Computer Vision

CSE/ECE 344/544

Message

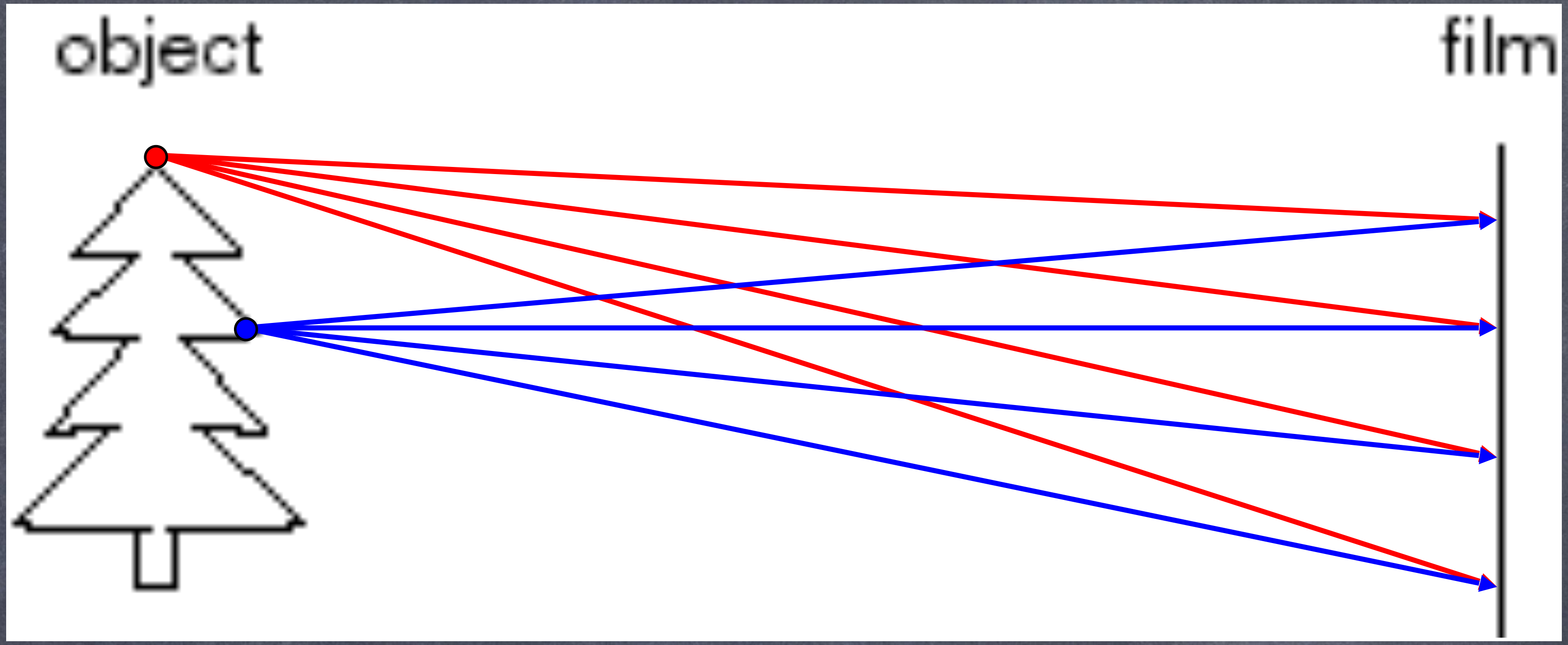
- If you do not pass the quiz but still want to do the course, then: (i) this is your research theme/area, (ii) you meet me by Monday (in between 2:30 - 3:30)
- I am ok to teach 10, 20, 50, 100 students if they are interested genuinely but I do not want 200 students just because CV/ML/IR is hot area and it looks good to have these courses in their resume.

Computer Vision

Image from: http://kirkh.deviantart.com/art/BioMech-Eye-168367549

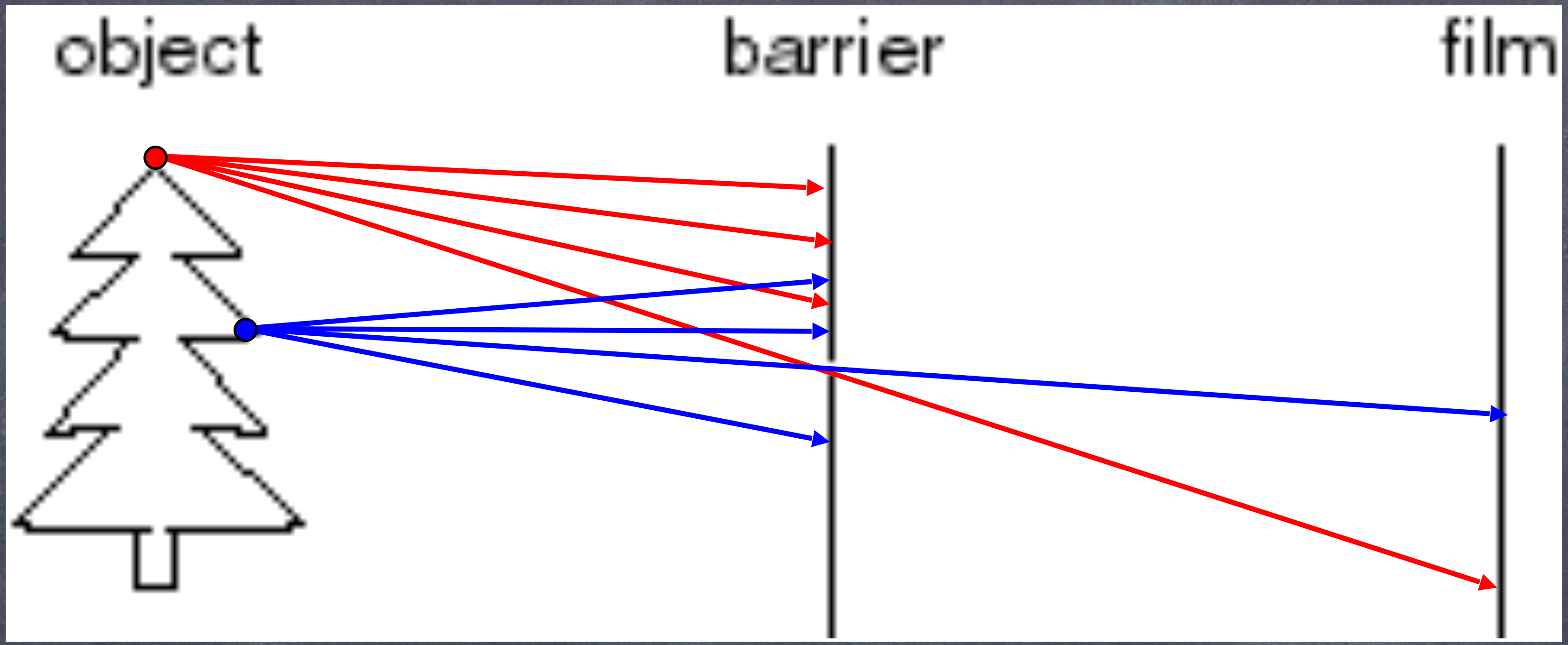
What is the first step in Image Processing and Computer Vision?

Image formation



- Let's design a camera
  - Idea 1: put a piece of film in front of an object
  - Do we get a reasonable image?

