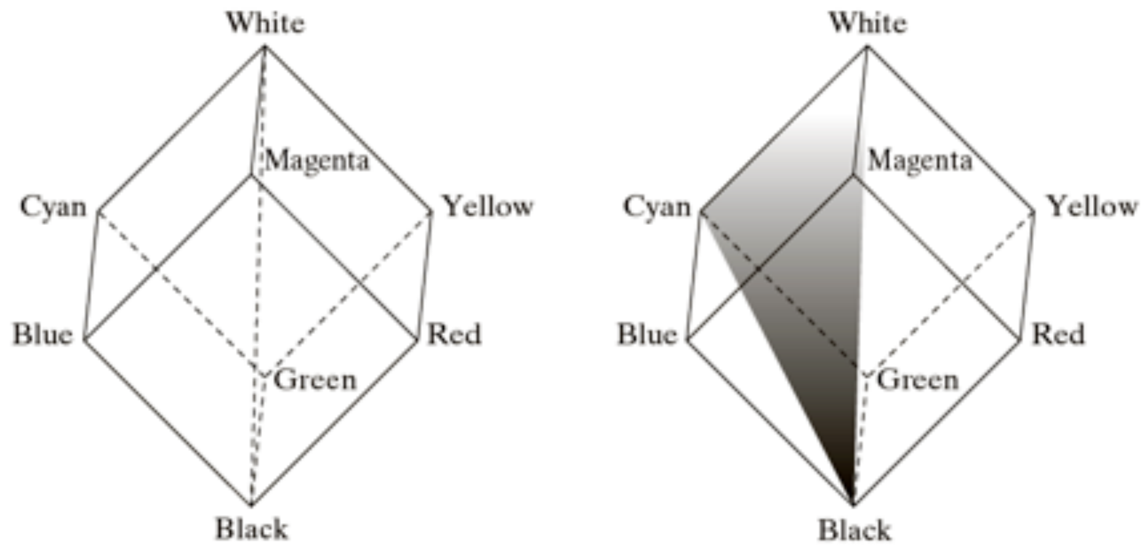


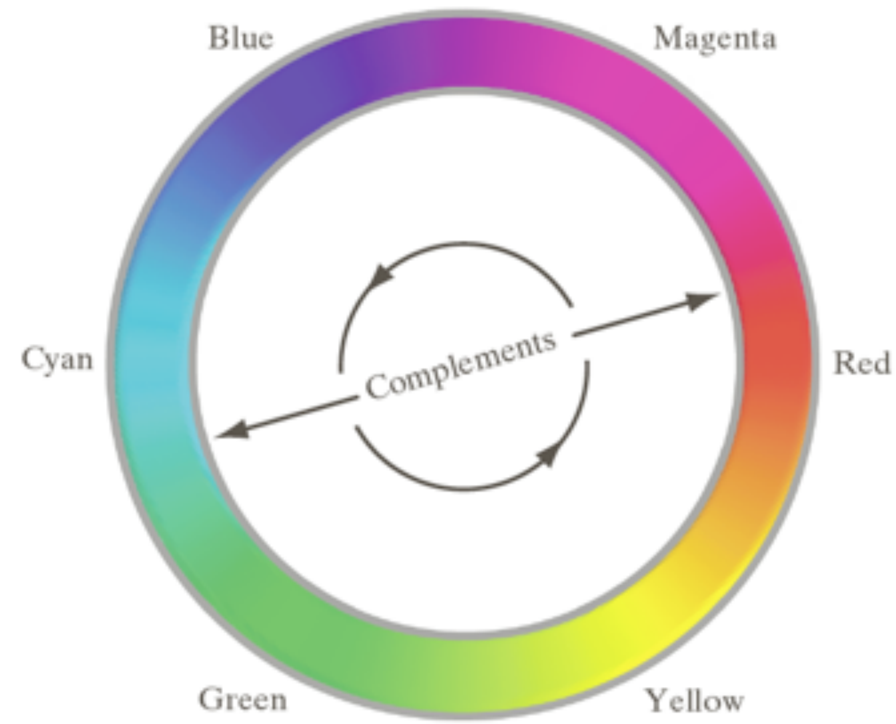
Color transformations

Color complements:

RGB <=> CMY

C		1-R
M	=	1-G
Y		1-B





Spatial techniques for smoothing and sharpening

Color smoothing:

$$\bar{\mathbf{c}}(x,y) = \frac{1}{K}\sum_{(s,t)\in S_{xy}} \mathbf{c}(x,y) = \begin{bmatrix} \frac{1}{K}\sum_{(s,t)\in S_{xy}} R(x,y) \\ \frac{1}{K}\sum_{(s,t)\in S_{xy}} G(x,y) \\ \frac{1}{K}\sum_{(s,t)\in S_{xy}} B(x,y) \end{bmatrix}$$

Using the Laplacian:

$$\nabla^2 \mathbf{c}(x,y) = \begin{bmatrix} \nabla^2 R(x,y) \\ \nabla^2 G(x,y) \\ \nabla^2 B(x,y) \end{bmatrix}$$

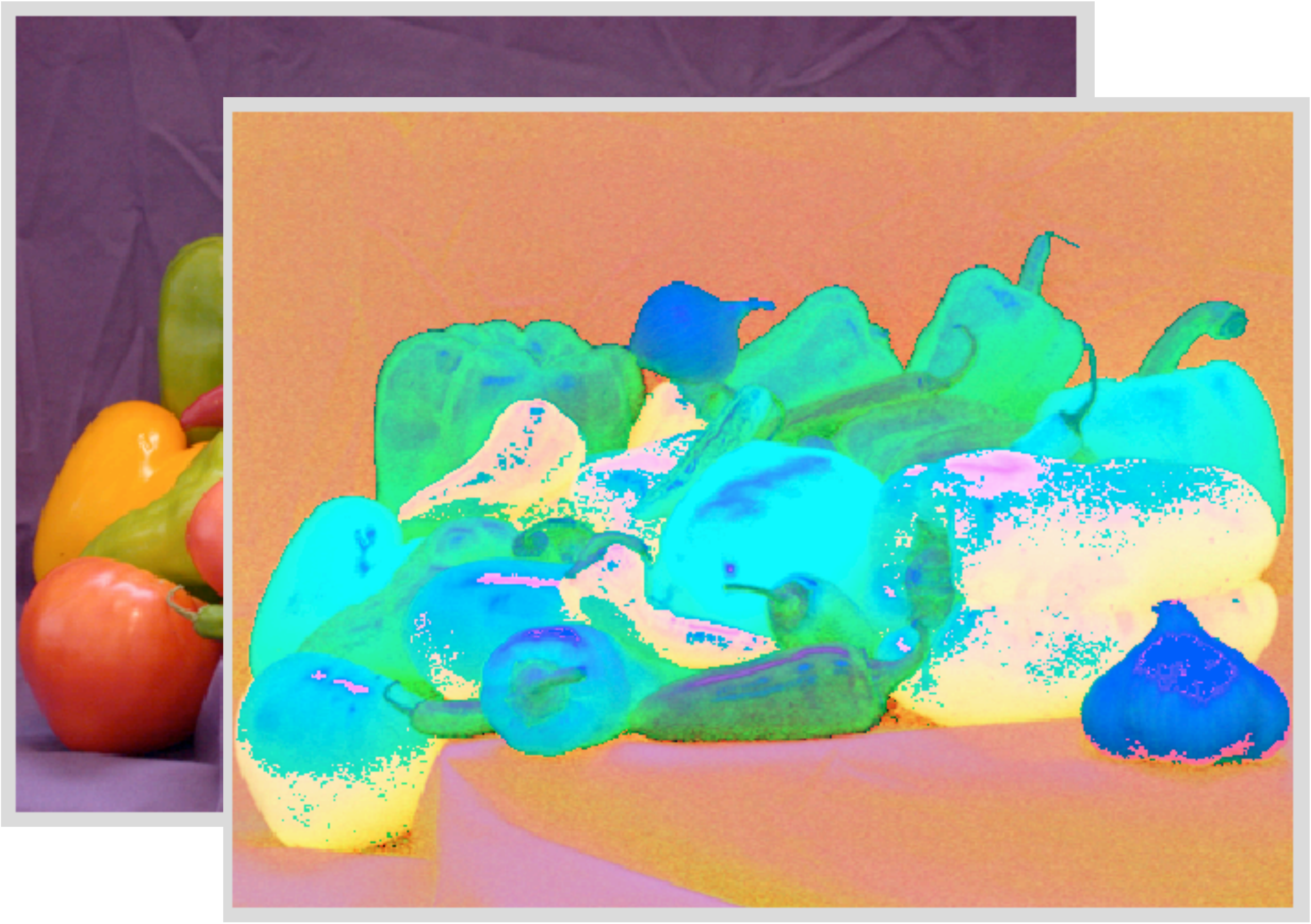
Similarly, one can apply the previous techniques using masks to each color channel R,G,B.

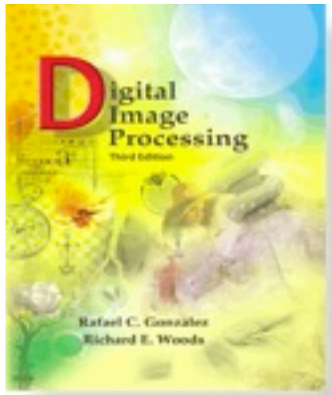
Segmentation based on color.

- Recall that a grayscale image can be segmented into two parts by thresholding.

- Similarly color images may be segmentet by one ore more thresholds in one of the color channels

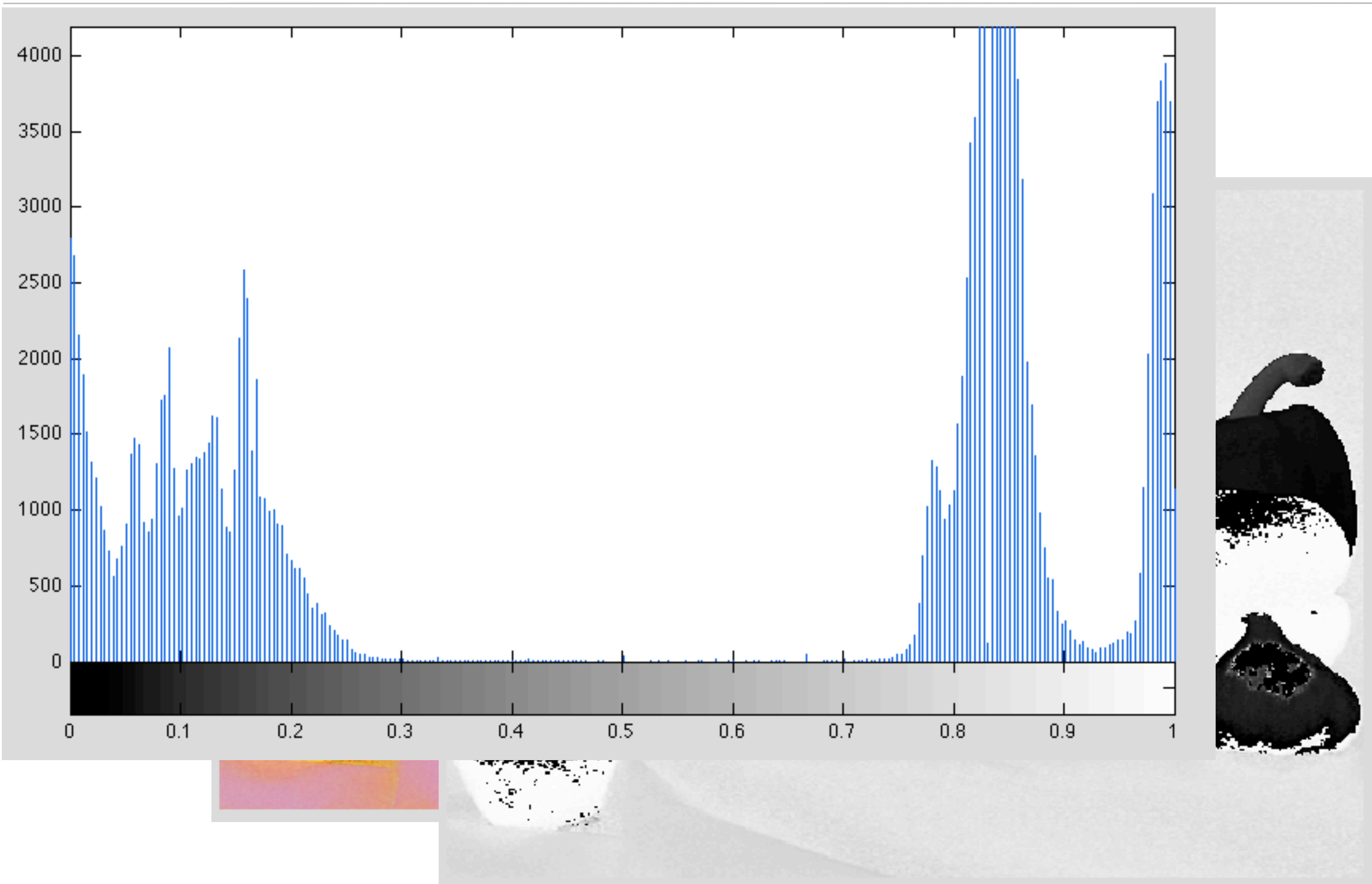
- In some applications, thresholding the **hue**-value may give good results.





Chapter 6

Color Image Processing



Segmentation based on color.

If we want to consider all the channels at the same time, a color is associated to a vector **a**. We must define an "enclosing" distance from **a**. This can be done using different distances.

$$D(\mathbf{z},\mathbf{a}) \leq D_0$$

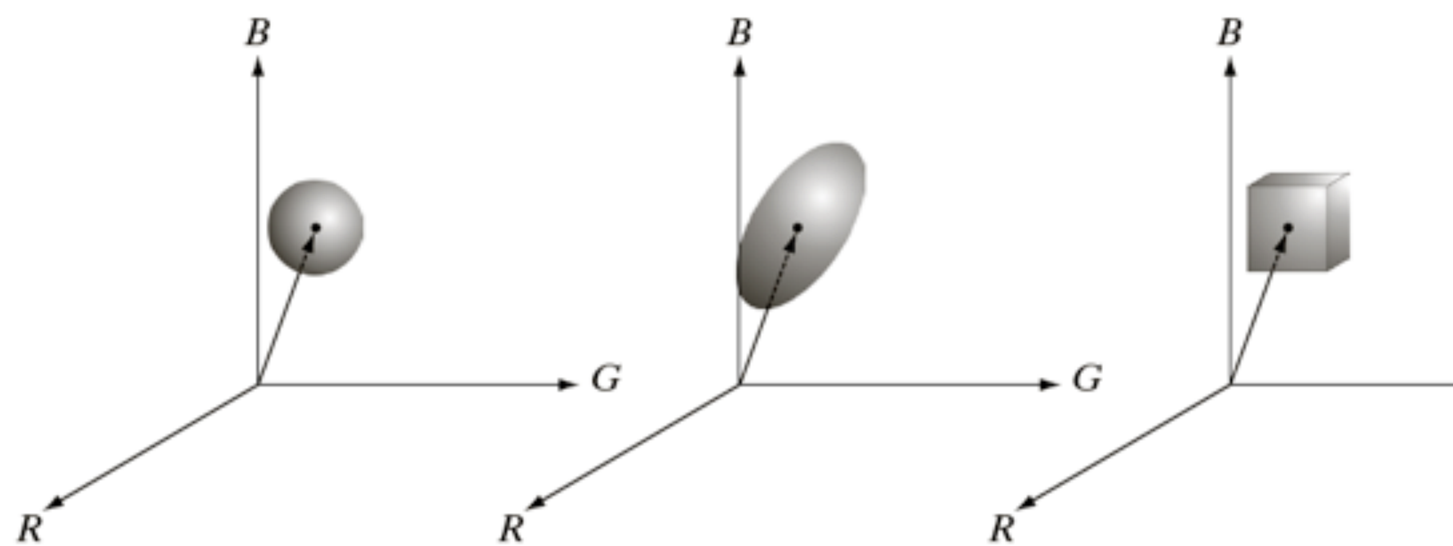
Euclidean norm

$$\|\mathbf{z}-\mathbf{a}\| = \sqrt{(\mathbf{z}-\mathbf{a})^T(\mathbf{z}-\mathbf{a})},$$

Anisotropic Euclidean norm

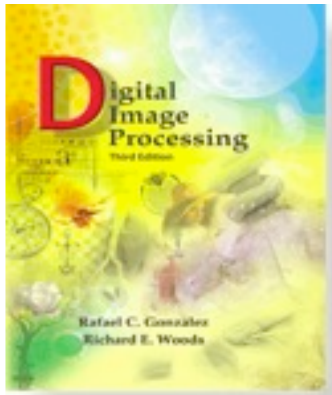
$$D(\mathbf{z},\mathbf{a}) = \sqrt{(\mathbf{z}-\mathbf{a})^T C^{-1}(\mathbf{z}-\mathbf{a})},$$

infinity norm



a b c
FIGURE 6.43 Three approaches for enclosing data regions for RGB vector segmentation.

$$\max\{|z_R - a_R|, |z_G - a_G|, |z_B - a_B|\} \leq D_0$$



Segmentation based on color.

If we want to consider all the channels at the same time, a color is associated to a vector **a**. We must define an "enclosing" distance from **a**. This can be done using different distances.

