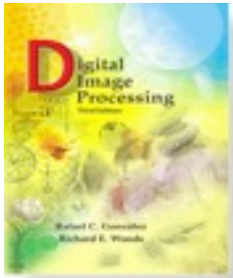
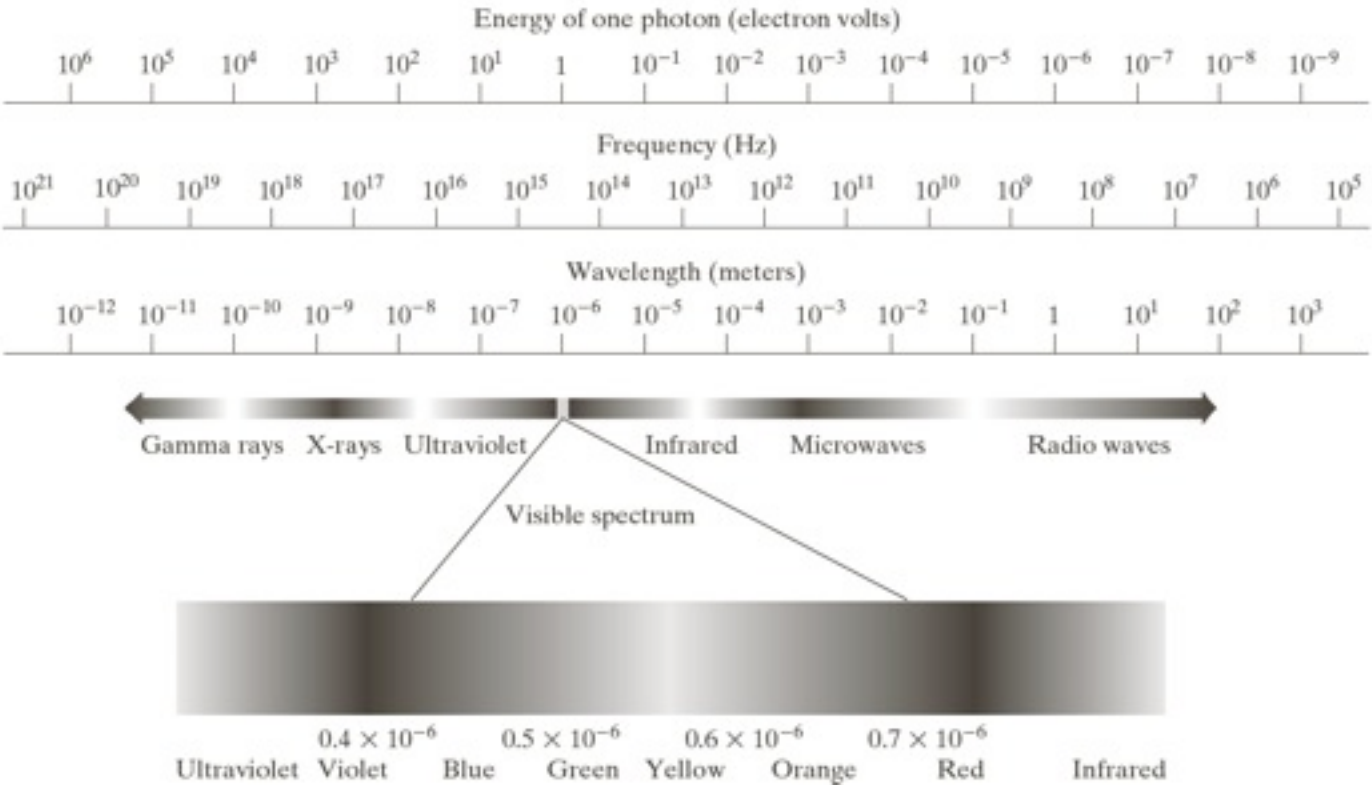
Last time:

- Images are generated by “capturing” the energy of electromagnetic waves.



Chapter 2
Digital Image Fundamentals



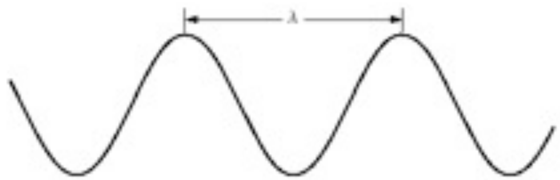
A wave is a periodic function of *period T* :

$$\phi(t+T) = \phi(t)$$

Spatial waves: $\phi(x+ct)$, where c is the speed and $\nu = 1/T$ is the frequency.

$$c = \frac{\text{distance covered in one period}}{\text{period length}} = \frac{\lambda}{T} = \lambda\nu$$

λ is the wavelength



Electromagnetic waves travel at the speed of light,

$$c \approx 300000 km/s = 3 \times 10^8 m/s$$

In terms of wavelength, $\phi(x) = \phi(x+\lambda)$

Energy of the wave: $E = h\nu = \dfrac{hc}{\lambda}$

where h is Planck's constant

$$h \approx 4.13 \times 10^{-15} [eV \cdot s]$$

Exercise:

Given the wavelength,
find the <u>frequency</u> and the <u>energy</u> of
visible green light.

Exercise:

Given the wavelength,
find the <u>frequency</u> and the <u>energy</u> of
visible green light.

According to NASA/LANDSAT thematic bands, visible green has a wavelength in the range $0.52 - 0.60\,\mu m$.

$$\lambda_{vg} \approx 0.55\,\mu m$$

$$c = \frac{\lambda}{T} = \lambda\nu \rightarrow \nu = \frac{c}{\lambda}$$

Exercise:

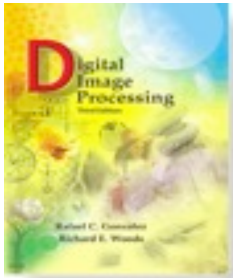
Given the wavelength,
find the frequency and the energy of
visible green light.

According to NASA/LANDSAT thematic bands, visible green has a wavelength in the range $0.52 - 0.60\,\mu m$.

$$\lambda_{vg} \approx 0.55\,\mu m$$

$$c = \frac{\lambda}{T} = \lambda\nu \rightarrow \nu = \frac{c}{\lambda}$$

$$\nu = \frac{3 \cdot 10^8\, m/s}{0.55 \cdot 10^{-6} m} = \frac{3}{0.55} 10^{14} \frac{1}{s} \approx 5.45 \cdot 10^{14} Hz$$



Exercise:

Given the wavelength,
find the <u>frequency</u> and the <u>energy</u> of
visible green light.

According to NASA/LANDSAT thematic bands, visible green has a wavelength in the range $0.52 - 0.60\,\mu m$.

$$\lambda_{vg} \approx 0.55\,\mu m$$

$$c = \frac{\lambda}{T} = \lambda\nu \rightarrow \nu = \frac{c}{\lambda}$$

$$\nu = \frac{3 \cdot 10^{8}\, m/s}{0.55 \cdot 10^{-6} m} = \frac{3}{0.55} 10^{14} \frac{1}{s} \approx 5.45 \cdot 10^{14}\, Hz$$

Energy:

$$\begin{aligned} E &= h\nu = 4.13 \cdot 10^{-15}\, eVs \cdot 5.45 \cdot 10^{14}\, Hz \\ &\approx 22 \cdot 10^{-1} eV = 2.2 eV \end{aligned}$$

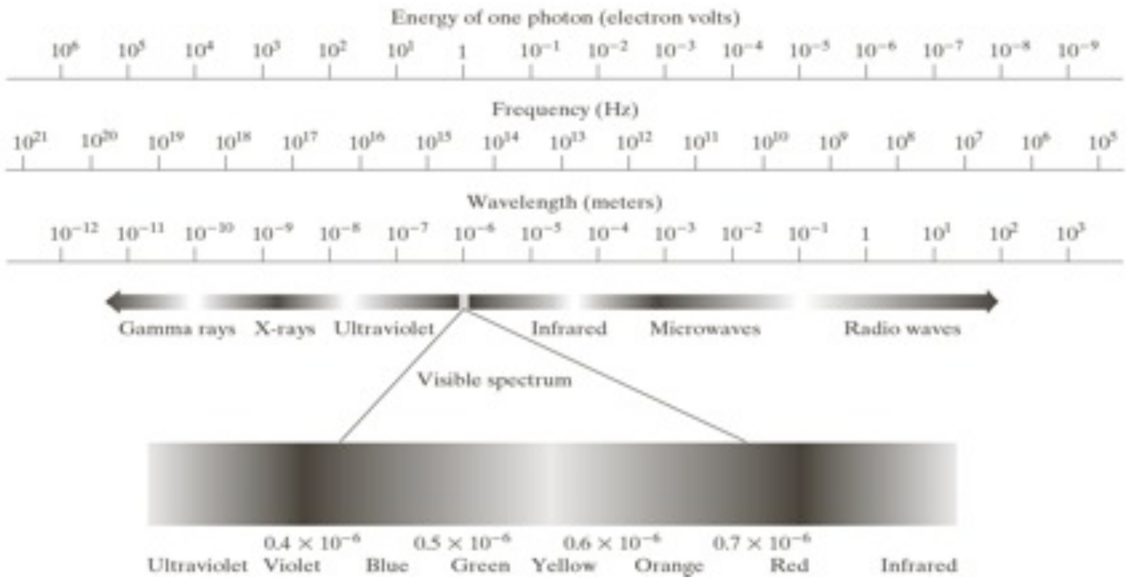
Exercise:

Given the wavelength,
find the frequency and the energy of visible green light.

