Session 2

Image processing fundamentals – segmentation, edge detection, boundary detection

Session delivered by:

Dr. Subarna Chatterjee
Subarna.cs.et@msruas.ac.in

Session Topics

- Digital Images
  - Digital Image Processing
  - Key Stages in Digital Image Processing
  - Digital Image Formation – Sampling & Quantization
  - Image Formats
- Image processing fundamentals
  - edge detection,
  - segmentation,
  - boundary detection
  - filters
  - morphological operations

Digital Image Fundamentals

- Human visual system
- A simple image model
- Sampling and quantization
- Color models and Color imaging

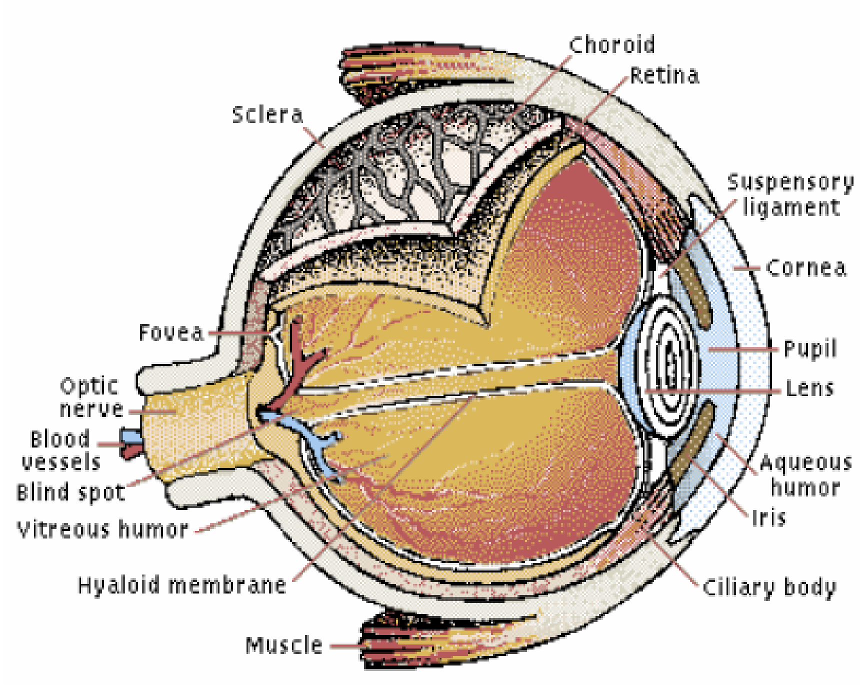
Human Visual System

- Brightness adaptation
- Brightness discrimination
- Weber ratio
- Mach band pattern
- Simultaneous contrast

  – c

Human Visual System

- Elements of visual perception



The amount of light entering the eye is controlled by the pupil, which dilates and contracts accordingly. The cornea and lens, whose shape is adjusted by the ciliary body, focus the light on the retina, where receptors convert it into nerve signals that pass to the brain.

Human Visual System

- Elements of visual perception
  - Cones
    - 6 – 7 million in each eye
    - Photopic or bright-light vision
    - Highly sensitive to color
  - Rods
    - 75 – 150 million
    - Not involved in color vision
    - Sensitive to low level of illumination (scotopic or dim-light vision)
  - An object appears brightly colored in daylight will be seen colorless in moonlight (why)

Human Visual System

- Image formation in the eye
  - Distance between center of lens and retina (focal length) vary between 14-17 mm.
  - Image length h = 17(mm) x (15/100)

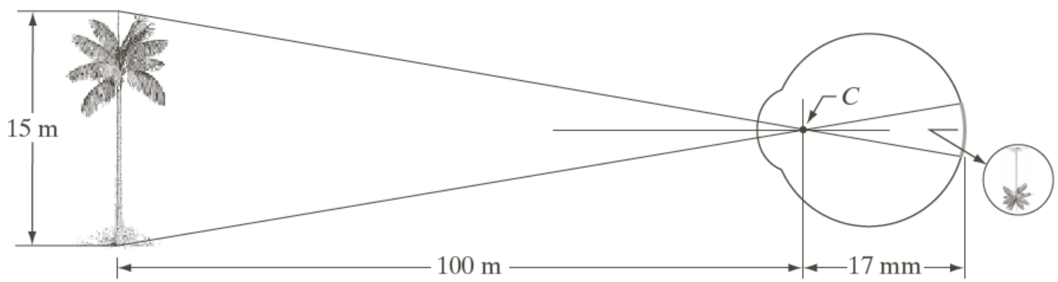
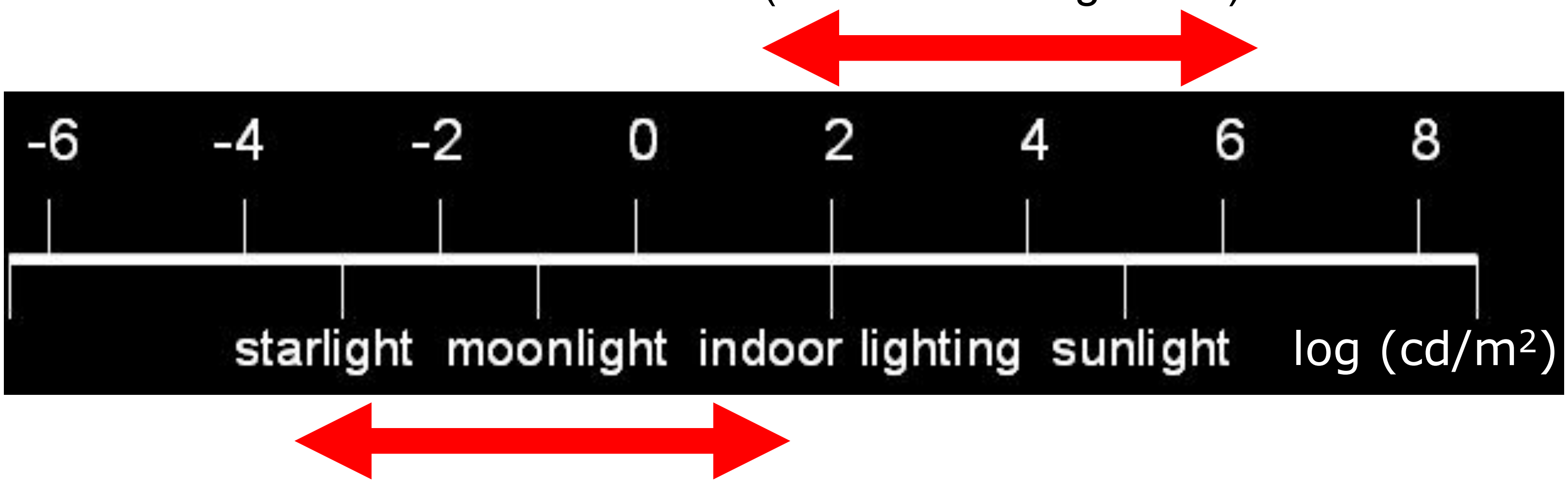


FIGURE 2.3 Graphical representation of the eye looking at a palm tree. Point C is the optical center of the lens.

Human Visual System

Human simultaneous luminance vision range
(5 orders of magnitude)

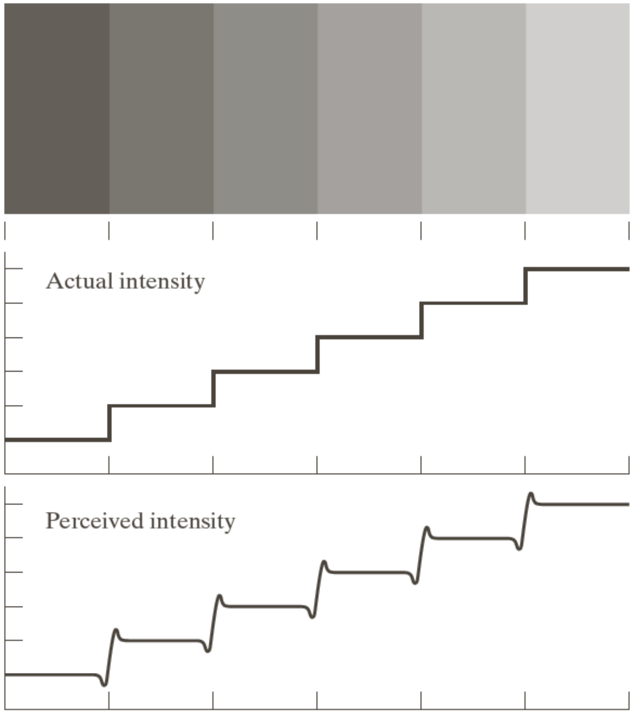


Human Visual System

- Brightness adaptation
  - HVS can adapt to light intensity range on the order of 10^{10}
  - Subjective brightness is a logarithmic function of the light intensity incident on the eye

Human Visual System

- Perceived brightness is not a simple function of intensity – Mach band pattern



a
b
c

FIGURE 2.7 Illustration of the Mach band effect. Perceived intensity is not a simple function of actual intensity.

Human Visual System

- Perceived brightness is not a simple function of intensity – Simultaneous contrast

a b c

FIGURE 2.8 Examples of simultaneous contrast. All the inner squares have the same intensity, but they appear progressively darker as the background becomes lighter.

A simple image model

- Two-dimensional light-intensity function

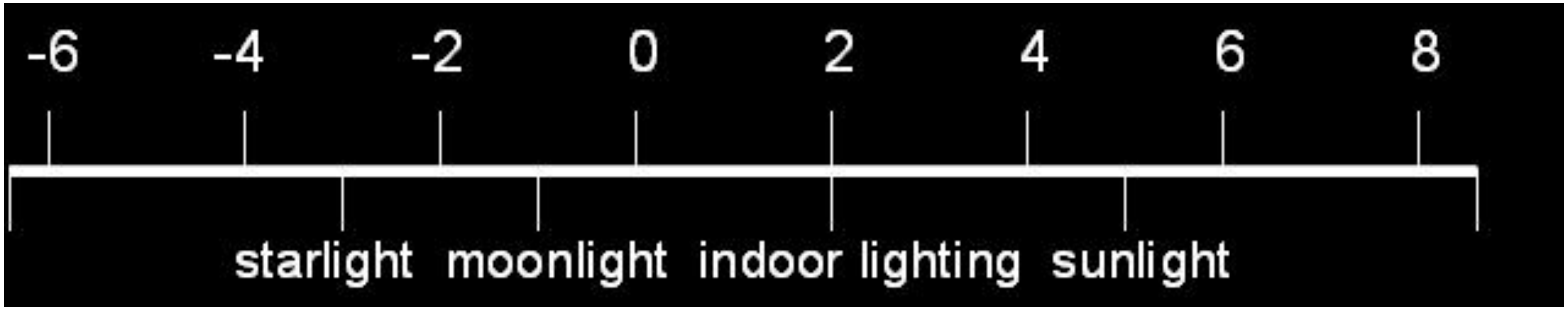
$f(x,y) = l(x,y)\ r(x,y)$

$l(x,y)$ - illumination component

$r(x,y)$ – reflectance component

A simple image model

- $I(x,y)$ - illumination range



- $r(x,y)$ – typical reflectance indixes
  - black velvet (0.01)
  - stainless steel (0.65)
  - white paint (0.80)
  - silver plate (0.90)
  - snow (0.93)

Sampling

- Digitization of the spatial coordinates, sample (x, y) at discrete values of (0, 0), (0, 1), ….

- f(x, y) is 2-D array

$$
f(x,y) = \begin{matrix} f(0,0) & f(0,1) & \cdots & f(0,M \quad 1) \\ f(1,0) & f(1,1) & & f(1,M \quad 1) \\ & & & \\ f(N \quad 1,0) & f(N \quad 1,1) & & f(N \quad 1,M \quad 1) \end{matrix}
$$

Quantization

- Digitization of the light intensity function
- Each f(i,j) is called a pixel
- The magnitude of f(i,j) is represented digitally with a fixed number of bits - quantization

Image Sensor

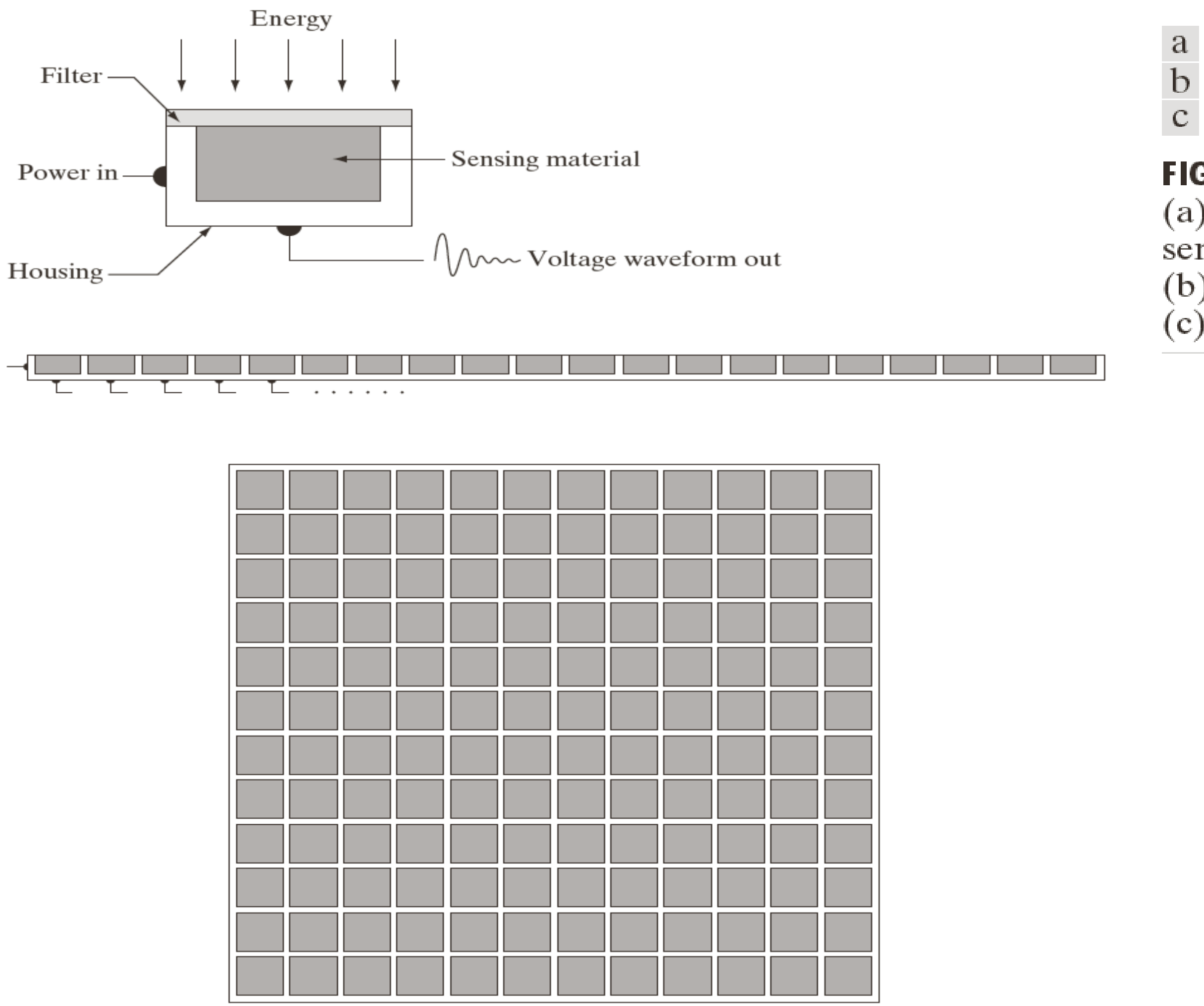


FIGURE 2.12
(a) Single imaging sensor.
(b) Line sensor.
(c) Array sensor.