$D(\mathbf{z},\mathbf{a}) \leq D_0$

Euclidean norm

$\|\mathbf{z}-\mathbf{a}\| = \sqrt{(\mathbf{z}-\mathbf{a})^T(\mathbf{z}-\mathbf{a})},$

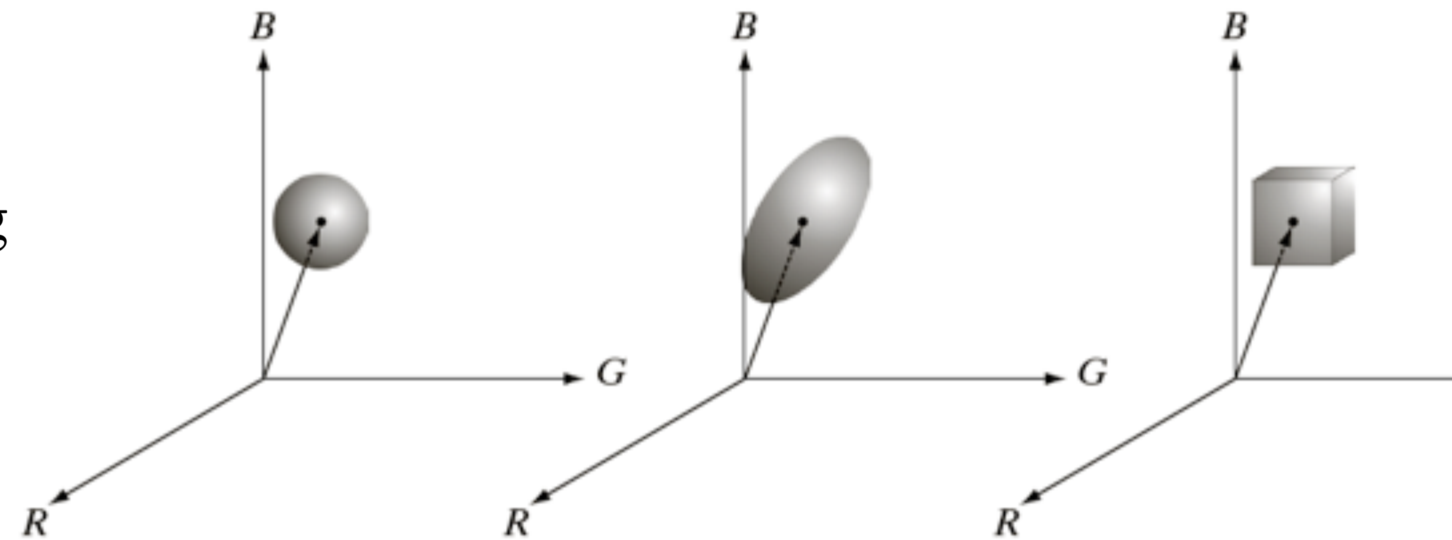
Anisotropic Euclidean norm

$D(\mathbf{z},\mathbf{a}) = \sqrt{(\mathbf{z}-\mathbf{a})^T C^{-1}(\mathbf{z}-\mathbf{a})},$

infinity norm

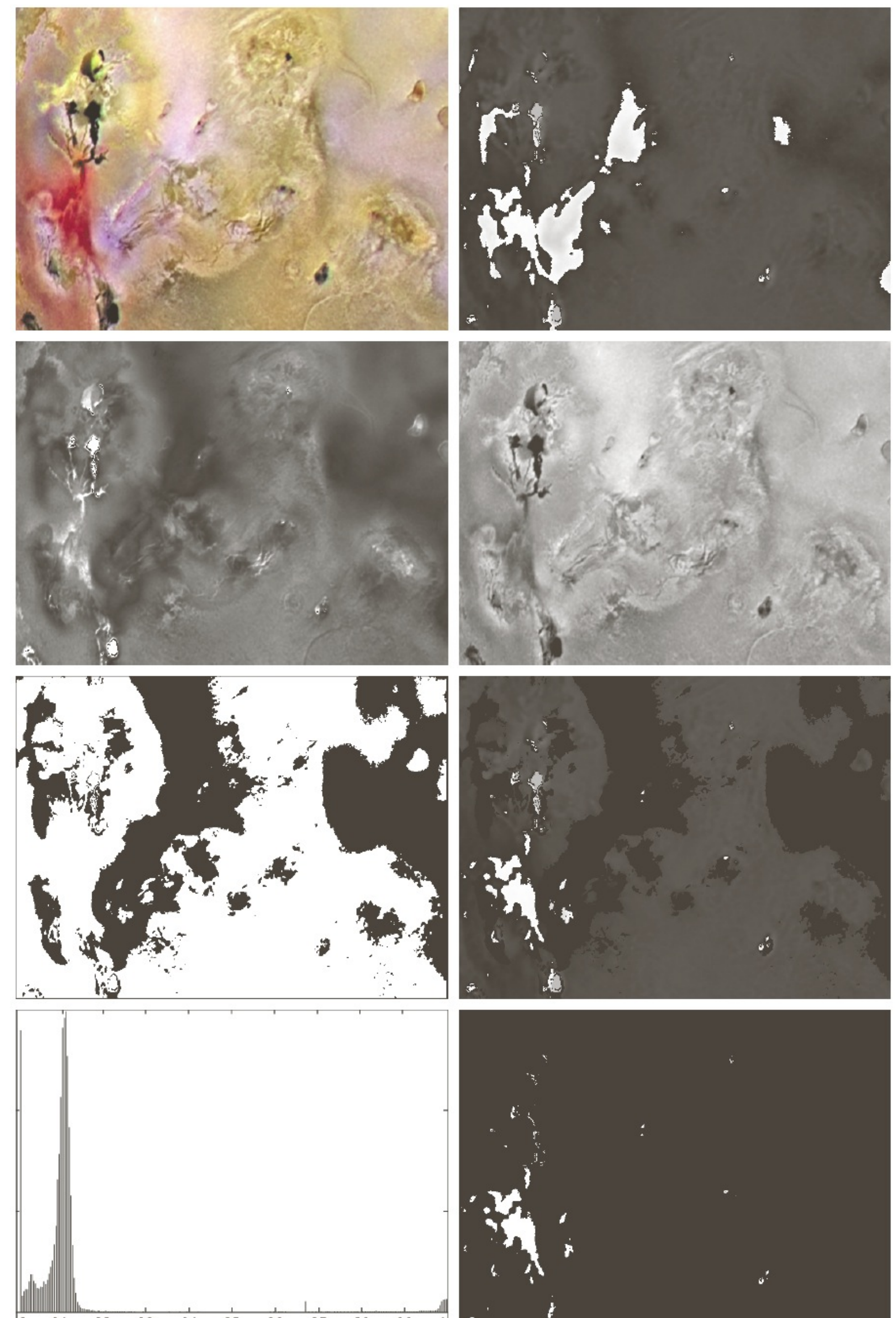
$\max\{|z_R - a_R|, |z_G - a_G|, |z_B - a_B|\} \leq D_0$

NB. The same can be done using HSV or other colormaps



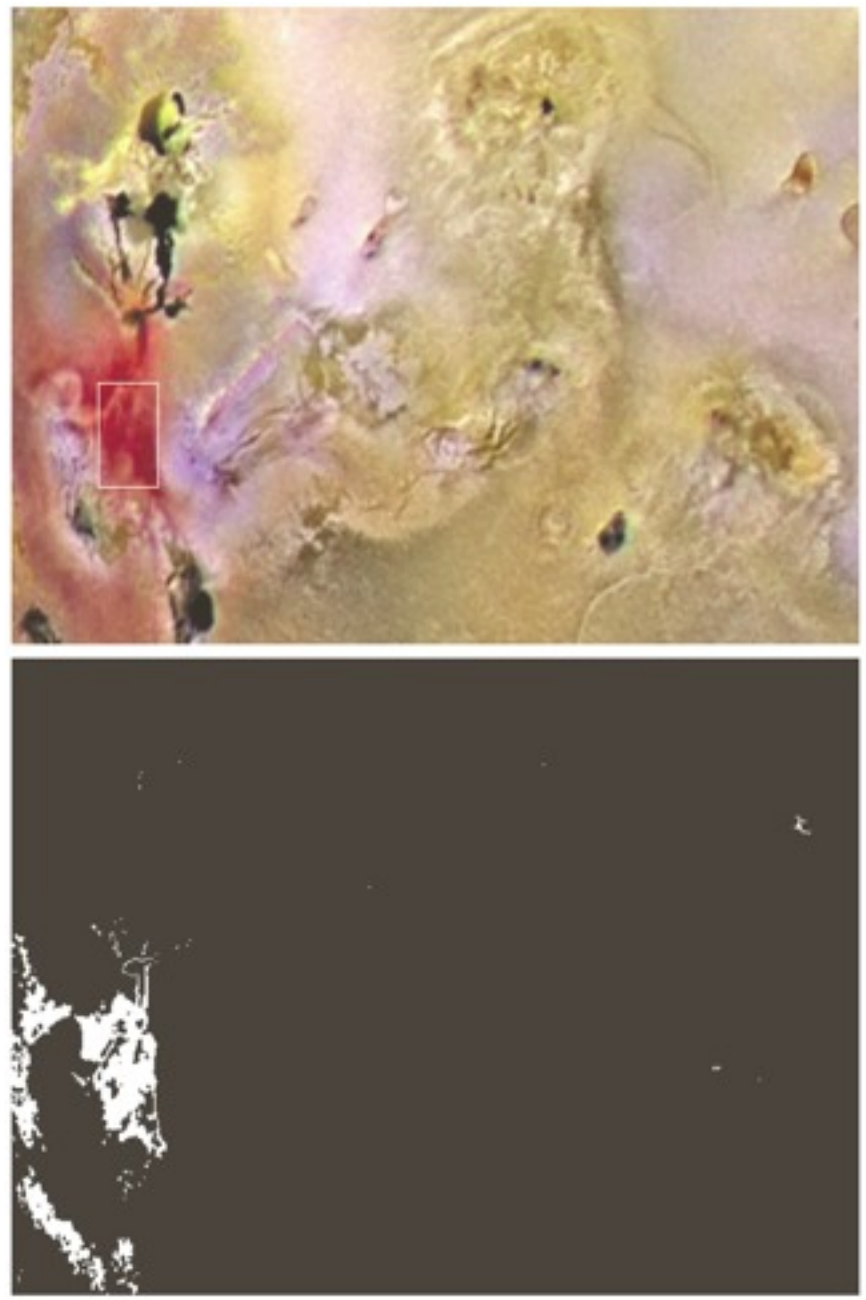
a b c

FIGURE 6.43 Three approaches for enclosing data regions for RGB vector segmentation.



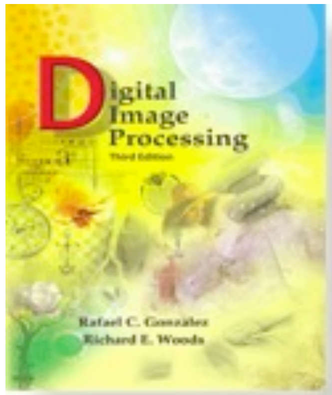
a b
c d
e f
g h

FIGURE 6.42 Image segmentation in HSI space. (a) Original. (b) Hue. (c) Saturation. (d) Intensity. (e) Binary saturation mask (black = 0). (f) Product of (b) and (e). (g) Histogram of (f). (h) Segmentation of red components in (a).



a
b

FIGURE 6.44 Segmentation in RGB space. (a) Original image with colors of interest shown enclosed by a rectangle. (b) Result of segmentation in RGB vector space. Compare with Fig. 6.42(h).



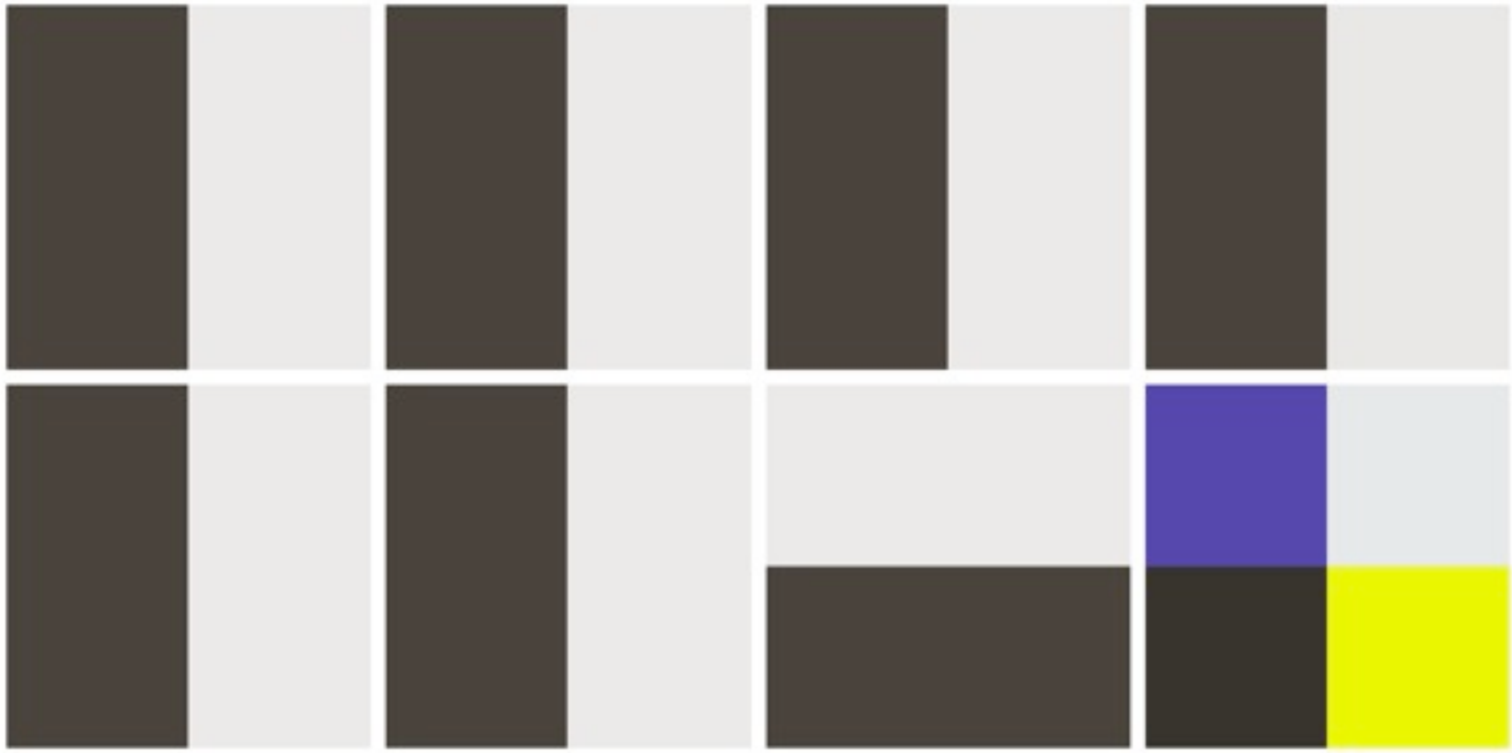
Color edge detection and color image gradient

The gradient is defined for scalar functions

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

The application of the gradient to each of the RGB channels can yield erroneous results.

We need a manner to extend the definition to vector functions (Di Zenzo, 1986).



a	b	c	d
e	f	g	h

FIGURE 6.45 (a)–(c) R, G, and B component images and (d) resulting RGB color image. (e)–(g) R, G, and B component images and (h) resulting RGB color image.

Denote by $\mathbf{r}, \mathbf{g}, \mathbf{b}$, the unit vectors on the RGB color space. Define

$$\begin{aligned}\mathbf{u} &= \frac{\partial R}{\partial x}\mathbf{r} + \frac{\partial G}{\partial x}\mathbf{g} + \frac{\partial B}{\partial x}\mathbf{b} \\ \mathbf{v} &= \frac{\partial R}{\partial y}\mathbf{r} + \frac{\partial G}{\partial y}\mathbf{g} + \frac{\partial B}{\partial y}\mathbf{b}\end{aligned}$$

Define also

$$\begin{aligned}g_{xx} &= \|\mathbf{u}\|^2 = \mathbf{u}^T\mathbf{u} \\ g_{yy} &= \|\mathbf{v}\|^2 = \mathbf{v}^T\mathbf{v} \\ g_{xy} &= \mathbf{u}^T\mathbf{v} = \langle \mathbf{u}, \mathbf{v} \rangle \qquad \text{(scalar product)}\end{aligned}$$

The direction of maximum change:

$$\theta(x,y) = \frac{1}{2}\tan^{-1}\left(\frac{2g_{x,y}}{g_{xx} - g_{yy}}\right)$$

Rate of change:

$$F_\theta(x,y) = \left(\frac{1}{2}(g_{xx} + g_{yy}) + (g_{xx} - g_{yy})\cos 2\theta + 2g_{xy}\sin 2\theta\right)^{\frac{1}{2}}$$

Note that the direction is defined up to a 90-degree value. This in facts defines 2 directions (maximum change, minimum change).

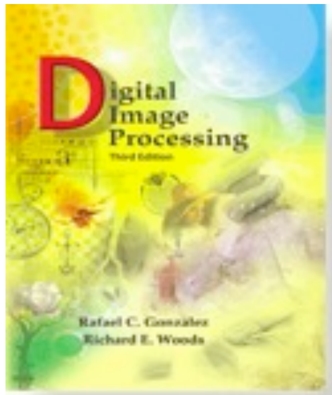
The g_{xx}, g_{yy}, g_{xy} can be computed using methods seen before (ex. Sobel masks).

a b
c d

FIGURE 6.46 (a) RGB image. (b) Gradient computed in RGB color vector space. (c) Gradients computed on a per-image basis and then added. (d) Difference between (b) and (c).

a b c

FIGURE 6.47 Component gradient images of the color image in Fig. 6.46. (a) Red component, (b) green component, and (c) blue component. These three images were added and scaled to produce the image in Fig. 6.46(c).



a b
c d

FIGURE 6.48 (a)–(c) Red, green, and blue component images corrupted by additive Gaussian noise of mean 0 and variance 800. (d) Resulting RGB image. [Compare (d) with Fig. 6.46(a).]

a b c

FIGURE 6.49 HSI components of the noisy color image in Fig. 6.48(d). (a) Hue. (b) Saturation. (c) Intensity.

a b
c d

FIGURE 6.50 (a) RGB image with green plane corrupted by salt-and-pepper noise. (b) Hue component of HSI image. (c) Saturation component. (d) Intensity component.

This concludes our study of color images and chapter 6 (if you wish to learn more, you can consult the book).

Next topic: image segmentation (Chapter 10).

Segmentation:
Is the process of decomposing the image into subregions/objects.

The level of detail (that defines the subregion or the object) depends on the application under consideration.