According to NASA/LANDSAT thematic bands, visible green has a wavelength in the range $0.52 - 0.60\,\mu m$.

$$\lambda_{vg} \approx 0.55\,\mu m$$

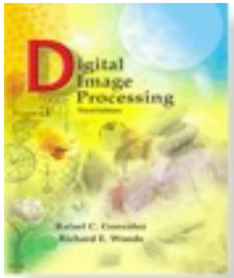
$$c = \frac{\lambda}{T} = \lambda\nu \rightarrow \nu = \frac{c}{\lambda}$$

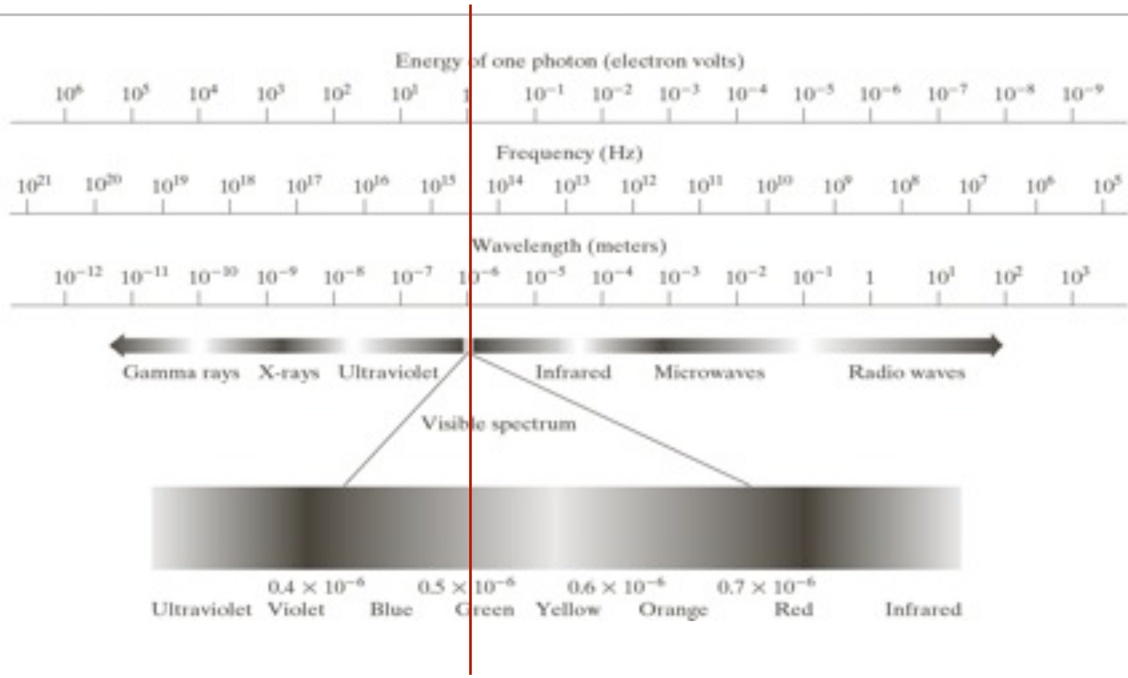


$$\nu = \frac{3 \cdot 10^8\,m/s}{0.55 \cdot 10^{-6} m} = \frac{3}{0.55} 10^{14} \frac{1}{s} \approx 5.45 \cdot 10^{14} Hz$$

Energy:

$$\begin{aligned} E &= h\nu = 4.13 \cdot 10^{-15}\,eV\,s \cdot 5.45 \cdot 10^{14} Hz \\ &\approx 22 \cdot 10^{-1} eV = 2.2 eV \end{aligned}$$





Exercise:

Given the wavelength,
find the frequency and the energy of visible green light.

According to NASA/LANDSAT thematic bands, visible green has a wavelength in the range $0.52 - 0.60\,\mu m$.

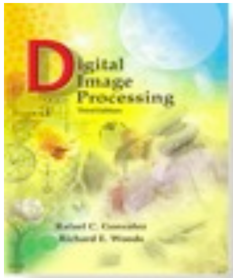
$$\lambda_{vg} \approx 0.55\,\mu m$$

$$c = \frac{\lambda}{T} = \lambda\nu \rightarrow \nu = \frac{c}{\lambda}$$

$$\nu = \frac{3 \cdot 10^8\,m/s}{0.55 \cdot 10^{-6} m} = \frac{3}{0.55} 10^{14} \frac{1}{s} \approx 5.45{\cdot}10^{14} Hz$$

Energy:

$$\begin{aligned} E &= h\nu = 4.13 \cdot 10^{-15}\,eV s \cdot 5.45 \cdot 10^{14} Hz \\ &\approx 22 \cdot 10^{-1} eV = 2.2 eV \end{aligned}$$



Light is a particular type of electromagnetic wave

The colors that humans perceive are determined by the light *reflected* by the object:

- all the light reflected: white object
- some components (of the visible spectrum) absorbed, some reflected: color (wavelength reflected).

- Light reflected/absorbed at the same rate for all wavelengths: *monochromatic* light.
Thus we speak of *intensity* or *gray level*

Properties of light sources/reflected light:

Chromatic light (colors): from 0.43 μm to 0.79 μm wavelength

Radiance: Total amount of energy out of the light source (Watts)

Luminance: Amount of light perceived from a light source (lumen) ex. stars

Brightness: earlier a synonymous of luminance, is now a subjective measurement of light perceived from a light source.

Example of image acquisition in the visible light spectrum (think about taking pictures with your camera)

Image model: $0 < f(x,y) < \infty$

$$f(x,y) = i(x,y)r(x,y)$$

illumination
- between 1 and ∞
- determined by the light source

reflectance
- between 0 (total absorption) and 1 (total reflectance)
- determined by the object

1. Illumination source emits light (all wavelengths in the visible spectrum)

2. Object absorbes some wavelengths and reflects other (color)

3. imaging system captures the energy of the reflected wavelengths

4. The output is obtained through sampling and digitalization

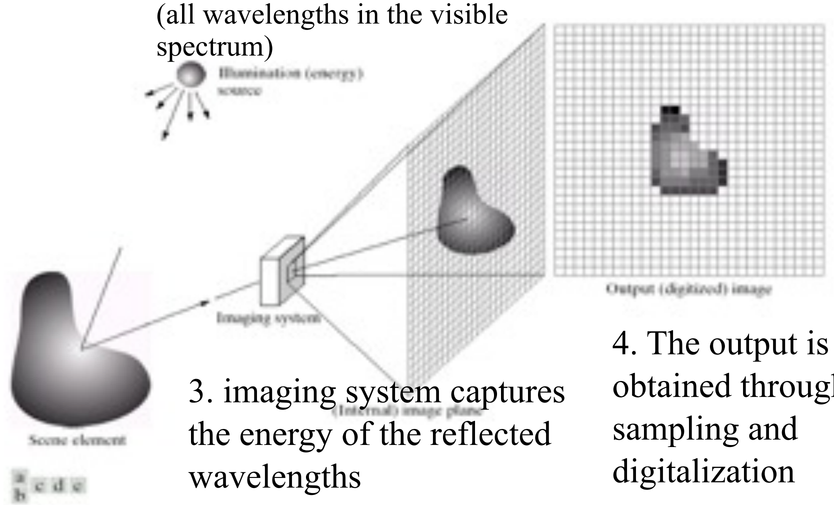
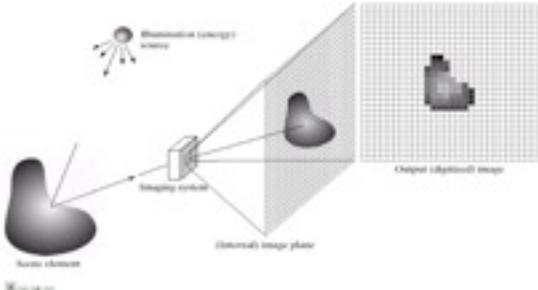
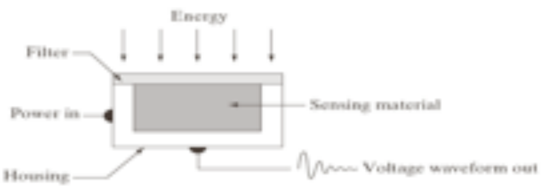
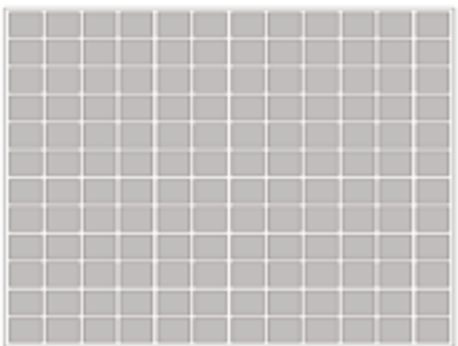