Image Acquisition

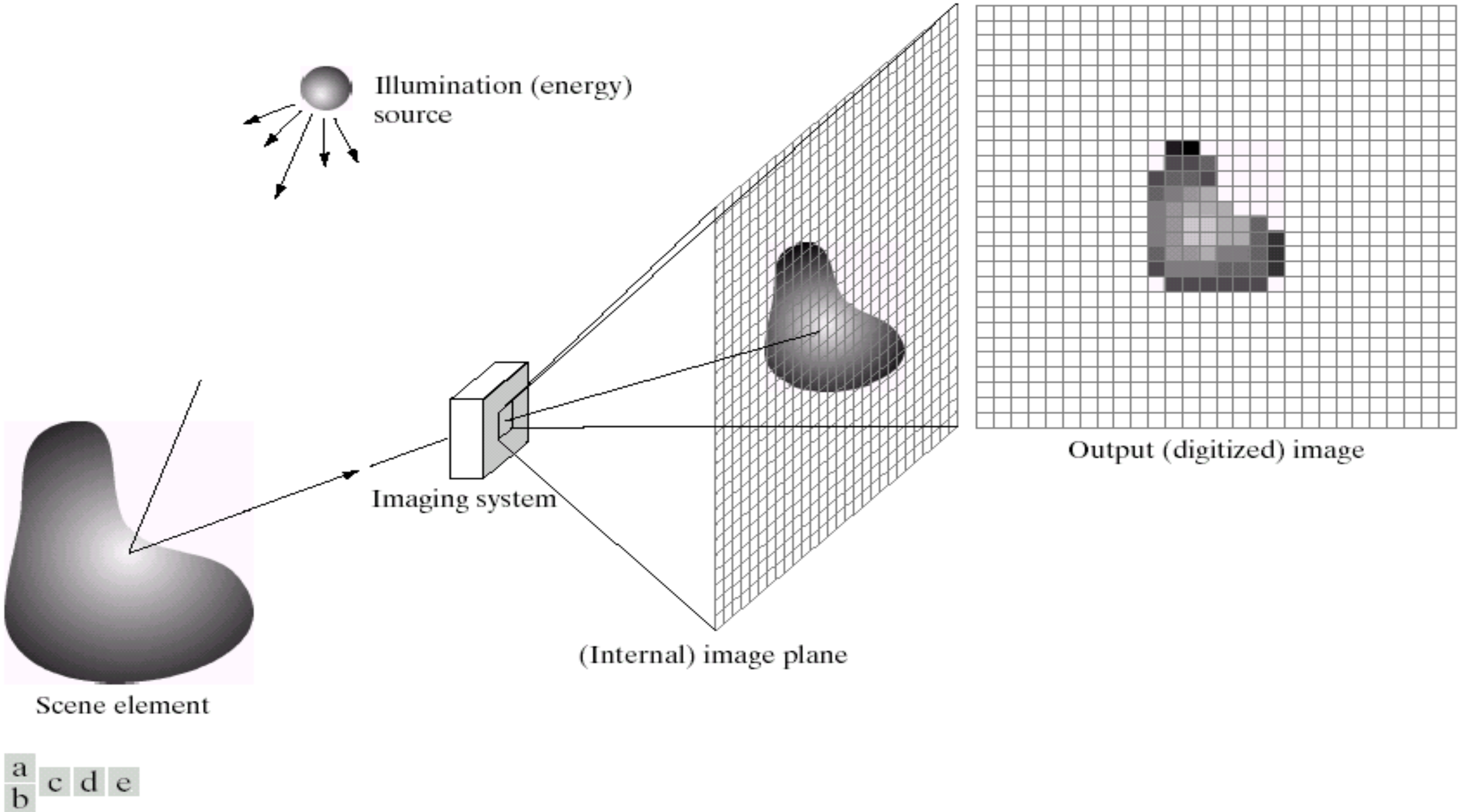
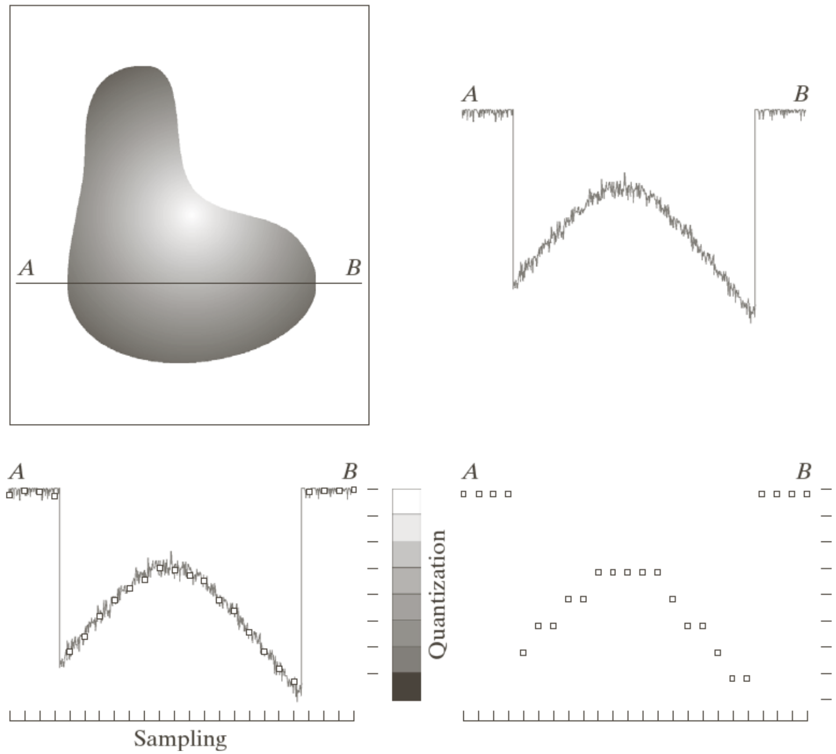


FIGURE 2.15 An example of the digital image acquisition process. (a) Energy ("illumination") source. (b) An element of a scene. (c) Imaging system. (d) Projection of the scene onto the image plane. (e) Digitized image.

Cont..



a b
c d

FIGURE 2.16 Generating a digital image. (a) Continuous image. (b) A scan line from A to B in the continuous image, used to illustrate the concepts of sampling and quantization. (c) Sampling and quantization. (d) Digital scan line.

Sampling and Quantization

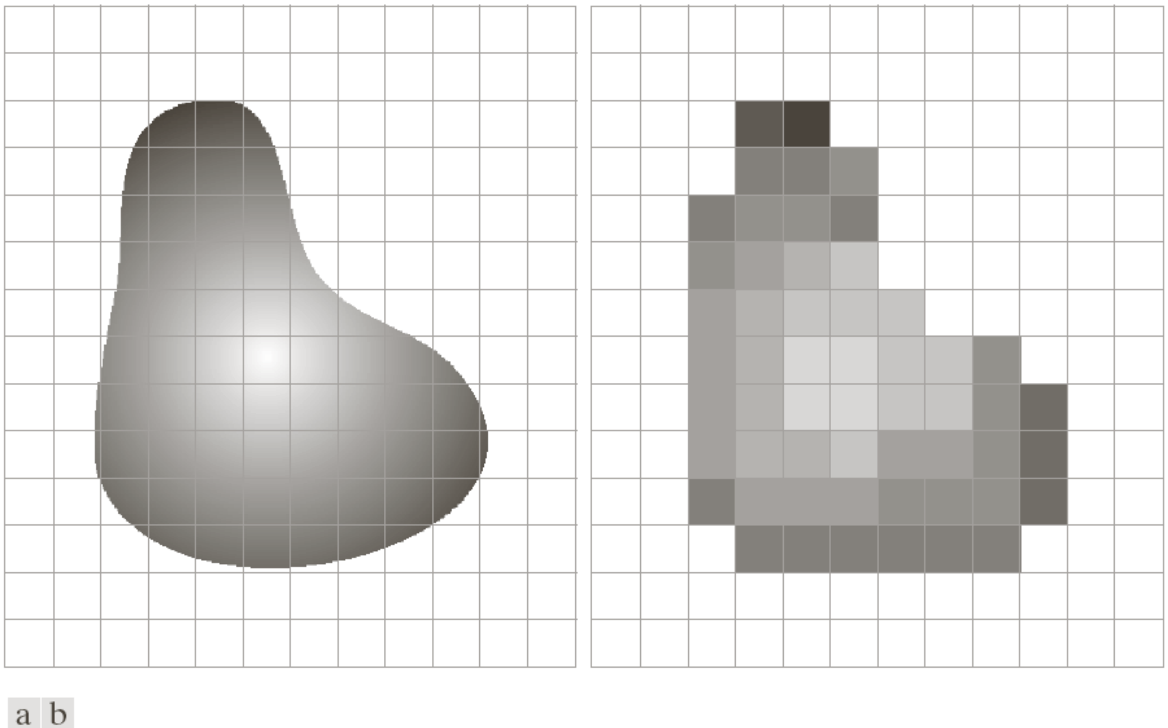


FIGURE 2.17 (a) Continuous image projected onto a sensor array. (b) Result of image sampling and quantization.

Sampling and Quantization

- How many samples to take?
  - Number of pixels (samples) in the image
  - Nyquist rate
- How many gray-levels to store?
  - At a pixel position (sample), number of levels of color/intensity to be represented

Sampling and Quantization

- How many samples to take?

a b
c d

FIGURE 2.20 Typical effects of reducing spatial resolution. Images shown at: (a) 1250 dpi, (b) 300 dpi, (c) 150 dpi, and (d) 72 dpi. The thin black borders were added for clarity. They are not part of the data.

Sampling and Quantization

- How many samples to take?
  - The Nyquist Rate
  - Samples must be taken at a rate that is twice the frequency of the highest frequency component to be reconstructed.
  - Under-sampling: sampling at a rate that is too coarse, i.e., is below the Nyquist rate.
  - Aliasing: artefacts that result from under-sampling.

Sampling and Quantization

- How many gray-levels to store?

e f
g h

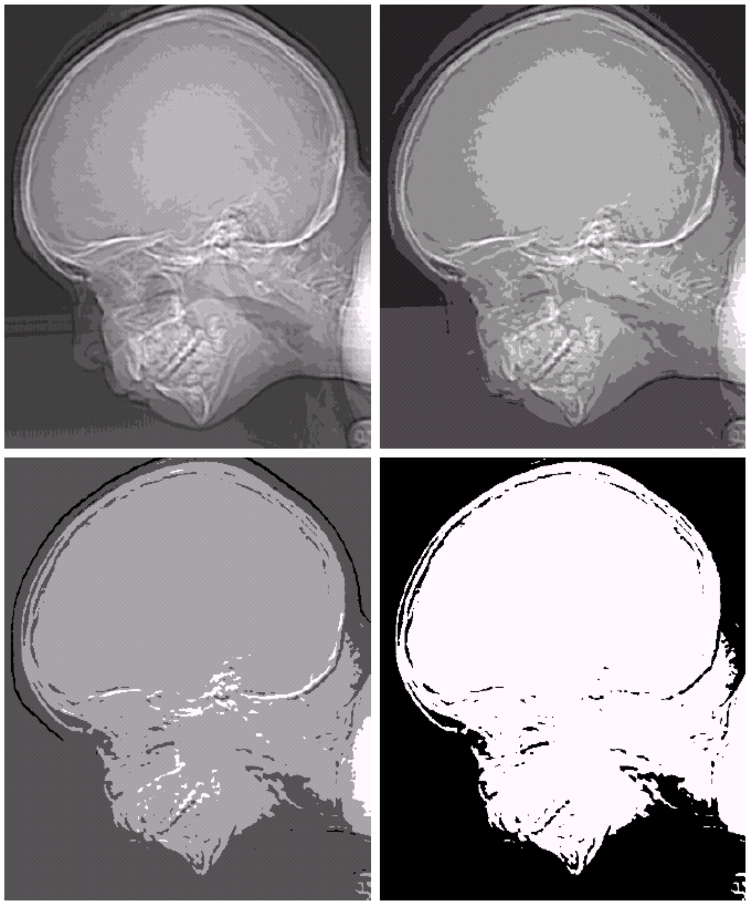
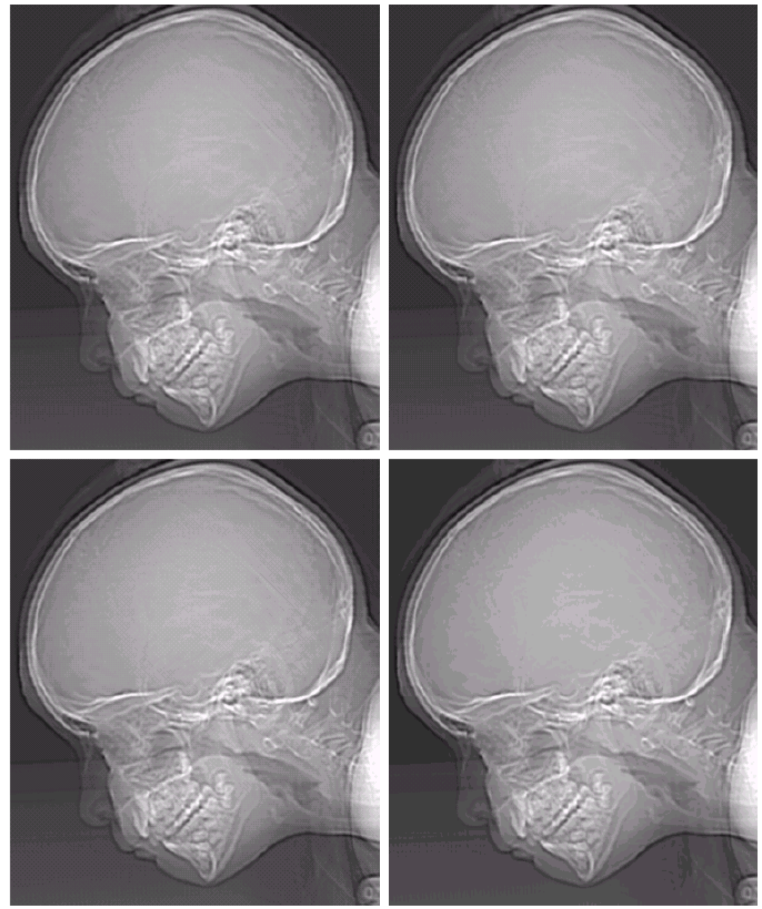


FIGURE 2.21 *(Continued)* (e)–(h) Image displayed in 16, 8, 4, and 2 gray levels. (Original courtesy of Dr. David R. Pickens, Department of Radiology & Radiological Sciences, Vanderbilt University Medical Center.)



a b
c d

FIGURE 2.21 (a) 452 × 374, 256-level image. (b)–(d) Image displayed in 128, 64, and 32 gray levels, while keeping the spatial resolution constant.

Sampling and Quantization

- Non-uniform sampling
- Non-uniform quantization

Basic relationships between pixels

- Neighbors
- Connectivity

Simple intensity processing

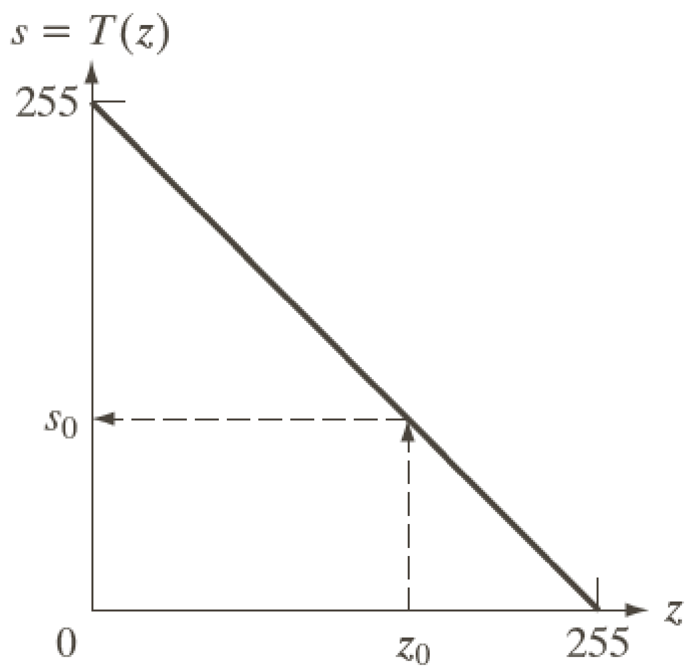


FIGURE 2.34 Intensity transformation function used to obtain the negative of an 8-bit image. The dashed arrows show transformation of an arbitrary input intensity value z_0 into its corresponding output value s_0.