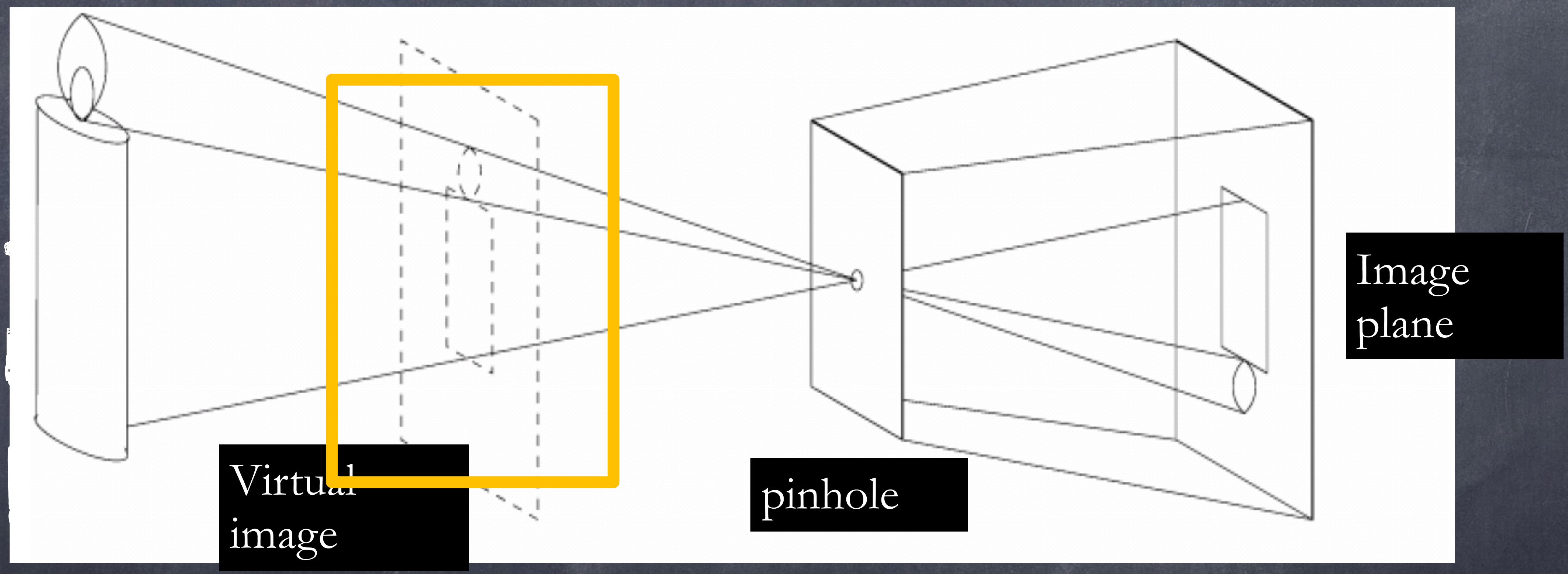
Pinhole camera



- Add a barrier to block off most of the rays
  - This reduces blurring
  - The opening is known as the **aperture**
  - How does this transform the image?

Slide by Steve Seitz

Pinhole camera



If we treat pinhole as a point, only one ray from any given point can enter the camera.

Fig from Forsyth and Ponce

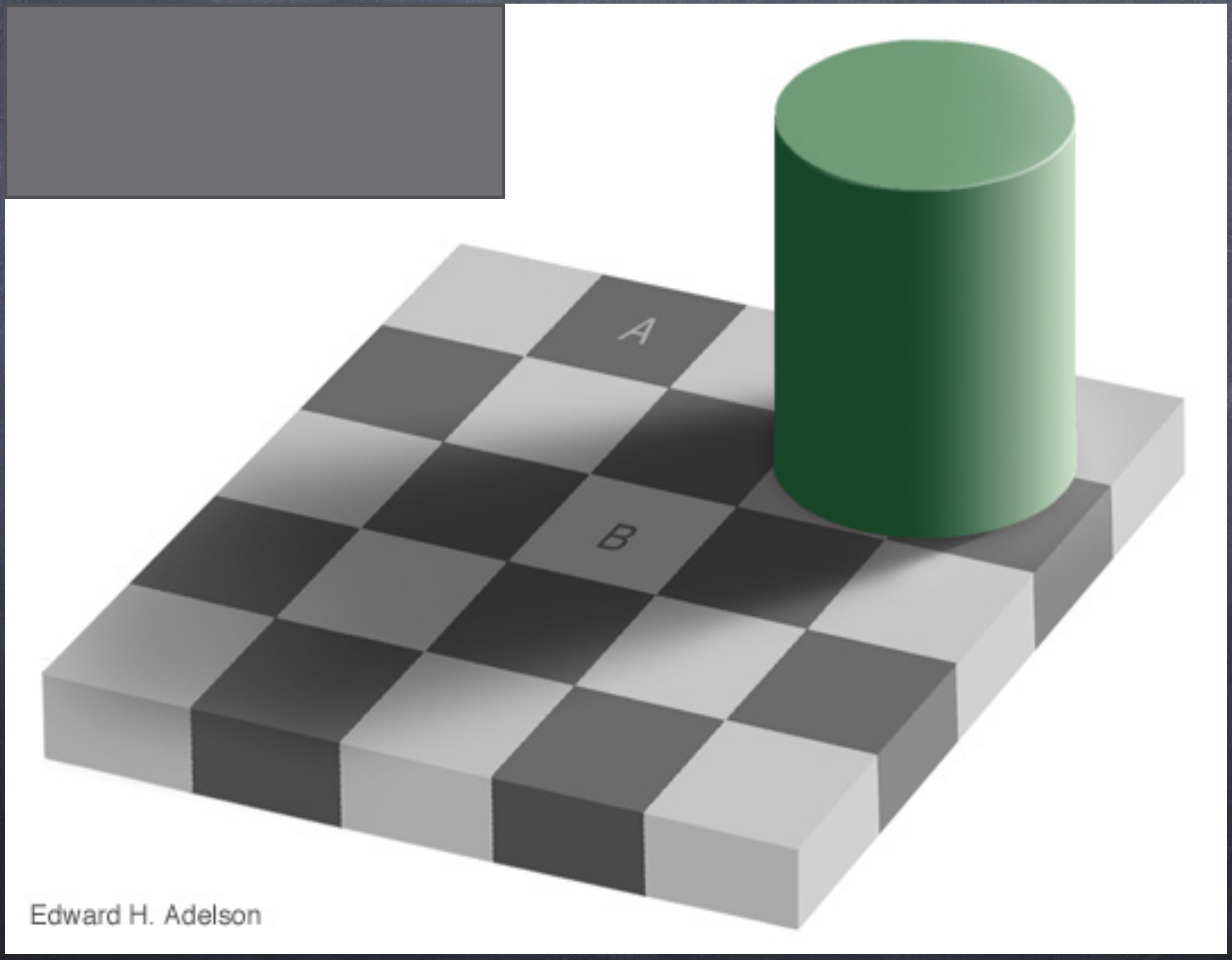
Perspective effects

Perspective effects

Physical parameters of image formation

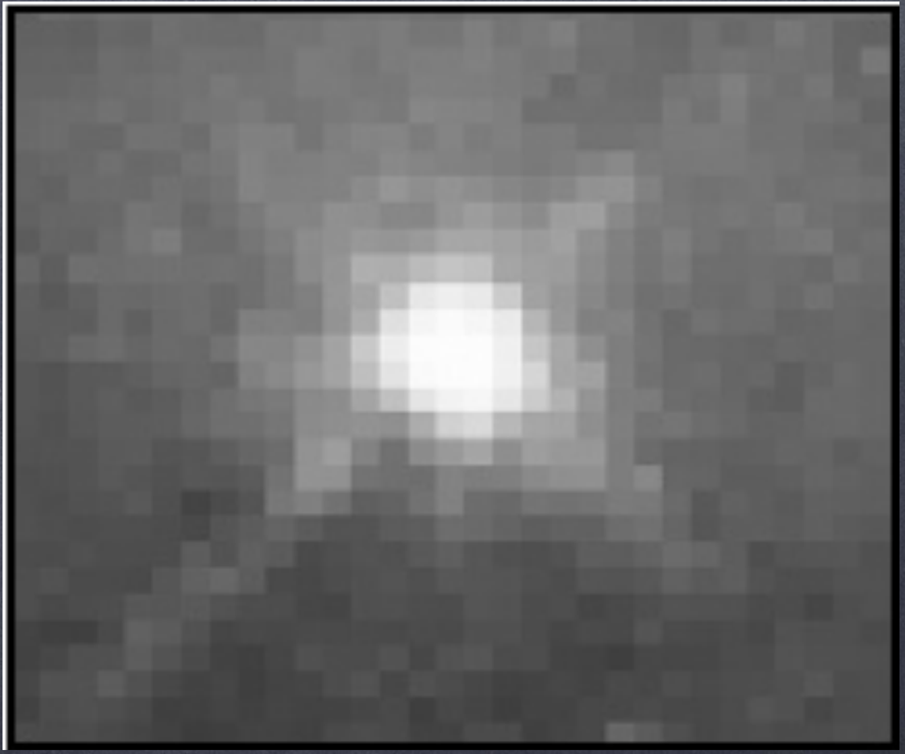
- Geometric
  - Type of projection
  - Camera pose
- Optical
  - Sensor's lens type
  - focal length, field of view, aperture
- Photometric
  - Type, direction, intensity of light reaching sensor
  - Surfaces' reflectance properties

Why is Vision Hard?