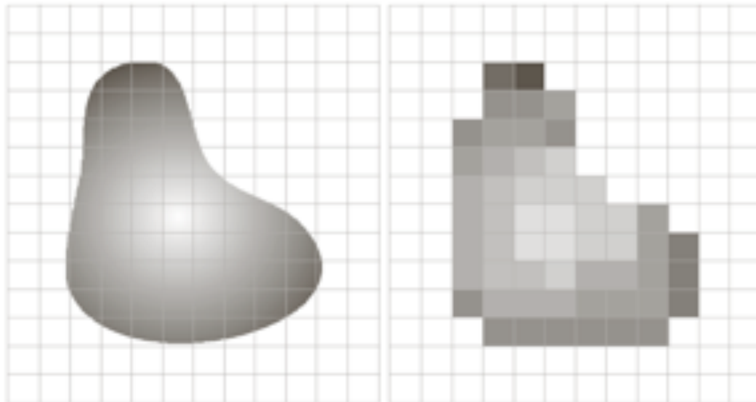
FIGURE 2.15 An example of the digital image acquisition process. (a) Energy ("illumination") source. (b) An element of a scene. (c) Imaging system. (d) Projection of the scene onto the image plane. (e) Digitized image.







Sampling and quantization



a b

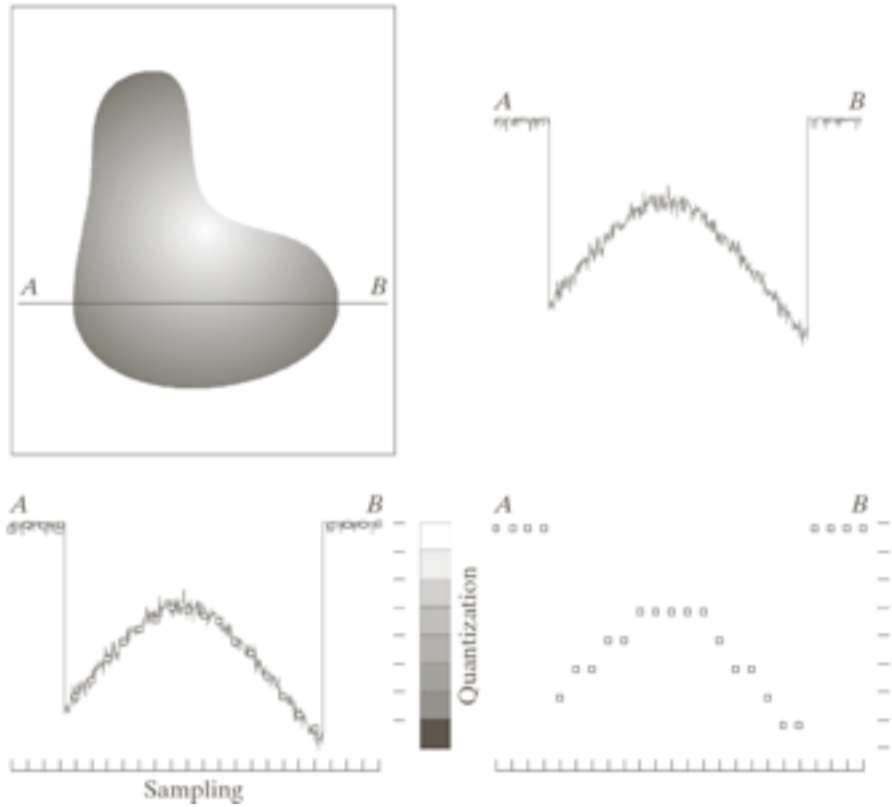
FIGURE 2.17 (a) Continuous image projected onto a sensor array. (b) Result of image sampling and quantization.

Sampling: digitizing the coordinate values
Quantization: digitizing the amplitude values

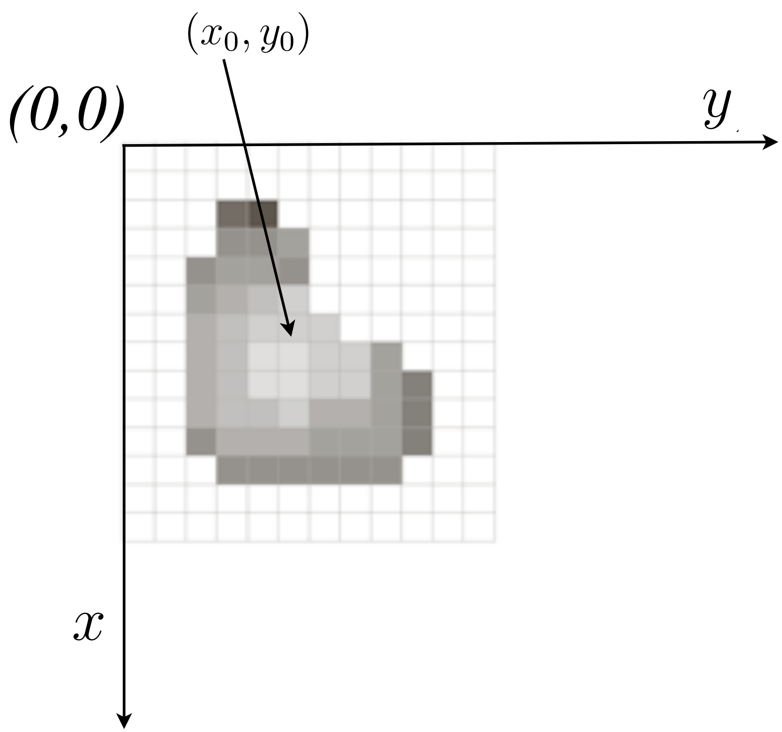
Image sampling and quantization:

From continuous (+noise)

to discrete (digitalized)



Because of sampling, the image will be described by a finite set of points (pixels).



$$f(x, y) = i(x, y)r(x, y)$$

In practice, for any point (x_0, y_0) of the image, we require

$$i_{min}r_{min} = L_{min} \leq \ell = f(x_0, y_0) \leq L_{max} = i_{max}r_{max}$$

where L_{max} is required to be finite

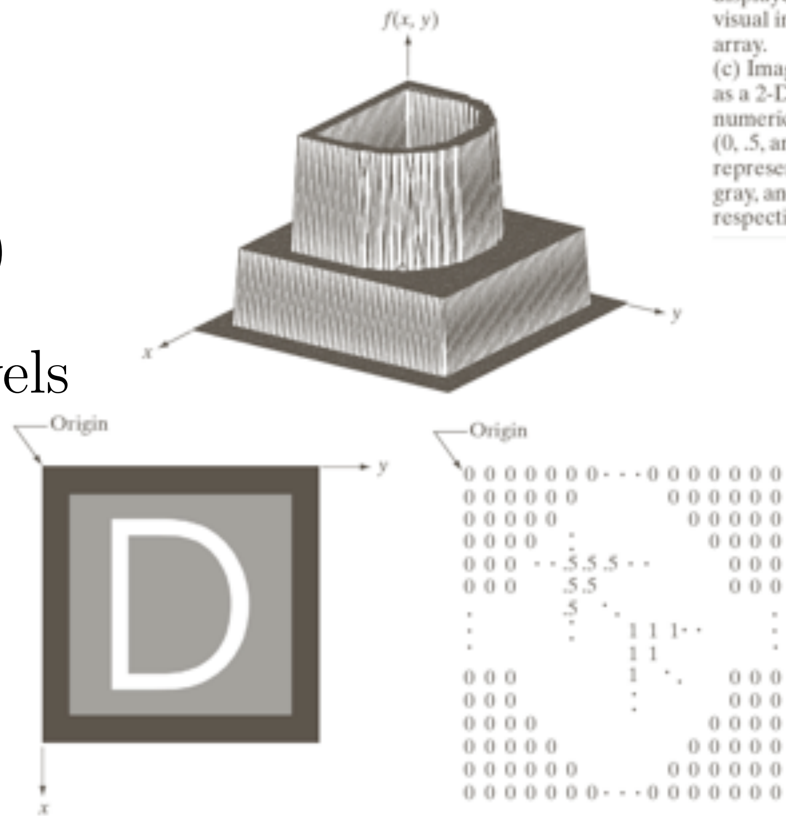
$[L_{min}, L_{max}]$ is the *intensity* scale (also *gray scale*) of the image.

Common practice: $[L_{min}, L_{max}]$ is shifted to $[0, L-1]$, where $\ell = 0$ is black and $\ell = L - 1$ is white.