Color Imaging

- Light

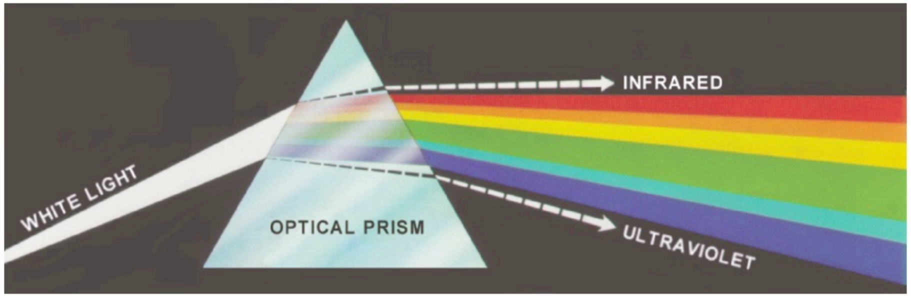


FIGURE 6.1 Color spectrum seen by passing white light through a prism. (Courtesy of the General Electric Co., Lamp Business Division.)

Color Imaging

- Visible light spectrum

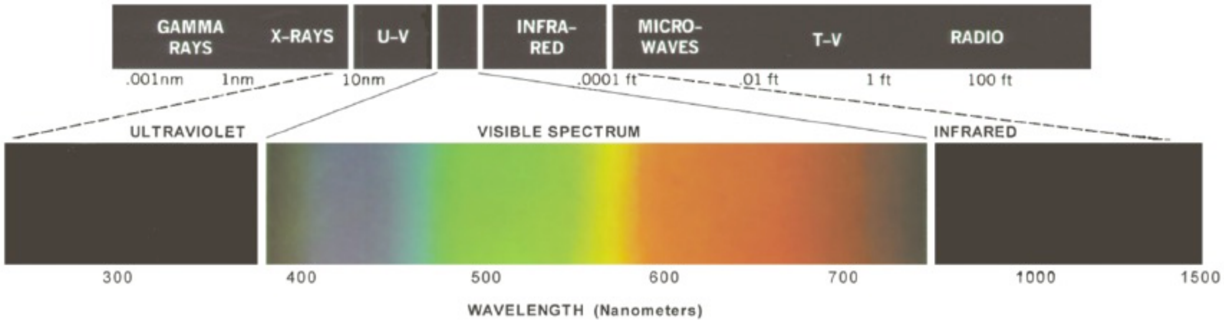


FIGURE 6.2 Wavelengths comprising the visible range of the electromagnetic spectrum. (Courtesy of the General Electric Co., Lamp Business Division.)

Color Imaging

- Trichromacy and human color vision

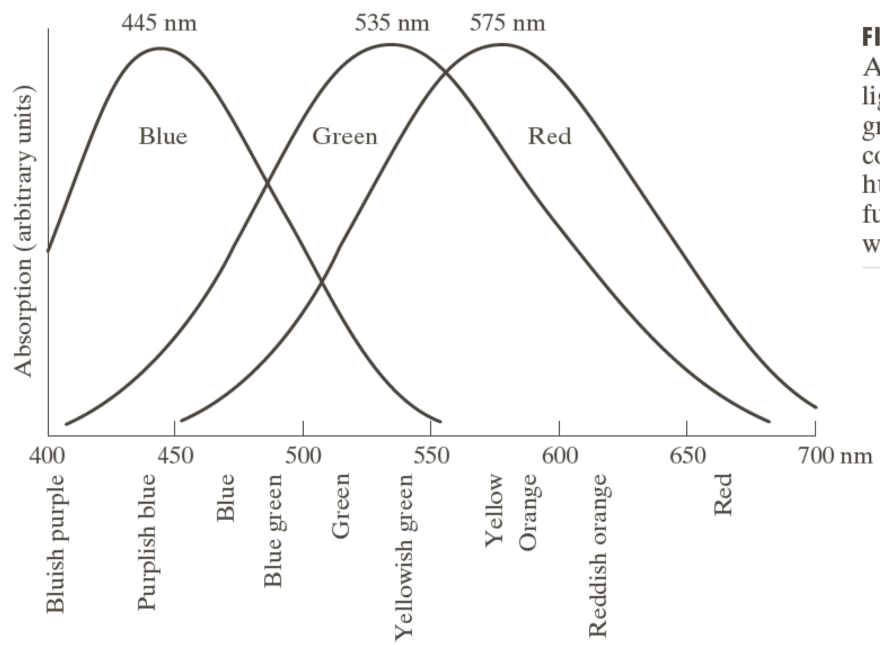
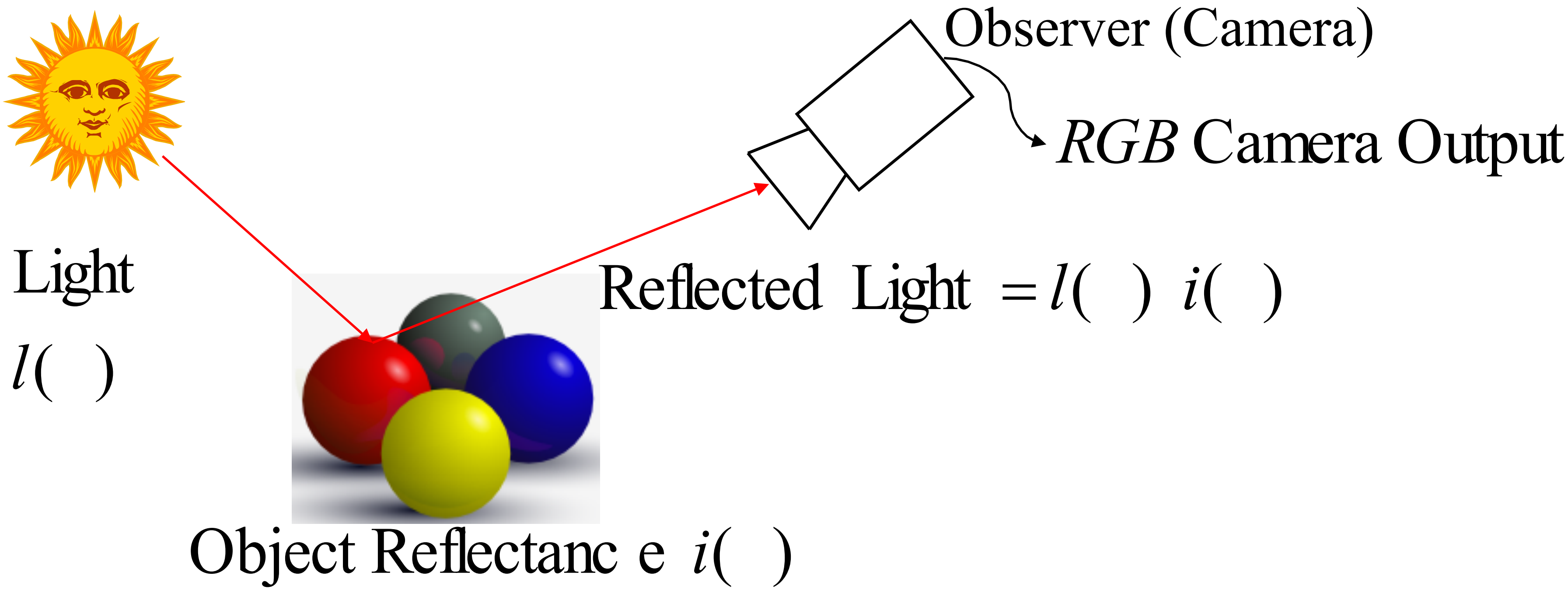


FIGURE 6.3 Absorption of light by the red, green, and blue cones in the human eye as a function of wavelength.

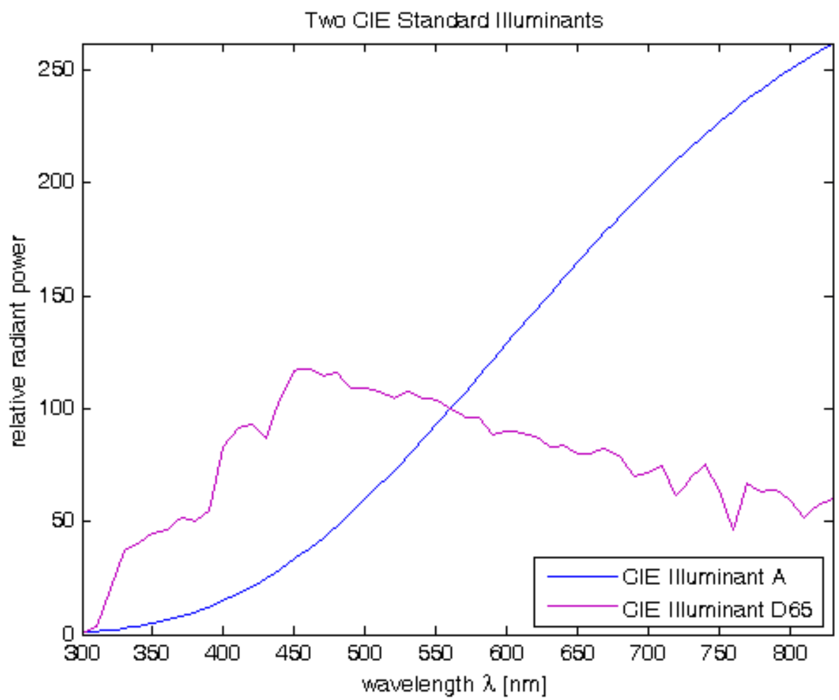
Color Imaging

- Color image formation (acquisition)



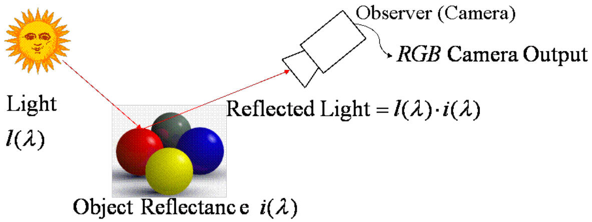
Color Imaging

- Power spectrum of standard illuminants



Color Imaging

- Color image formation (acquisition)



$$R(x,y) = \quad l(\quad)\ i(x,y,\quad)\ F_R(\quad)d$$

$$G(x,y) = \quad l(\quad)\ i(x,y,\quad)\ F_G(\quad)d$$

$$B(x,y) = \quad l(\quad)\ i(x,y,\quad)\ F_B(\quad)d$$

Color filters
Of the sensor

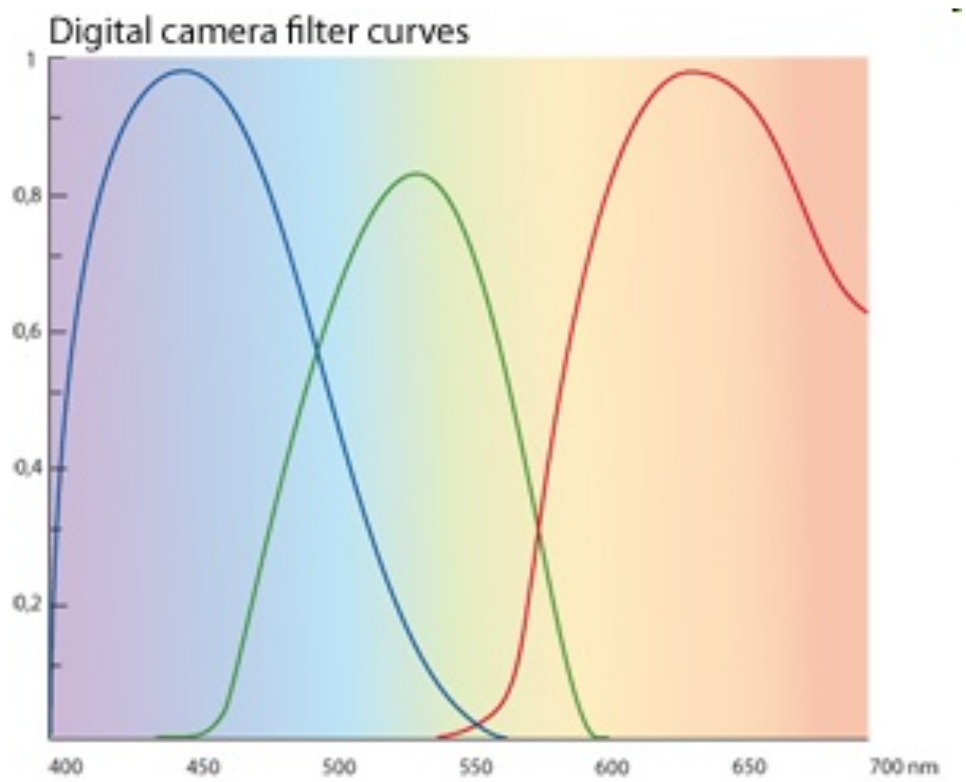
Color Imaging

- Color image formation (acquisition)

$$R(x,y) = \ l(\ )\ i(x,y,\ )\ F_R(\ )d$$

$$G(x,y) = \ l(\ )\ i(x,y,\ )\ F_G(\ )d$$

$$B(x,y) = \ l(\ )\ i(x,y,\ )\ F_B(\ )d$$



Color Imaging

- The RGB Color Model
  - R, G, B at 3 axis ranging in [0 1] each
  - Gray scale along the diagonal
  - If each component is quantized into 256 levels [0:255], the total number of different colors that can be produced is $(2^8)^3 = 2^{24} = 16,777,216$ colors.

Color Imaging

- The RGB Color Model

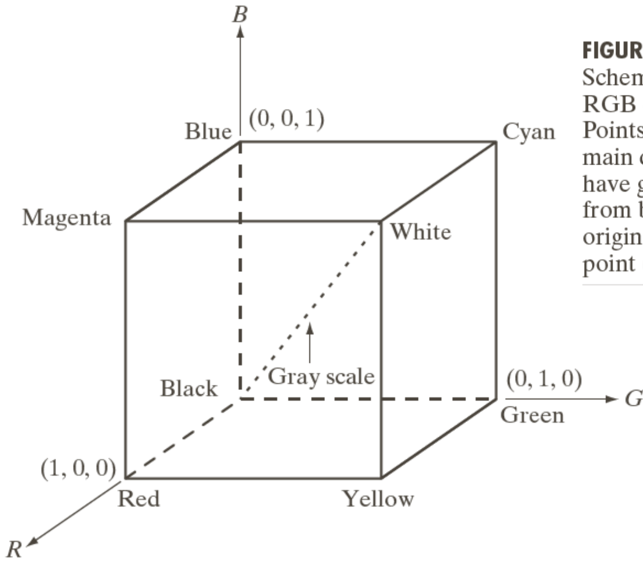


FIGURE 6.7 Schematic of the RGB color cube. Points along the main diagonal have gray values, from black at the origin to white at point (1, 1, 1).

FIGURE 6.8 RGB 24-bit color cube.

Color Imaging

- The YIQ Color Model
  - Video (NTSC) standard
  - Y encodes luminance; I and Q encode chrominance (“color”)
  - Black and white TV shows only the Y channel
  - Backward compatibility; efficiency

$$\begin{cases} Y &= 0.299 \times R + 0.587 \times G + 0.114 \times B \\ I &= 0.596 \times R - 0.274 \times G - 0.322 \times B \\ Q &= 0.212 \times R - 0.523 \times G + 0.311 \times B \end{cases}$$

Color Imaging

- Color Models, YCbCr

$$
\begin{cases}
Y &= 0.2989 \times R + 0.5866 \times G + 0.1145 \times B \\
Cb &= -0.1688 \times R - 0.3312 \times G + 0.5000 \times B \\
Cr &= 0.5000 \times R - 0.4184 \times G - 0.0816 \times B
\end{cases}
$$

Color Imaging

- Color image representation (in RGB space)

Red Green Blue

Color Imaging

- Color image representation (in RGB space)

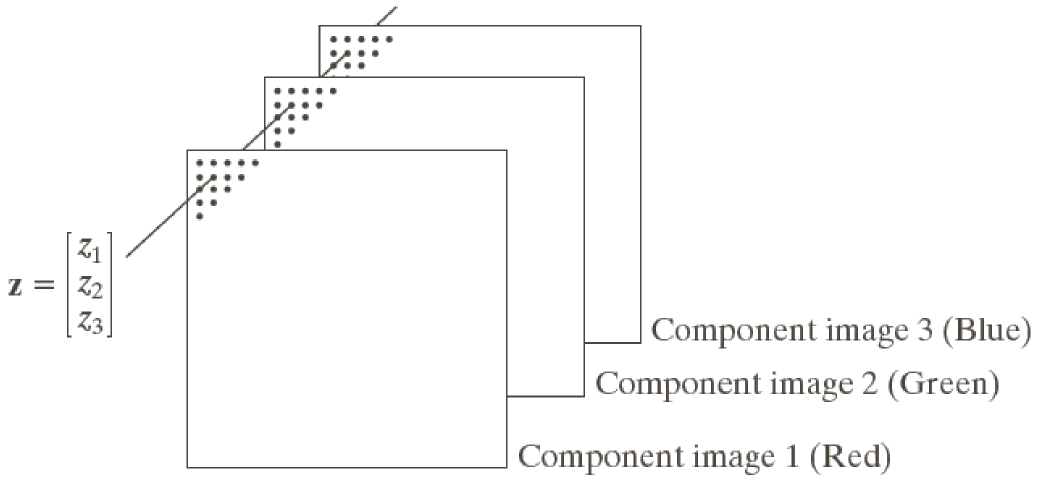


FIGURE 2.38 Formation of a vector from corresponding pixel values in three RGB component images.