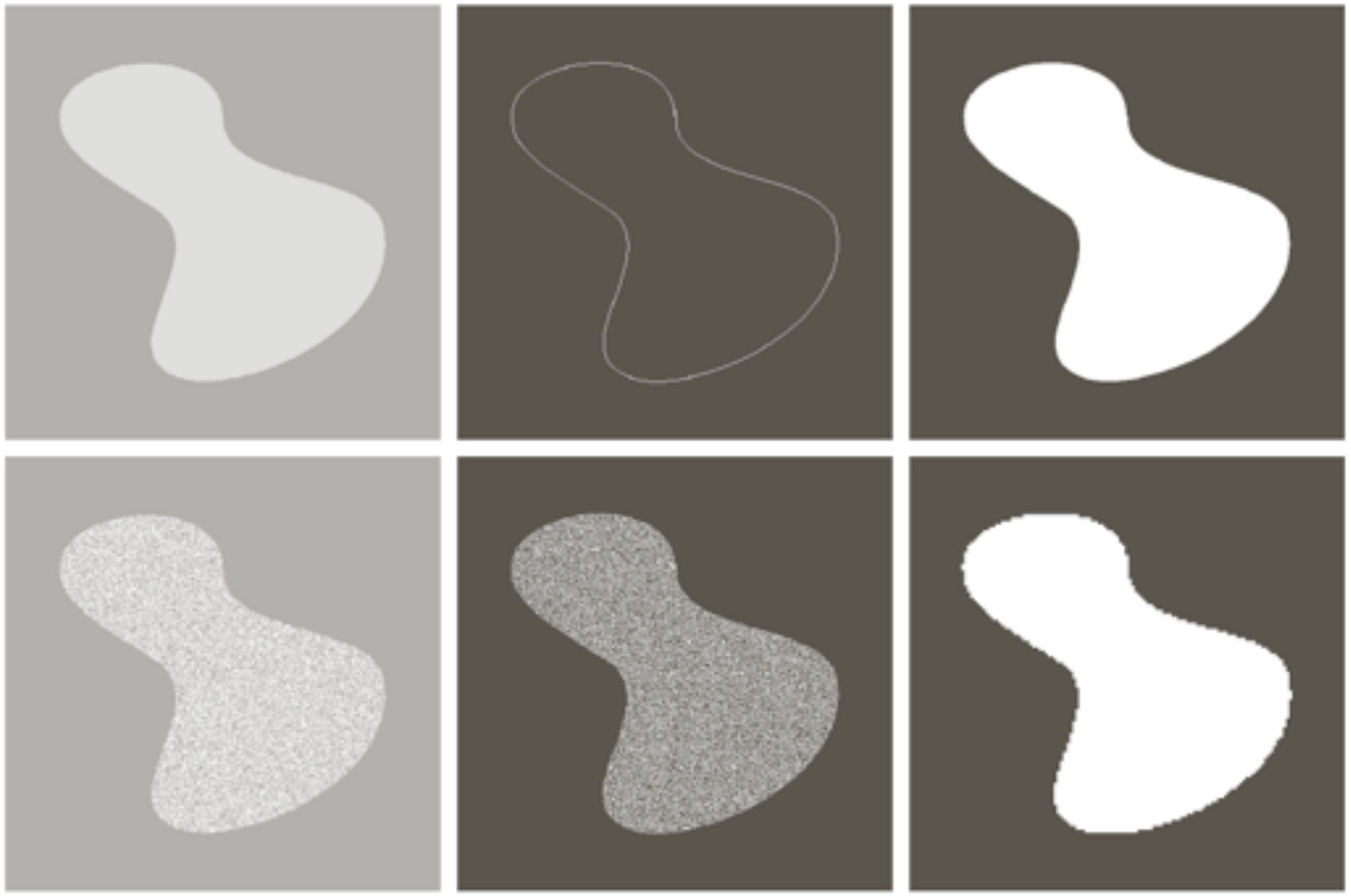
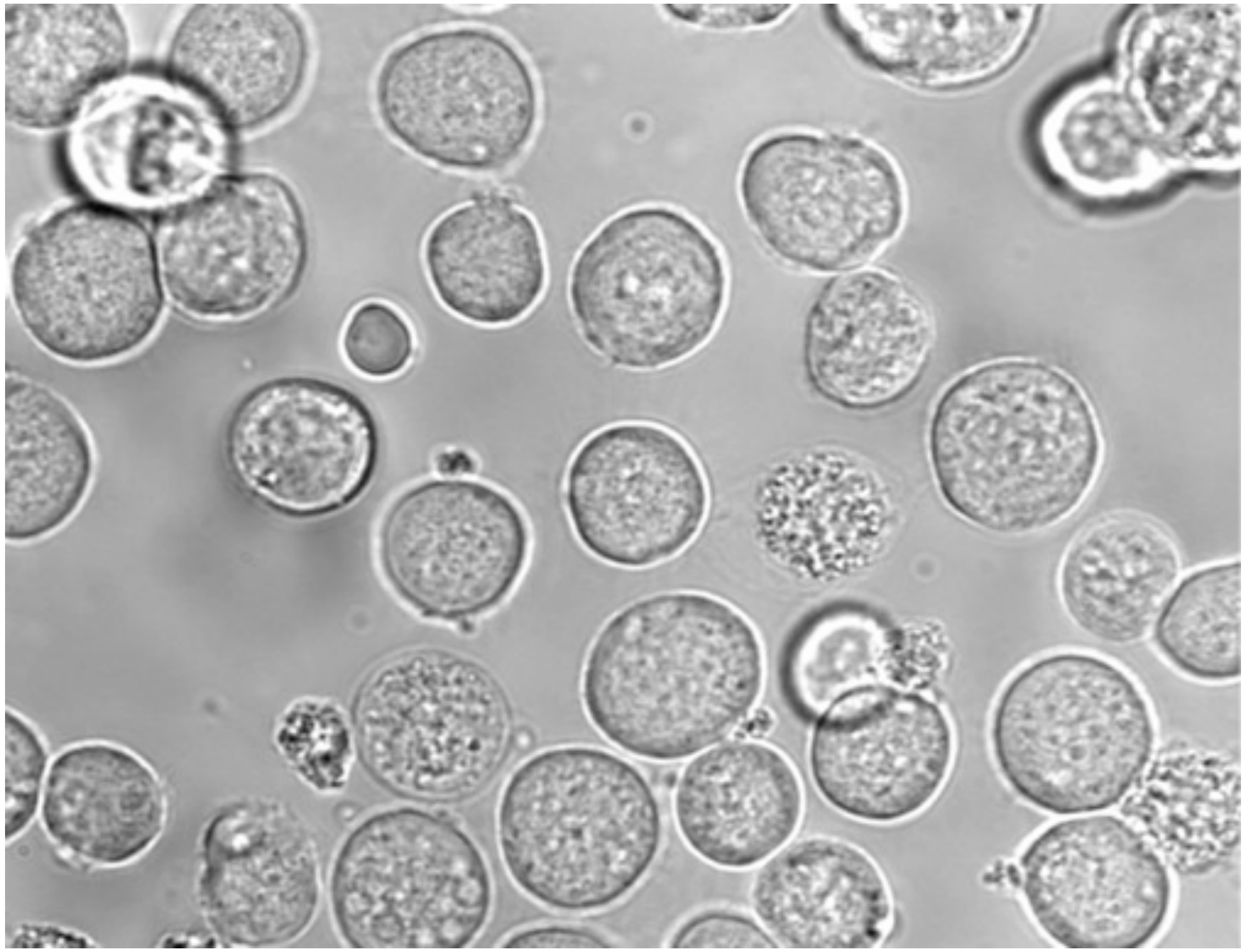
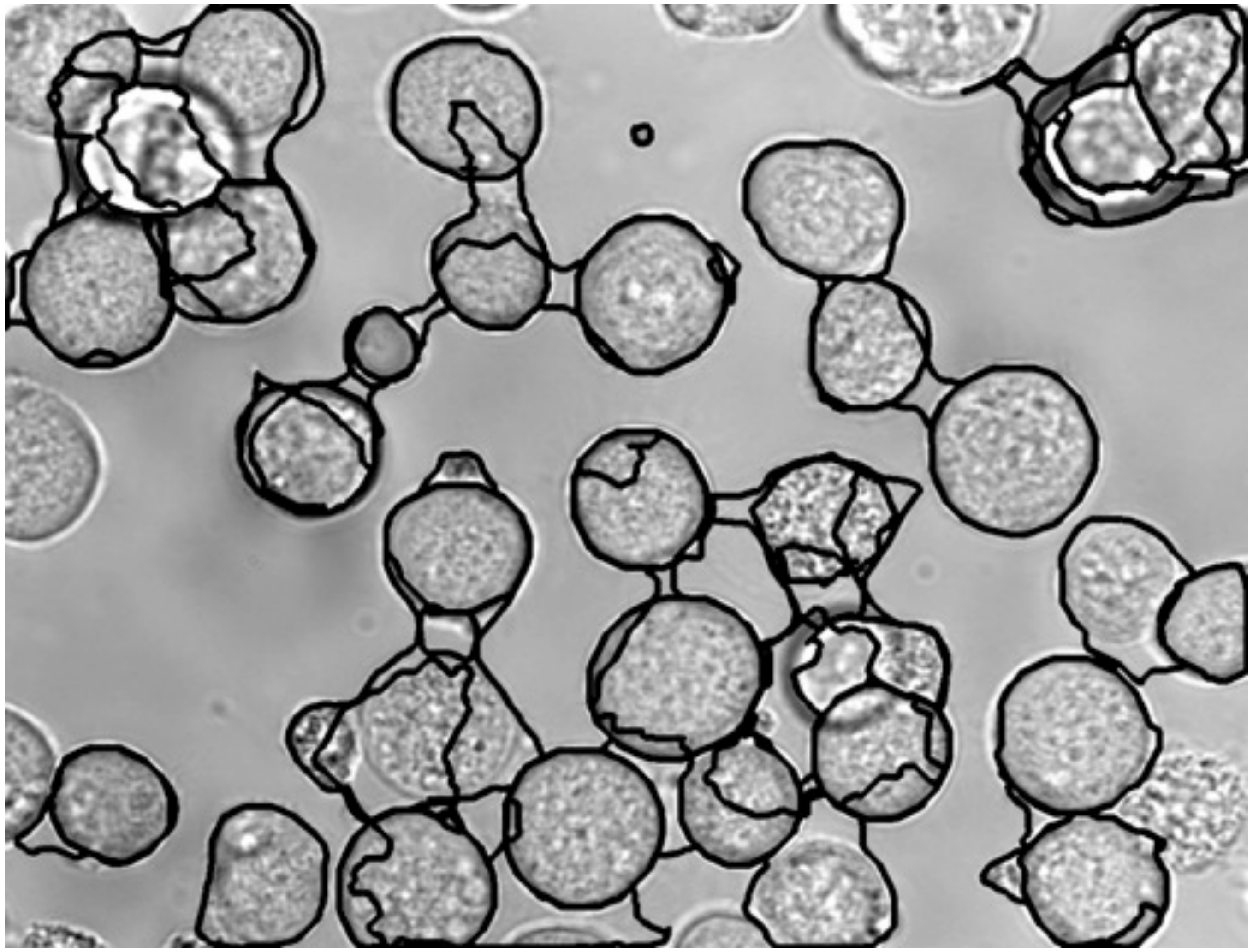


FIGURE 10.1 (a) Image containing a region of constant intensity. (b) Image showing the boundary of the inner region, obtained from intensity discontinuities. (c) Result of segmenting the image into two regions. (d) Image containing a textured region. (e) Result of edge computations. Note the large number 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.

Examples:



Examples:



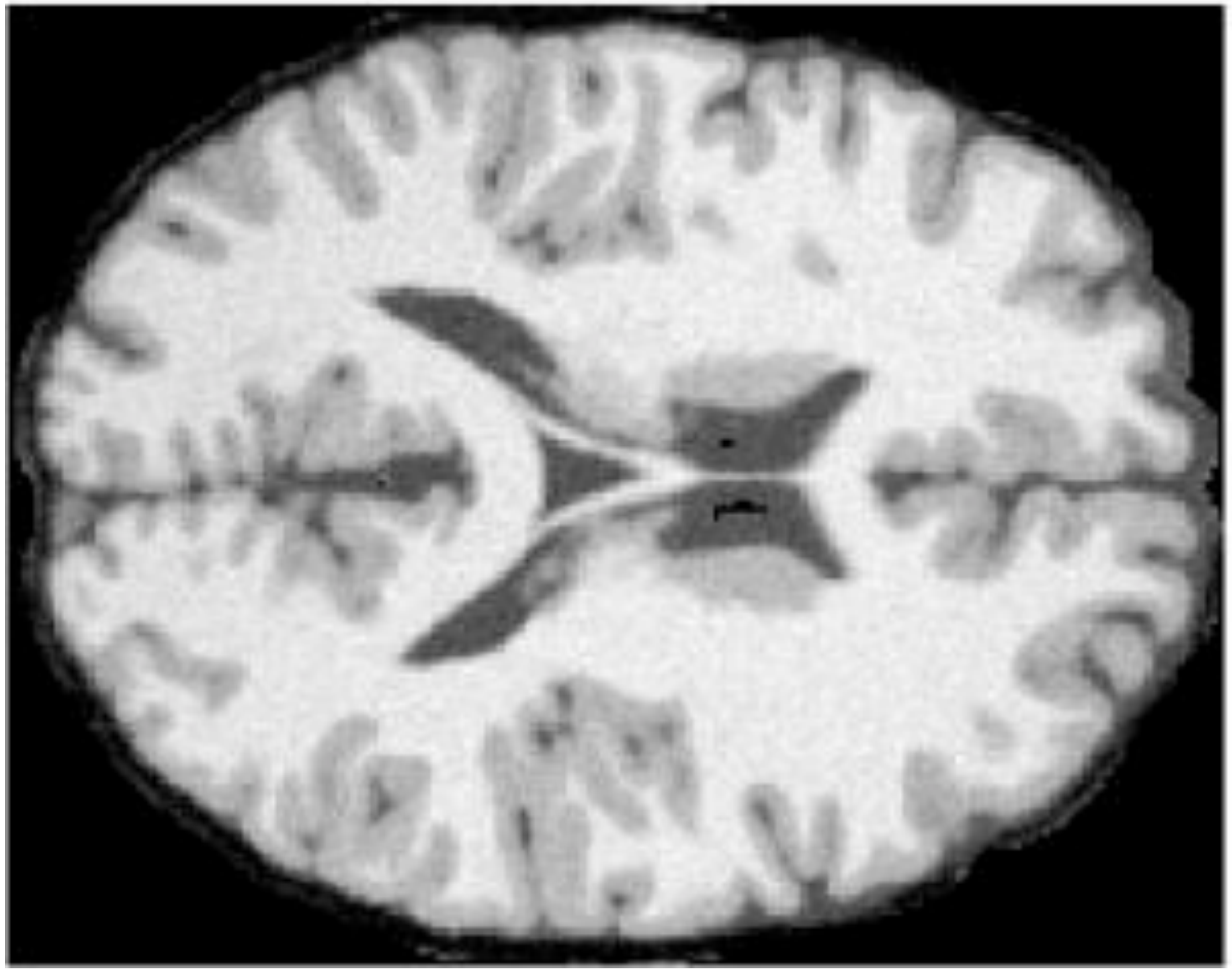
Examples:

5
10
15
20
25
30
35
40
45
50
55
20
40
60
80
100
120
140
160
180
200
220

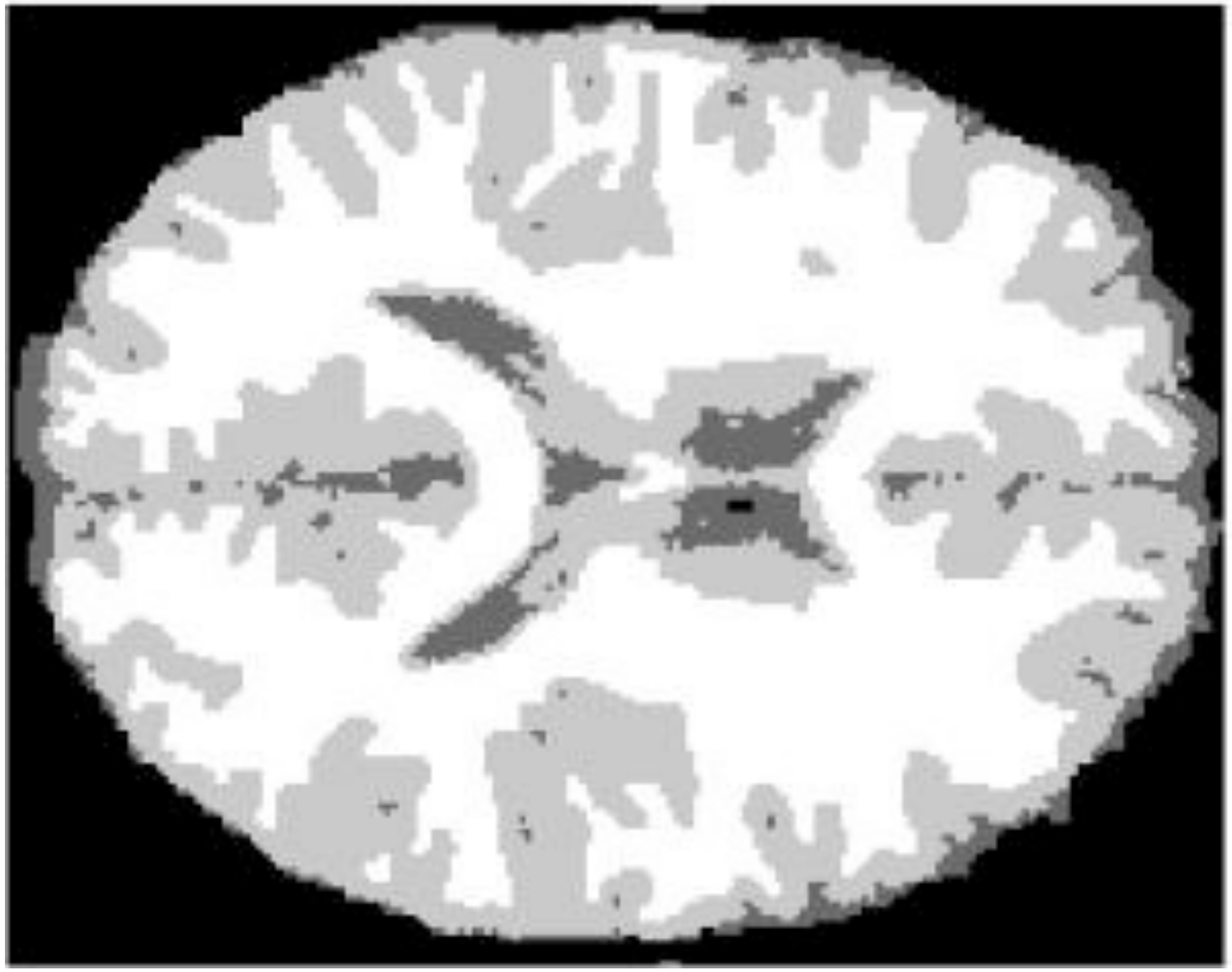
Examples:


BP:74388
5
10
15
20
25
30
35
40
45
50
55
20
40
60
80
100
120
140
160
180
200
220

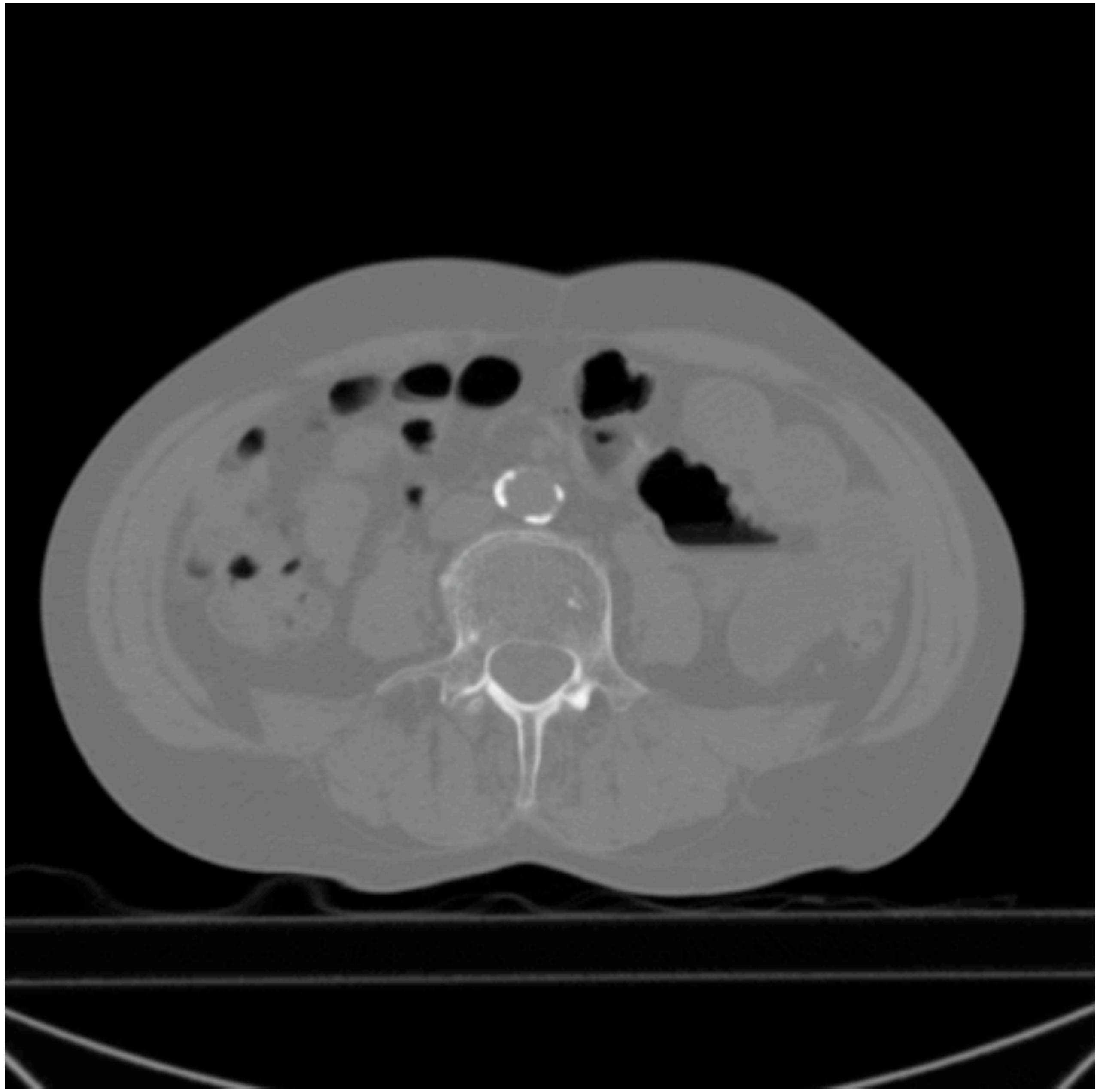
Examples:



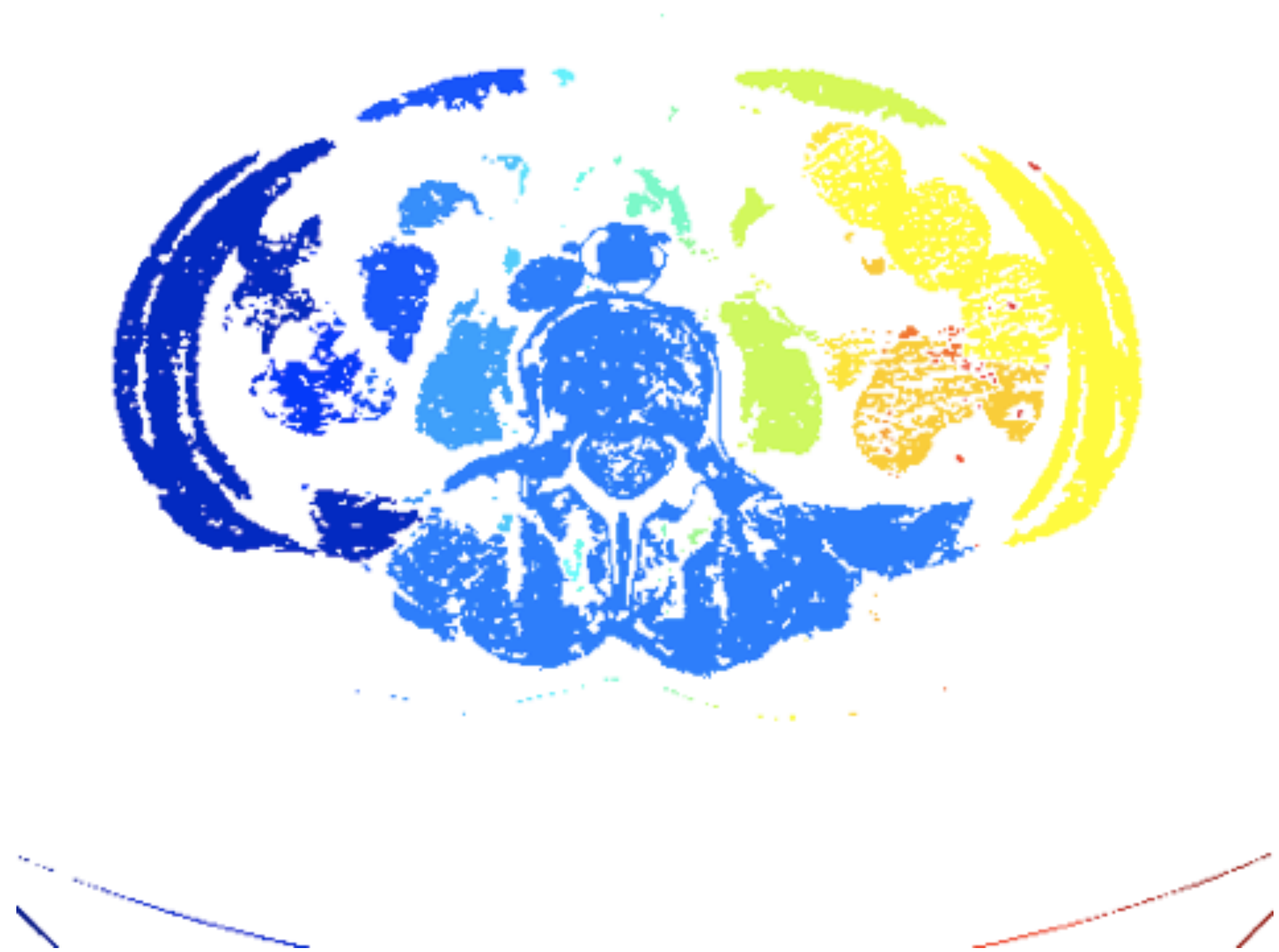
Examples:



Examples:



Examples:



We will consider 2 types of segmentation:

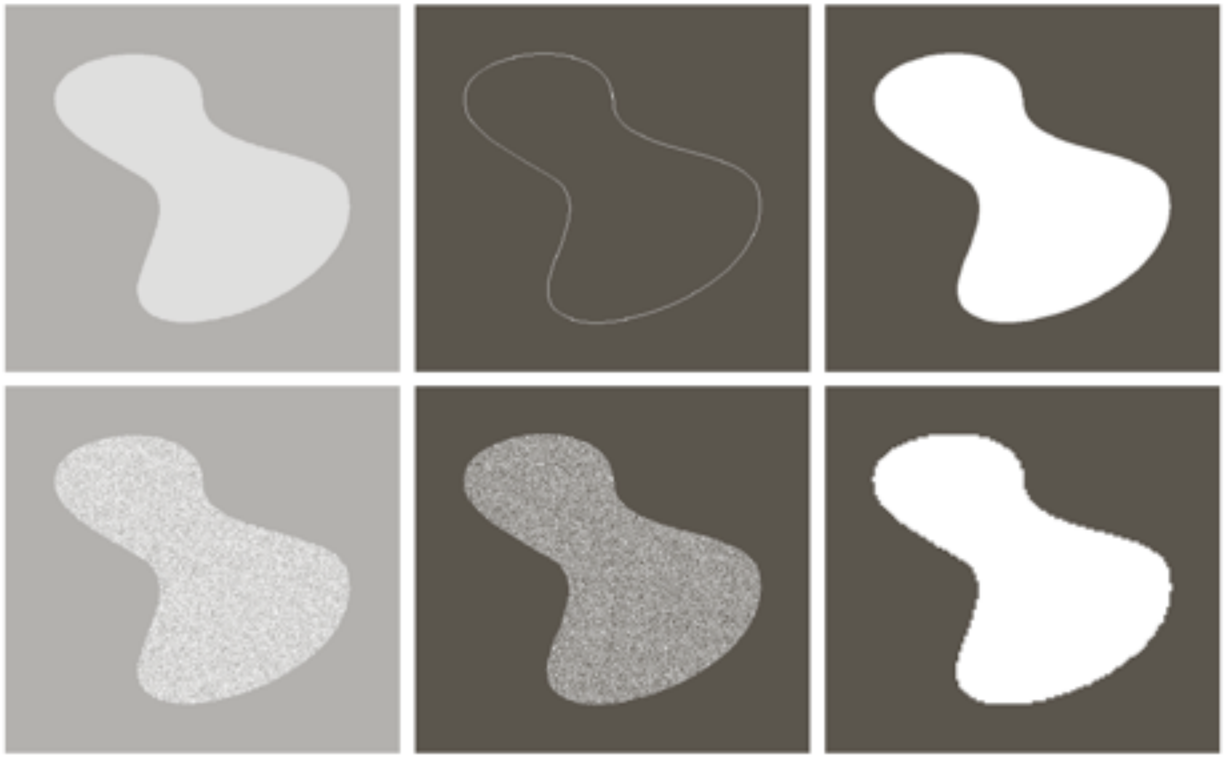
- based on intensity values (discontinuity)

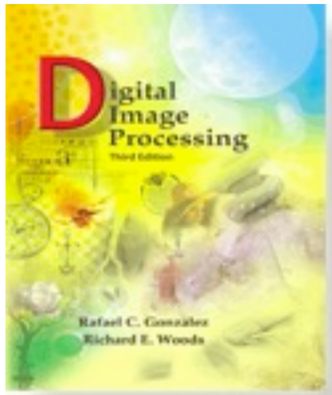
- based on similarity properties (according to some predefined criteria)

We will consider 2 types of segmentation:

- based on intensity values (discontinuity)

- based on similarity properties (according to some predefined criteria)





Foundaments

Denote by R the whole image region. The segmentation produces a partition of R in n subregions, $R_i, i = 1, \dots, n$, such that

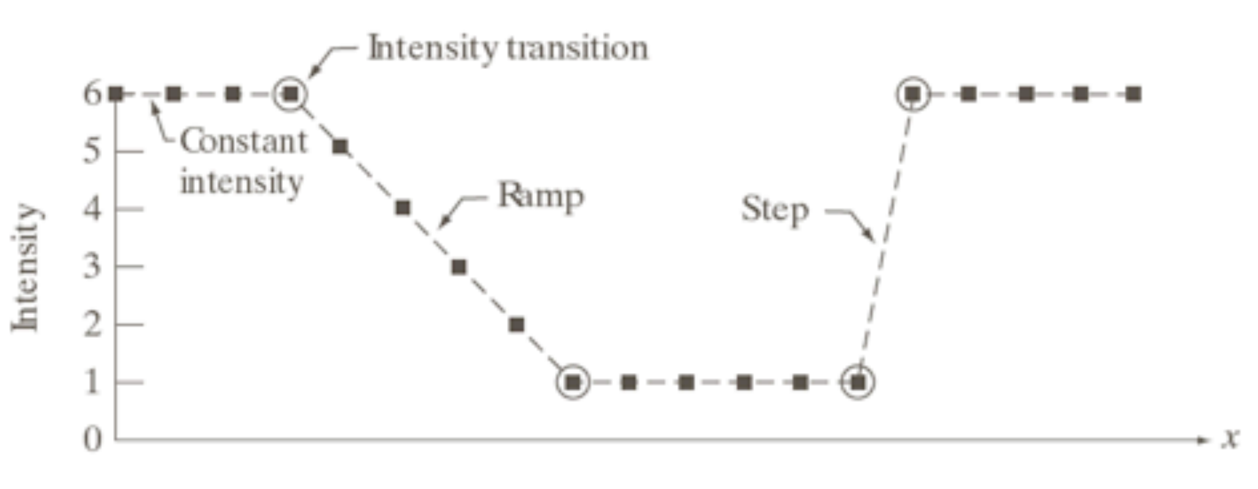
- $\cup_{i=1,\dots,n} R_i = R$
- $R_i \cap R_j = \emptyset$ for $i \neq i$ (regions are disjoint)
- $Q(R_i) =$ TRUE, for $i = 1, 2, \dots, n$ (Q is some condition that the regions must satisfy)
- $Q(R_i \cup R_j) =$ FALSE for adjacent regions

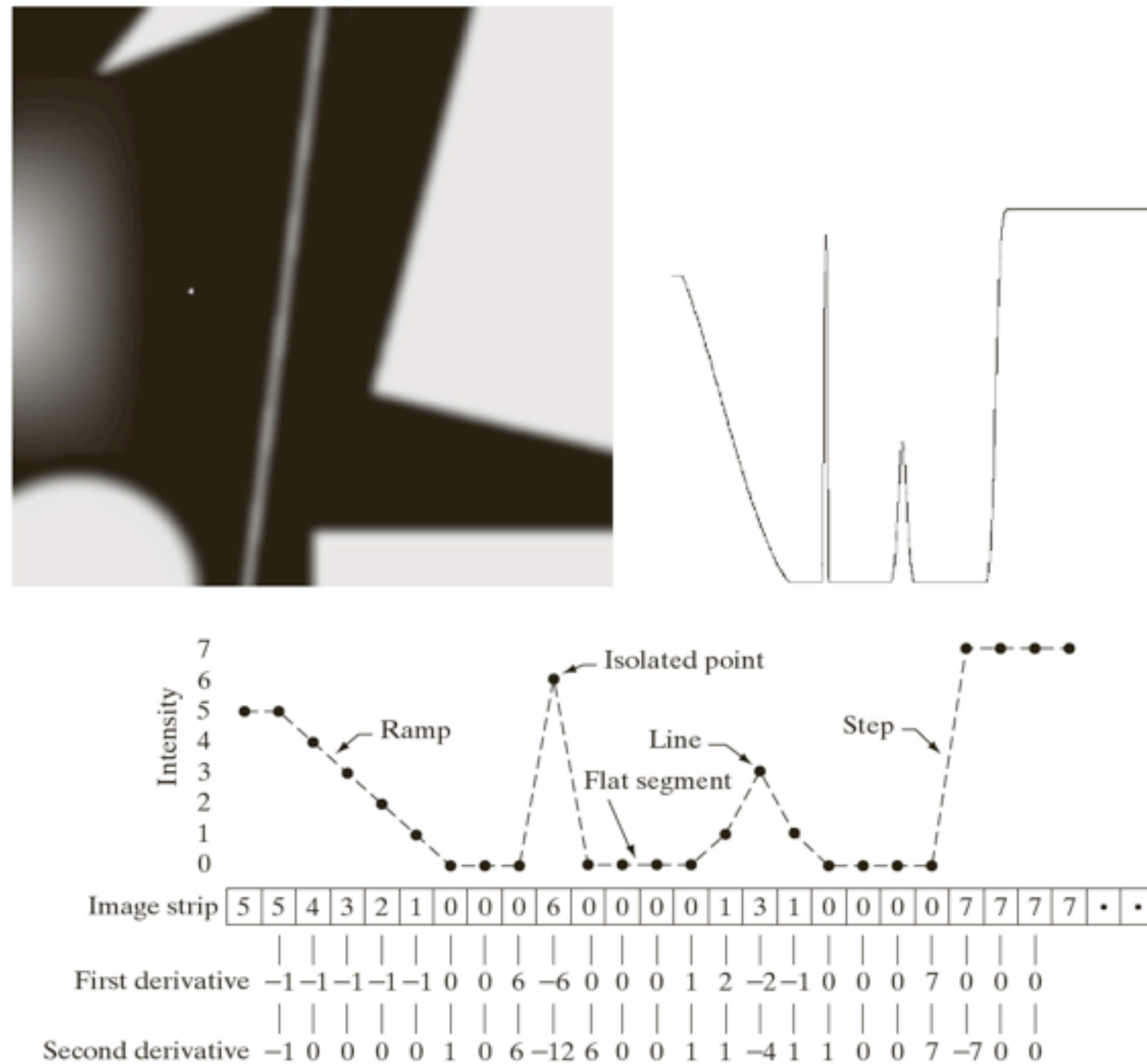
Intensity based methods:

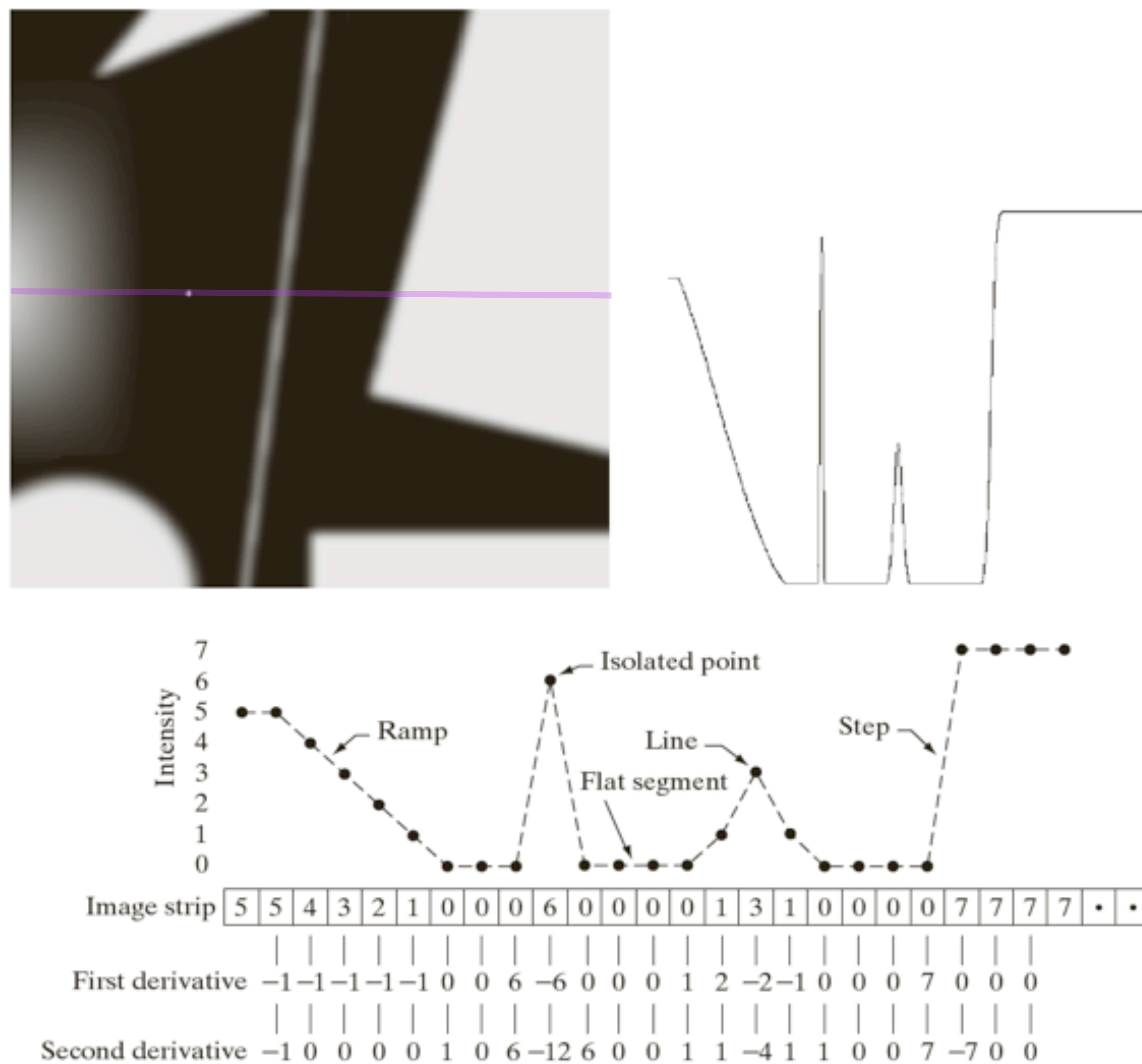
use the discontinuity of the signal (edges)

→ derivatives (first and second derivative)

Recall from Lecture 6 (on smoothing and sharpening)

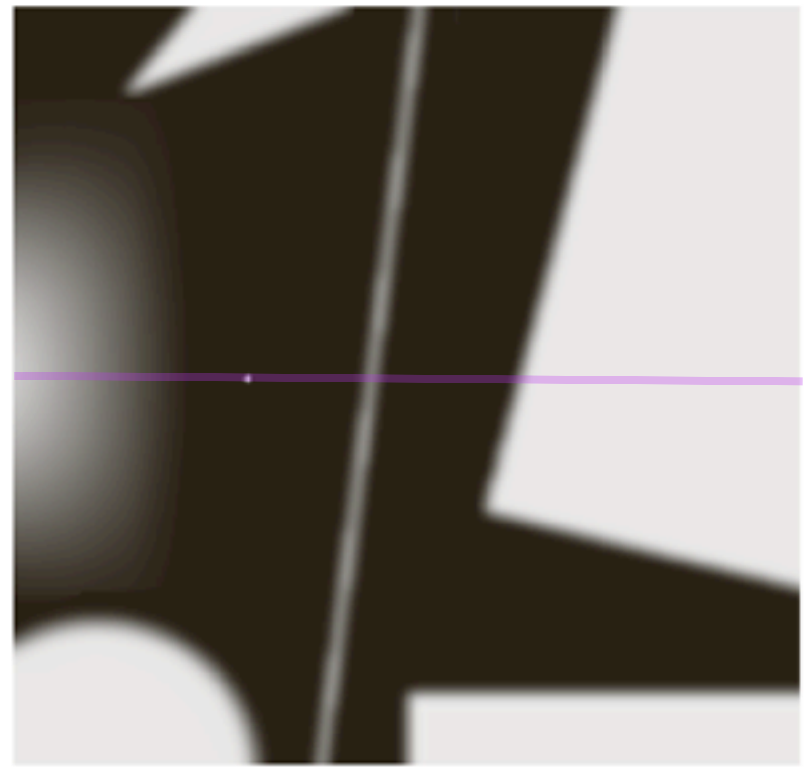


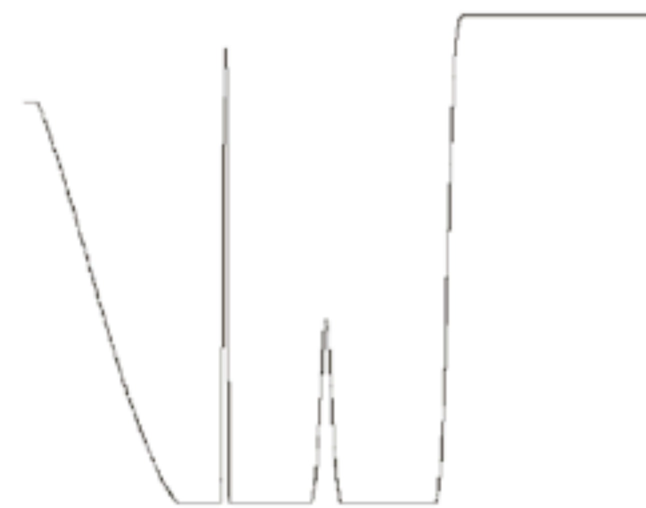

Intensity
7
6
5
4
3
2
1
0
Isolated point
Ramp
Line
Flat segment
Step
Image strip 5 5 4 3 2 1 0 0 0 6 0 0 0 0 1 3 1 0 0 0 0 7 7 7 7 • •
First derivative −1 −1 −1 −1 −1 0 0 6 −6 0 0 0 1 2 −2 −1 0 0 0 7 0 0 0
Second derivative −1 0 0 0 0 1 0 6 −12 6 0 0 1 1 −4 1 1 0 0 7 −7 0 0


Intensity
7
6
5
4
3
2
1
0
Isolated point
Ramp
Line
Flat segment
Step
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

Derivative operator:

1. Must be 0 on constant regions
2. Is nonzero on ramps/ steps
3. Is nonzero on the onset of ramps/steps





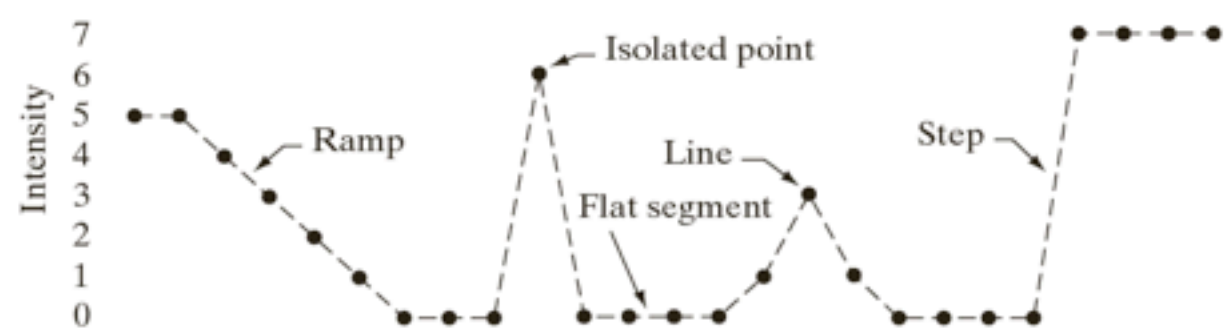
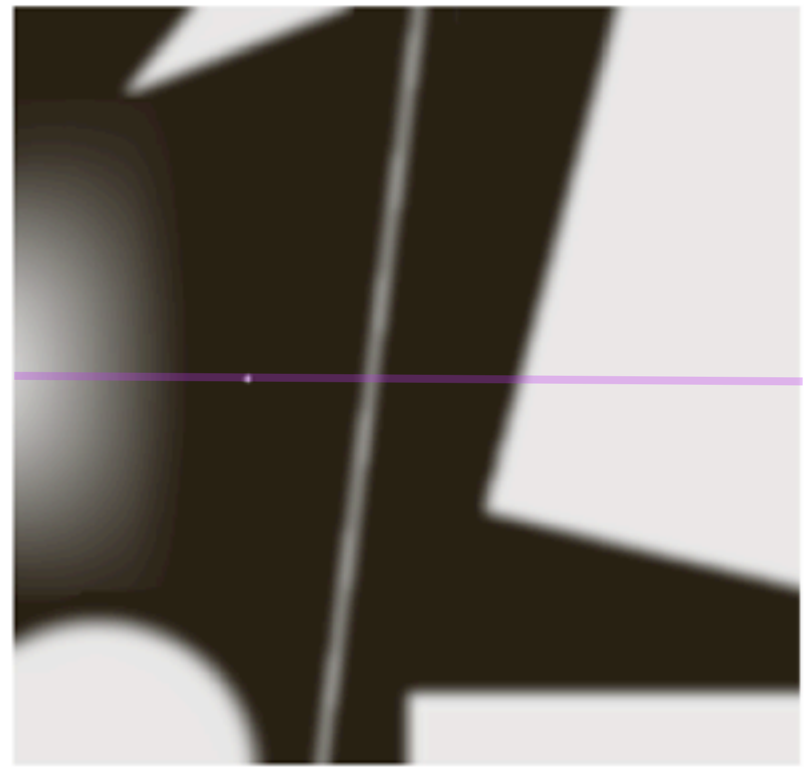
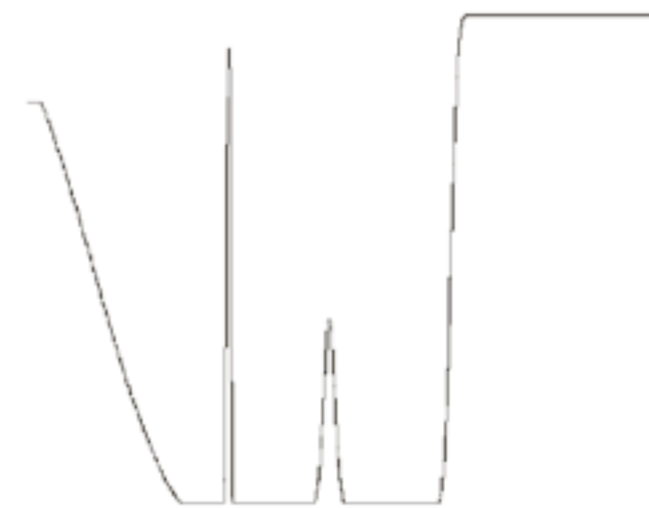


Image strip	5	5	4	3	2	1	0	0	0	6	0	0	0	0	1	3	1	0	0	0	0	7	7	7	7	•	•
First derivative		−1	−1	−1	−1	−1	0	0	6	−6	0	0	0	1	2	−2	−1	0	0	0	7	0	0	0			
Second derivative		−1	0	0	0	0	1	0	6	−12	6	0	0	1	1	−4	1	1	0	0	7	−7	0	0			



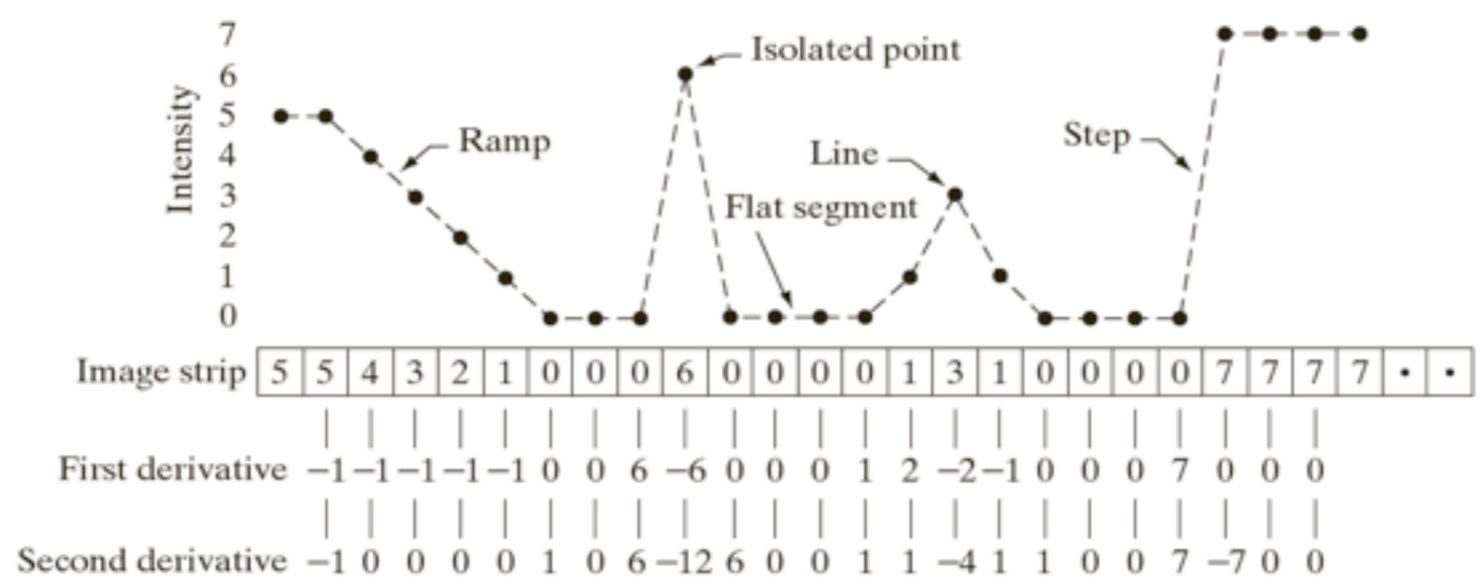


Derivative operator:

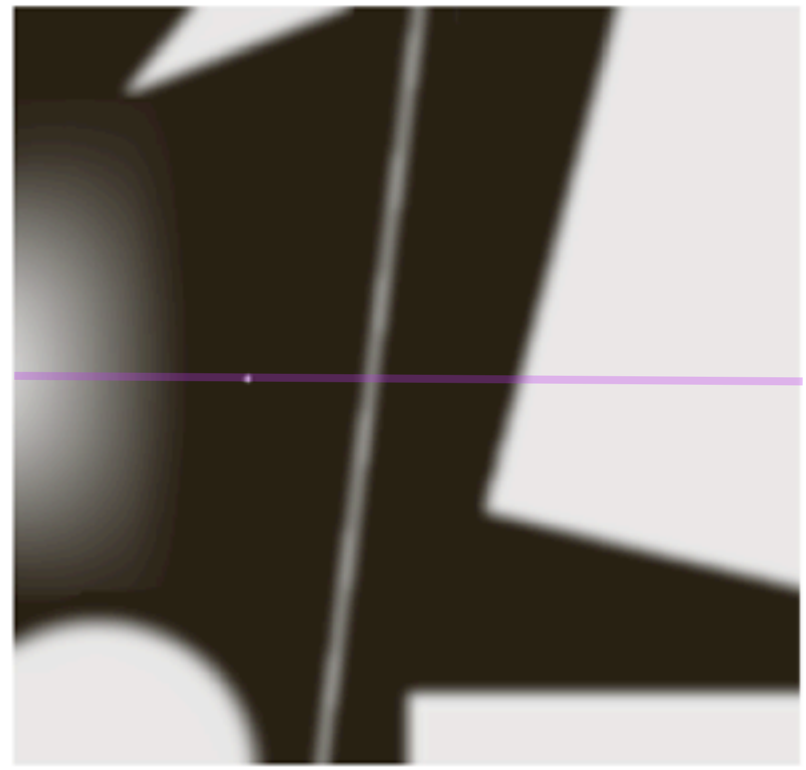
1. Must be 0 on constant regions
2. Is nonzero on ramps/ steps
3. Is nonzero on the onset of ramps/steps

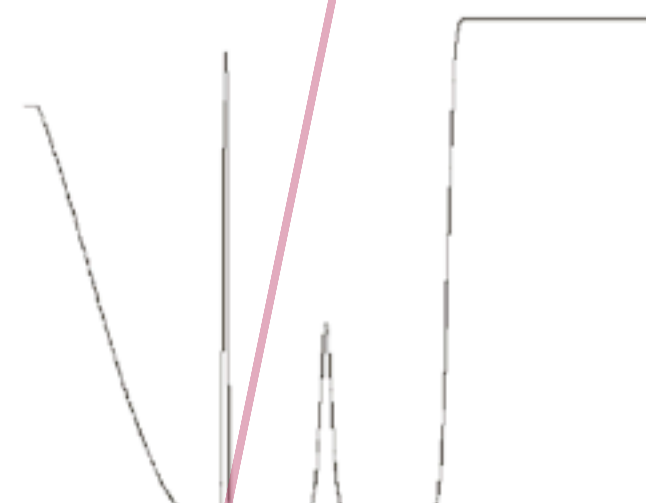
Second derivative operator:

1. must be 0 on constant regions
2. must be nonzero on the onset of steps/ramps
3. must be 0 along ramps with constant slope



This is a “roof” edge



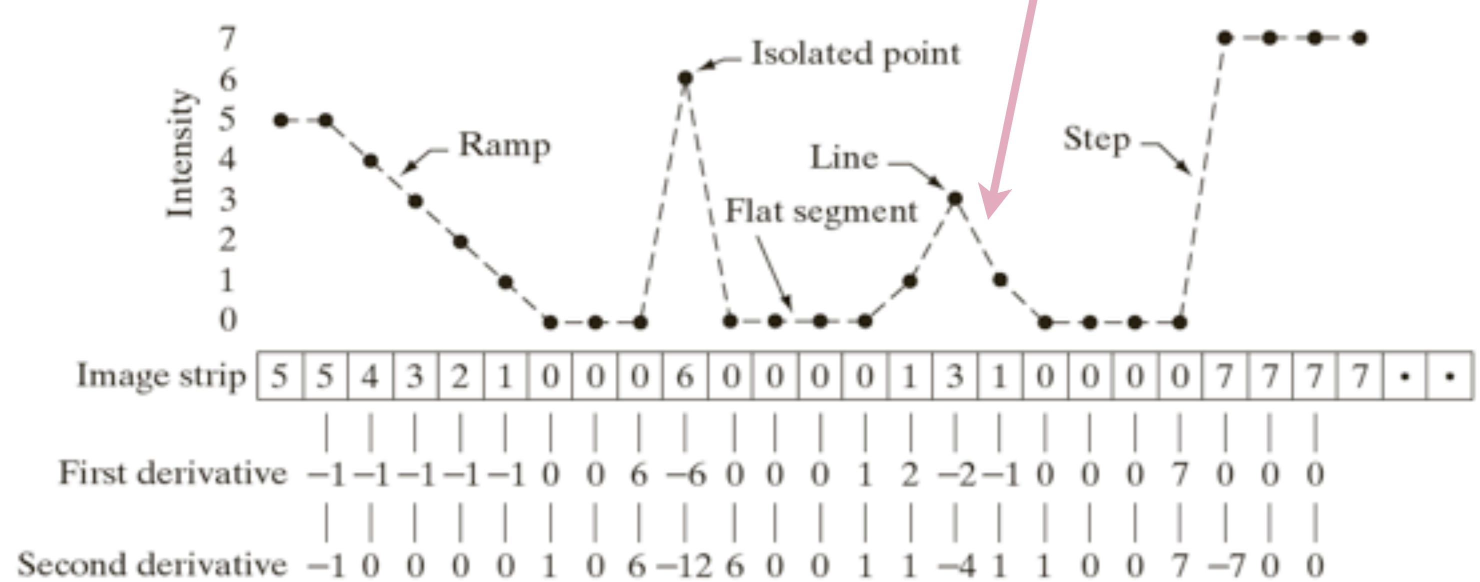


Derivative operator:

1. Must be 0 on constant regions
2. Is nonzero on ramps/ steps
3. Is nonzero on the onset of ramps/steps

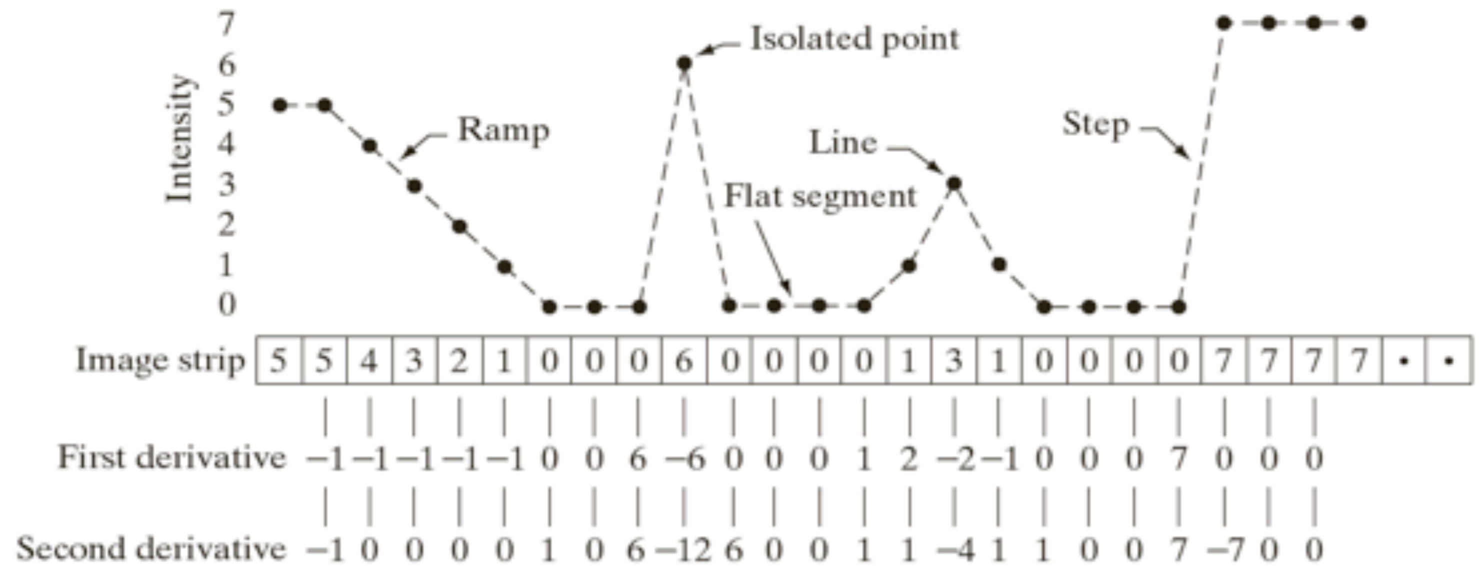
Second derivative operator:

1. must be 0 on constant regions
2. must be nonzero on the onset of steps/ramps
3. must be 0 along ramps with constant slope



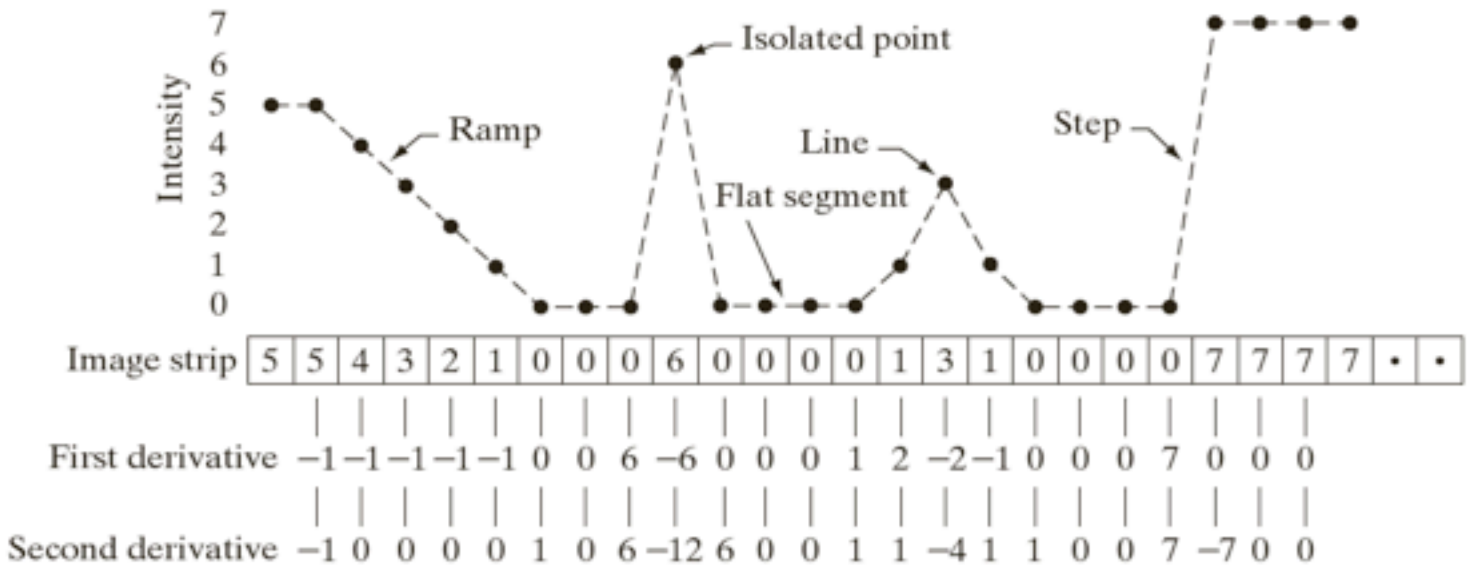
$$\frac{\partial f}{\partial x} \approx f(x+1) - f(x),$$

$$\frac{\partial^2 f}{\partial x^2} \approx f(x-1) - 2f(x) + f(x+1)$$



$$\frac{\partial f}{\partial x} \approx f(x+1) - f(x),$$

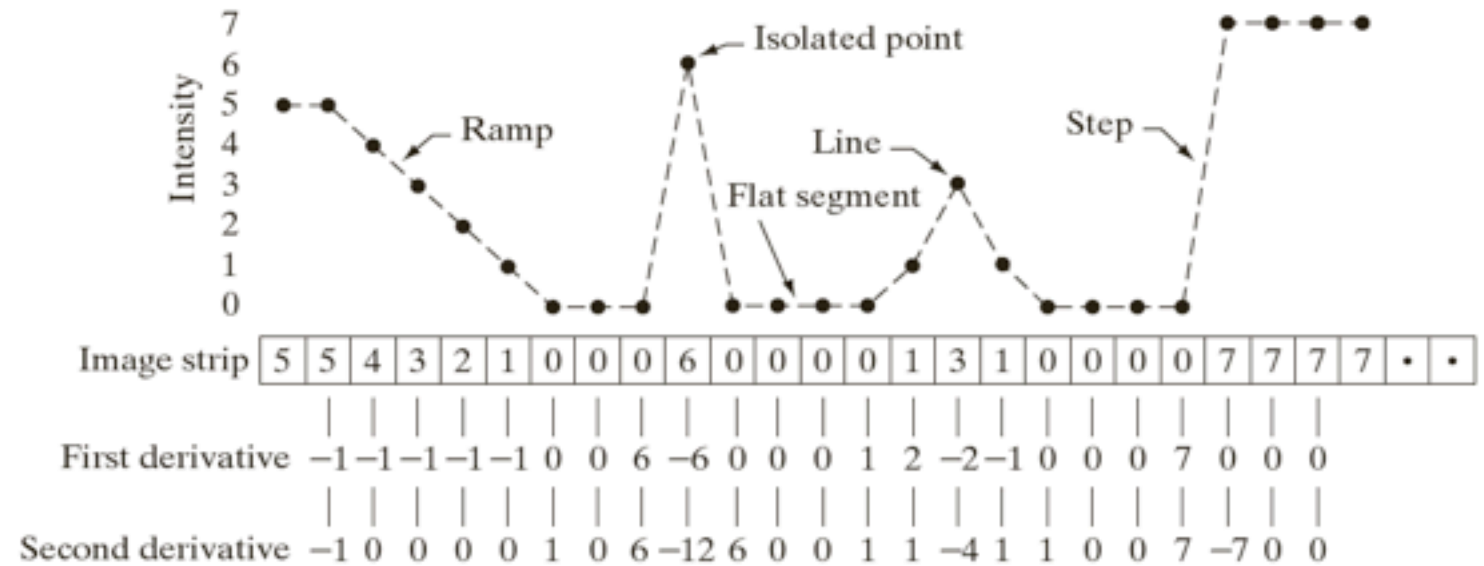
$$\frac{\partial^2 f}{\partial x^2} \approx f(x-1) - 2f(x) + f(x+1)$$