Continuous $f(s,t) \rightarrow f(x,y)$ Discrete

$f(x,y)$ a 2-D array with

- M $(x = 0, 1, \ldots, M-1)$ rows
- N columns $(y = 0, 1, \ldots, N-1)$
- $[0, L-1]$ (discrete) intensity levels



a
b c

FIGURE 2.18 (a) Image plotted as a surface. (b) Image displayed as a visual intensity array. (c) Image shown as a 2-D numerical array (0, .5, and 1 represent black, gray, and white, respectively).

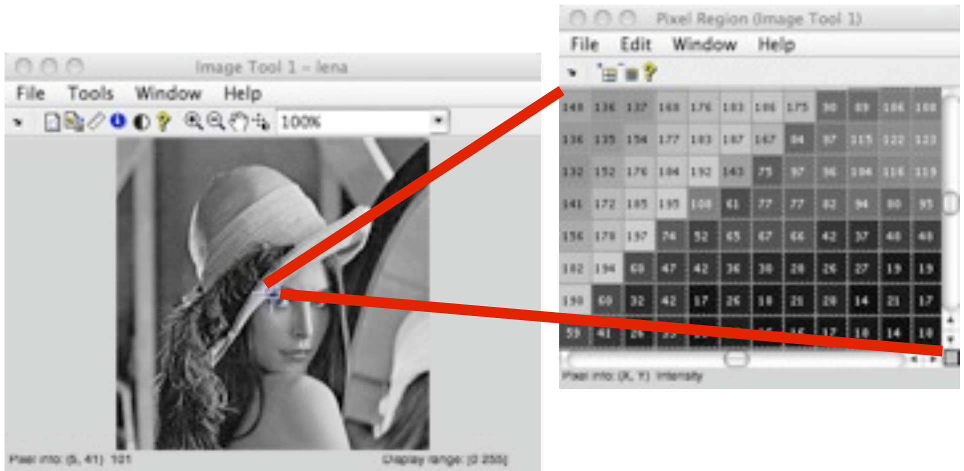
It is typical to choose:

$$L = 2^k$$

and the values between [*0, L-1*] to be equally spaced.

k is the number of bits used for the representation of the intensity.

Ex. jpeg uses typically 8 bits, (2^8=256), giving an intensity range between 0 and 255.



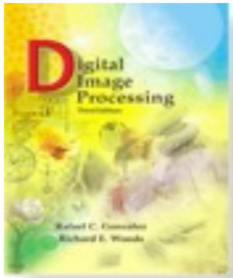


TABLE 2.1
Number of storage bits for various values of N and k.

N/k	1 ($L = 2$)	2 ($L = 4$)	3 ($L = 8$)	4 ($L = 16$)	5 ($L = 32$)	6 ($L = 64$)	7 ($L = 128$)	8 ($L = 256$)
32	1,024	2,048	3,072	4,096	5,120	6,144	7,168	8,192
64	4,096	8,192	12,288	16,384	20,480	24,576	28,672	32,768
128	16,384	32,768	49,152	65,536	81,920	98,304	114,688	131,072
256	65,536	131,072	196,608	262,144	327,680	393,216	458,752	524,288
512	262,144	524,288	786,432	1,048,576	1,310,720	1,572,864	1,835,008	2,097,152
1024	1,048,576	2,097,152	3,145,728	4,194,304	5,242,880	6,291,456	7,340,032	8,388,608
2048	4,194,304	8,388,608	12,582,912	16,777,216	20,971,520	25,165,824	29,369,128	33,554,432
4096	16,777,216	33,554,432	50,331,648	67,108,864	83,886,080	100,663,296	117,440,512	134,217,728
8192	67,108,864	134,217,728	201,326,592	268,435,456	335,544,320	402,653,184	469,762,048	536,870,912

$$L = 2^k$$

Each intensity value in a pixel is stored as a word of lenght k. The number of bits to store an image is:

$$b = MNk \qquad (b = N^2k, \quad \text{if } M = N)$$

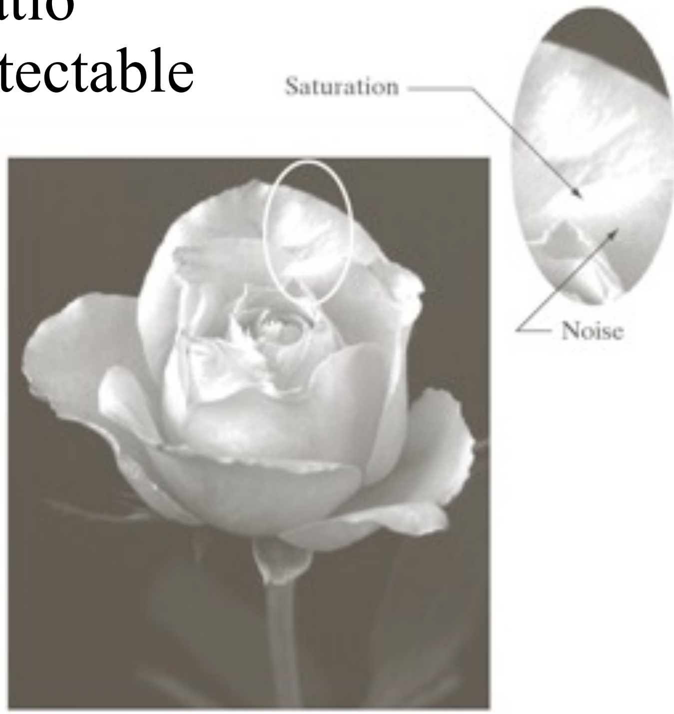
Some attributes of the imaging system/images

Dynamic range of an imaging system: ratio between the maximum and minimum detectable intensity level of the system.

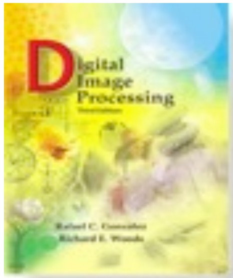
Saturation: highest value beyond which intensity levels are clipped (to a constant value)

Noise: grainy texture pattern

Contrast: difference in intensity between the highest and lowest intensity level in an image



(high contrast vs dull images)



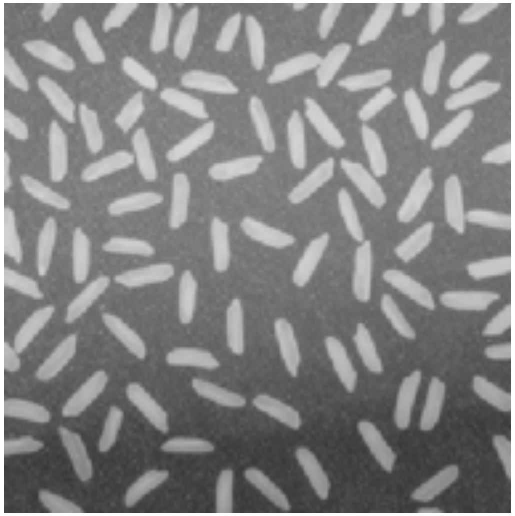
Spatial resolution:
is the measure of the smallest discernible detail in an image.

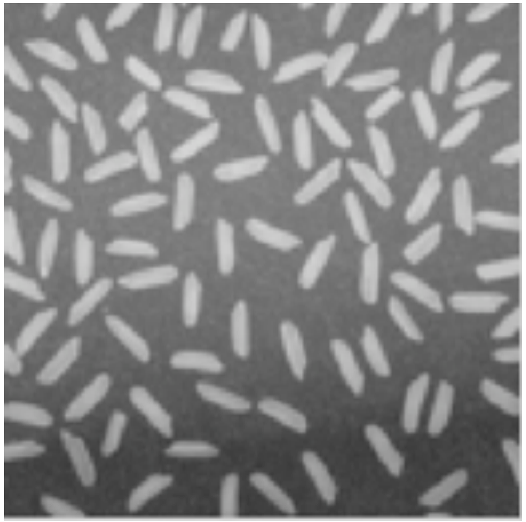
Relates number of pixels to spatial dimension of the image.

High spatial resolution: very detailed image
Low spatial resolution: poor detailed image

a b
c d

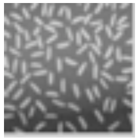
FIGURE 2.20 Typical effects of reducing spatial resolution. Images shown at: (a) 1250 dpi, (b) 300 dpi, (c) 150 dpi, and (d) 72 dpi. The thin black borders were added for clarity. They are not part of the data.