Edge Detection

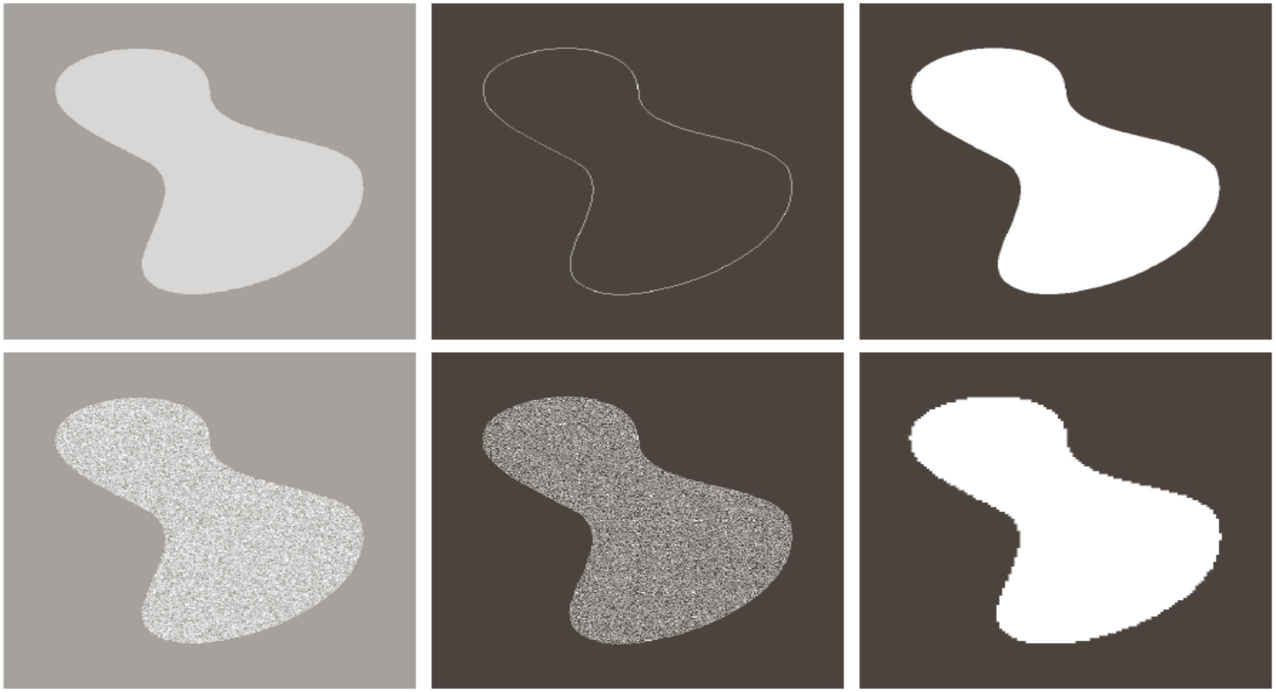
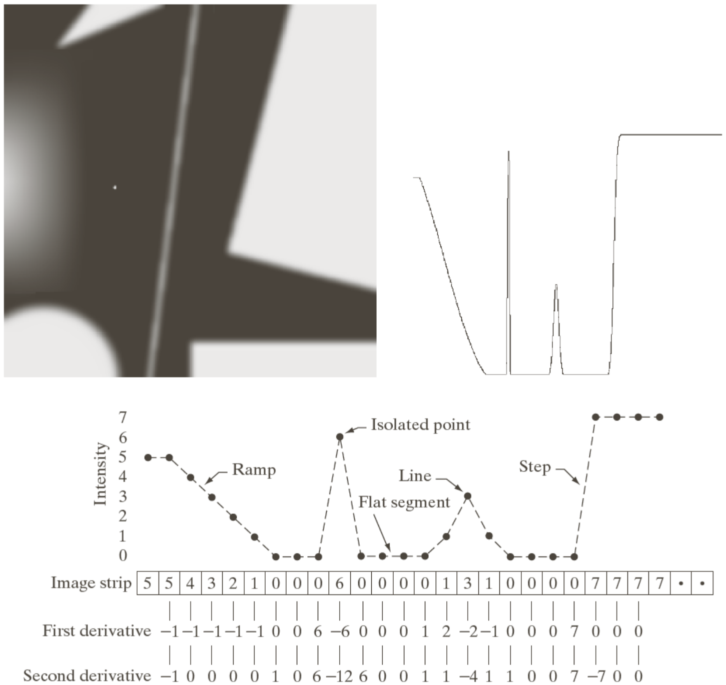


FIGURE 10.1 (a) Image containing a region of constant intensity. (b) Image showing the boundary of the inner region, obtained from intensity discontinuities. (c) Result of segmenting the image into two regions. (d) Image containing a textured region. (e) Result of edge computations. Note the large number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

Cont..

- Detection of discontinuities
  - Points
  - Lines
  - Edges

Cont..



a b
c

FIGURE 10.2 (a) Image. (b) Horizontal intensity profile through the center of the image, including the isolated noise point. (c) Simplified profile (the points are joined by dashes for clarity). The image strip corresponds to the intensity profile, and the numbers in the boxes are the intensity values of the dots shown in the profile. The derivatives were obtained using Eqs. (10.2-1) and (10.2-2).

Cont..

- Detection of discontinuities

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

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

FIGURE 10.3 A general 3×3 spatial filter mask.

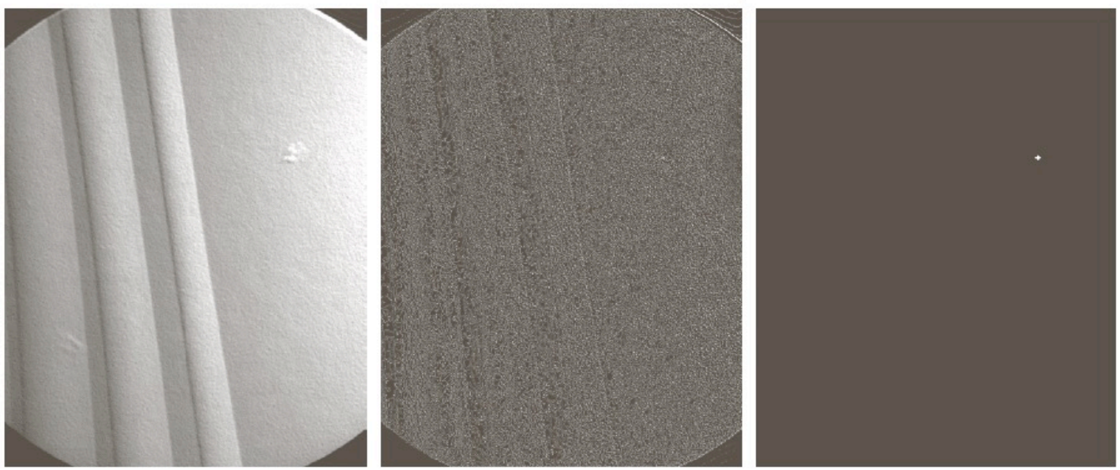
Z_i corresponding pixel values

Cont..

- Point detection

1	1	1
1	−8	1
1	1	1

$|R| \quad T$



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.)

Cont..

- Line detection

−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°

FIGURE 10.6 Line detection masks. Angles are with respect to the axis system in Fig. 2.18(b).

Cont..

- Line detection

−1	−1	−1
2	2	2
−1	−1	−1

Horizontal

2	−1	−1
−1	2	−1
−1	−1	2

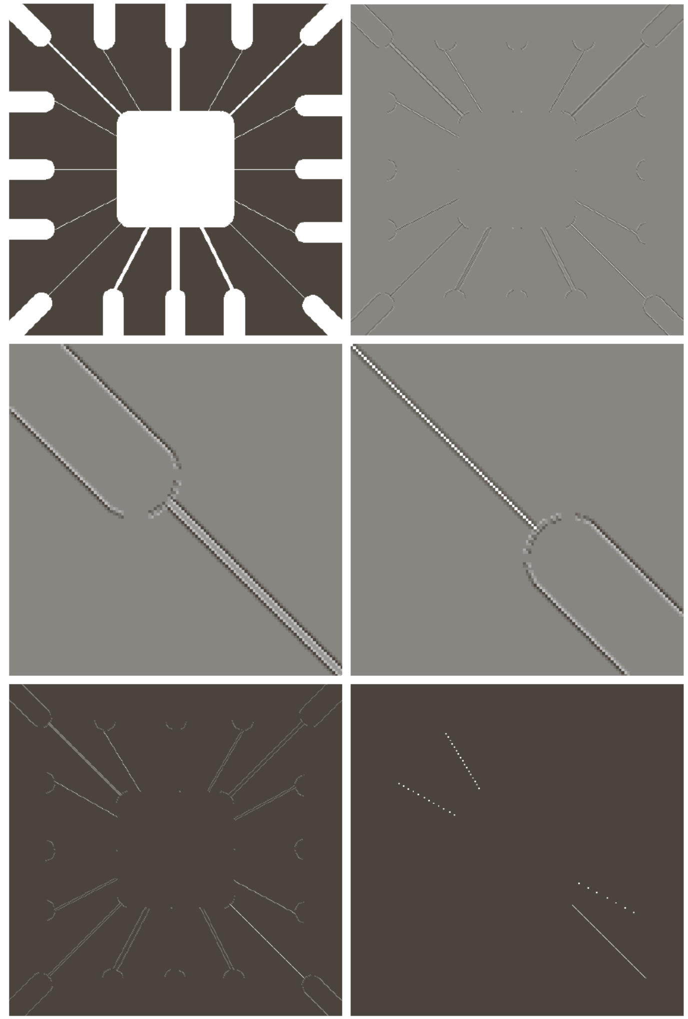
+45°

−1	2	−1
−1	2	−1
−1	2	−1

Vertical

−1	−1	2
−1	2	−1
2	−1	−1

−45°

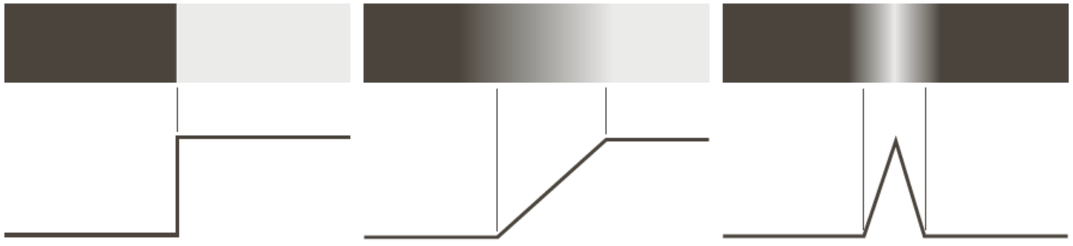


a b
c d
e f

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.)

Cont..

- Edge detection



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.

Edge Detection

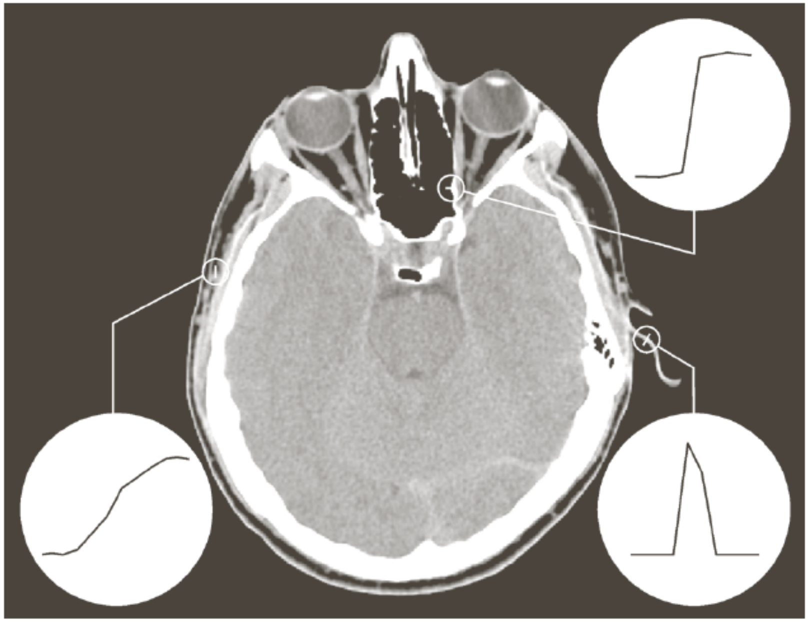


FIGURE 10.9 A 1508 × 1970 image showing (zoomed) actual ramp (bottom, left), step (top, right), and roof edge profiles. The profiles are from dark to light, in the areas indicated by the short line segments shown in the small circles. The ramp and "step" profiles span 9 pixels and 2 pixels, respectively.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

Edge Detection

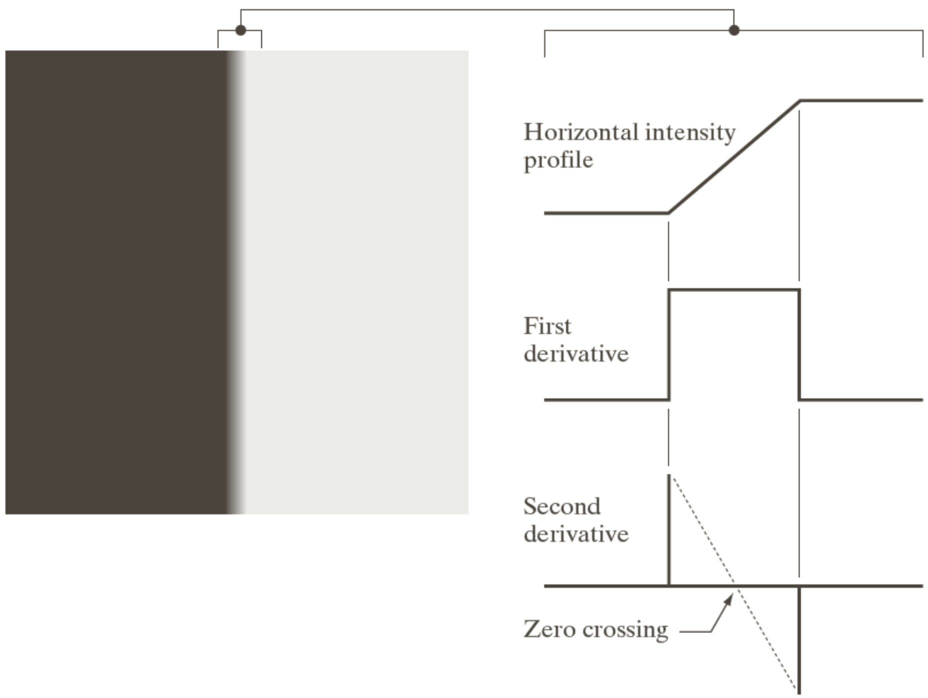


FIGURE 10.10 (a) Two regions of constant intensity separated by an ideal vertical ramp edge. (b) Detail near the edge, showing a horizontal intensity profile, together with its first and second derivatives.