We see that:

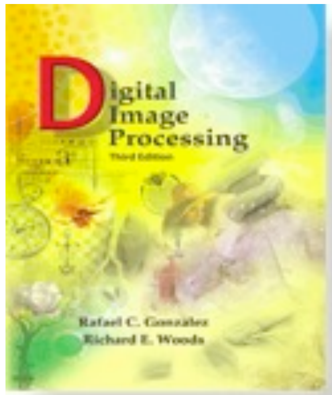
$$\frac{\partial f}{\partial x} \approx f(x+1) - f(x).$$

$$\frac{\partial^2 f}{\partial x^2} \approx f(x-1) - 2f(x) + f(x+1)$$



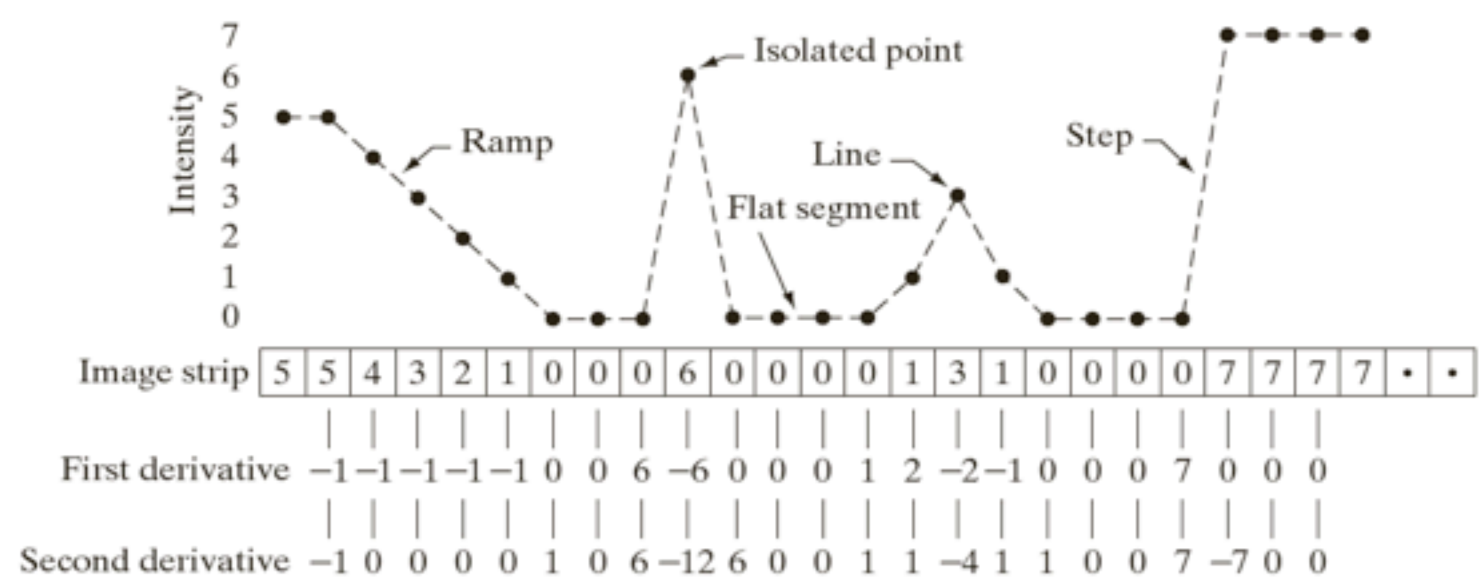
We see that:

- First order derivative produce thicker edges in an image



$$\frac{\partial f}{\partial x} \approx f(x+1) - f(x)$$

$$\frac{\partial^2 f}{\partial x^2} \approx f(x-1) - 2f(x) + f(x+1)$$

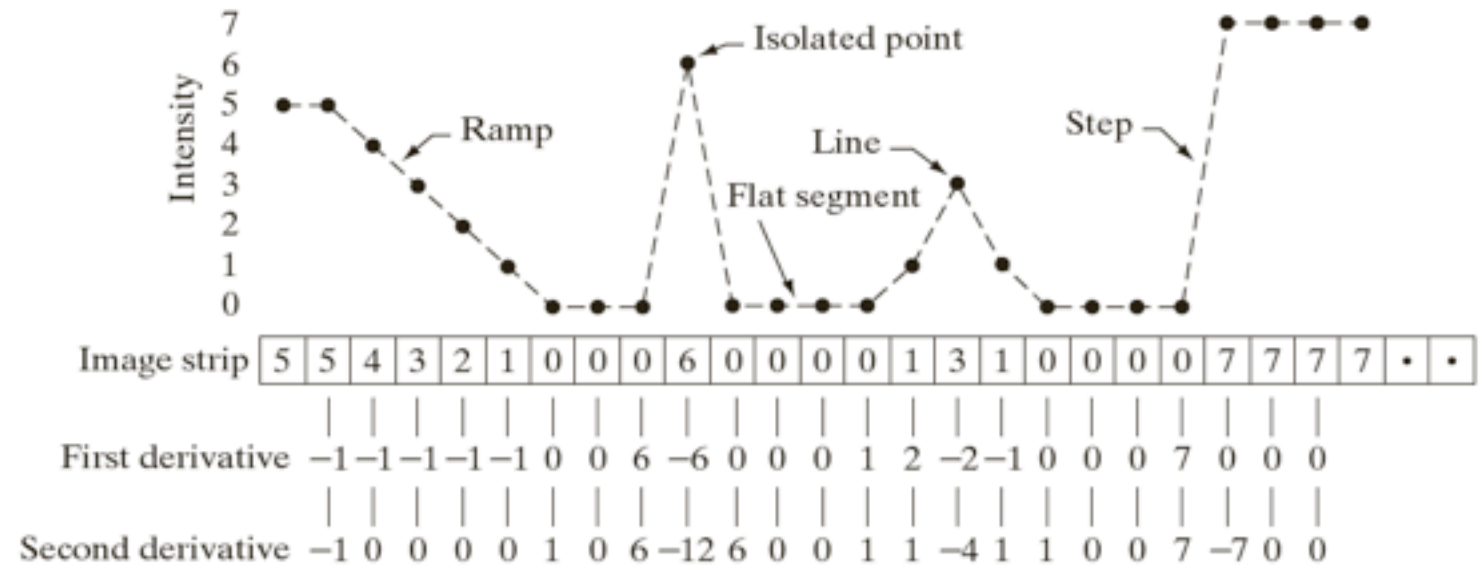


We see that:

- First order derivative produce thicker edges in an image
- Second order derivative have high response to detail (points, noise...)

$$\frac{\partial f}{\partial x} \approx f(x+1) - f(x)$$

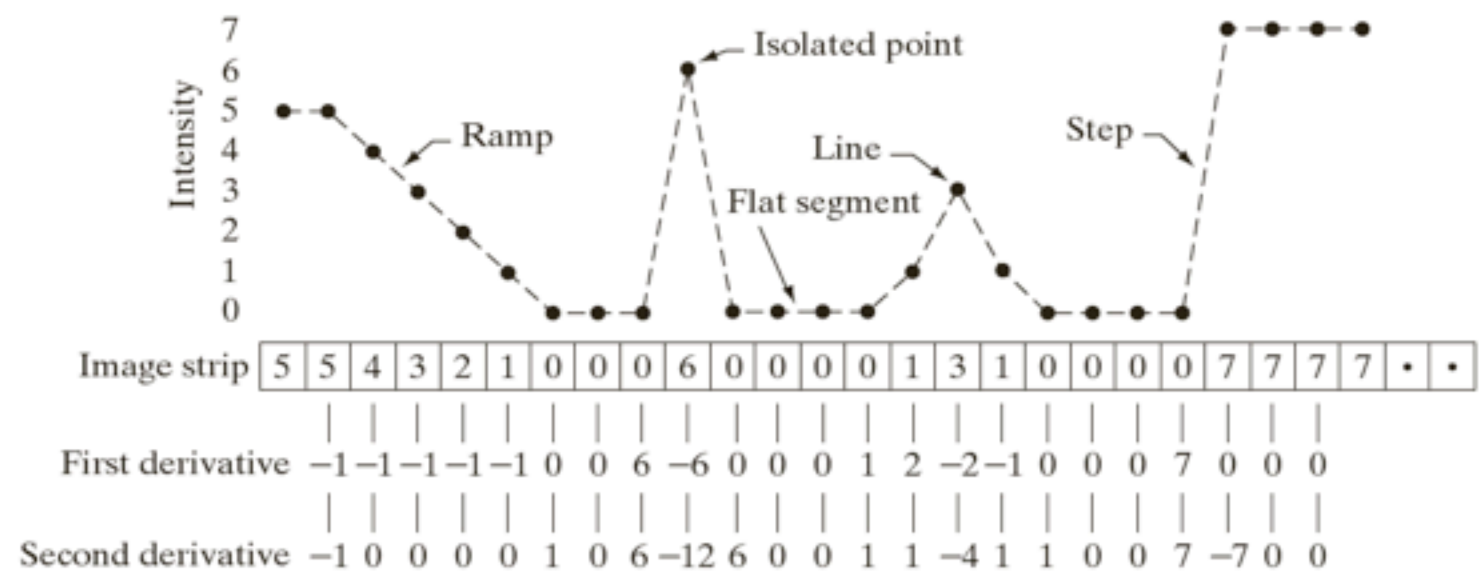
$$\frac{\partial^2 f}{\partial x^2} \approx f(x-1) - 2f(x) + f(x+1)$$



We see that:
- First order derivative produce thicker edges in an image
- Second order derivative have high response to detail (points, noise...)
- Second order derivative produce double-edge response at ramps/steps

$$\frac{\partial f}{\partial x} \approx f(x+1)-f(x),$$

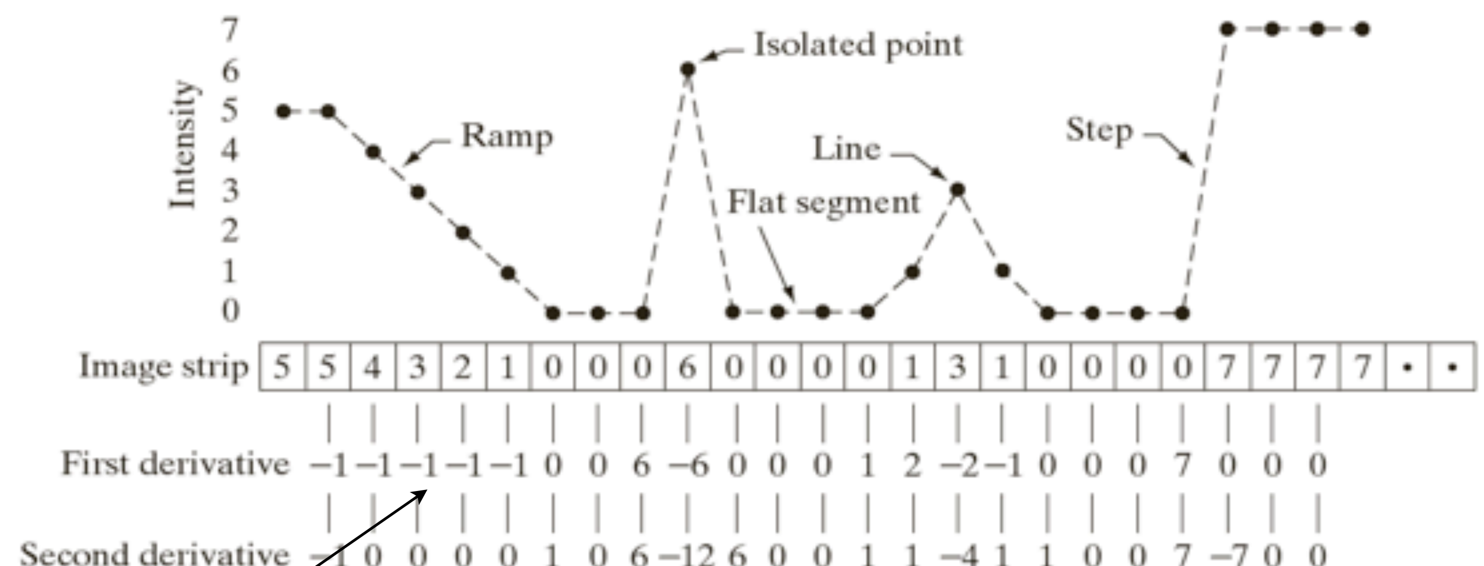
$$\frac{\partial^2 f}{\partial x^2} \approx f(x-1)-2f(x)+f(x+1)$$



We see that:
- First order derivative produce thicker edges in an image
- Second order derivative have high response to detail (points, noise...)
- Second order derivative produce double-edge response at ramps/steps
- The sign of the second derivative can be used to determine transitions dark->light or light -> dark

$$\frac{\partial f}{\partial x} \approx f(x+1) - f(x),$$

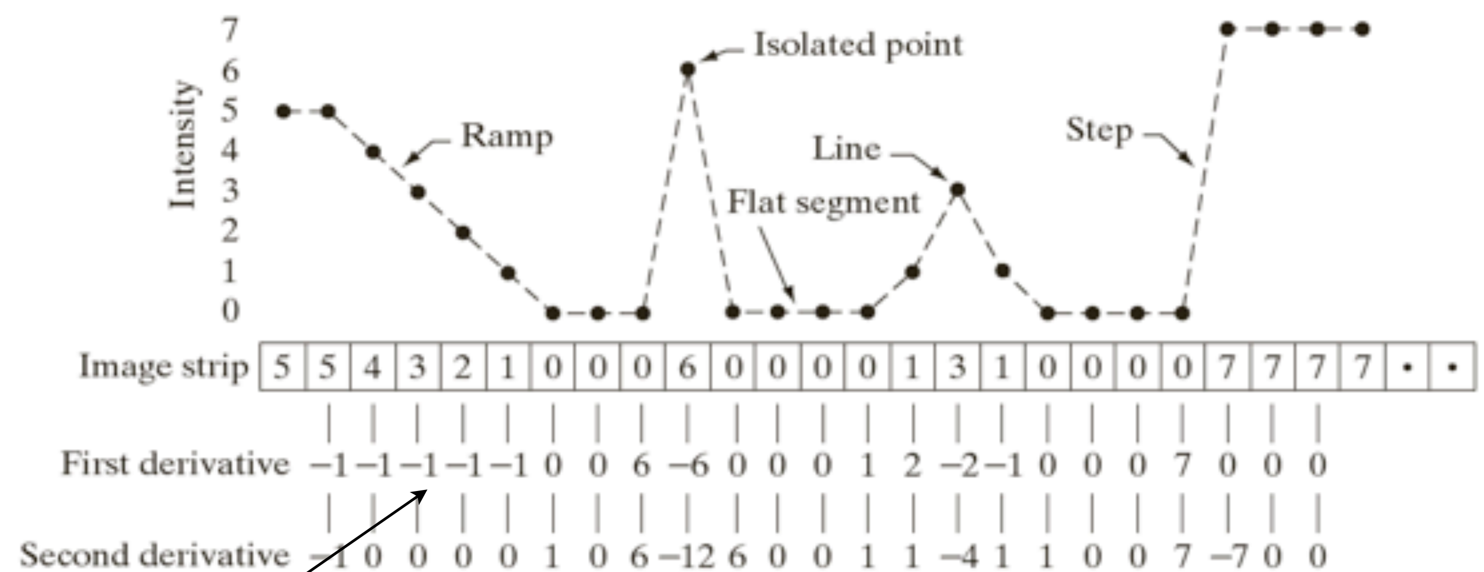
$$\frac{\partial^2 f}{\partial x^2} \approx f(x-1) - 2f(x) + f(x+1)$$



We see that:
- First order derivative produce thicker edges in an image
- Second order derivative have high response to detail (points, noise...)
- Second order derivative produce double-edge response at ramps/steps
- The sign of the second derivative can be used to determine transitions dark->light or light -> dark

$$\frac{\partial f}{\partial x} \approx f(x+1) - f(x),$$

$$\frac{\partial^2 f}{\partial x^2} \approx f(x-1) - 2f(x) + f(x+1)$$

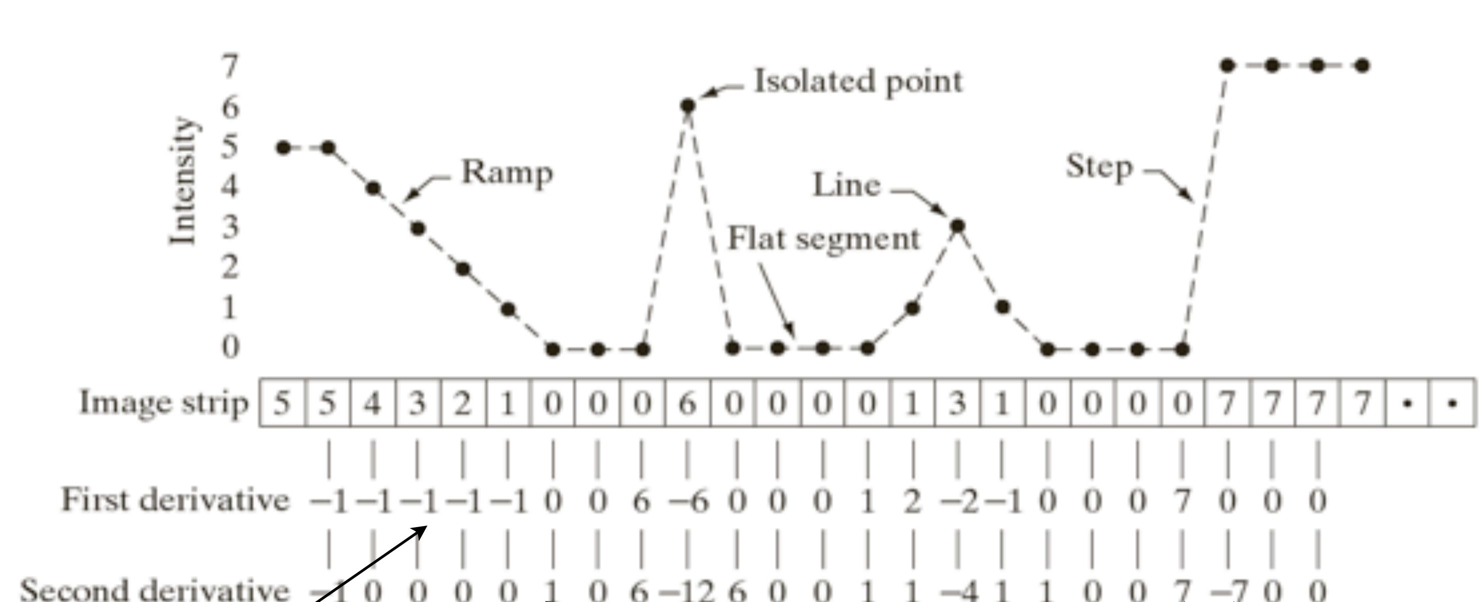


We see that:

- First order derivative produce thicker edges in an image
- Second order derivative have high response to detail (points, noise...)
- Second order derivative produce double-edge response at ramps/steps
- The sign of the second derivative can be used to determine transitions dark->light or light -> dark

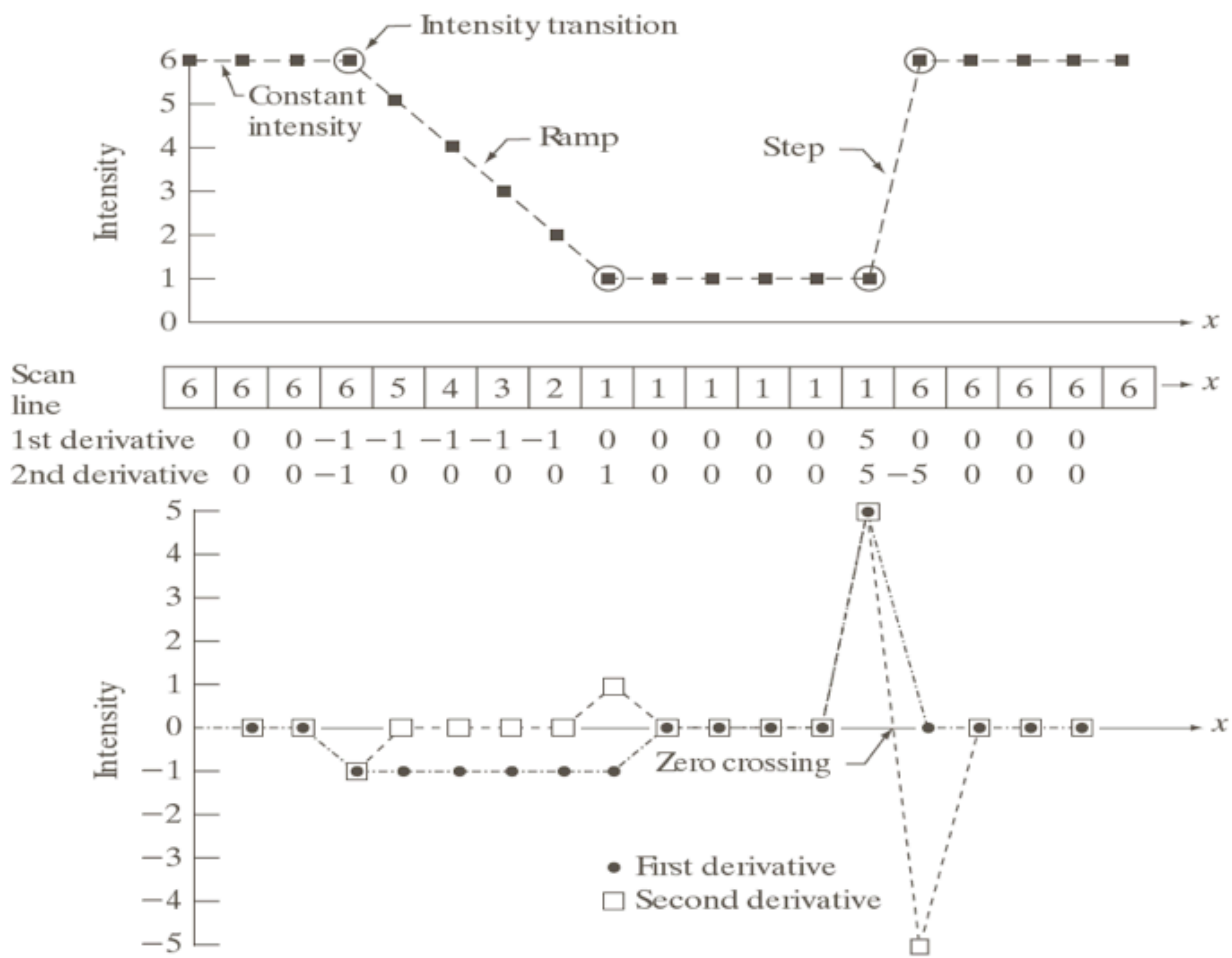
$$\frac{\partial f}{\partial x} \approx f(x+1) - f(x),$$

$$\frac{\partial^2 f}{\partial x^2} \approx f(x-1) - 2f(x) + f(x+1)$$

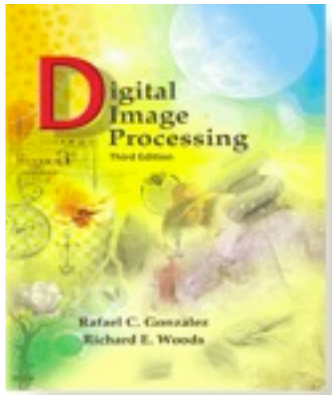


We see that:

- First order derivative produce thicker edges in an image
- Second order derivative have high response to detail (points, noise...)
- Second order derivative produce double-edge response at ramps/steps
- The sign of the second derivative can be used to determine transitions dark->light or light -> dark



Intensity transition
Constant intensity
Ramp
Step
Intensity
x
Scan line
6 6 6 6 5 4 3 2 1 1 1 1 1 1 6 6 6 6 6
1st derivative 0 0 −1 −1 −1 −1 −1 0 0 0 0 0 5 0 0 0 0
2nd derivative 0 0 −1 0 0 0 0 1 0 0 0 0 5 −5 0 0 0
Zero crossing
First derivative
Second derivative

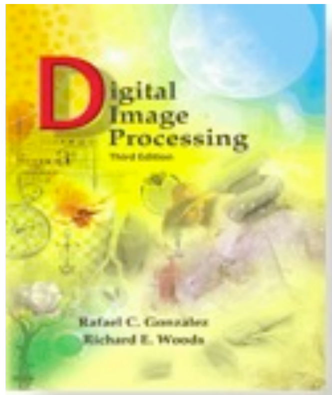


To apply first/second order derivatives, we have seen how to use filter masks:

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

$$R = \sum_{i=1}^{9} w_i z_i$$

(R is the response of the mask, here done by correlation)



Detection of isolated points: use second derivative

Laplacian mask

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

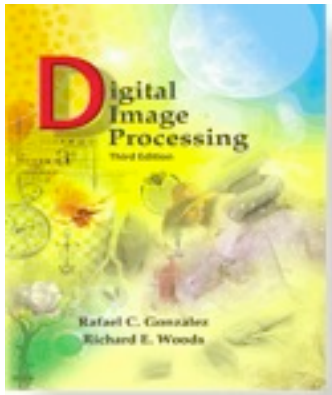
$$\nabla^2 f \approx f(x-1,y) + f(x+1,y) + f(x,y-1) + f(x,y+1) - 4f(x,y)$$

0	1	0
1	−4	1
0	1	0

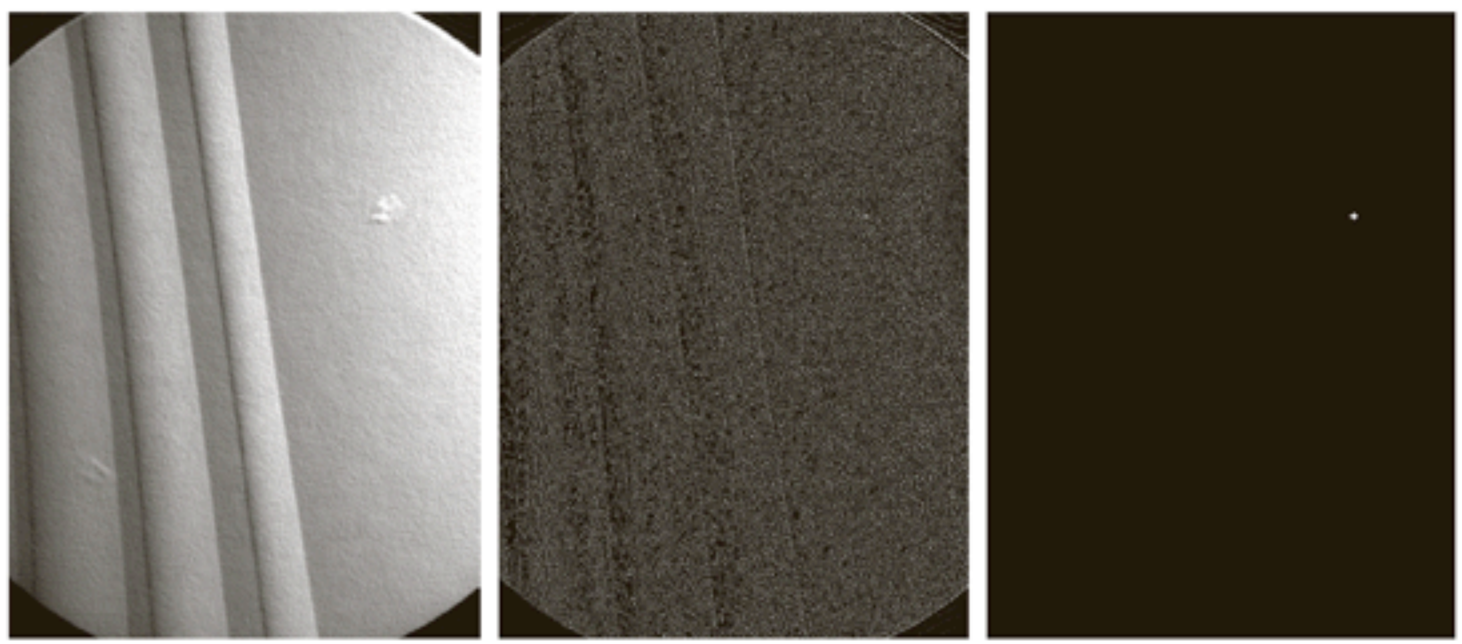
1	1	1
1	−8	1
1	1	1

Set

$$g(x,y) = \begin{cases} 1 & \text{if } |R(x,y)| \geq T \\ 0 & \text{otherwise} \end{cases} \qquad (T \text{ threshold value})$$



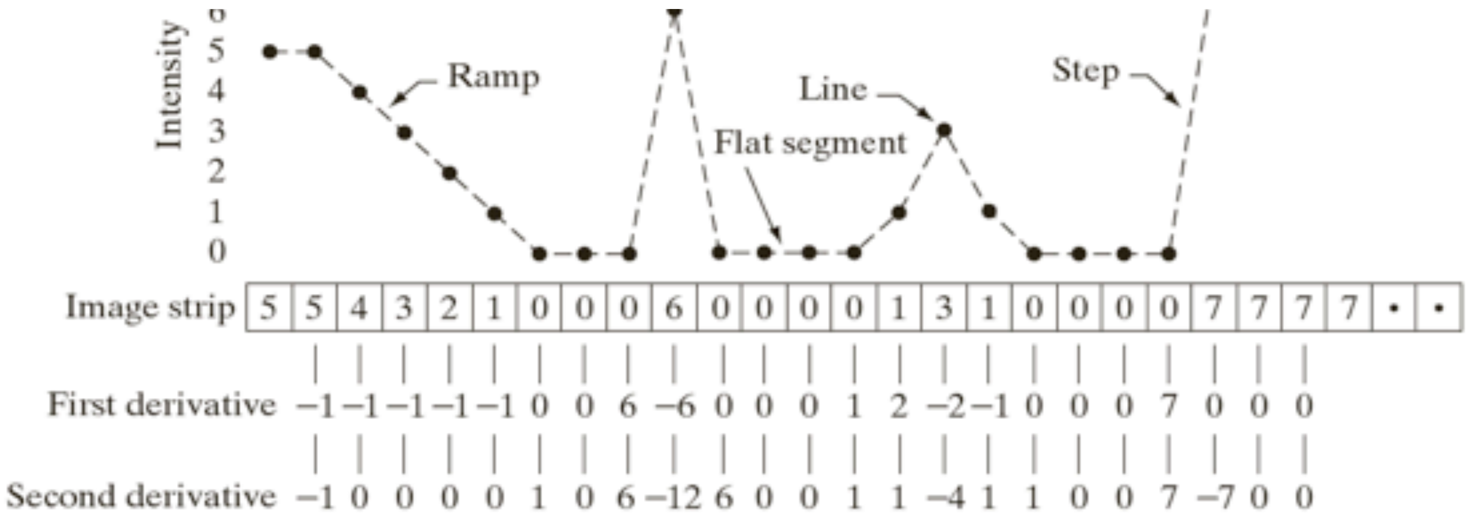
1	1	1
1	−8	1
1	1	1



a
b c d

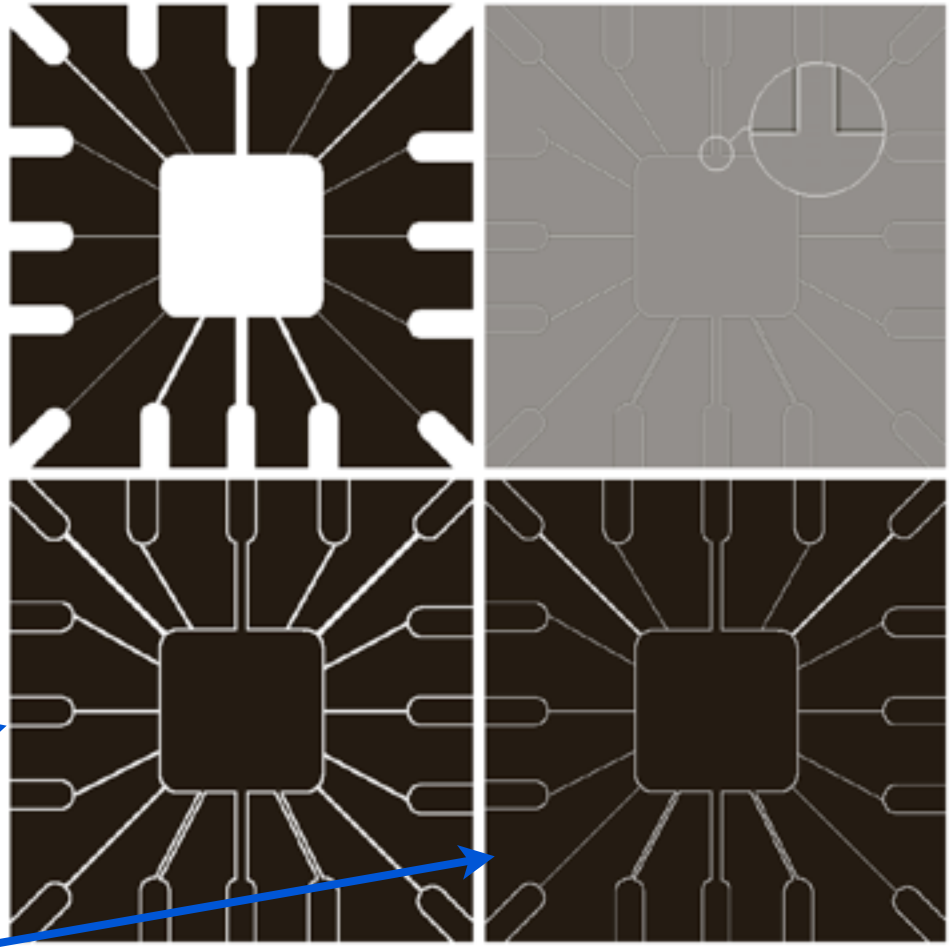
FIGURE 10.4 (a) Point detection (Laplacian) mask. (b) X-ray image of turbine blade with a porosity. The porosity contains a single black pixel. (c) Result of convolving the mask with the image. (d) Result of using Eq. (10.2-8) showing a single point (the point was enlarged to make it easier to see). (Original image courtesy of X-TEK Systems, Ltd.)

Detection of edges: second and first derivative



Second derivatives give thinner lines (and generally higher response)

Need to handle properly the double-line effect



Taking the abs(nabla f) gives thicker lines

only positive values of laplacian

NB. The Laplacian is *isotropic*, i.e. it picks up equally well edges in all the directions.

If we are interested in some *specific* directions for the edges, it is useful to use other types of masks (anisotropic).

Horizontal

−1	−1	−1
2	2	2
−1	−1	−1

+45°

2	−1	−1
−1	2	−1
−1	−1	2

Vertical

−1	2	−1
−1	2	−1
−1	2	−1

−45°

−1	−1	2
−1	2	−1
2	−1	−1

The preferred direction is weighted by a higher value.
The sum of the mask coefficients is 0 (consistent with constant images)

NB. The Laplacian is *isotropic*, i.e. it picks up equally well edges in all the directions.

If we are interested in some *specific* directions for the edges, it is useful to use other types of masks (anisotropic).

−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

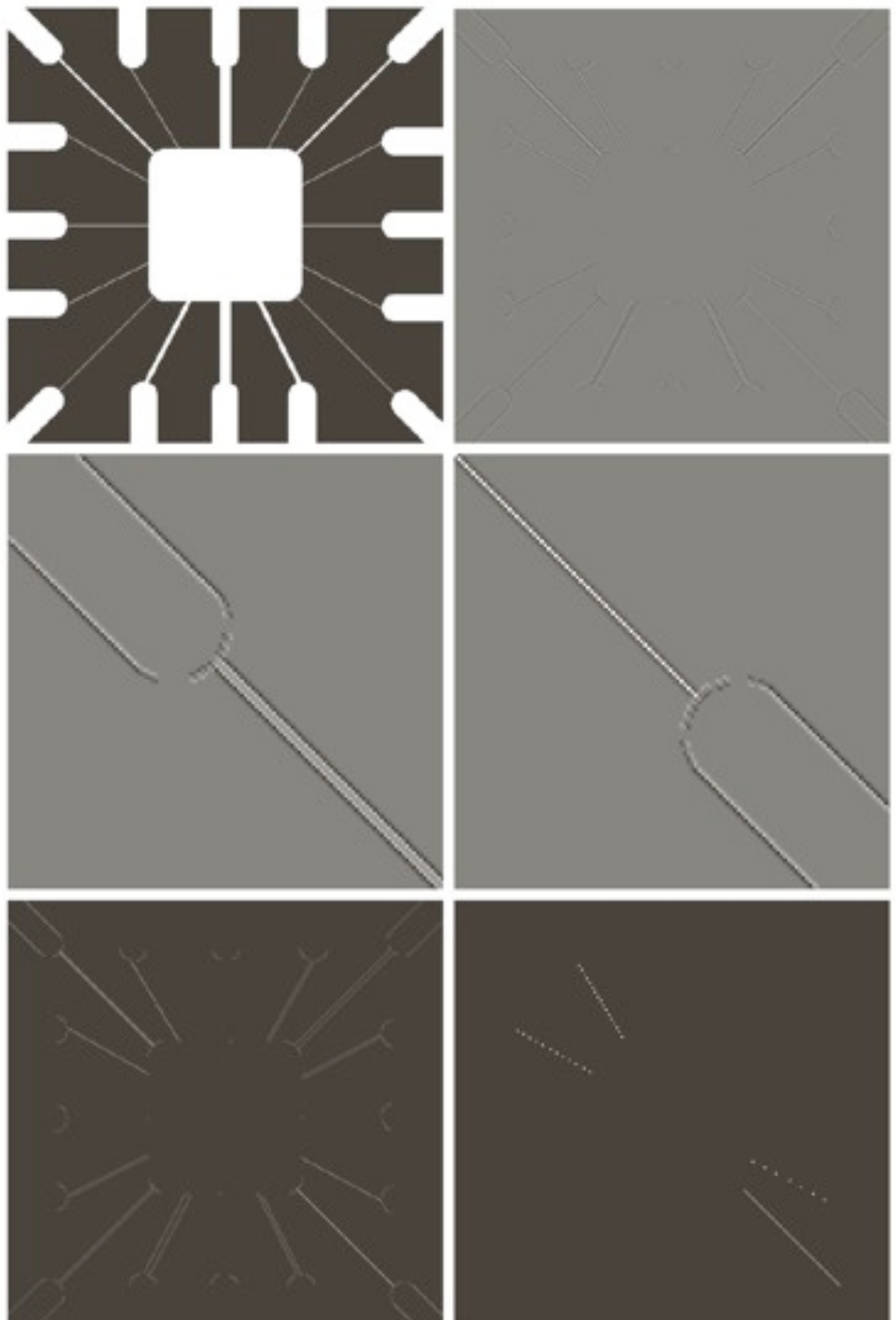
Vertical

−1	−1	2
−1	2	−1
2	−1	−1

−45°

The preferred direction is weighted by a higher value.
The sum of the mask coefficients is 0 (consistent with constant images)

Thresholding: $$g(x, y) = \begin{cases} 1 & \text{if } |R(x, y)| \geq T \\ 0 & \text{otherwise} \end{cases}$$



a b
c d
e f

FIGURE 10.7 (a) Image of a wire-bond template. (b) Result of processing with the $+45°$ line detector mask in Fig. 10.6. (c) Zoomed view of the top left region of (b). (d) Zoomed view of the bottom right region of (b). (e) The image in (b) with all negative values set to zero. (f) All points (in white) whose values satisfied the condition $g \geq T$, where g is the image in (e). (The points in (f) were enlarged to make them easier to see.)