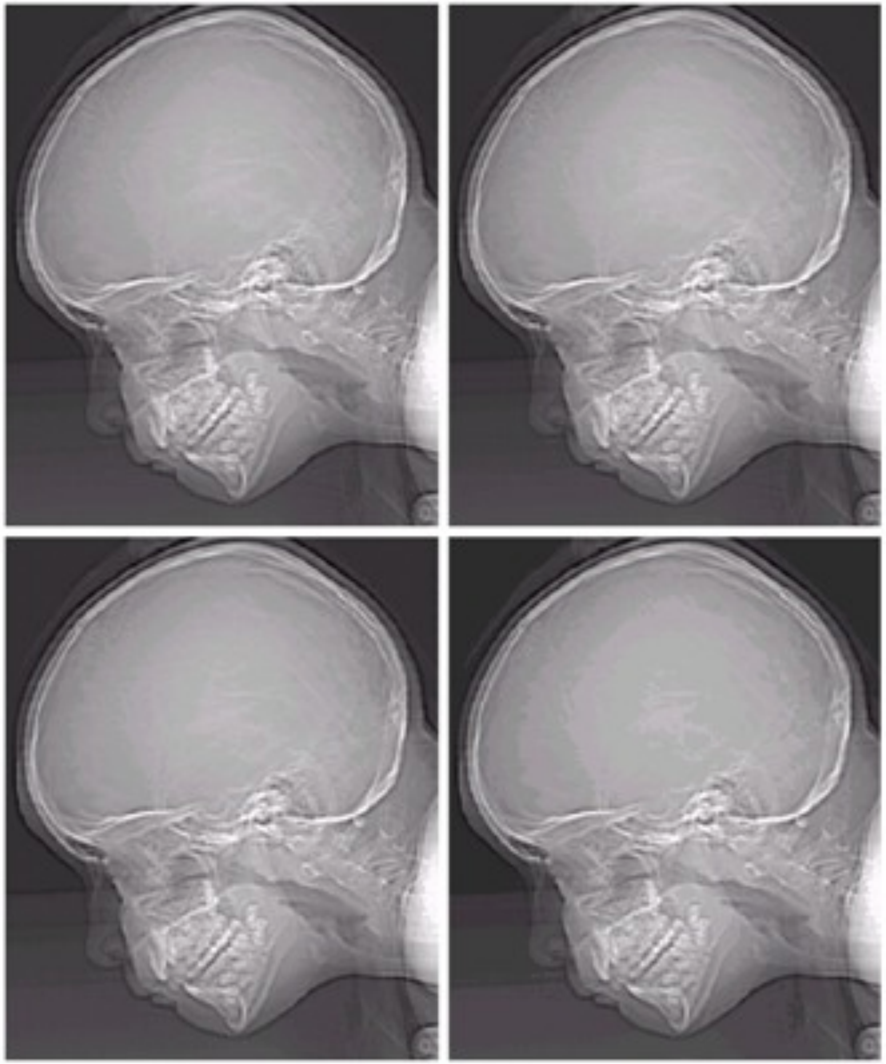
The same image of rice ('rice.png') at initial spatial resolution, halved, and halved again resolution.
Above: The pixel size is doubled (the image is shown at the same spatial dimension)

Below: The pixel size is kept constant (the spatial dimension of the image is halved)



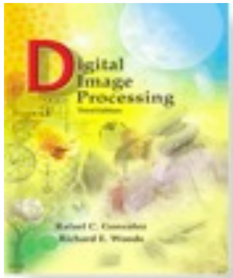
Intensity resolution: spatial resolution fixed, reduce *k*, the number of intensity levels.
[0, L-1]=[0, 2^k]

Low intensity resolution might result in false contours



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.



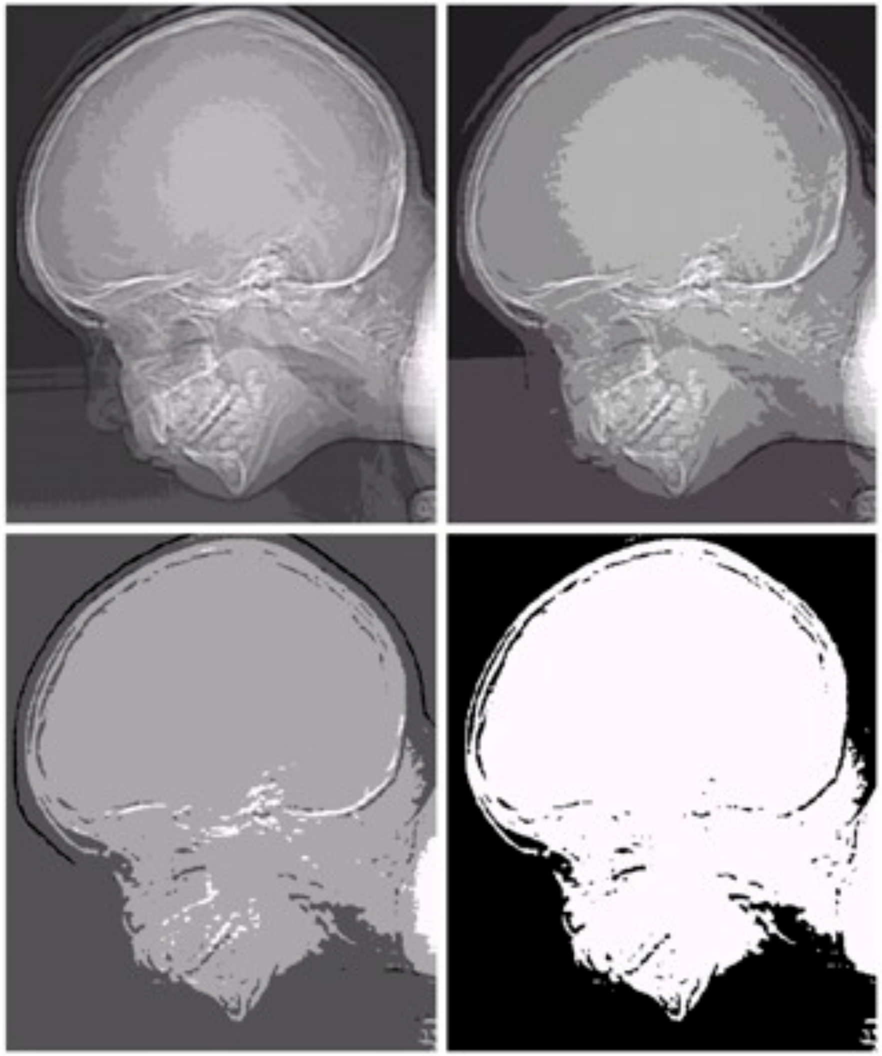
Intensity resolution: spatial resolution fixed, reduce *k*, the number of intensity levels.

[0, L-1]=[0, 2^k]

Low intensity resolution might result in false contours

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.)



What are the optimal values of N and k ?

No general rule,
might depend on the
level of detail of the image.

a b c

FIGURE 2.22 (a) Image with a low level of detail. (b) Image with a medium level of detail. (c) Image with a relatively large amount of detail. (Image (b) courtesy of the Massachusetts Institute of Technology.)

Isopreference curves: on the same line two points correspond to different images that have the same preference rating.

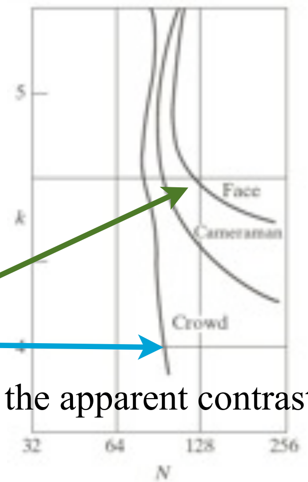


FIGURE 2.23 Typical isopreference curves for the three types of images in Fig. 2.22.

For a fixed number of pixels
the perceived image is independent
of the number of levels used

low k is tends to increase the apparent contrast, perceived as an improved quality

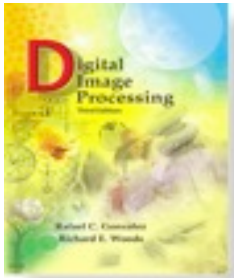


Image interpolation

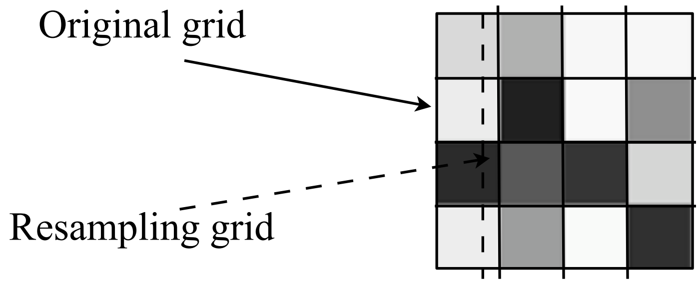
Is used for zooming, shrinking, rotating, geometric corrections, etc.
(resampling methods)

Interpolation: estimate values at unknown locations using known data values.

Example: Given a 64 x 64 image, we want to zoom in and create a 128 x 128 image.