Edge Detection

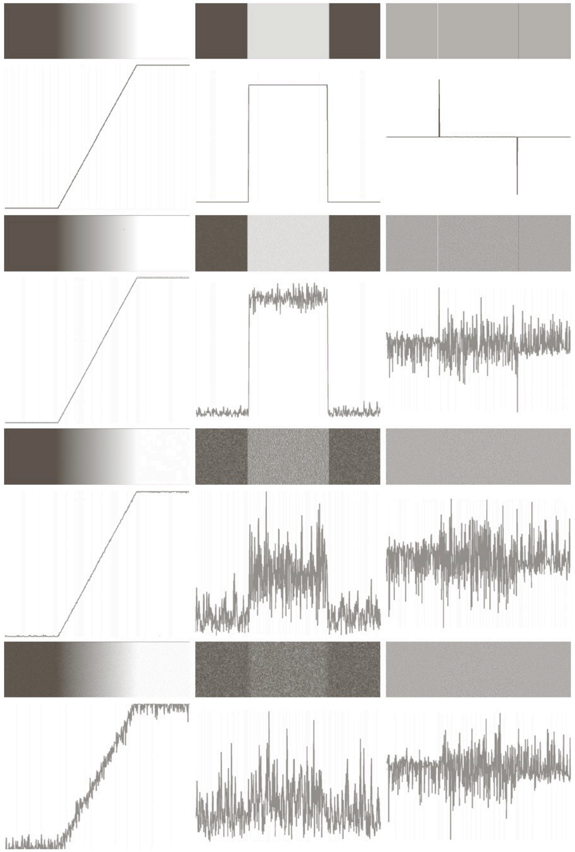


FIGURE 10.11 First column: Images and intensity profiles of a ramp edge corrupted by random Gaussian noise of zero mean and standard deviations of 0.0, 0.1, 1.0, and 10.0 intensity levels, respectively. Second column: First-derivative images and intensity profiles. Third column: Second-derivative images and intensity profiles.

Edge Detection

– Gradient operators

$$f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{f}{x} \\ \frac{f}{y} \end{bmatrix}$$

– Magnitude of the gradient

$$f = \left(G^2{}_x + G^2{}_y\right)^{1/2} \qquad f \quad \left|G_x\right| + \left|G_y\right|$$

– Direction of the gradient vector

$$(x, y) = \tan^{\;1} \frac{G_y}{G_x}$$

Edge Detection

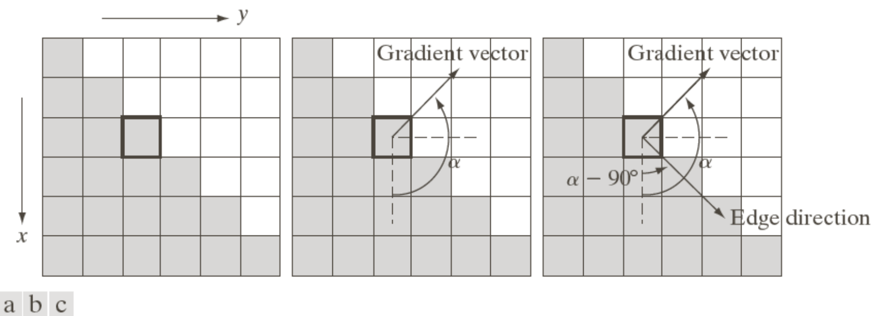


FIGURE 10.12 Using the gradient to determine edge strength and direction at a point. Note that the edge is perpendicular to the direction of the gradient vector at the point where the gradient is computed. Each square in the figure represents one pixel.

Edge Detection

G_y

−1	1

G_x

−1
1

a b

FIGURE 10.13 One-dimensional masks used to implement Eqs. (10.2-12) and (10.2-13).

Edge Detection

0	1	1
−1	0	1
−1	−1	0

−1	−1	0
−1	0	1
0	1	1

Prewitt

0	1	2
−1	0	1
−2	−1	0

−2	−1	0
−1	0	1
0	1	2

Sobel

a b
c d

FIGURE 10.15 Prewitt and Sobel masks for detecting diagonal edges.

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

G_x

−1	0
0	1

G_y

0	−1
1	0

Roberts

−1	−1	−1
0	0	0
1	1	1

−1	0	1
−1	0	1
−1	0	1

Prewitt

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

Sobel

a
b c
d e
f g

FIGURE 10.14 A 3 × 3 region of an image (the z's are intensity values) and various masks used to compute the gradient at the point labeled z_5.

Edge Detection

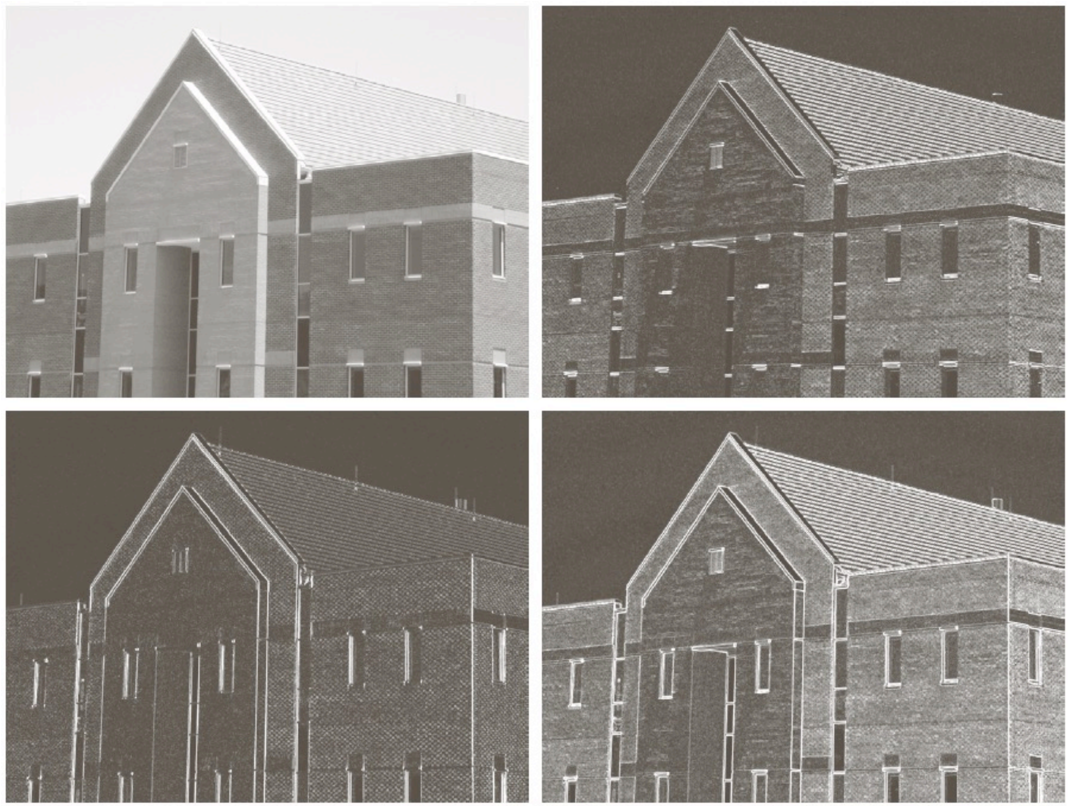
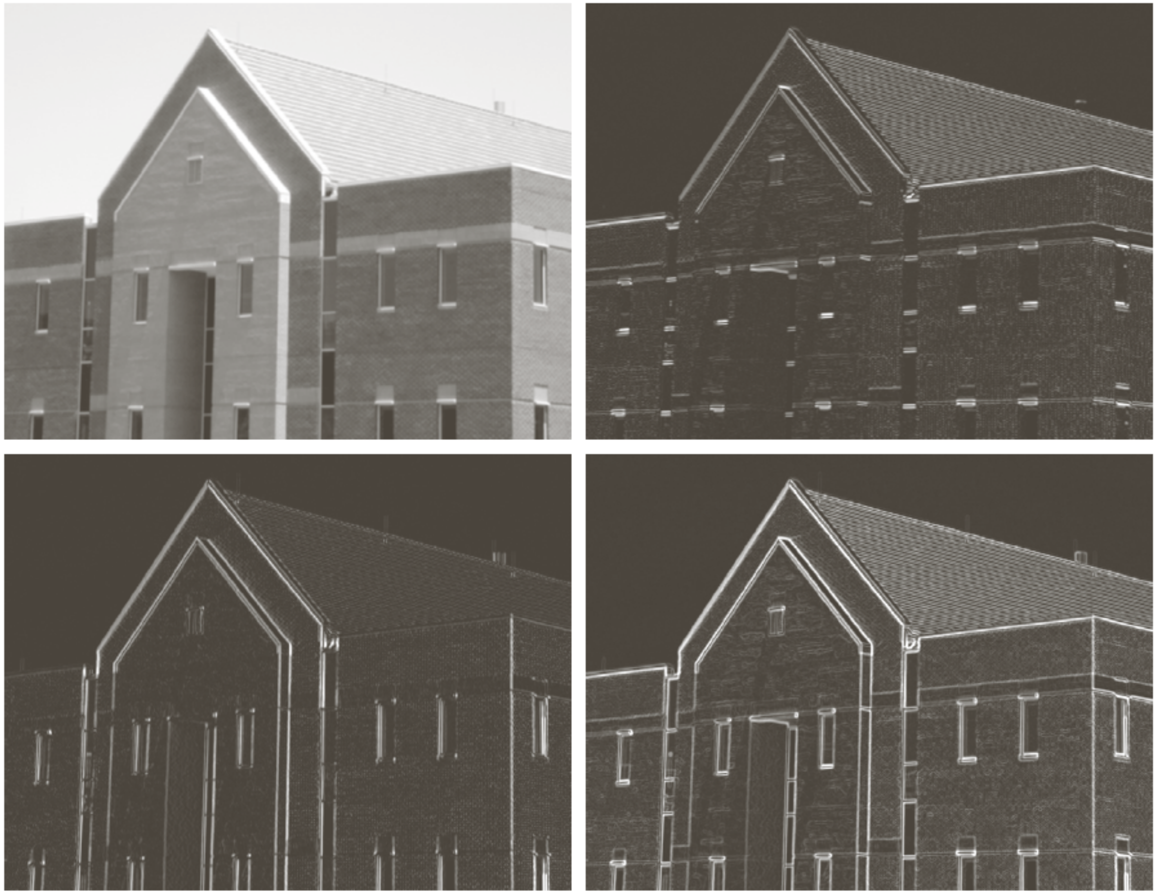


FIGURE 10.16 (a) Original image of size 834×1114 pixels, with intensity values scaled to the range [0, 1]. (b) $|g_x|$, the component of the gradient in the x-direction, obtained using the Sobel mask in Fig. 10.14(f) to filter the image. (c) $|g_y|$, obtained using the mask in Fig. 10.14(g). (d) The gradient image, $|g_x| + |g_y|$.

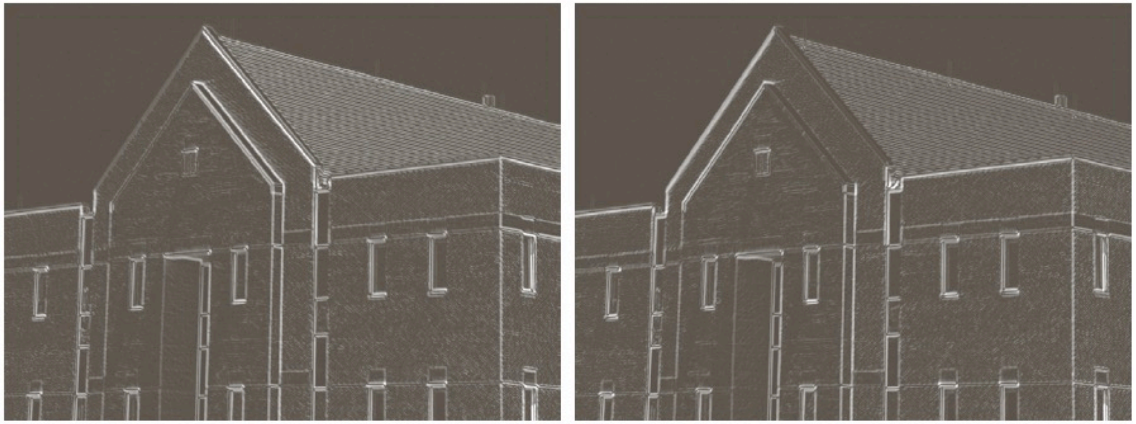
Edge Detection



a b
c d

FIGURE 10.18 Same sequence as in Fig. 10.16, but with the original image smoothed using a 5×5 averaging filter prior to edge detection.

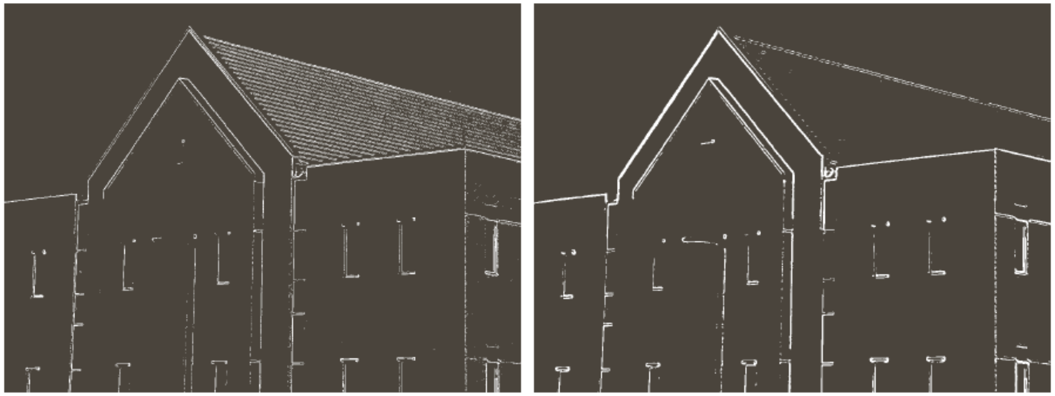
Edge Detection



a b

FIGURE 10.19 Diagonal edge detection. (a) Result of using the mask in Fig. 10.15(c). (b) Result of using the mask in Fig. 10.15(d). The input image in both cases was Fig. 10.18(a).

Edge Detection



a b

FIGURE 10.20 (a) Thresholded version of the image in Fig. 10.16(d), with the threshold selected as 33% of the highest value in the image; this threshold was just high enough to eliminate most of the brick edges in the gradient image. (b) Thresholded version of the image in Fig. 10.18(d), obtained using a threshold equal to 33% of the highest value in that image.

Edge Detection

– Laplacian

- Laplacian of a 2d function f(x,y) is a 2^{nd} order derivative defined as

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

- Masks used to compute Lapla

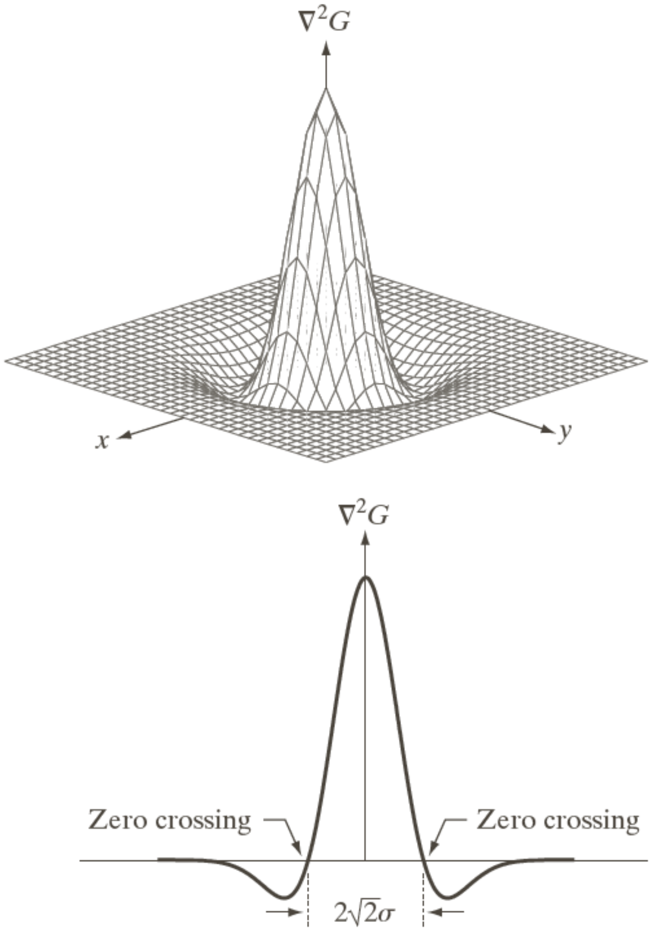
0	–1	0
–1	4	–1
0	–1	0

–1	–1	–1
–1	8	–1
–1	–1	–1

Edge Detection

- Laplacian of Gaussian (LoG)
  - Because these kernels are approximating a second derivative measurement on the image, they are very sensitive to noise. To counter this, the image is often Gaussian smoothed before applying the Laplacian filter. This pre-processing step reduces the high frequency noise components prior to the differentiation step.
  - In fact, since the convolution operation is associative, we can convolve the Gaussian smoothing filter with the Laplacian filter first, and then convolve this hybrid filter with the image to achieve the required result. Doing things this way has two advantages:
    - Since both the Gaussian and the Laplacian kernels are usually much smaller than the image, this method usually requires far fewer arithmetic operations.
    - The LoG (`Laplacian of Gaussian') kernel can be precalculated in advance so only one convolution needs to be performed at run-time on the image.