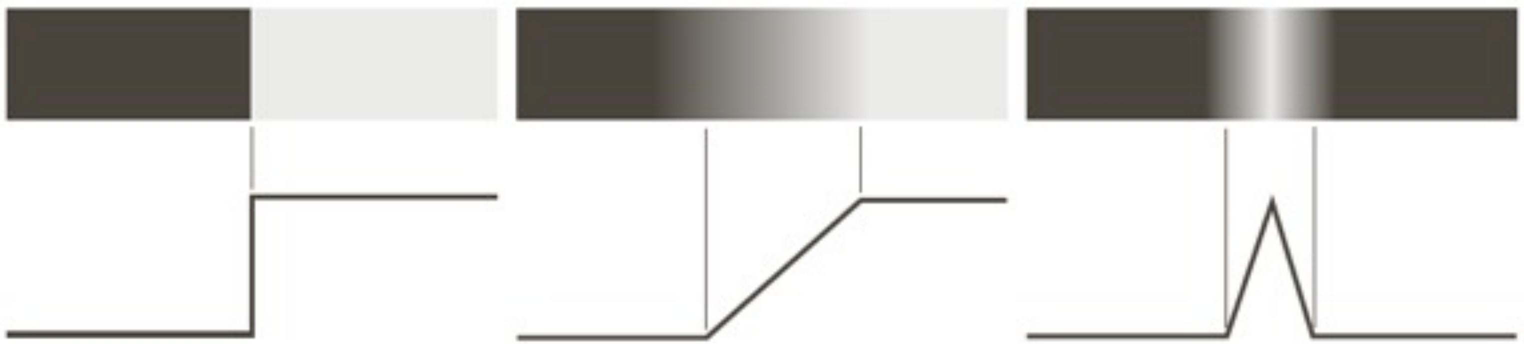
Edge models

ideal edges: *step edges*
(clean, distance 1 pixel)

typical of computed-generated images

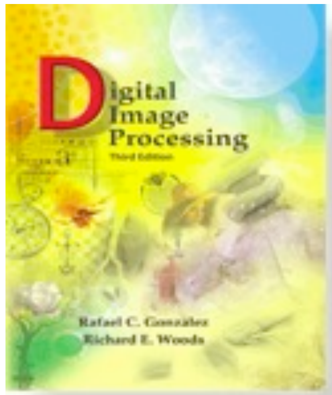
In practice, edges in digital images are not clean: they appear rather as a *ramp.*
misfocussing, blurring, noise of the imaging system…

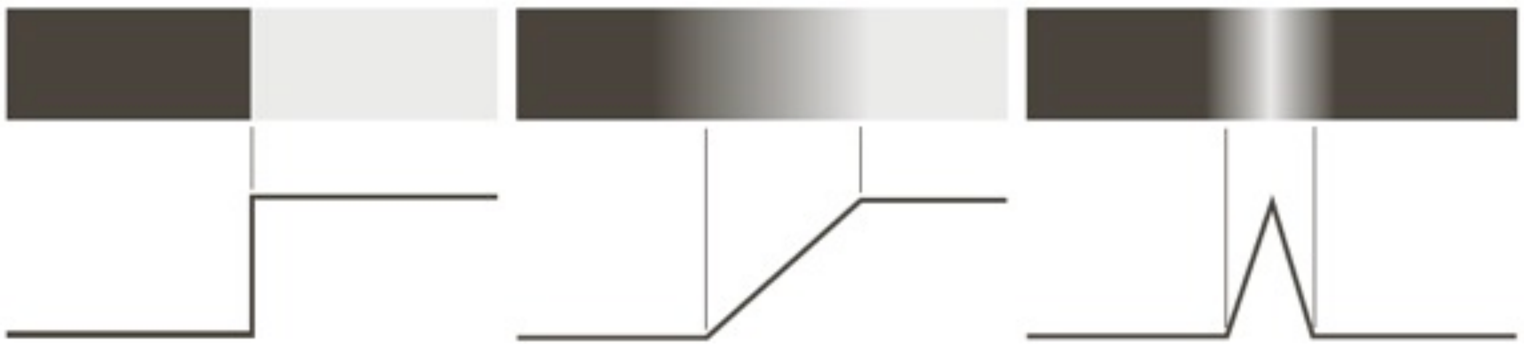
The more the blurring, the larger the ramp is. The slope of the ramp is inversely proportional to the blurring



a b c

FIGURE 10.8 From left to right, models (ideal representations) of a step, a ramp, and a roof edge, and their corresponding intensity profiles.





a b c

FIGURE 10.8 From left to right, models (ideal representations) of a step, a ramp, and a roof edge, and their corresponding intensity profiles.

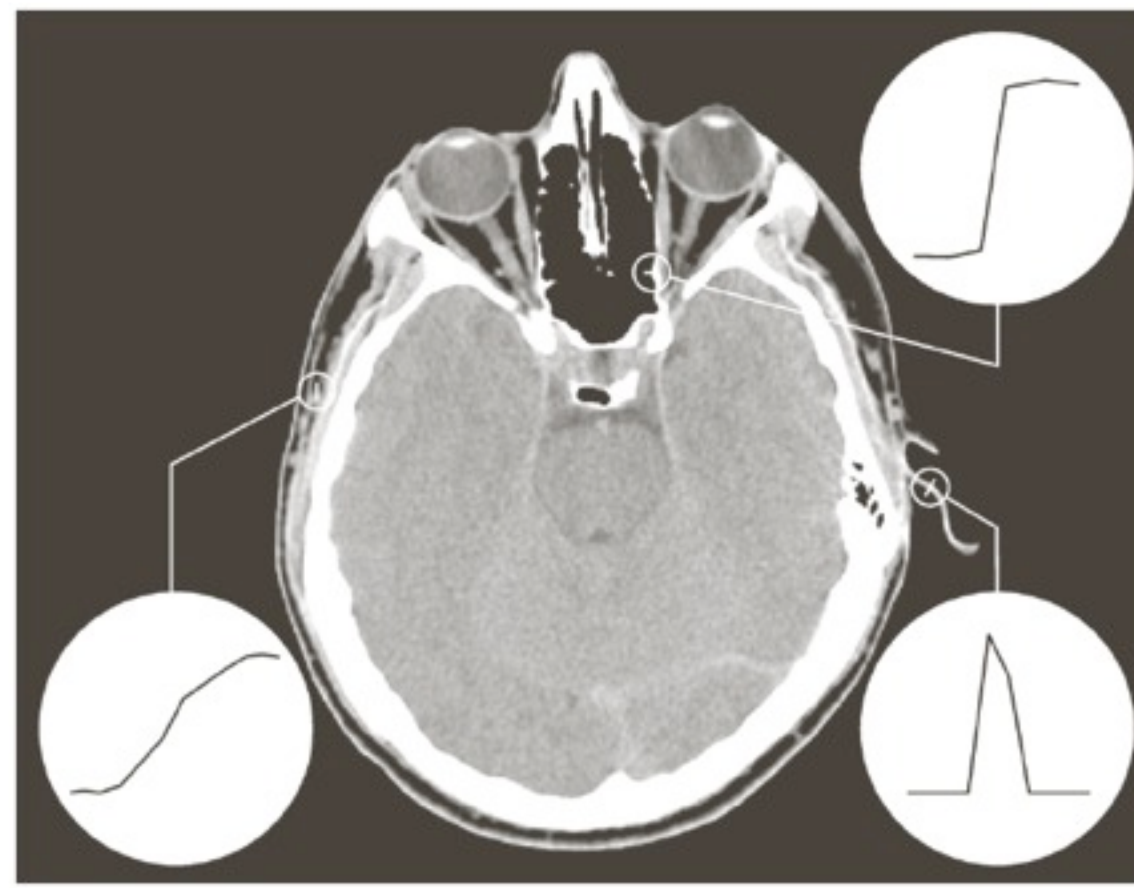
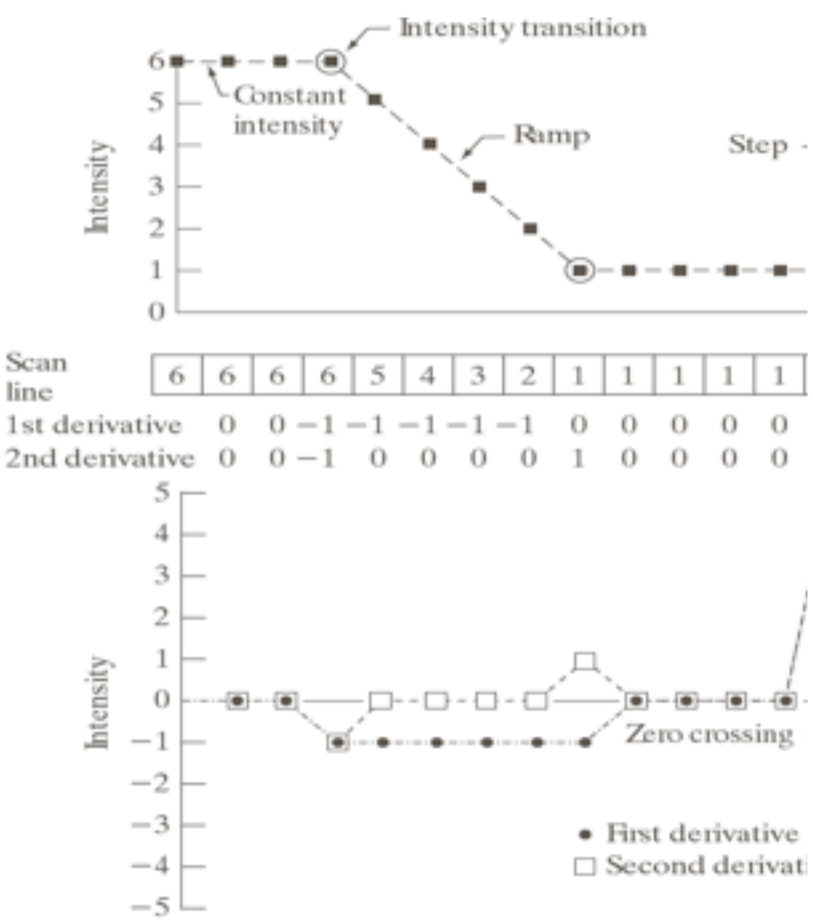
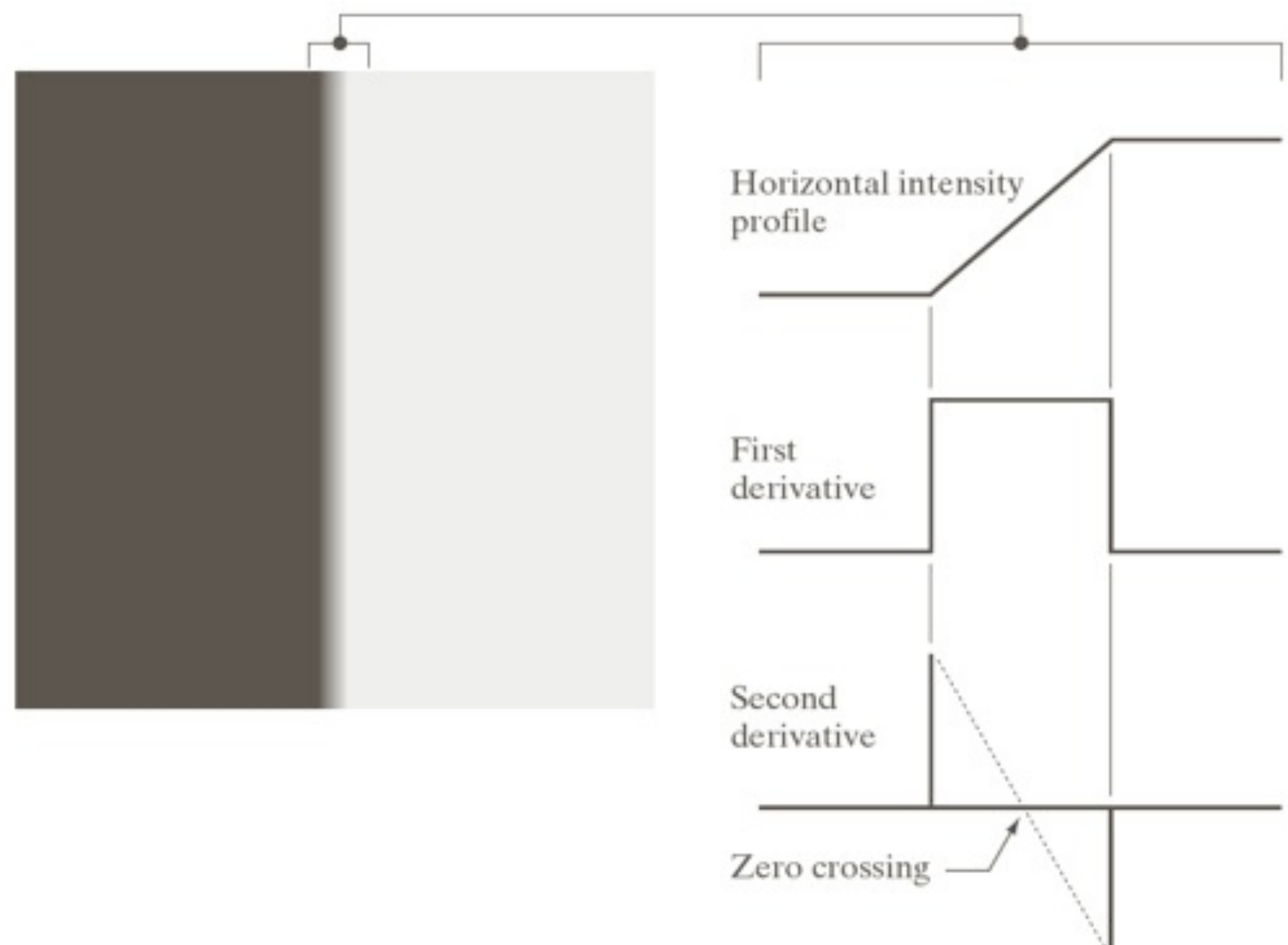


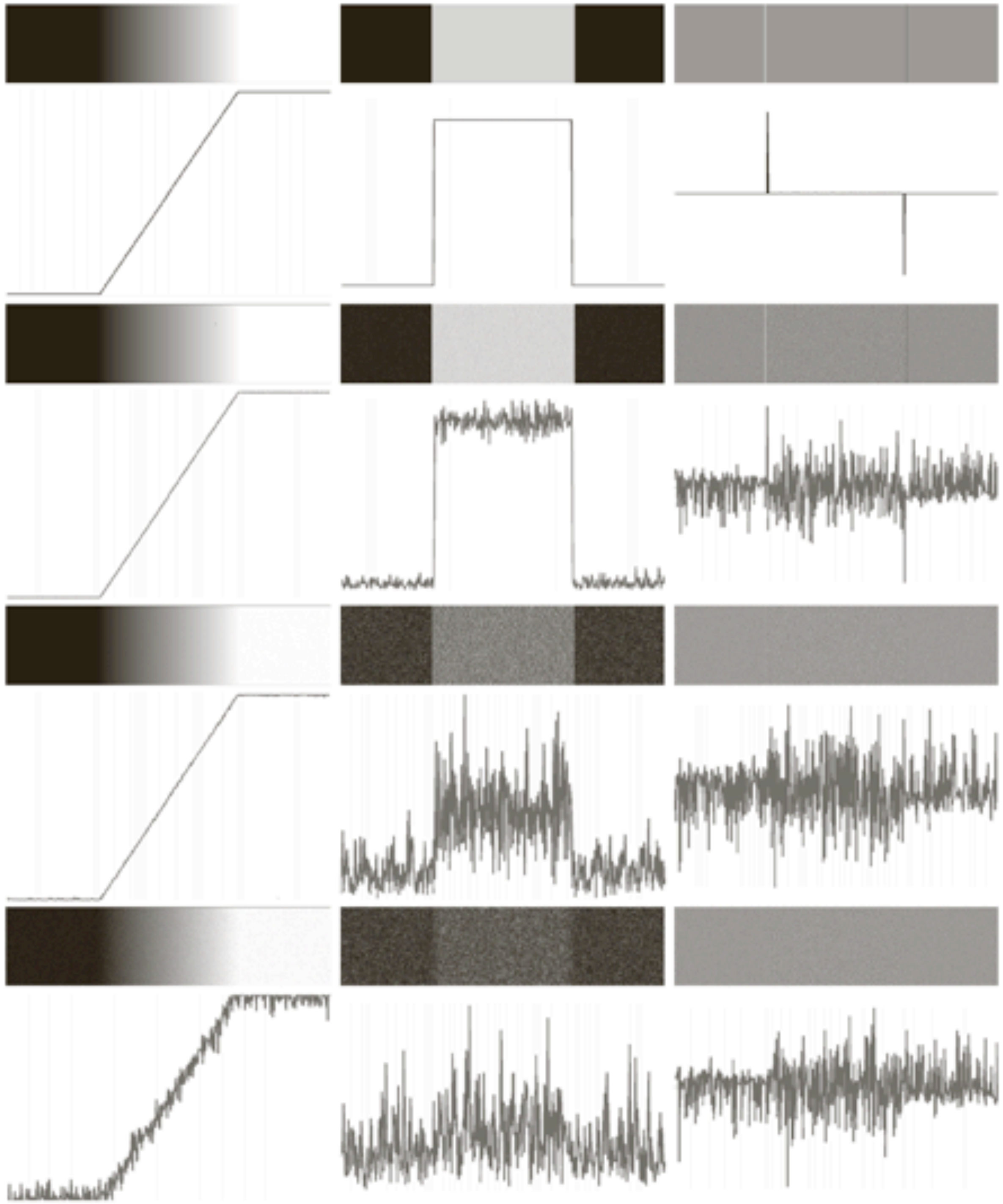
FIGURE 10.9 A 1508 × 1970 image showing (zoomed) actual ramp (bottom, left), step (top, right), and roof edge profiles. The profiles are from dark to light, in the areas indicated by the short line segments shown in the small circles. The ramp and "step" profiles span 9 pixels and 2 pixels, respectively.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

Ideal ramps:





- The magnitude of the first derivative can be used to detect a ramp (edge).
- The sign of the second derivative to decide if the point lies on the dark side or light side



If the ramp is contaminated by noise, the derivative operators can have problems.

Three fundamental steps:

1. Smooth the image (to reduce the noise)
2. Detect candidate edge points
3. Localize the edge