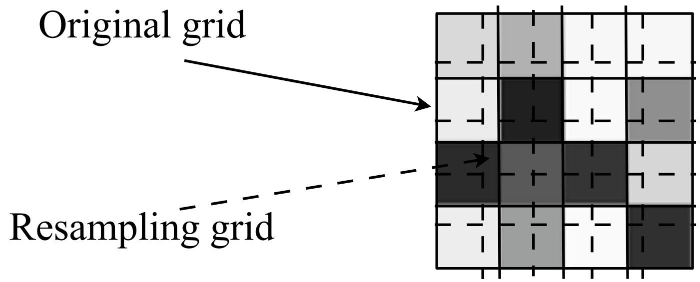
Types of interpolation:

Nearest neighbor: find the closest pixel in the original grid and assign its value



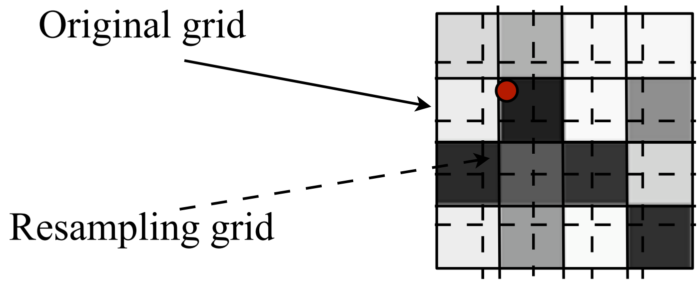
Types of interpolation:

Nearest neighbor: find the closest pixel in the original grid and assign its value



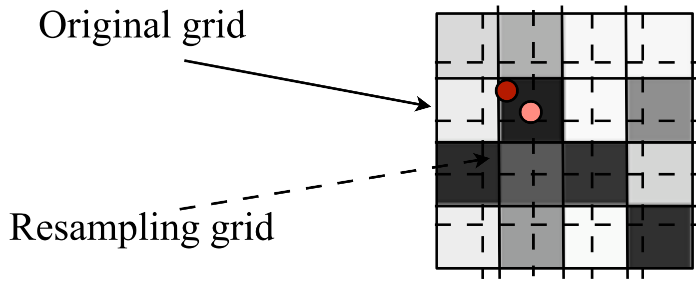
Types of interpolation:

Nearest neighbor: find the closest pixel in the original grid and assign its value



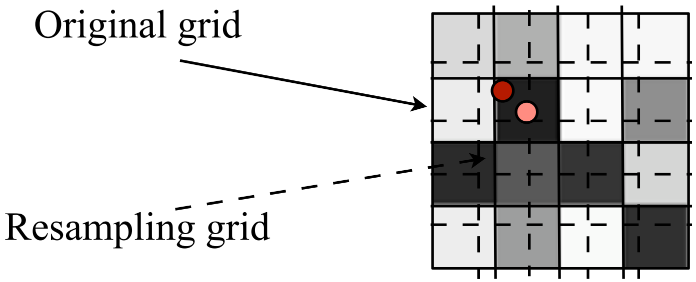
Types of interpolation:

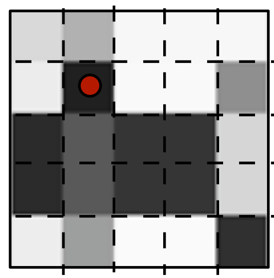
Nearest neighbor: find the closest pixel in the original grid and assign its value

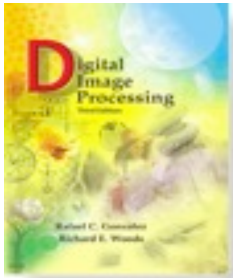


Types of interpolation:

Nearest neighbor: find the closest pixel in the original grid and assign its value

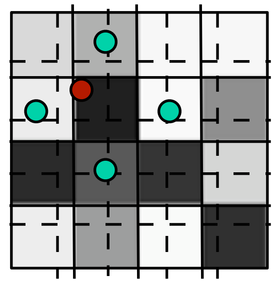


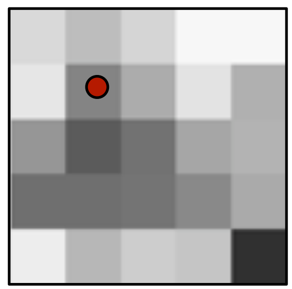


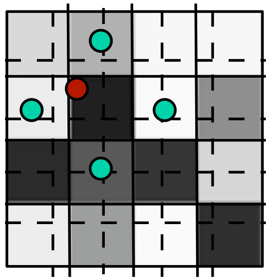


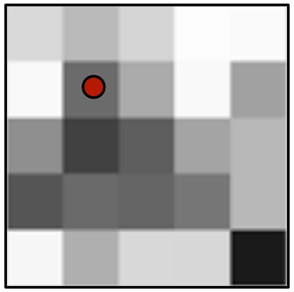
Bilinear interpolation: $v(x, y) = ax + by + cxy + d$

the coefficients *a, b, c, d* are computed using 4 neighbors



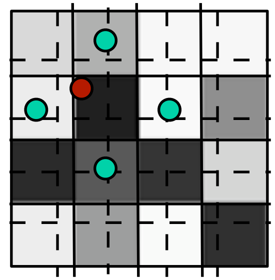


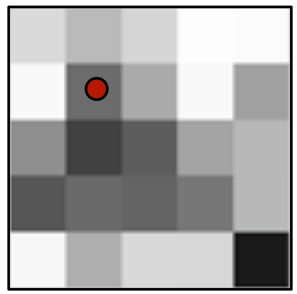




Bicubic interpolation: $$v(x,y)=\sum_{i=0}^{3}\sum_{j=0}^{3}a_{i,j}x^i y^j$$

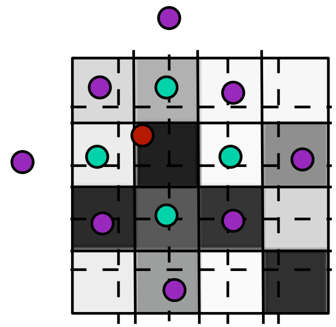
the coefficients $a_{i,j}$ are computed using 16 nearest neighbors.

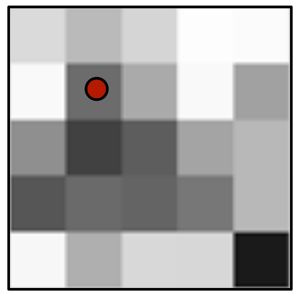


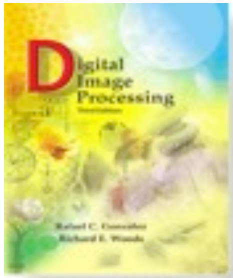


Bicubic interpolation: $v(x,y) = \sum_{i=0}^{3}\sum_{j=0}^{3} a_{i,j}x^i y^j$

the coefficients $a_{i,j}$ are computed using 16 nearest neighbors.



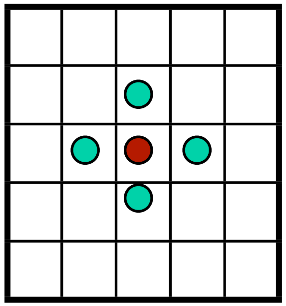




Matlab tips: the command `imresize`

```
>> help imresize
 IMRESIZE Resize image.
    B = IMRESIZE(A, SCALE) returns an image that is SCALE times the
    size of A, which is a grayscale, RGB, or binary image.

    B = IMRESIZE(A, [NUMROWS NUMCOLS]) resizes the image so that it has
    the specified number of rows and columns.  Either NUMROWS or NUMCOLS
    may be NaN, in which case IMRESIZE computes the number of rows or
    columns automatically in order to preserve the image aspect ratio.


    To control the interpolation method used by IMRESIZE, add a METHOD
    argument to any of the syntaxes above, like this:

        IMRESIZE(A, SCALE, METHOD)
        IMRESIZE(A, [NUMROWS NUMCOLS], METHOD),
        IMRESIZE(X, MAP, M, METHOD)
        IMRESIZE(X, MAP, [NUMROWS NUMCOLS], METHOD)


    METHOD can be a string naming a general interpolation method:

        'nearest'    - nearest-neighbor interpolation

        'bilinear'   - bilinear interpolation

        'bicubic'    - cubic interpolation; the default method
```

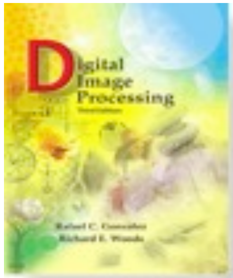
Relation between the pixels



Given a pixel p of coordinates (x, y)

$$N_4(p) = \{(x-1, y), (x+1, y), (x, y-1), (x, y+1)\}$$

4-neighbors of p

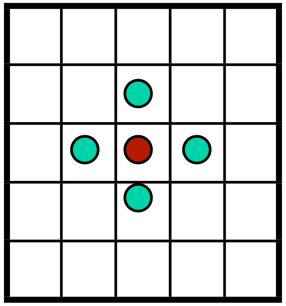


Relation between the pixels

Given a pixel p of coordinates (x, y)