Edge Detection



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.

Edge Detection and Image Segmentation

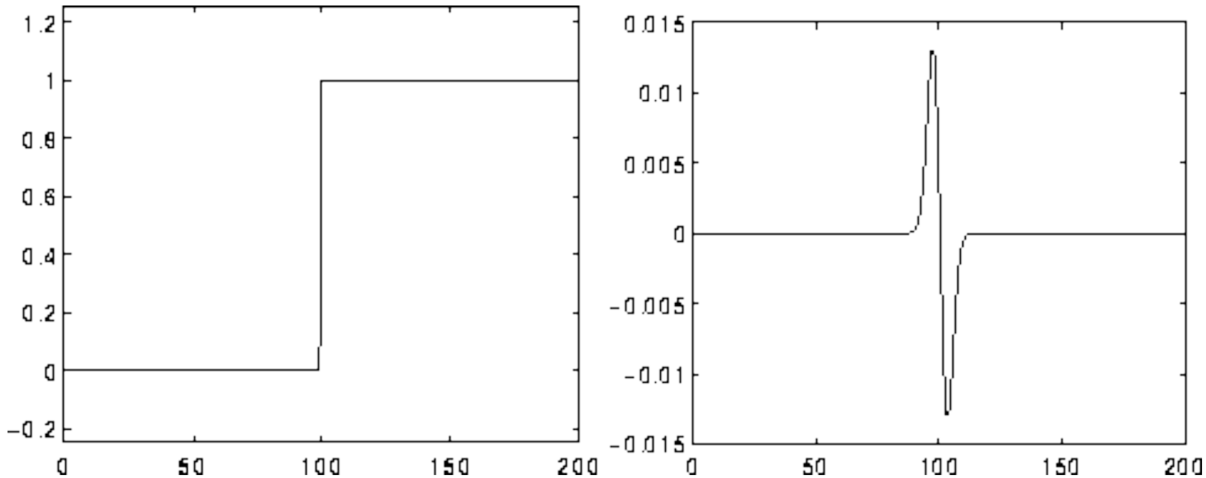
- Edge detection
  - Zero Crossing Detector (http://homepages.inf.ed.ac.uk/rbf/HIPR2/zeros.htm)
  - The zero crossing detector looks for places in the Laplacian of an image where the value of the Laplacian passes through zero - i.e. points where the Laplacian changes sign. Such points often occur at `edges' in images - i.e. points where the intensity of the image changes rapidly, but they also occur at places that are not as easy to associate with edges.
  - It is best to think of the zero crossing detector as some sort of feature detector rather than as a specific edge detector.

Edge Detection and Image Segmentation

- Edge detection
  - Zero Crossing Detector
  - The core of the zero crossing detector is the Laplacian of Gaussian filter, `edges' in images give rise to zero crossings in the LoG output.

Edge Detection and Image Segmentation

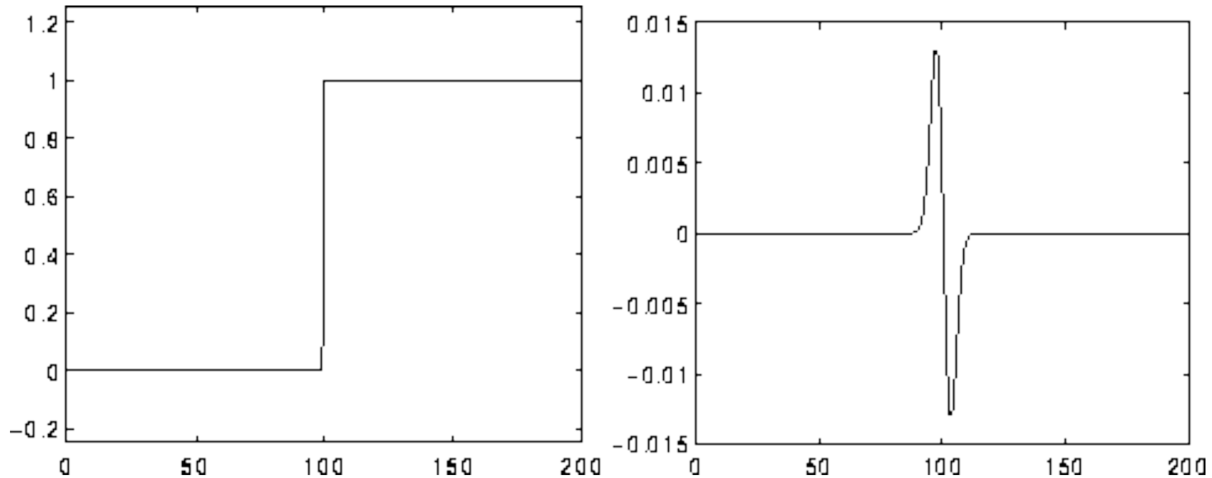
- Edge detection
  - Zero Crossing Detector



Response of 1-D LoG filter to a step edge. The left hand graph shows a 1-D image, 200 pixels long, containing a step edge. The right hand graph shows the response of a 1-D LoG filter with Gaussian standard deviation 3 pixels.

Edge Detection and Image Segmentation

- Edge detection
  - Zero Crossing Detector



Response of 1-D LoG filter to a step edge. The left hand graph shows a 1-D image, 200 pixels long, containing a step edge. The right hand graph shows the response of a 1-D LoG filter with Gaussian standard deviation 3 pixels.

Edge Detection and Image Segmentation

- Edge detection
  - Zero Crossing Detector

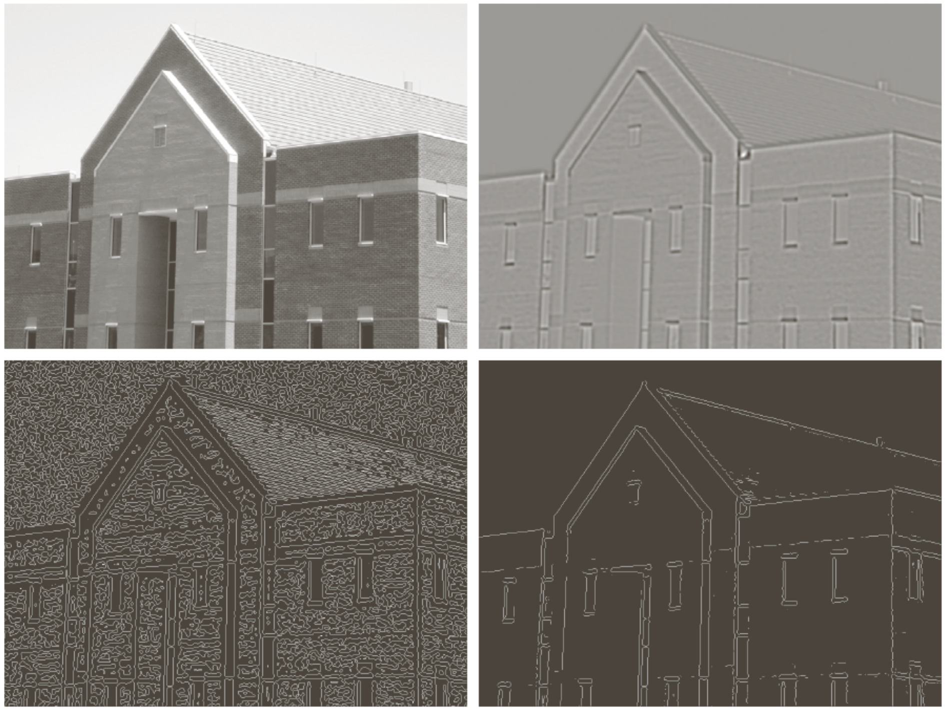


FIGURE 10.22 (a) Original image of size 834 × 1114 pixels, with intensity values scaled to the range [0, 1]. (b) Results of Steps 1 and 2 of the Marr-Hildreth algorithm using $\sigma = 4$ and $n = 25$. (c) Zero crossings of (b) using a threshold of 0 (note the closed-loop edges). (d) Zero crossings found using a threshold equal to 4% of the maximum value of the image in (b). Note the thin edges.

Edge Detection and Image Segmentation

- Edge detection
  - Canny Edge Detector (http://homepages.inf.ed.ac.uk/rbf/HIPR2/canny.htm)
  - The Canny operator works in a multi-stage process.
  - First of all the image is smoothed by Gaussian convolution.
  - Then a simple 2-D first derivative operator (somewhat like the Roberts Cross) is applied to the smoothed image to highlight regions of the image with high first spatial derivatives. Edges give rise to ridges in the gradient magnitude image.
  - The algorithm then tracks along the top of these ridges and sets to zero all pixels that are not actually on the ridge top so as to give a thin line in the output, a process known as non-maximal suppression.
  - The tracking process exhibits hysteresis controlled by two thresholds: T1 and T2, with T1 > T2.
  - Tracking can only begin at a point on a ridge higher than T1. Tracking then continues in both directions out from that point until the height of the ridge falls below T2.
  - This hysteresis helps to ensure that noisy edges are not broken up into multiple edge fragments.

Edge Detection and Image Segmentation

- Region Segmentation
  - Region-based segmentation methods attempt to partition or group regions according to common image properties. These image properties consist of
    1. Intensity values from original images, or computed values based on an image operator
    2. Textures or patterns that are unique to each type of region
    3. Spectral profiles that provide multidimensional image data
  - Elaborate systems may use a combination of these properties to segment images, while simpler systems may be restricted to a minimal set on properties depending of the type of data available.

Edge Detection and Image Segmentation

- Region Segmentation
  - Thresholding

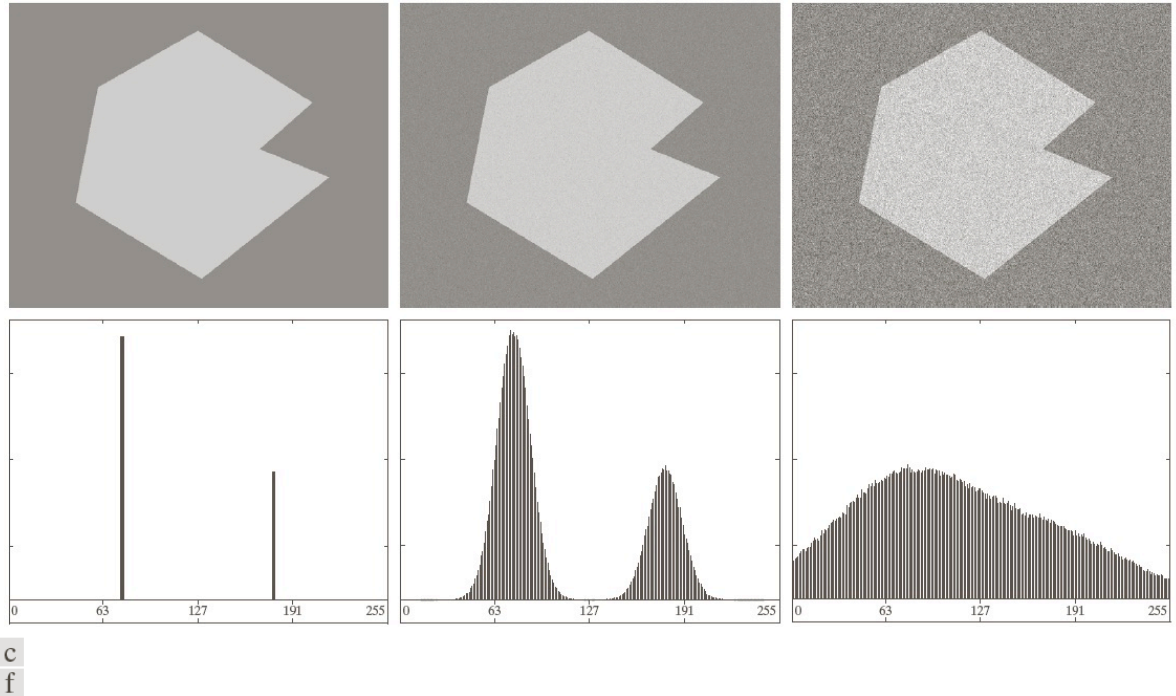


FIGURE 10.36 (a) Noiseless 8-bit image. (b) Image with additive Gaussian noise of mean 0 and standard deviation of 10 intensity levels. (c) Image with additive Gaussian noise of mean 0 and standard deviation of 50 intensity levels. (d)–(f) Corresponding histograms.

Edge Detection and Image Segmentation

- Region Segmentation
  - Thresholding

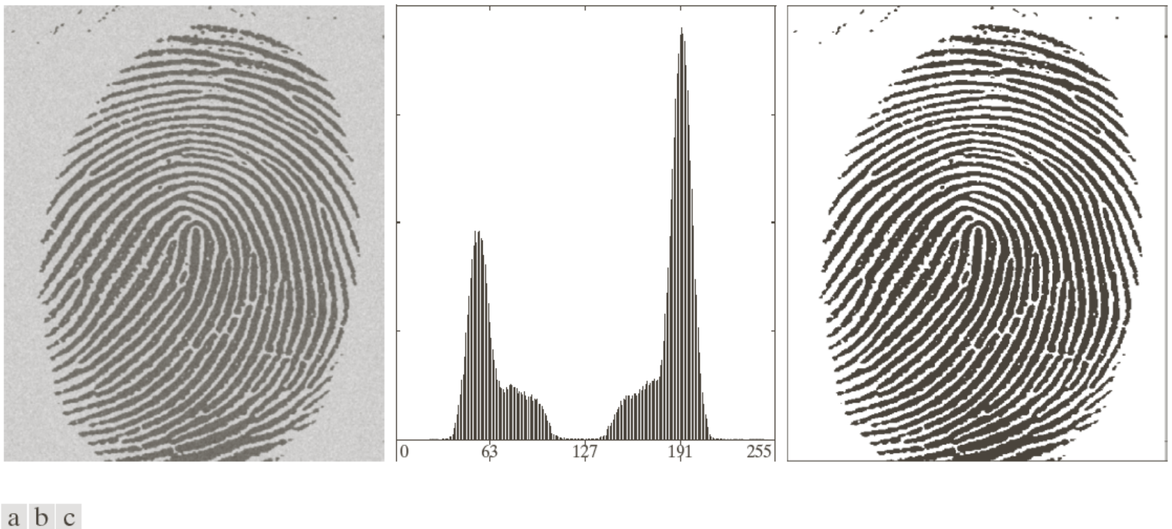


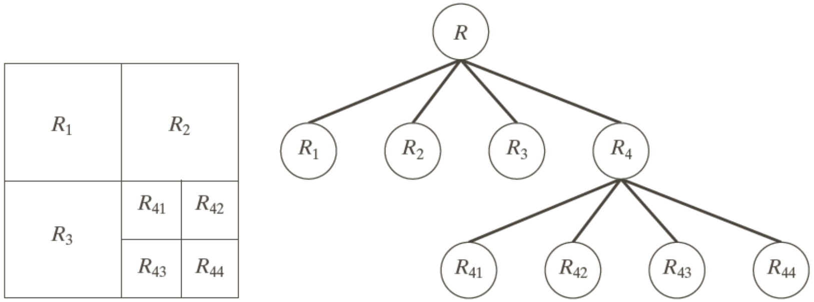
FIGURE 10.38 (a) Noisy fingerprint. (b) Histogram. (c) Segmented result using a global threshold (the border was added for clarity). (Original courtesy of the National Institute of Standards and Technology.)