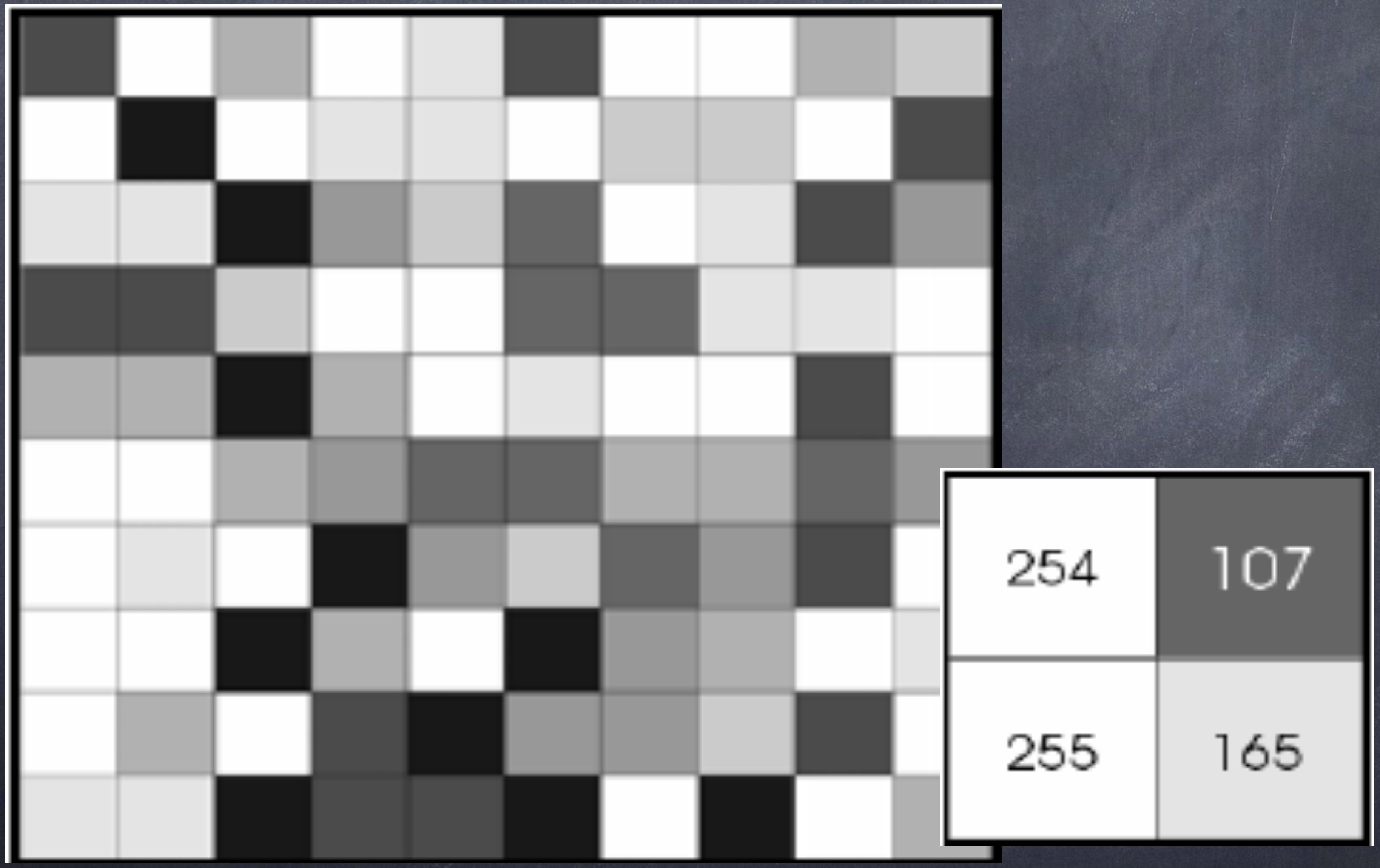


What is an image?

- An image is two dimensional function, $f(x,y)$, where x and y are spatial coordinates, and the amplitude of f at any pair of coordinates (x,y) is called the intensity or grey level of the image at that point.



Digital Image?



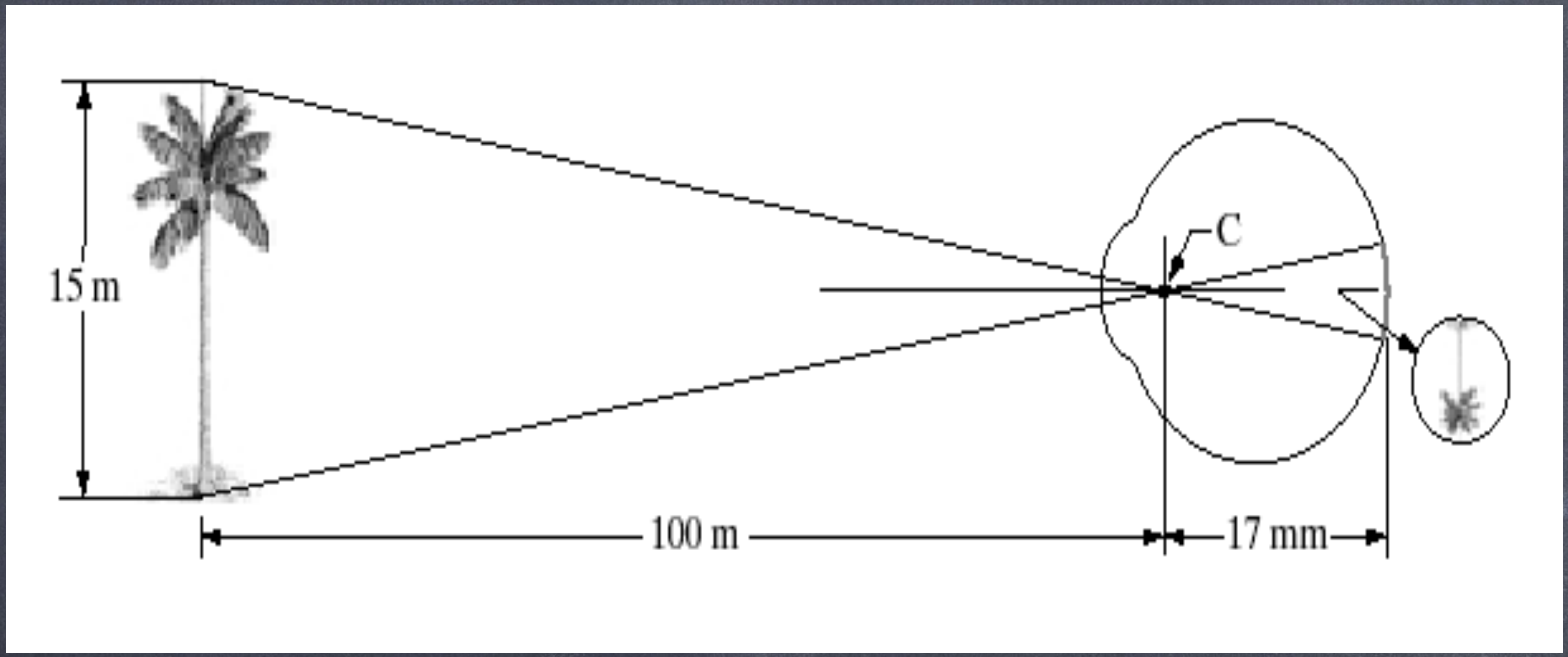
Fundamental Steps

- Fundamental steps are
  - Image/video Acquisition
  - Enhancement, Restoration and Reconstruction
  - Segmentation
  - Representation and Description
  - Applications: e.g. Object Recognition

Acquisition Tools

- Human Eye
- Ordinary Camera
- X-Ray Machine
- Positron Emission Tomography, MRI, etc.
- Infrared Imaging
- Geophysical Imaging