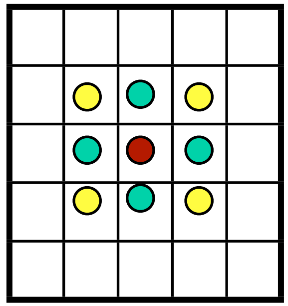
$$N_4(p) = \{(x-1, y), (x+1, y), (x, y-1), (x, y+1)\}$$

4-neighbors of p



$$N_4(p) \cup N_D(p) = N_8(p), \text{ 8-neighbors of } p.$$

Relation between the pixels

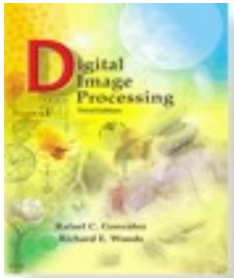


Given a pixel p of coordinates (x, y)

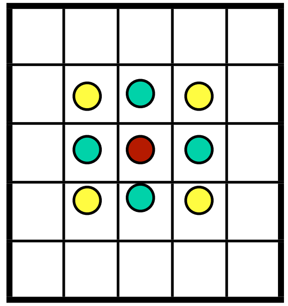
$$N_4(p) = \{(x-1, y), (x+1, y), (x, y-1), (x, y+1)\}$$

4-neighbors of p

$$N_4(p) \cup N_D(p) = N_8(p), \text{ 8-neighbors of } p.$$



Relation between the pixels



Given a pixel p of coordinates (x, y)

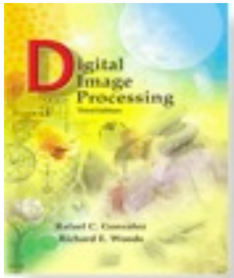
$$N_4(p) = \{(x-1, y), (x+1, y), (x, y-1), (x, y+1)\}$$

4-neighbors of p

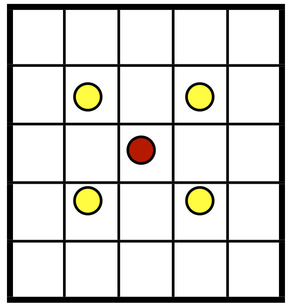
$$N_4(p) \cup N_D(p) = N_8(p), \text{ 8-neighbors of } p.$$

$$N_D(p) = \{(x-1, y-1), (x-1, y+1), (x+1, y-1), (x+1, y+1)\}$$

Diagonal neighbors of p



Relation between the pixels



Given a pixel p of coordinates (x, y)

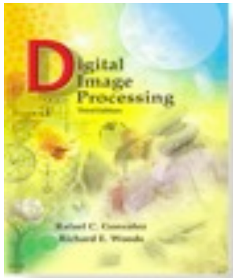
$$N_4(p) = \{(x-1, y), (x+1, y), (x, y-1), (x, y+1)\}$$

4-neighbors of p

$$N_4(p) \cup N_D(p) = N_8(p), \text{ 8-neighbors of } p.$$

$$N_D(p) = \{(x-1, y-1), (x-1, y+1), (x+1, y-1), (x+1, y+1)\}$$

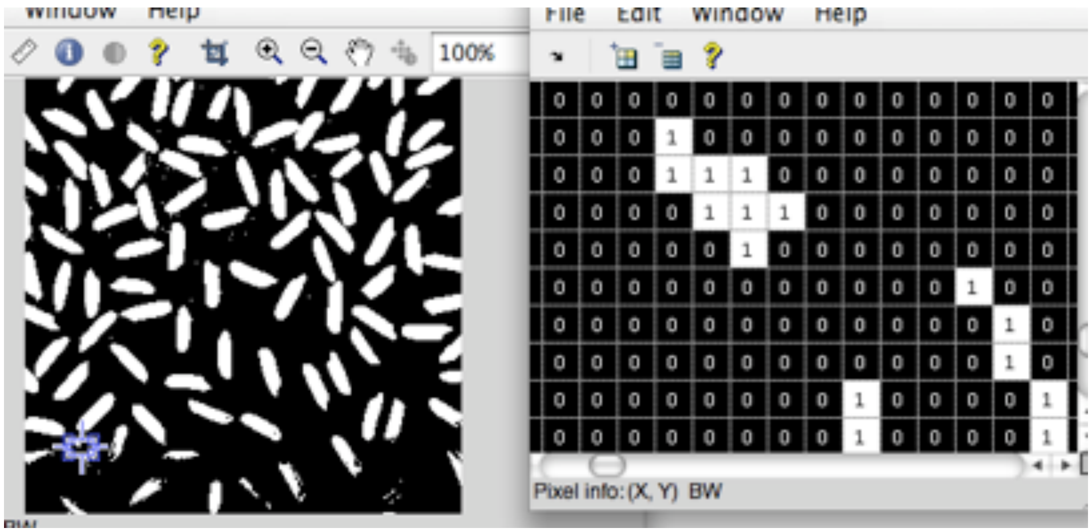
Diagonal neighbors of p



Adjacency

Using the concept of pixel neighbours, we can define the concept of adjacency. This is particularly useful when dealing with *binary* images, (i.e. images consisting of 0=black/1=white).

However, the concept can be generalized to non-binary images.



Example of a binary image

Define a set of intensity values, V.
For binary images, V = {1}.
For gray scale images, V is a subset of the image values.

0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	0	0	0	0	0	0	0	0
0	0	0	0	1	1	1	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	0	0	0	0	0	1	0	0	0	0	1

4-adjacency: p and q are adjacent if they have values in V and q is is in $N_4(p)$

8-adjacency: p and q are adjacent if they have values in V and q is is in $N_8(p)$

Occasionally, the 8-adjacency can cause troubles when defining a *path* between two pixels.

mixed adjacency: *p* and *q* have values in V. Moreover either *q* is in $N_4(p)$ or *q* is in $N_D(p)$ and $N_4(p) \cap N_4(q)$ has no points in V.

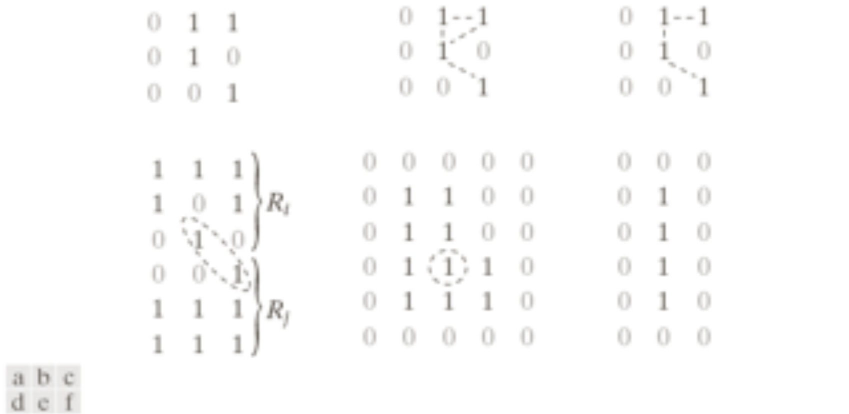


FIGURE 2.25 (a) An arrangement of pixels. (b) Pixels that are 8-adjacent (adjacency is shown by dashed lines; note the ambiguity). (c) *m*-adjacency. (d) Two regions that are adjacent if 8-adjecency is used. (e) The circled point is part of the boundary of the 1-valued pixels only if 8-adjacency between the region and background is used. (f) The inner boundary of the 1-valued region does not form a closed path, but its outer boundary does.

A digital *path* from pixel p with coordinates (x, y) to a pixel q of coordinates (s, t) is a sequence of pixels

$$(x, y) = (x_0, y_0), (x_1, y_1), \ldots, (x_n y_n) = (s, t)$$

n is the path length.

If the first point equals the last point, the path is *closed*.

The type and length of path depends on the adjacency specified.

Connected components

Two pixels are connected if there is a path between the two.

Given a subset S of pixels of an image and a pixel *p* in S, a *connected component* of S is the set of pixels that are connected to *p*. S is connected if it consists of a single connected component.

A subset R of pixels is called a region if it is connected.

Two regions are adjacent if their union is connected.

N.B. The definitions depend on the adjacency!