Edge Detection and Image Segmentation

- Region Splitting and Merging
  - The basic idea of region splitting is to break the image into a set of disjoint regions which are coherent within themselves:
    - Initially take the image as a whole to be the area of interest.
    - Look at the area of interest and decide if all pixels contained in the region satisfy some similarity constraint.
    - If TRUE then the area of interest corresponds to a region in the image.
    - If FALSE split the area of interest (usually into four equal sub-areas) and consider each of the sub areas as the area of interest in turn.
    - This process continues until no further splitting occurs. In the worst case this happens when the areas are just one pixel in size.
    - This is a divide and conquer or top down method.
  - If only a splitting schedule is used then the final segmentation would probably contain many neighbouring regions that have identical or similar properties.
  - Thus, a merging process is used after each split which compares adjacent regions and merges them if necessary. Algorithms of this nature are called split and merge algorithms.

Edge Detection and Image Segmentation

- Region Splitting and Merging



a b

FIGURE 10.52 (a) Partitioned image. (b) Corresponding quadtree. *R* represents the entire image region.

Edge Detection and Image Segmentation

- Region Splitting and Merging

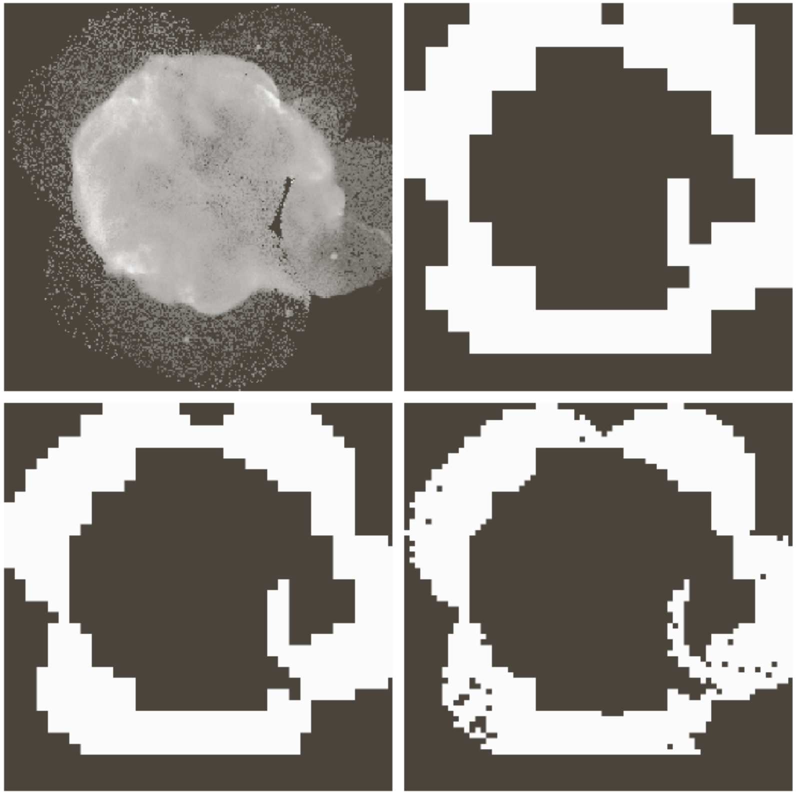


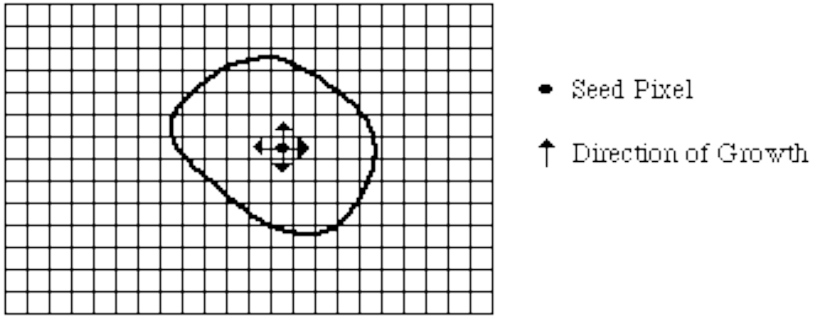
FIGURE 10.53 (a) Image of the Cygnus Loop supernova, taken in the X-ray band by NASA's Hubble Telescope. (b)–(d) Results of limiting the smallest allowed quadregion to sizes of 32×32, 16×16, and 8×8 pixels, respectively. (Original image courtesy of NASA.)

Edge Detection and Image Segmentation

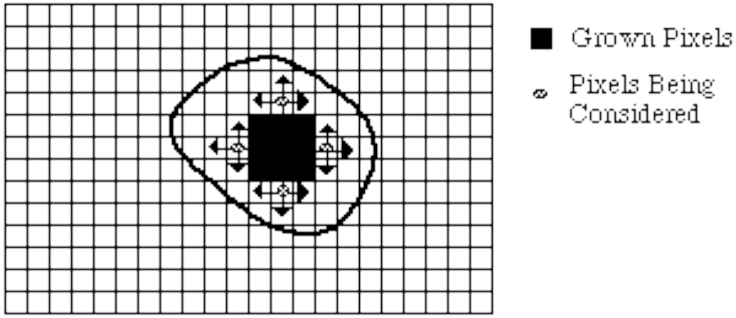
- Region Growing
  - Region growing approach is the opposite of the split and merge approach:
    - An initial set of small areas are iteratively merged according to similarity constraints.
    - Start by choosing an arbitrary seed pixel and compare it with neighbouring pixels.
    - Region is grown from the seed pixel by adding in neighbouring pixels that are similar, increasing the size of the region.
    - When the growth of one region stops we simply choose another seed pixel which does not yet belong to any region and start again.
    - This whole process is continued until all pixels belong to some region.
    - A bottom up method.

Edge Detection and Image Segmentation

- Region Growing



(a) Start of Growing a Region



(b) Growing Process After a Few Iterations

Edge Detection and Image Segmentation

- Region Growing
- However starting with a particular seed pixel and letting this region grow completely before trying other seeds biases the segmentation in favour of the regions which are segmented first.
- This can have several undesirable effects:
  - Current region dominates the growth process -- ambiguities around edges of adjacent regions may not be resolved correctly.
  - Different choices of seeds may give different segmentation results.
  - Problems can occur if the (arbitrarily chosen) seed point lies on an edge.

Edge Detection and Image Segmentation

- Region Growing
- To counter the above problems, simultaneous region growing techniques have been developed.
  - Similarities of neighbouring regions are taken into account in the growing process.
  - No single region is allowed to completely dominate the proceedings.
  - A number of regions are allowed to grow at the same time.
  - similar regions will gradually coalesce into expanding regions.
  - Control of these methods may be quite complicated but efficient methods have been developed.
  - Easy and efficient to implement on parallel computers.

Edge Detection and Image Segmentation

- Region Growing

- To counter the above problems, simultaneous region growing techniques have been developed.

  - Similarities of neighbouring regions are taken into account in the growing process.
  - No single region is allowed to completely dominate the proceedings.
  - A number of regions are allowed to grow at the same time.
  - similar regions will gradually coalesce into expanding regions.
  - Control of these methods may be quite complicated but efficient methods have been developed.
  - Easy and efficient to implement on parallel computers.

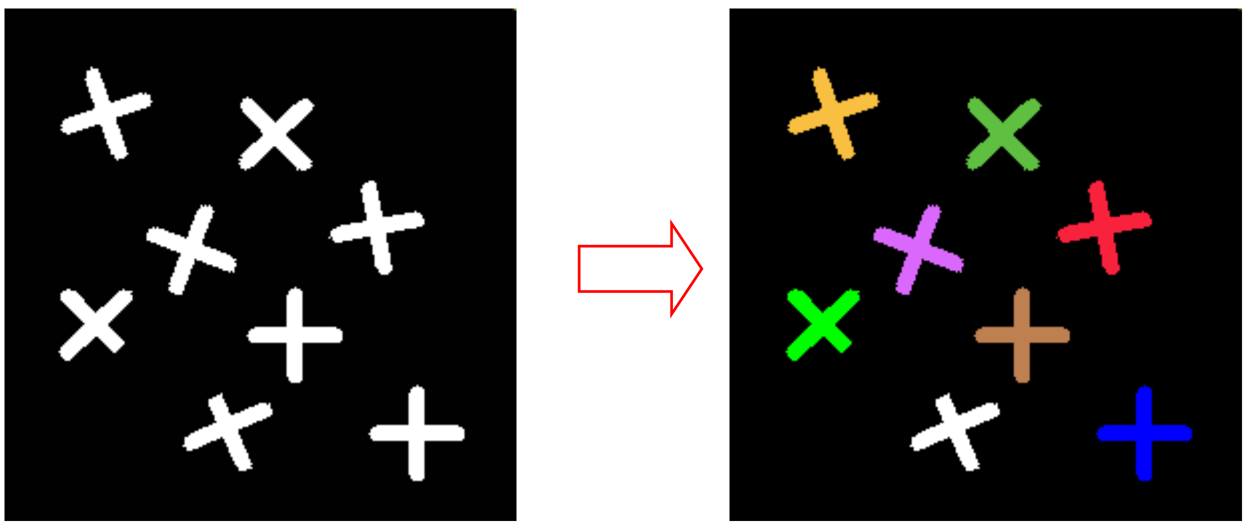
Edge Detection and Image Segmentation

- Advanced Image Segmentation Methods

Segment the foreground objects

Identify Individual Object

- Connected Component Labeling: Give pixels belonging to the same object the same label (grey level or color)



Identify Individual Object

- Connected component labelling works by scanning an image, pixel-by-pixel (from top to bottom and left to right) in order to identify **connected** pixel regions, i.e. regions of adjacent pixels which share the same set of intensity values.

Identify Individual Object

- Connected Pixels
  - Two pixels p and q are said to be connected if there is a sequence of foreground (1) pixels

$$p_0, p_1, \ldots p_n$$

  - Such that

$$p_0 = p$$

$$p_n = q$$

$$p_i \text{ is a neighbor of } p_{i\ 1}$$

Identify Individual Object

- Pixel Neighbors

0	1	0
1	p	1
0	1	0

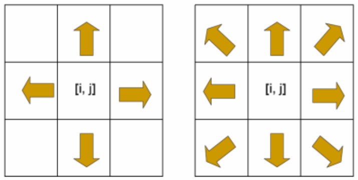
Four nearest neighbors

1	1	1
1	p	1
1	1	1

Eight nearest neighbors

Identify Individual Object

- 4- and 8-Connected Pixels
  - When only the four nearest neighbors are considered part of the neighborhood, then pixels p and q are said to be "4-connected"
  - When the 8 nearest neighbors are considered part of the neighborhood, then pixels p and q are said to be "8-connected"



Labelling Algorithm

- The connected components labeling operator scans the image by moving along a row until it comes to a point p (where p denotes the pixel to be labeled at any stage in the scanning process) for which V={1}. When this is true, it examines the four neighbors of p which have already been encountered in the scan (in the case of 8-connectivity, the neighbors (i) to the left of p, (ii) above it, and (iii and iv) the two upper diagonal terms). Based on this information, the labeling of p occurs as follows:
  - If all four neighbors are 0, assign a new label to p, else
  - if only one neighbor has V={1}, assign its label to p, else
  - if more than one of the neighbors have V={1}, assign one of the labels to p and make a note of the equivalences.
- After completing the scan, the equivalent label pairs are sorted into equivalence classes and a unique label is assigned to each class.
- As a final step, a second scan is made through the image, during which each label is replaced by the label assigned to its equivalence classes. For display, the labels might be different graylevels or colors.

Labelling Algorithm

- Example (8-connectivity)

1	1		1	1	
	1	1			
		1	1		
1				1	
1					
1	1	1			

Original Binary Image

1	1		2	2	
	1	1			
		1	1		
3				1	
3					
3	3	3			

After first scan

Equivalent labels

{1, 2}

Labelling Algorithm

- Example (8-connectivity)

1	1		2	2	
	1	1			
		1	1		
3				1	
3					
3	3	3			

Equivalent labels

{1, 2}

After 1st scan

1	1		1	1	
	1	1			
		1	1		
2				1	
2					
2	2	2			

Final label

1: {1, 2}
2: {3}

After 2nd scan
(two individual objects/regions have been identified)

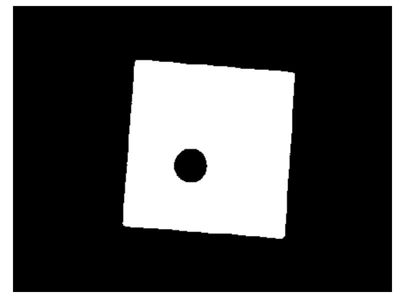
Describe Objects

- Some useful features can be extracted once we have connected components, including
  - Area
  - Centroid
  - Extremal points, bounding box
  - Circularity
  - Spatial moments

Describe Objects

- Area

$$A = \underset{(x,y)\quad R}{\qquad 1}$$



- Centroid

$$\bar{x} = \frac{1}{A} \underset{(x,y)\quad R}{\qquad x}$$

$$\bar{y} = \frac{1}{A} \underset{(x,y)\quad R}{\qquad y}$$

Digital Filters

In image processing filters are mainly used to suppress:

- high frequencies in the image
  (i.e. smoothing the image)
- low frequencies
  (i.e. enhancing or detecting edges in the image)

An image can be filtered either in the frequency or in the spatial domain.

Linear Filtering

Low Pass Filters ("smoothing”):

- remove high spatial frequency noise from a digital image.

Reconstruction filtering
Enhancement filtering

Moving Window Operations

- affects one pixel of the image at a time, changing its value by some function of a "local" region of pixels ("covered" by the window).

Neighborhood-averaging filters
Median filters
Mode filters

Cont..

Neighborhood-averaging filters :

- replace the value of each pixel, a[i,j] say, by a weighted-average of the pixels in some neighborhood around it.

(i.e.: ***a*** weighted sum of **a[i+p,j+q],** with **p = -k** to **k, q = -k** to **k** for some positive k; the weights are non-negative with the highest weight on the **p = q = 0** term. If all the weights are equal then this is a mean filter is "linear")

Median filters:

- replaces each pixel value by the median of its neighbors.

(i.e. the value such that 50% of the values in the neighborhood are above, and 50% are below.)

Mode filters:

- each pixel value is replaced by its most common neighbor. This is a particularly useful filter for classification procedures where each pixel corresponds to an object which must be placed into a class

Smoothing Operations

Linear Filters:

- Uniform filter

- Triangular filter

- Gaussian filter

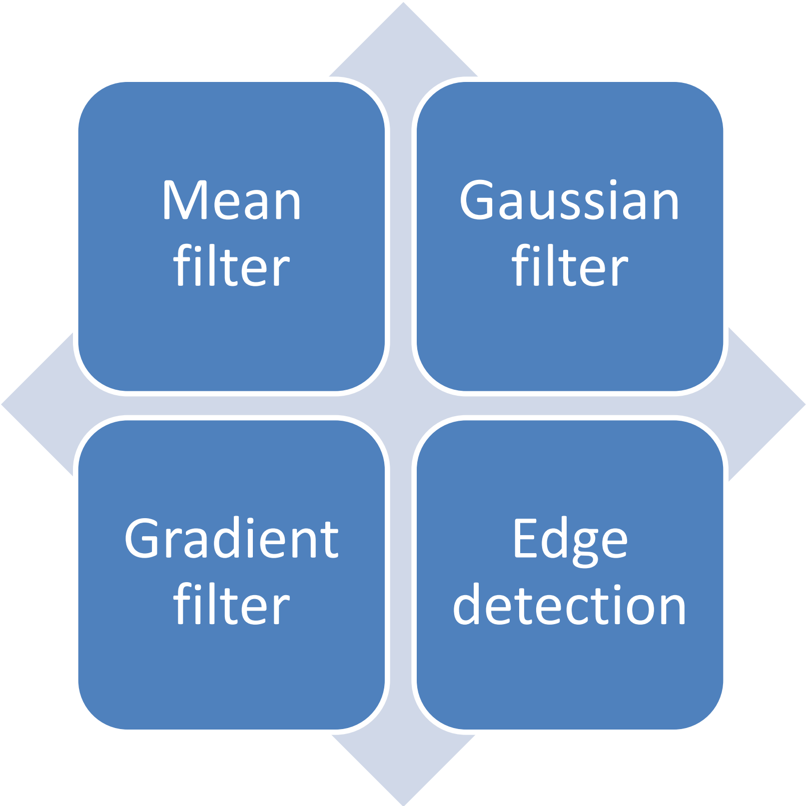
Non-Linear Filters:

- Median filter

- Kuwahara filter

Linear Filters

Linear filters are operations, which convert the source image via a linear transformation to the result image. The new value of a pixel is calculated by a weighted sum of the pixel values in its neighbourhood.