Image Formation in the Eye



Electromagnetic spectrum

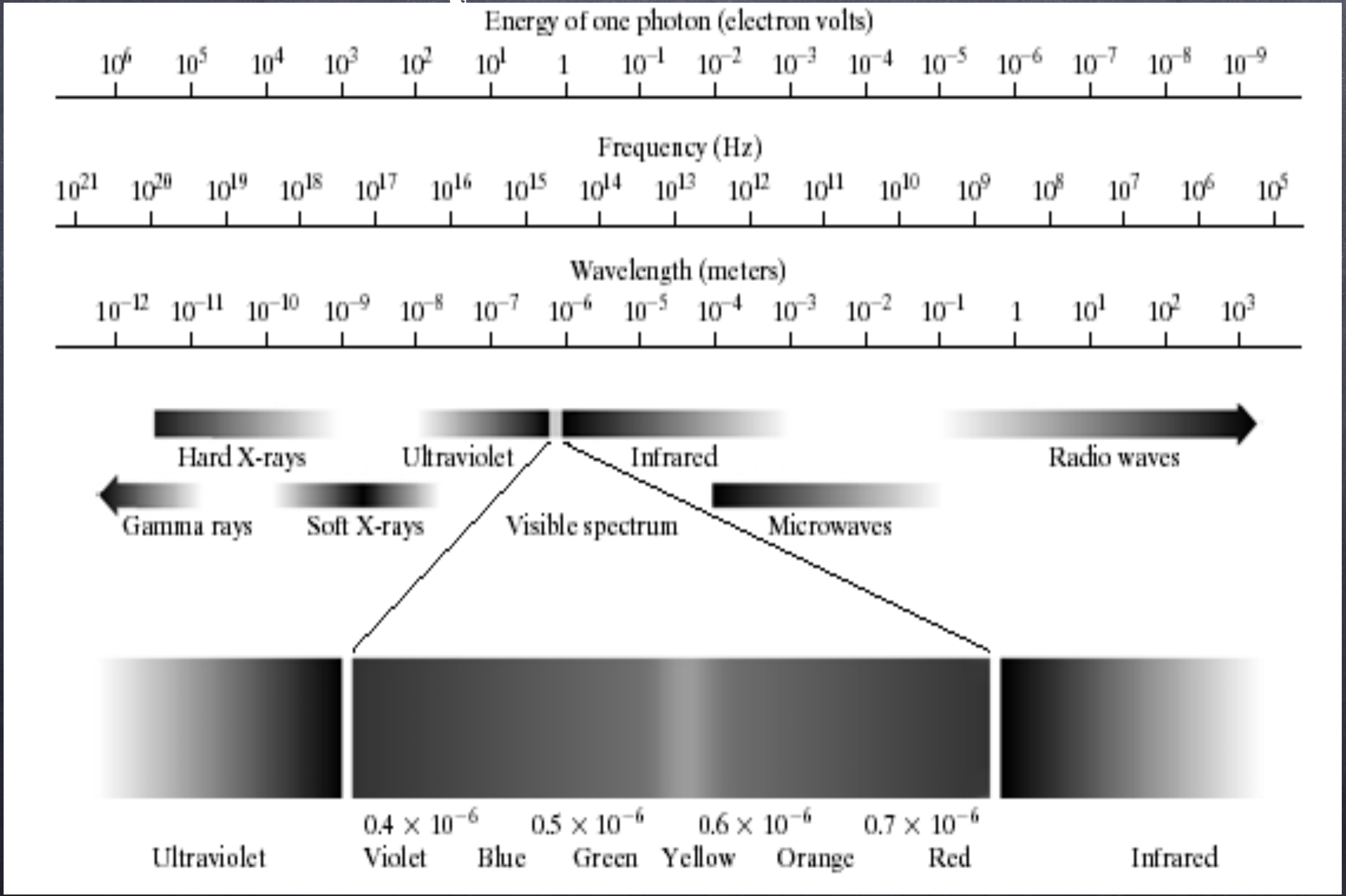


Image Acquisition …
Depends on background

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.

Image Formation

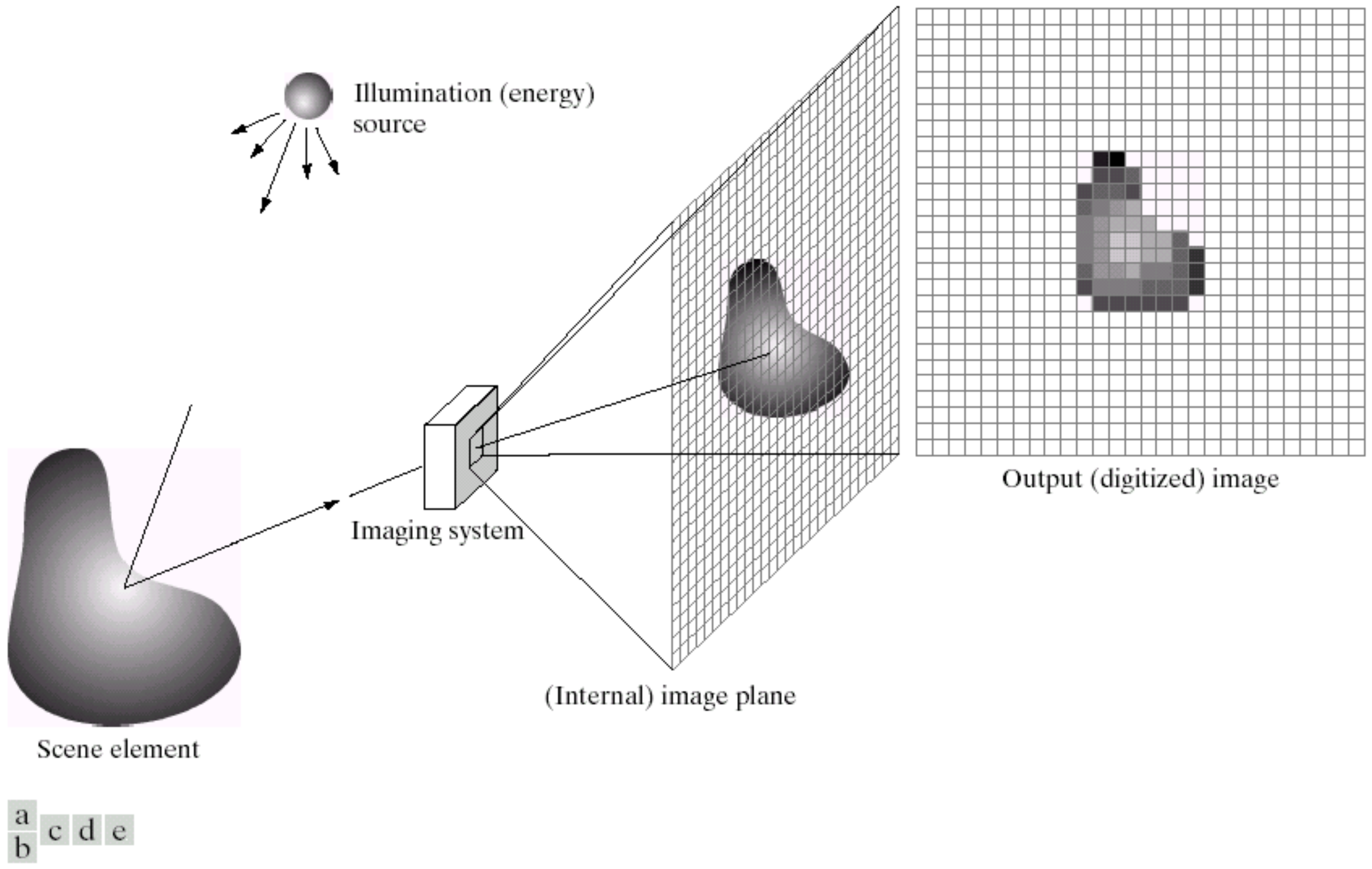
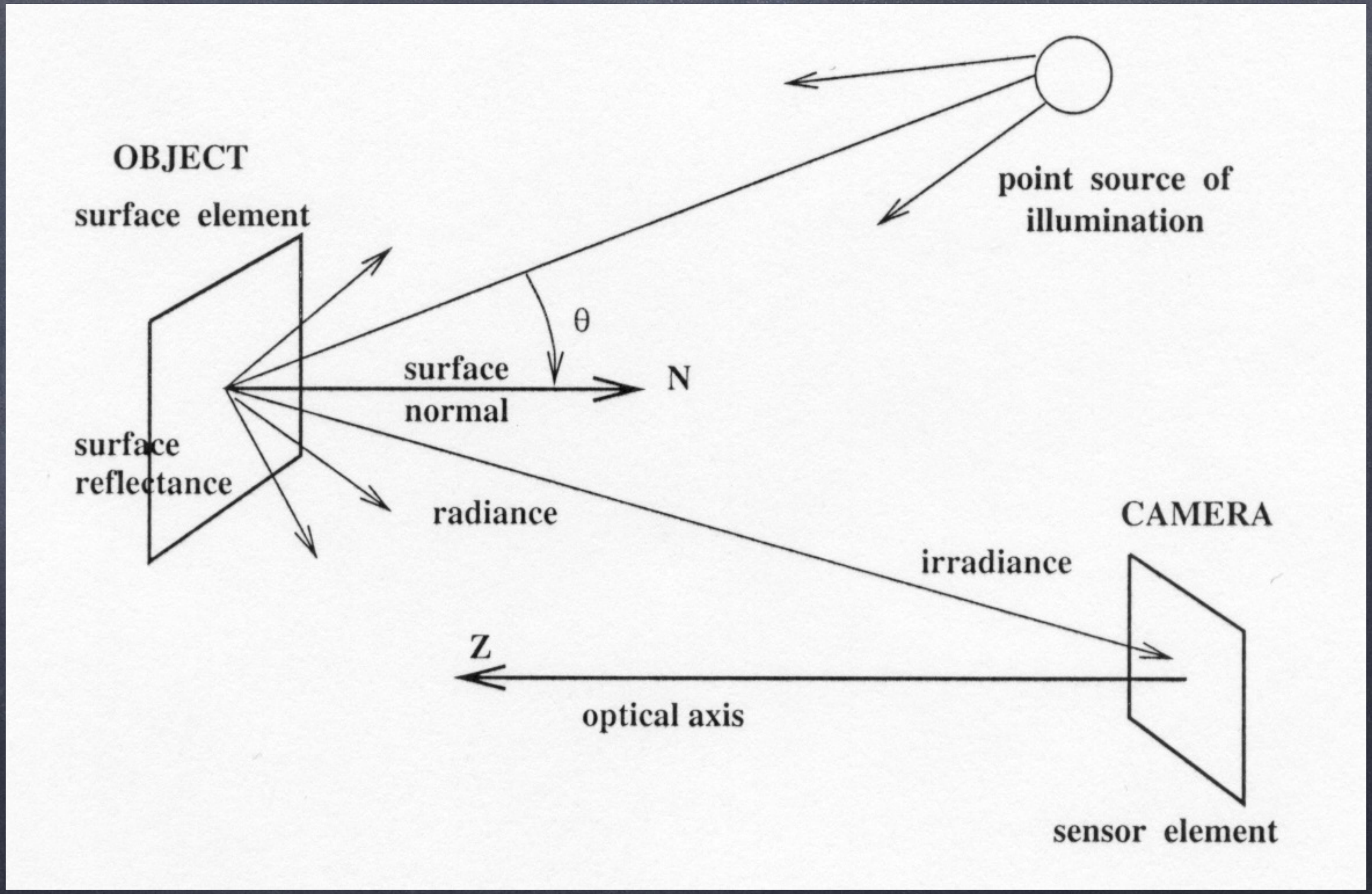