Connected component in Matlab: bwlabel

```
BW = logical(
[1 1 1 0 0 0 0 0
 1 1 1 0 1 1 0 0
 1 1 1 0 0 0 1 0
 1 1 1 0 0 0 1 0
 1 1 1 0 0 0 1 0
 1 1 1 0 0 1 1 0
 1 1 1 0 0 0 0 0]);

L = bwlabel(BW,4)
L =

     1     1     1     0     0     0     0     0
     1     1     1     0     2     2     0     0
     1     1     1     0     2     2     0     0
     1     1     1     0     0     0     3     0
     1     1     1     0     0     0     3     0
     1     1     1     0     0     0     3     0
     1     1     1     0     0     3     3     0
     1     1     1     0     0     0     0     0
```

Connected component in Matlab: `bwlabel`

```
BW = logical(
[1 1 1 0 0 0 0 0
 1 1 1 0 1 1 0 0
 1 1 1 0 0 0 1 0
 1 1 1 0 0 0 1 0
 1 1 1 0 0 0 1 0
 1 1 1 0 0 1 1 0
 1 1 1 0 0 0 0 0]);
```

```
L = bwlabel(BW,4)
L =

     1     1     1     0     0     0     0     0
     1     1     1     0     2     2     0     0
     1     1     1     0     2     2     0     0
     1     1     1     0     0     0     3     0
     1     1     1     0     0     0     3     0
     1     1     1     0     0     0     3     0
     1     1     1     0     0     3     3     0
     1     1     1     0     0     0     0     0
```

```
L = bwlabel(BW,8)
L =

     1     1     1     0     0     0     0     0
     1     1     1     0     2     2     0     0
     1     1     1     0     2     2     0     0
     1     1     1     0     0     0     2     0
     1     1     1     0     0     0     2     0
     1     1     1     0     0     0     2     0
     1     1     1     0     0     2     2     0
     1     1     1     0     0     0     0     0
```

Connected component in Matlab: `bwlabel`

```
BW = logical(
[1 1 1 0 0 0 0 0
 1 1 1 0 1 1 0 0
 1 1 1 0 0 0 1 0
 1 1 1 0 0 0 1 0
 1 1 1 0 0 0 1 0
 1 1 1 0 0 1 1 0
 1 1 1 0 0 0 0 0]);
```

These regions are connected only in 8 or m-adjacency

```
L = bwlabel(BW,4)
L =

     1     1     1     0     0     0     0     0
     1     1     1     0     2     2     0     0
     1     1     1     0     2     2     0     0
     1     1     1     0     0     0     3     0
     1     1     1     0     0     0     3     0
     1     1     1     0     0     0     3     0
     1     1     1     0     0     3     3     0
     1     1     1     0     0     0     0     0
```

```
L = bwlabel(BW,8)
L =

     1     1     1     0     0     0     0     0
     1     1     1     0     2     2     0     0
     1     1     1     0     2     2     0     0
     1     1     1     0     0     0     2     0
     1     1     1     0     0     0     2     0
     1     1     1     0     0     0     2     0
     1     1     1     0     0     2     2     0
     1     1     1     0     0     0     0     0
```

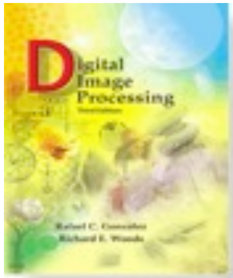
Example: we wish to count the number of rice grains in the rice picture.

The rice grains correspond to the "connected" components of the region.
However, we have to first remove some small pixels that are not rice grains.

(this can be done by removing all regions that have less than, say, 10 pixels)

```
img2 = bwareaopen(bw, 10); figure, imshow(img2)
```

The difference of the two images is just "noise"



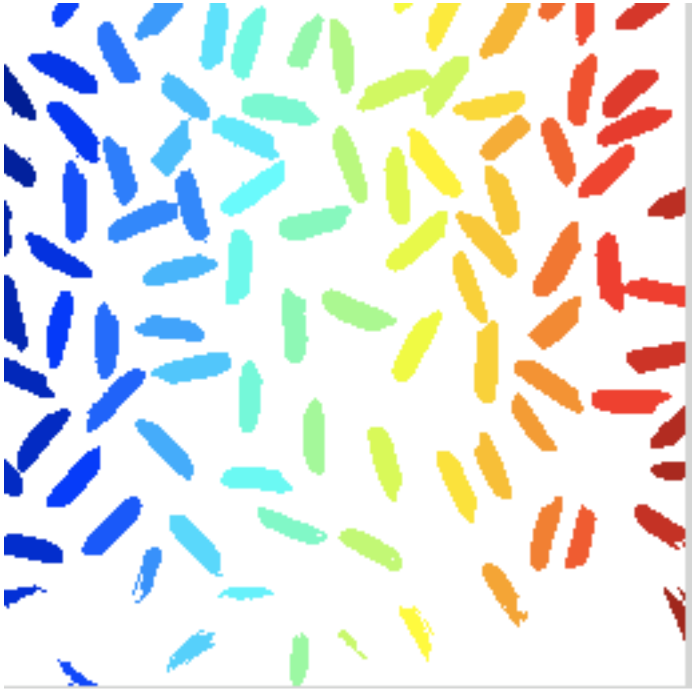
Count the connected components

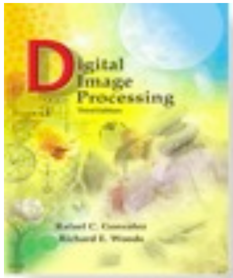
```
>> [lb2, num] = bwlabel(img2);
>> num

num =

    90
```

There are 90 rice grains in the picture.

Then we can visualize any of the rice grains (connected components)

```
figure, imshow(label2rgb(lb2))
```

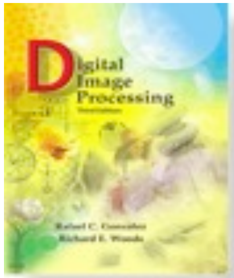




Returning to definitions….

Given a subset S of pixels, we define as *complement* S^C the set of points that are not in S.

```
BWC = 1-BW

BWC =

     0     0     0     1     1     1     1     1
     0     0     0     1     0     0     1     1
     0     0     0     1     0     0     1     1
     0     0     0     1     1     1     0     1
     0     0     0     1     1     1     0     1
     0     0     0     1     1     1     0     1
     0     0     0     1     1     0     0     1
     0     0     0     1     1     1     1     1
```



Background and foreground:

Assume the image has K disjoint regions R_k, $k = 1, \dots, K$ (not touching the border). Denote R_u the *union* of the regions and by $(R_u)^C$ the complement.

NB.For regions touching the borders, it is customary to add zero pixels around the full image.

R_u is called *foreground*

$(R_u)^C$ is called *background*

Boundary and edges:

The *boundary* of a region R is the set of all points that are adjacent to R^C

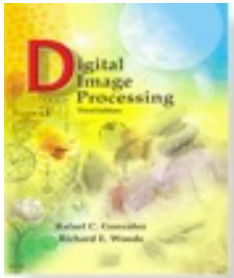
NB. The definition of boundary is dependent on the adjacency chosen. Default: 8-pixels.

Inner border: boundary of the region's foreground

Outer border: boundary of the regions' background

Outer borders form closed paths (important for border-following algorithms).

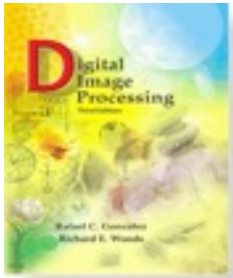
Edge: is different from border as edge is related to pixels with derivative values over a given threshold (local concept, compared to boundary, that is a global concept). In binary images, the two definitions are the same.



Distance measures

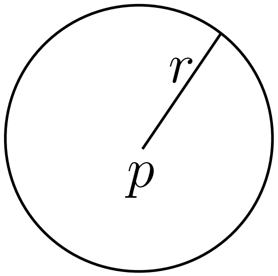
A distance function (or metric) is a function $D(p, q)$ where:

- $D(p, q) \geq 0$ and $D(p, q) = 0$ if and only if $p = q$ (positivity)
- $D(p, q) = D(q, p)$ (symmetry)
- $D(p, r) \leq D(p, q) + D(q, r)$ (triangle inequality).



Assume p has coordinates (x, y) and q has coordinates (s, t).

Euclidean distance: $D_e(p, q) = \sqrt{(x-s)^2 + (y-t)^2}$



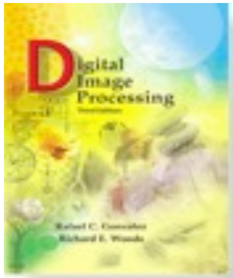
City-block distance or D_4 distance: $D_4(p, q) = |x - s| + |y - t|$.

(1-norm in vector spaces)

```
    2
  2 1 2
2 1 0 1 2
  2 1 2
    2
```

Chessboard distance or D_8 distance: $D_8(p, q) = \max\{|x - s|, |y - t|\}$ (infinity distance)

2	2	2	2	2
2	1	1	1	2
2	1	0	1	2
2	1	1	1	2
2	2	2	2	2



Chessboard distance or D_8 distance: $D_8(p,q) = \max\{|x-s|,|y-t|\}$
(infinity distance)

$$\begin{array}{ccccc}
2 & 2 & 2 & 2 & 2 \\
2 & 1 & 1 & 1 & 2 \\
2 & 1 & 0 & 1 & 2 \\
2 & 1 & 1 & 1 & 2 \\
2 & 2 & 2 & 2 & 2
\end{array}$$

The definition of distance involves only the coordinates of the points hence is *independent* of adjacency.

There is however another distance *Dm,* defined as the length of the shortest path between the points.