Mean Filter

- Mean filtering is a simple, method of smoothing images, i.e. reducing the amount of intensity variation between one pixel and the next. It is often used to reduce noise in images.
- The idea is to replace each pixel value in an image with the mean (`average') value of its neighbors, including itself. This filter is a low-pass filter.
- Dis-adv of its usage is that edges get blurred.

$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$

original | mean, radius 1 | mean, radius 3 | mean, radius 5

Gaussian Filter

- It is a 2-D conv operator that is used to `blur' images and remove detail and noise. It uses a kernel that represents the shape of a Gaussian (`bell-shaped') hump.
- The Gaussian distribution in 1-D has the form:

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$$

where σ is the standard deviation of the distribution.

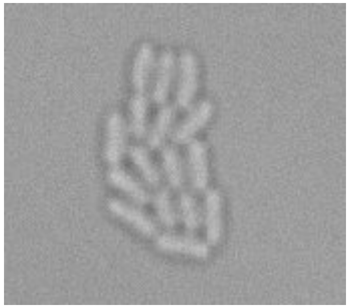
- In 2-D, an isotropic (*i.e.* circularly symmetric) Gaussian has the form:

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

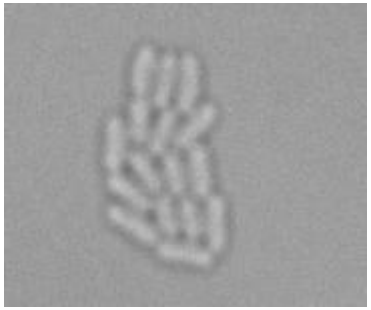
- Disadvantage of its usage is that edges get blurred.

Gaussian Filter

- This operation has 2 parameters. 1st one is the 'radius', which actually is the half side length of a square defining the neighbourhood of a pixel. The radius defines which pixels are actually regarded when calculating the mean. 2nd parameter is σ , which describes the width of the gaussian distribution. It determines how the pixels are weighted to calculate the final value. With a big sigma the result of the gaussian filter is similiar to the result of the mean filter of the same radius.
- This filter can be used to supress noise in the image (e.g. radius 3, sigma 0.7).



original



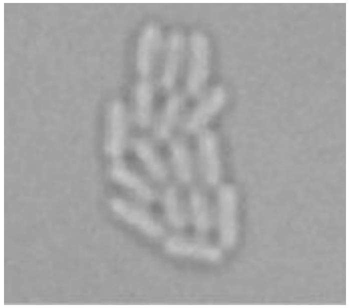
gaussian,
radius 5,
sigma 0.5

gaussian,
radius 5,
sigma 1

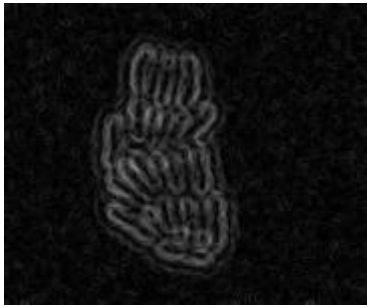
gaussian,
radius 5,
sigma 3

Gradient Filter

- The gradient filter calculates the norm of the gradient operator executed on the image intensity function. The gradient operator is the sum of the partial deviations of a function in x- and y-direction. Here the gradient operator is approximated using the sobel operator. This filter is a high-pass filter, i.e. the result is an image which is bright in regions where are edges (big changes of the brightness) in the original image, and dark in regions where are uniform areas in the original image.



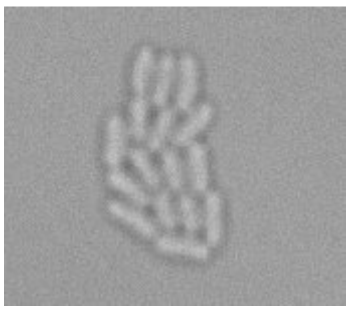
original



gradient

Edge Detection

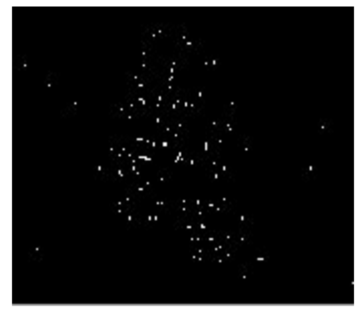
- With the edge detection operation different methods of edge detection can be executed on the image.
- Sobel, Prewitt, Roberts, Log, and Canny method. The only parameter for all these methods is the 'level', which controls how strong the edges are which are recognized. With level > 1 only the stronger edges are detected, with level < 1 also weaker edges are detected.



original

sobel, level 1

prewitt, level 1



roberts, level 1

log, level 1

canny, level 1

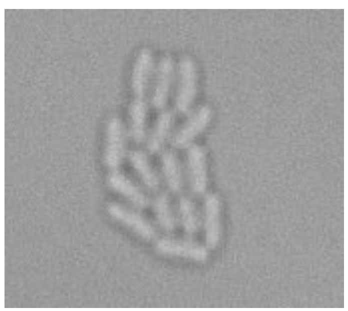
Non-Linear Filter

- Nonlinear filters are operations, which convert the source image via a non-linear transformation to the result image.

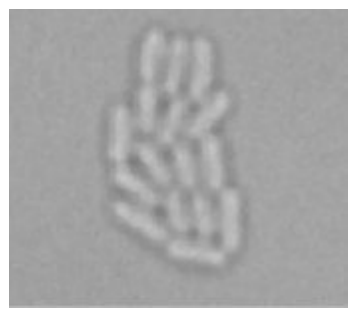
Median filter

Median Filter

- When using the median filter, the new value of a pixel is the median of the pixel values in its neighbourhood. This filter is a low-pass filter, i.e. it can be used to inhibit noise.
- Dis-adv of its usage is that edges get blurred. The only parameter of this operation is the 'radius', which actually is the half side length of a square defining the neighbourhood of a pixel. The bigger the radius, the better noise is inhibited, and compared to the mean filter the edges get less blurred.
- This filter can be used with a small radius (e.g. 1 pixel) to suppress noise in the image.



original



median, radius 1

median, radius 3

median, radius 5

Median Filter

$$\hat{f}(x,y) = \underset{(s,t) \in S_{xy}}{median}\{g(s,t)\}$$

- The median is calculated by first sorting all the pixel values from the surrounding neighborhood into numerical order
- Replacing the pixel being considered with the middle pixel value.
- If the neighborhood under consideration contains an even number of pixels, the average of the two middle pixel values is used
- Excellent at noise removal, without the smoothing effects that can occur with other smoothing filters
- Particularly good when salt and pepper noise is present

Contd..

123	125	126	130	140
122	124	126	127	135
118	120	150	125	134
119	115	119	123	133
111	116	110	120	130

Neighbourhood values:

115, 119, 120, 123, 124,
125, 126, 127, 150

Median value: 124

Morphological Filters

- Morphological filters are operations which affect the shape of structures in the image.
- They mostly work with minimum- and maximum operators and thus belong to the group of non-linear filters.
- The minimum- and maximum operations are executed on the neighbourhood of each pixel, defined by a structuring element.
- A structuring element is a binary matrix whose elements with 1 define neighbourhood pixels, elements with 0 are ignored.
- To determine which pixels belong to the neighbourhood of a certain pixel, the structuring element is centred over this pixel and serves as a mask.

Morphological Operations

Dilation

Erosion

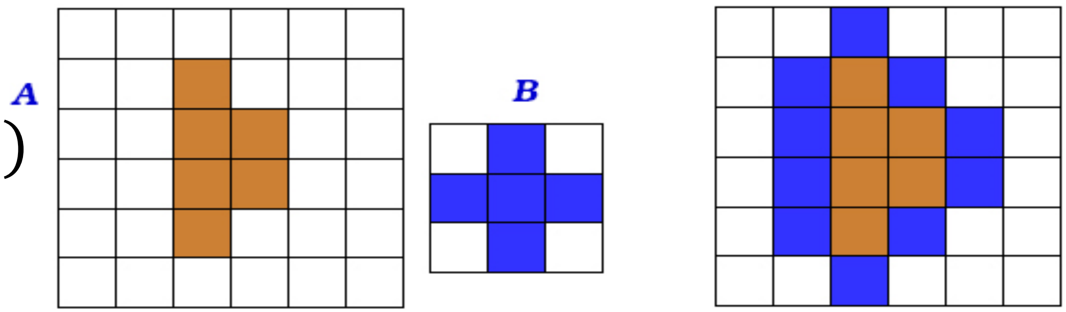
Opening

Closing

Dilation

- The only parameter for this operation is a structuring element, defining the neighbourhood of a pixel. It is possible to select simple structuring elements with disk, square, octagon or diamond shapes and indicate their size. It is possible to create more complex, arbitrary structuring elements too.

$$D(A,B) = A \oplus B = \bigcup_{\beta \in B} (A + \beta)$$



- This operation is mostly used in the editing of binary images.

original　　radius 1　　radius 3　　radius 5

Example for Dilation

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	1	1

Output Image

	1	0							

Example for Dilation

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	1	1

Output Image

	1	0	1						

Example for Dilation

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	1	1

Output Image

	1	0	1	1					

Example for Dilation

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	1	1

Output Image

	1	0	1	1	1				

Example for Dilation

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	1	1

Output Image

	1	0	1	1	1	1			

Example for Dilation

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	1	1

Output Image

	1	0	1	1	1	1	1		

Example for Dilation

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	1	1

Output Image

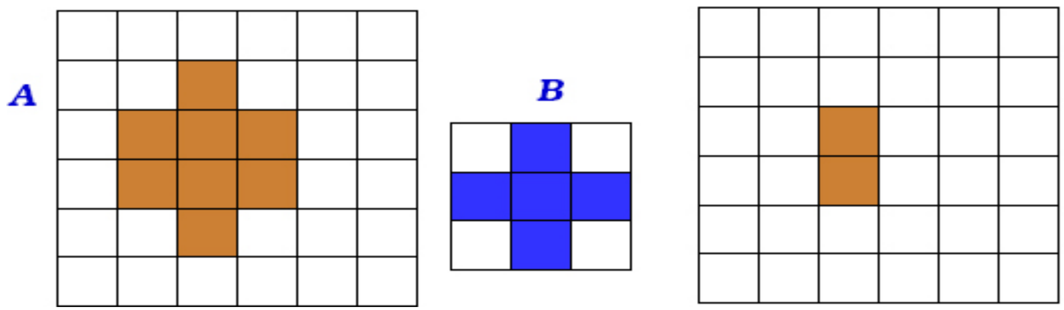
	1	0	1	1	1	1	1	1	

The object gets bigger and holes are filled!

Erosion

- The erosion is the complementary operation to the dilation. The only parameter for this operation is a structuring element, defining the neighbourhood of a pixel. It is possible to select simple structuring elements with disk, square, octagon or diamond shapes and indicate their size.

$$E(A,B) = A \ominus B = \bigcap_{\beta \in B} (A - \beta)$$



- This operation is mostly used in the editing of binary images.

original　radius 1　radius 3　radius 5

Example for Erosion

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	1	1

$$g(x) = f(x) \ominus SE$$

Output Image

	0								

Fit: If all '1's in the SE overlap with input => output = 1, otherwise output = 0

$$f(x) \ominus SE_2 \quad (f(x) \ominus SE_1) \ominus SE_1$$

$$SE_2 = 2 \; SE_1$$

Example for Erosion

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	1	1

Output Image

	0	0							

Example for Erosion

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	1	1

Output Image

	0	0	0						

Example for Erosion

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	1	1

Output Image

	0	0	0	0					

Example for Erosion

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	1	1

Output Image

	0	0	0	0	1				

Example for Erosion

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	1	1

Output Image

	0	0	0	0	1	0			

Example for Erosion

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	1	1

Output Image

	0	0	0	0	1	0	0		

Example for Erosion

Input image

1	0	0	0	1	1	1	0	1	1

Structuring Element

1	1	1

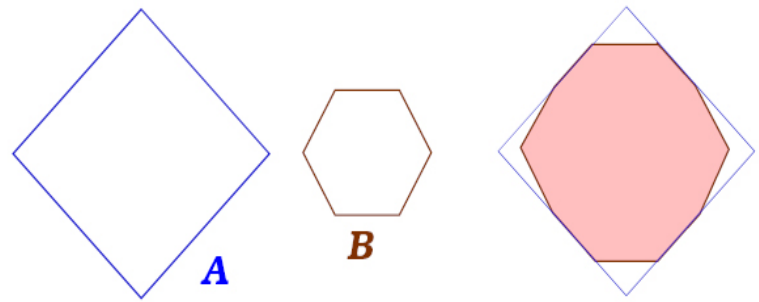
Output Image

	0	0	0	0	1	0	0	0	

The object gets smaller

Opening

- Morphological opening performs an erosion operation followed by a dilation operation using the same structuring element. The effect of this operation is that small objects or image details a removed while the size and the shape of larger objects in the image are preserved. It is basically a low-pass filter.
- $O(A, B) = A \circ B = D((E(A, B), B)$
  $= (A \ominus B) \oplus B$
- Roll B on the inside of the boundary of A.
- This operation is mostly used in the editing of binary images.



original

radius 1

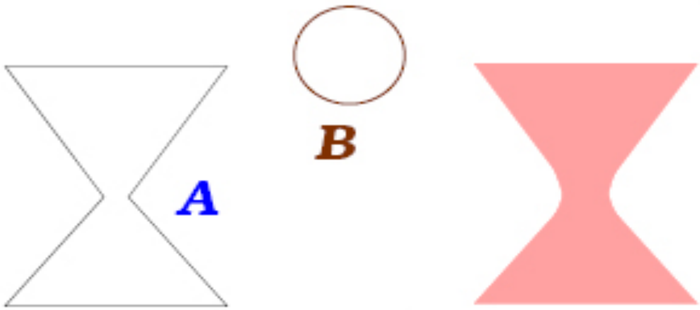
radius 3

- Morphological opening performs an erosion operation followed by a dilation operation using the same structuring element. The effect of this operation is that small objects or image details a removed while the size and the shape of larger objects in the image are preserved. It is basically a low-pass filter.
- $O(A, B) = A \circ B = D((E(A, B), B)$
  $= (A \ominus B) \oplus B$
- Roll B on the inside of the boundary of A.
- This operation is mostly used in the editing of binary images.

radius 5

Closing

- Morphological closing performs a dilation operation followed by an erosion operation using the same structuring element. The effect of this operation is that holes in objects are closed while the size and the shape of the object is preserved. It is also possible that several small nearby objects are combined to one large object.
- $C(A,B) = A \bullet B = E((D(A,-B),-B)$
  $= (A \oplus B) \ominus B$
- Roll B on the outside of the boundary of A.
- This operation is mostly used in the editing of binary im

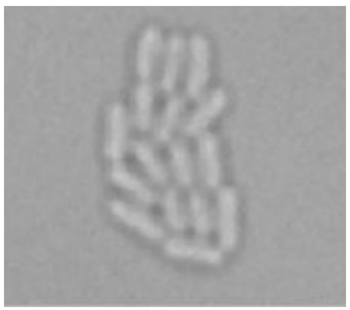


Threshold

- The simplest segmentation method is the conversion of a gray-scale image to a binary image using a threshold value.
- Pixels brighter than the threshold are white pixels in the resulting image, pixels darker than the threshold are black pixels.
- This method is useful if the brightness of the objects differs significantly from the brightness of the background.
- There are several improvements of this method where the threshold value is selected automatically. As a parameter of this operation it is possible to select the method.
- Supported methods are: manual selection of the threshold value, otsu, iterative, local entropy, joint entropy, relative entropy, renyis entropy-method.

Threshold

- When using manual selection of the threshold value it is possible to select a threshold value between 0 (black) and 1 (white). When using one of the other methods a factor can be selected to modify the automatically determined value.



original

manual, level 0.64

otsu

iterative

local entropy

relative entropy

renyis entropy

Session Summary

- Digital image is a matrix of pixel values.
- Digital image processing is a technique to extract or manipulate information from a digital image.
- Various image processing techniques used are edge detection, morphological operations, noise filtering etc.