FIGURE 2.15 An example of the digital image acquisition process. (a) Energy ("illumination") source. (b) An element of a scene. (c) Imaging system. (d) Projection of the scene onto the image plane. (e) Digitized image.

*f(x,y) = reflectance(x,y) * illumination(x,y)*

A Simple model of image formation

- The scene is illuminated by a single source.
- The scene reflects radiation towards the camera.
- The camera senses it via chemicals on film (or via sensors).

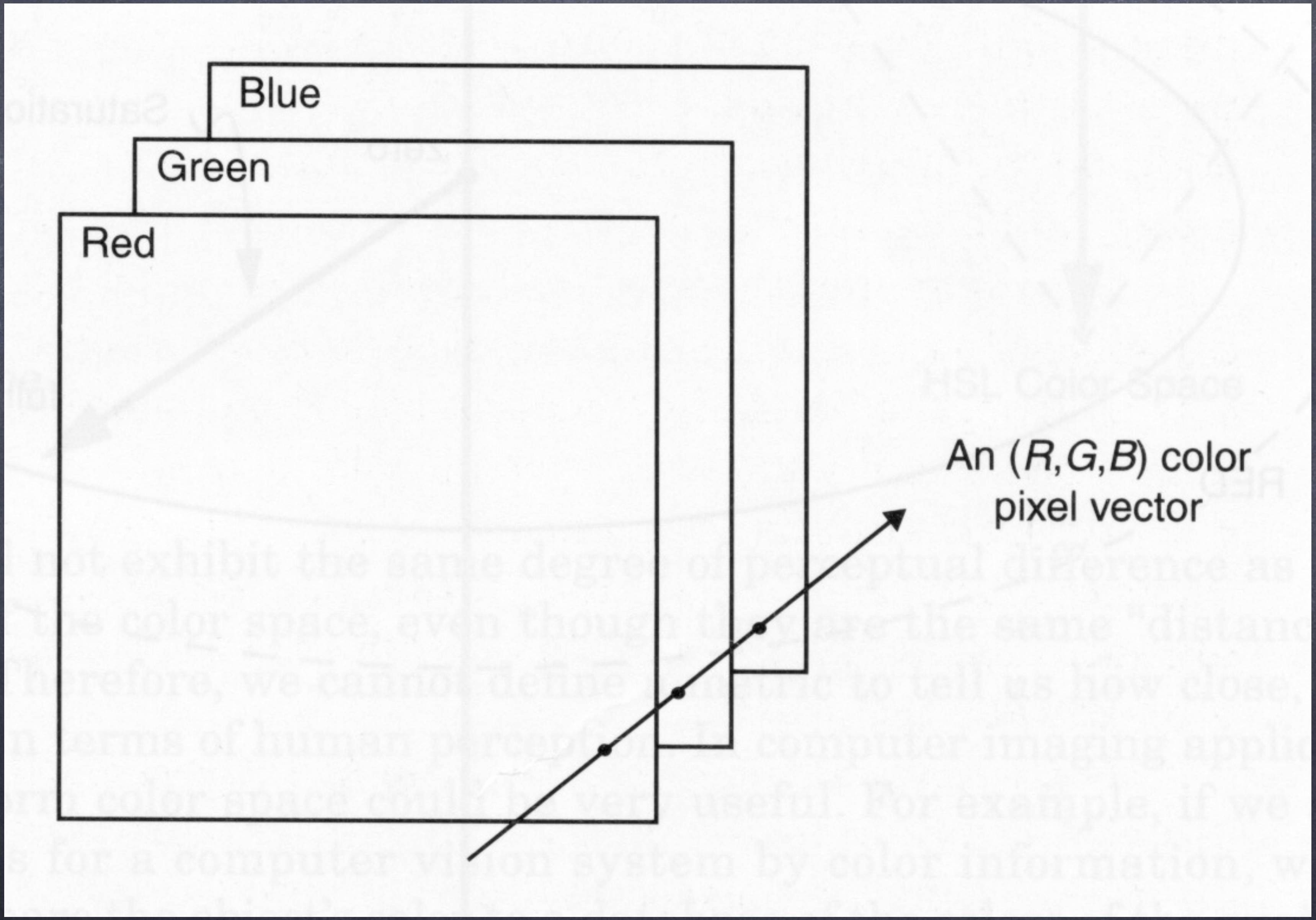


What is an image?

- We can think of an image as a function, f
    - f(x, y) gives the intensity at position (x, y)
    - Realistically, we expect the image only to be defined over a rectangle, with a finite range
    - A color image is just three functions pasted together. We can write this as a "vector-valued" function:

$$f(x,y) = \begin{bmatrix} r(x,y) \\ g(x,y) \\ b(x,y) \end{bmatrix}$$

Color image



Sampling and Quantization

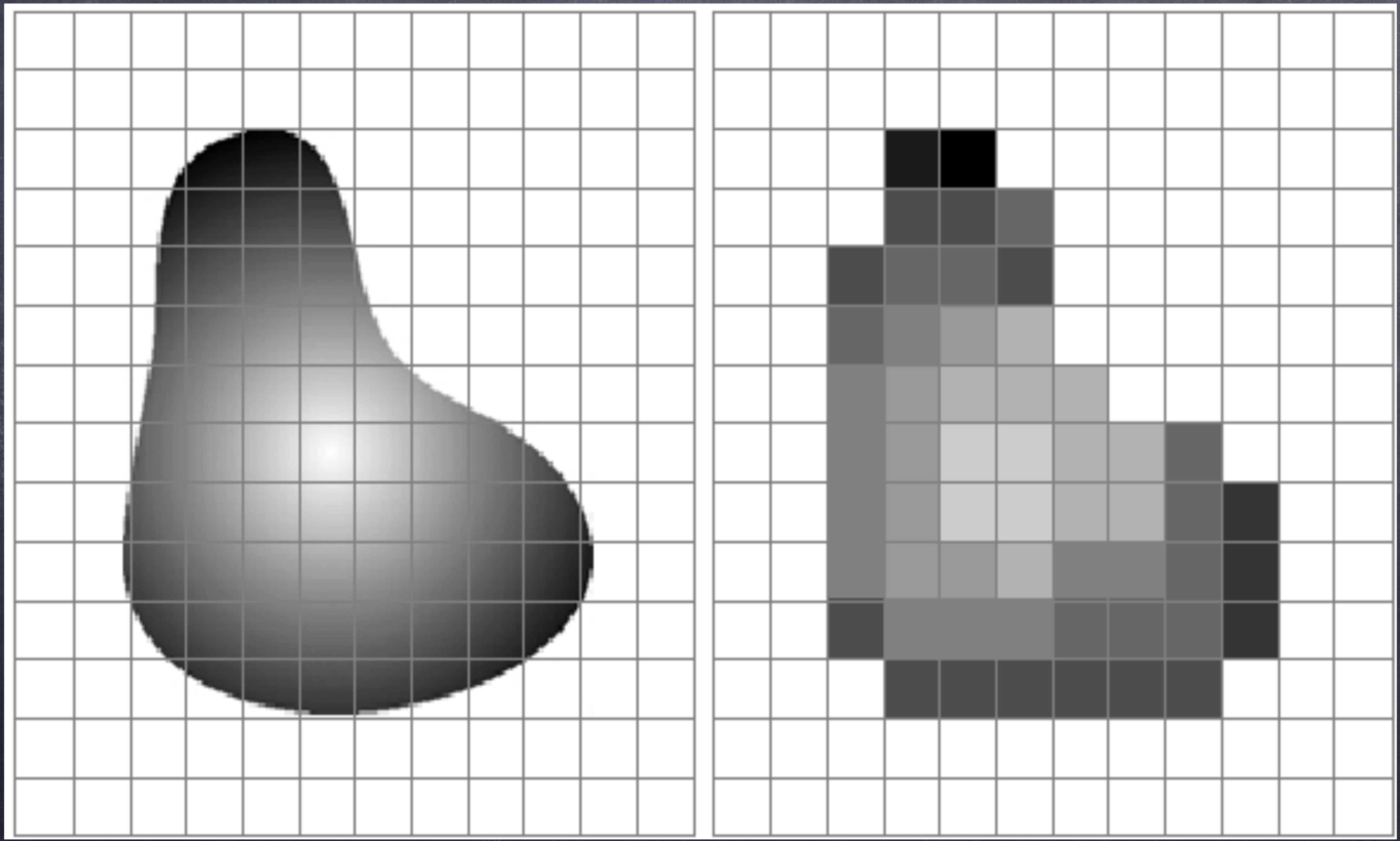


Image before sampling and quantization

Result of sampling and quantization

Image Sampling and Quantization

- The output of most sensors is a continuous voltage waveform whose amplitude and spatial behavior are related to the physical phenomenon being sensed.

- To create a digital image, we need to convert the continuous sensed data into digital form. This involves two processes: *sampling* and *quantization*.

- Digitizing the coordinate values is called *sampling*.
- Digitizing the amplitude values is called *quantization*.

Sampling and Quantization

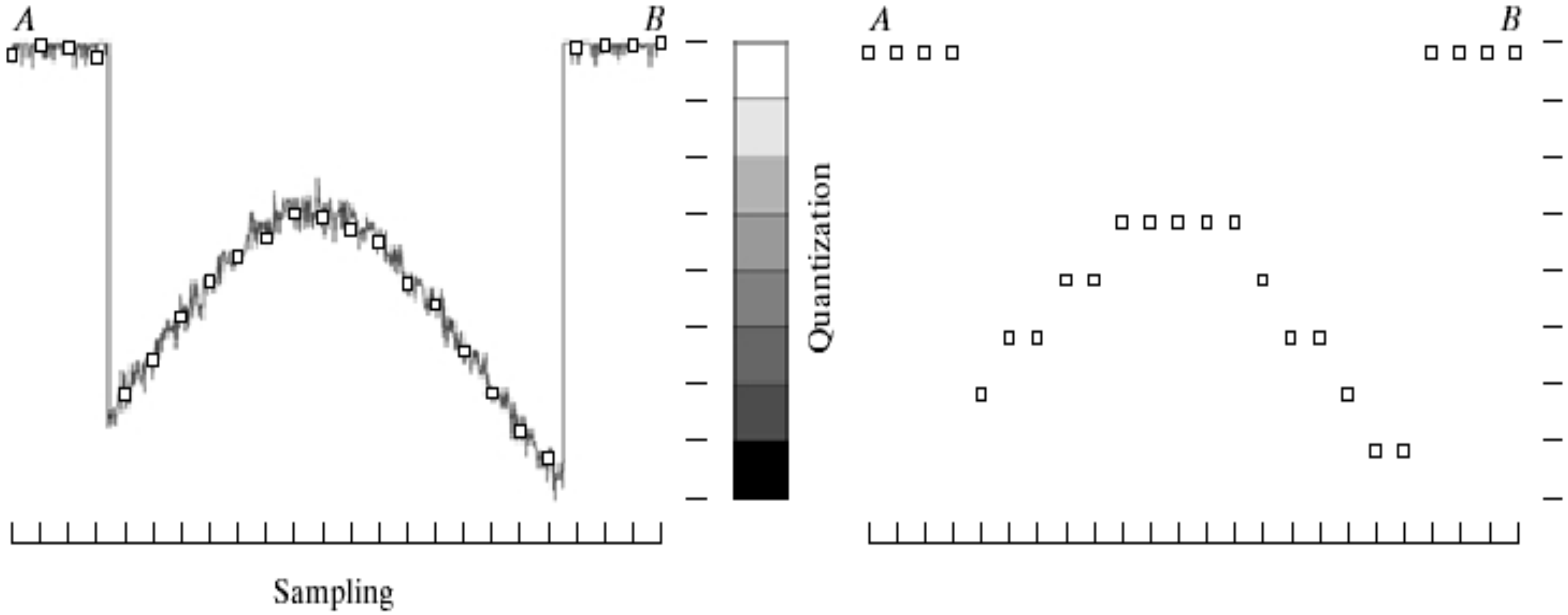
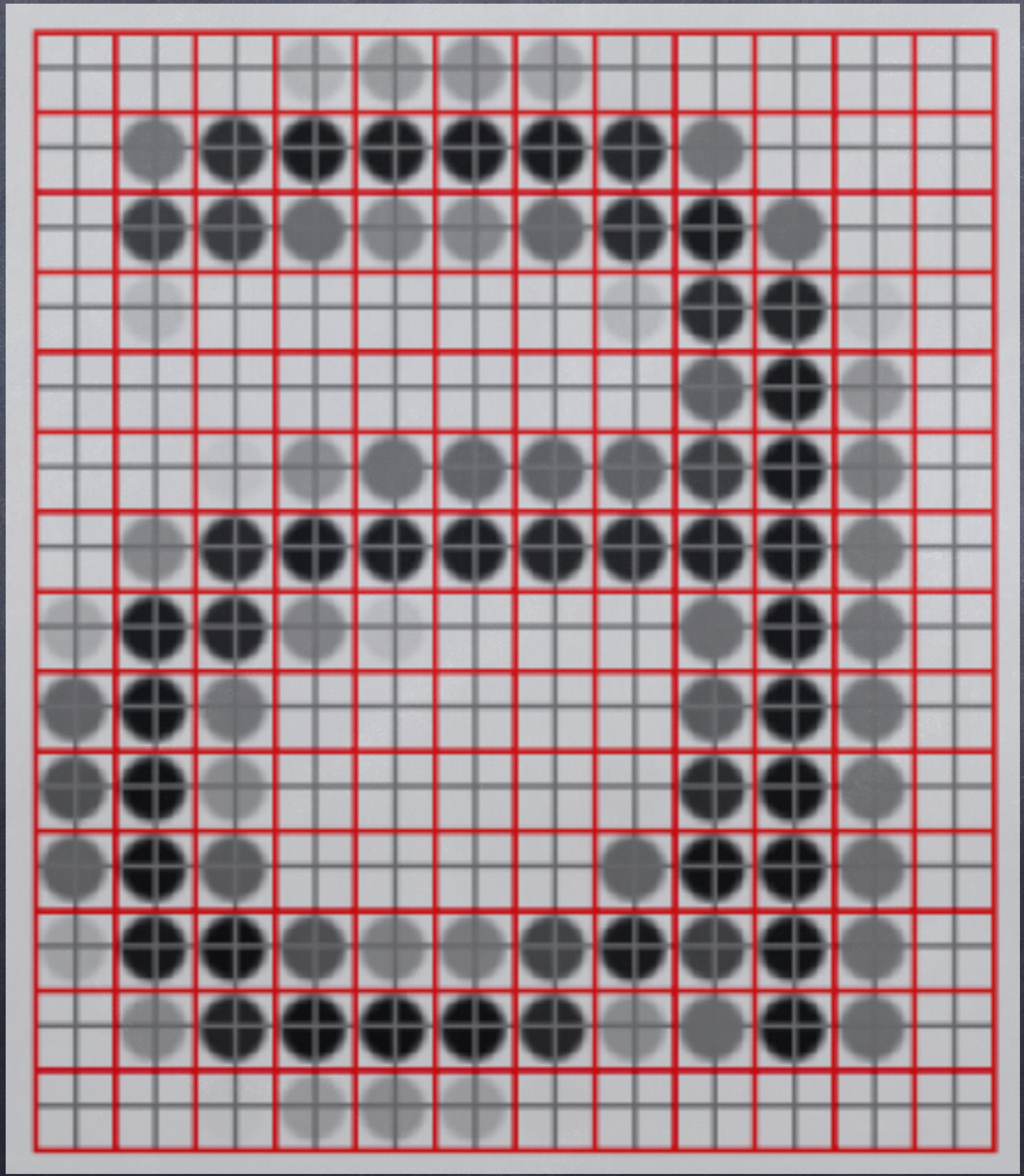


Image sampling



Quantization

- Quantization is the procedure of constraining the value of a function at a sampling point to a predetermined finite set of discrete values.
- If we want to specify the temperature of Delhi, ranging from 0°C to 50°C, up to a precision of 0.1°C
- we must be able to represent 501 possible values, which require 9 bits to represent one sample.
- On the other hand, if we only need a precision of 1°C, we only have 51 possible values requiring 6 bits for the representation.
- For image processing, higher precision give higher image quality but requires more bits in the representation of the samples.

Sampling and Quantization

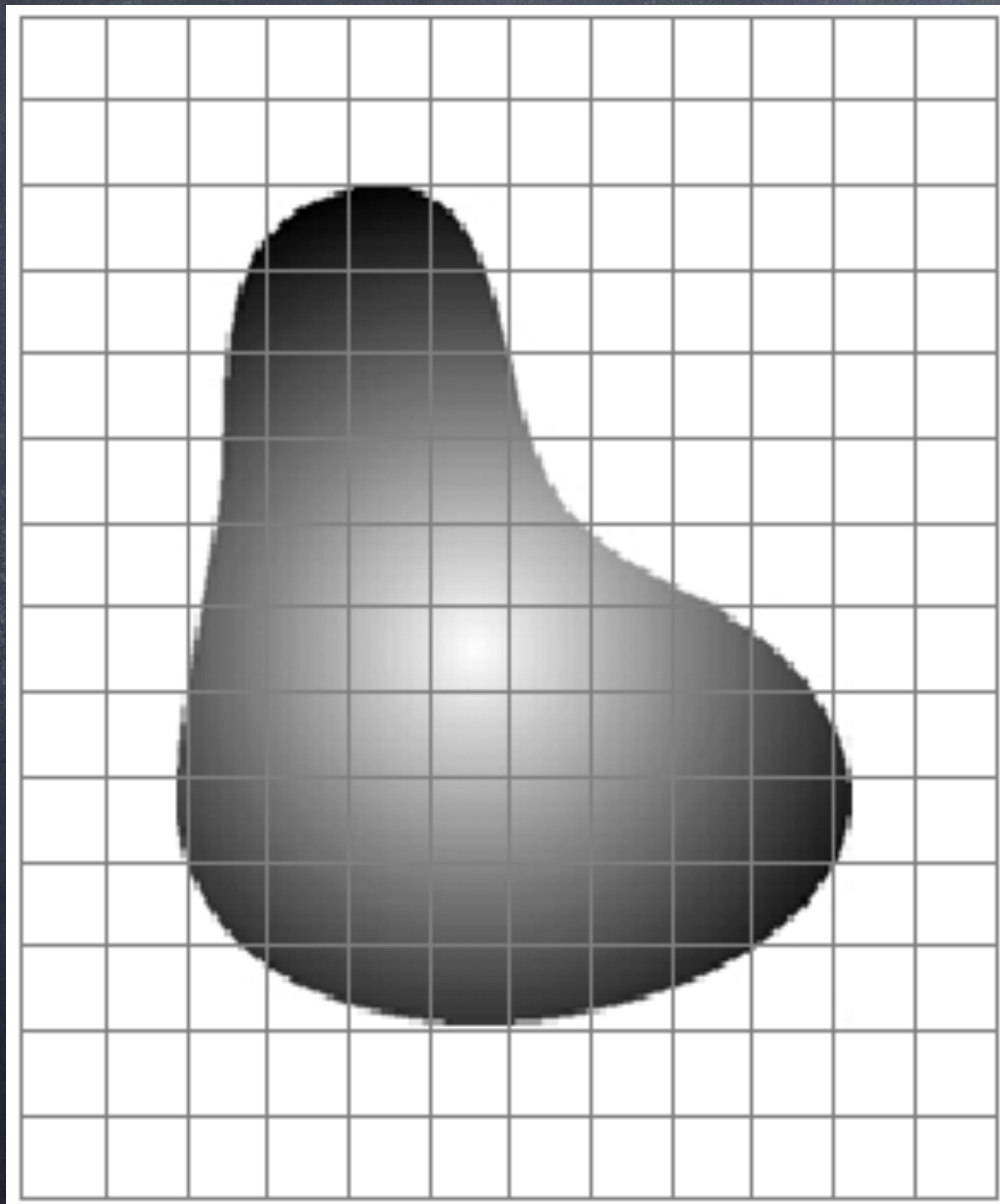
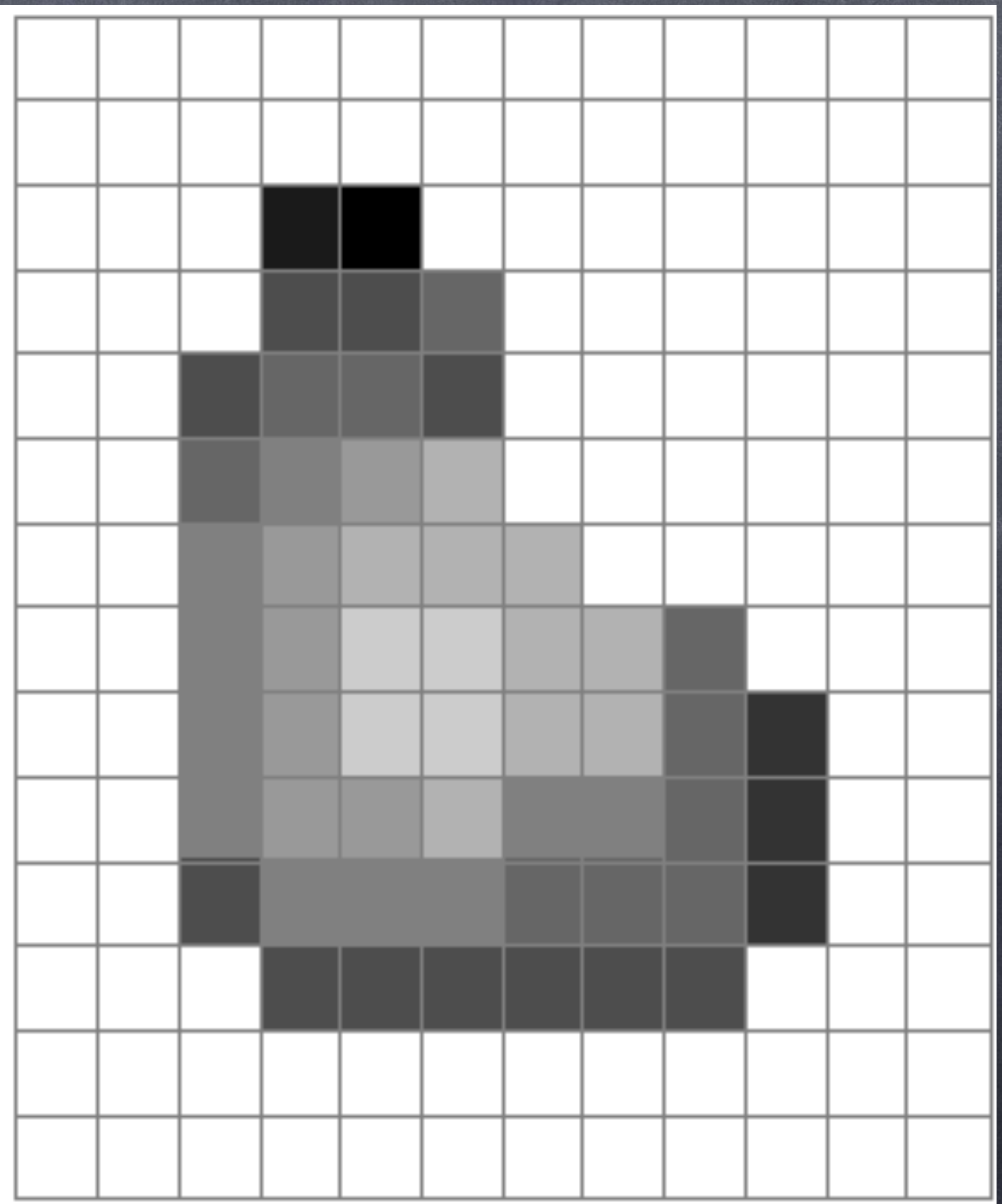


Image before sampling and quantization



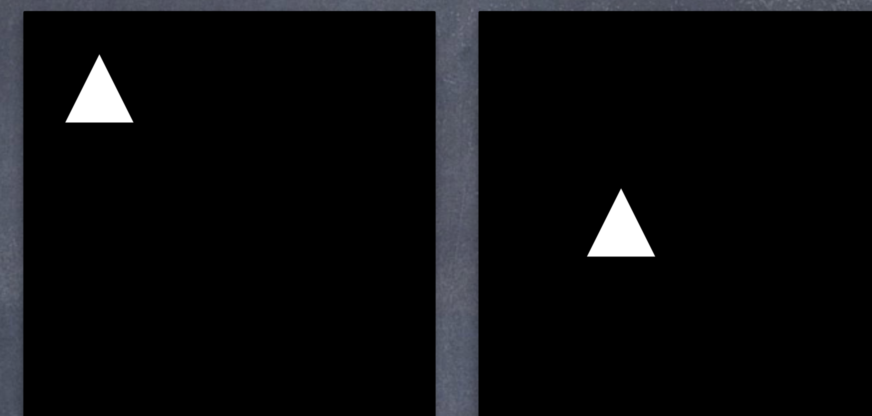
Result of sampling and quantization

Image Pixel Values

0	3	2	5	4	7	6	9	8
3	0	1	2	3	4	5	6	7
2	1	0	3	2	5	4	7	6
5	2	3	0	1	2	3	4	5
4	3	2	1	0	3	2	5	4
7	4	5	2	3	0	1	2	3
6	5	4	3	2	1	0	3	2
9	6	7	4	5	2	3	0	1
8	7	6	5	4	3	2	1	0

Example: Big triangles vs. Little triangles

Big triangles



Little triangles

?

Slide credit: Devi Parikh

Example: Big triangles vs. Little triangles

Slide credit: Devi Parikh

Example: Big triangles vs. Little triangles

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

Sum = 16

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

Sum = 13

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

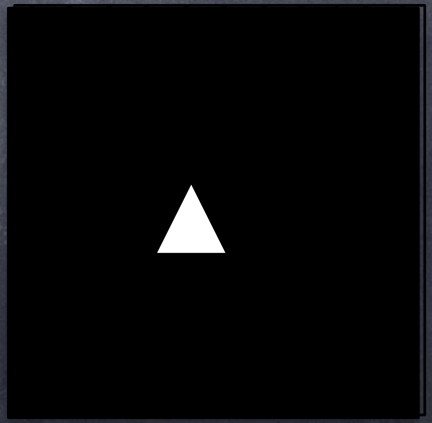
Sum = 2

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

Sum = 3

<u>Rule</u>

If sum > 10
    Answer = Big triangle
Else
    Answer = Little triangle



Little triangle
Sum = 2

Slide credit: Devi Parikh

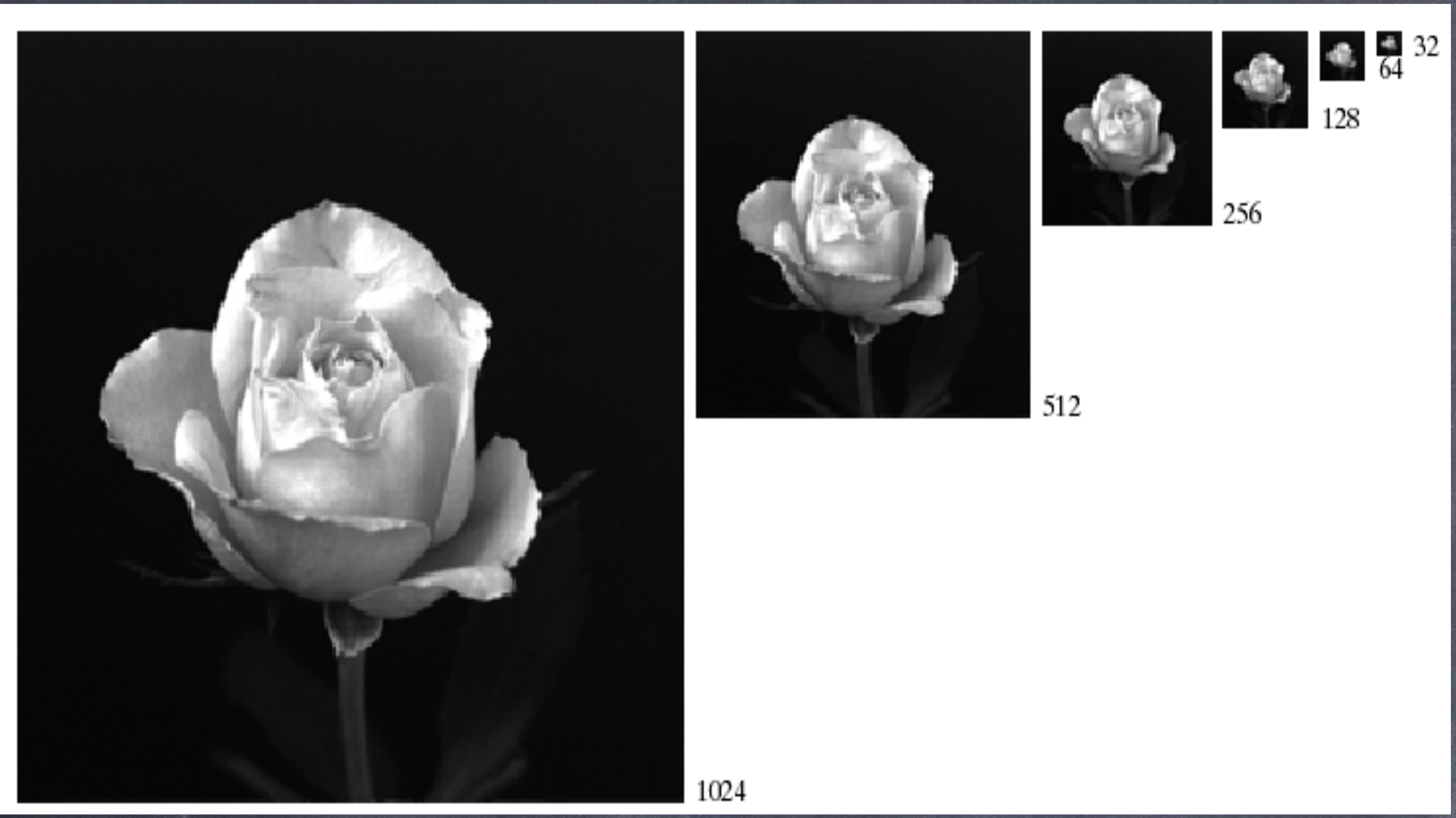
Image Size

- Due to processing, storage, and sampling hardware considerations, the number of gray levels typically is an integer power of 2:
- $L = 2^K$
- Where k is number of bits require to represent a grey value
- The discrete levels should be equally spaced and that they are integers in the interval [0, L-1].
- The number of bits, b, required to store a digitized image is
- $b=M*N*k$.
- Where M, N represent size of image

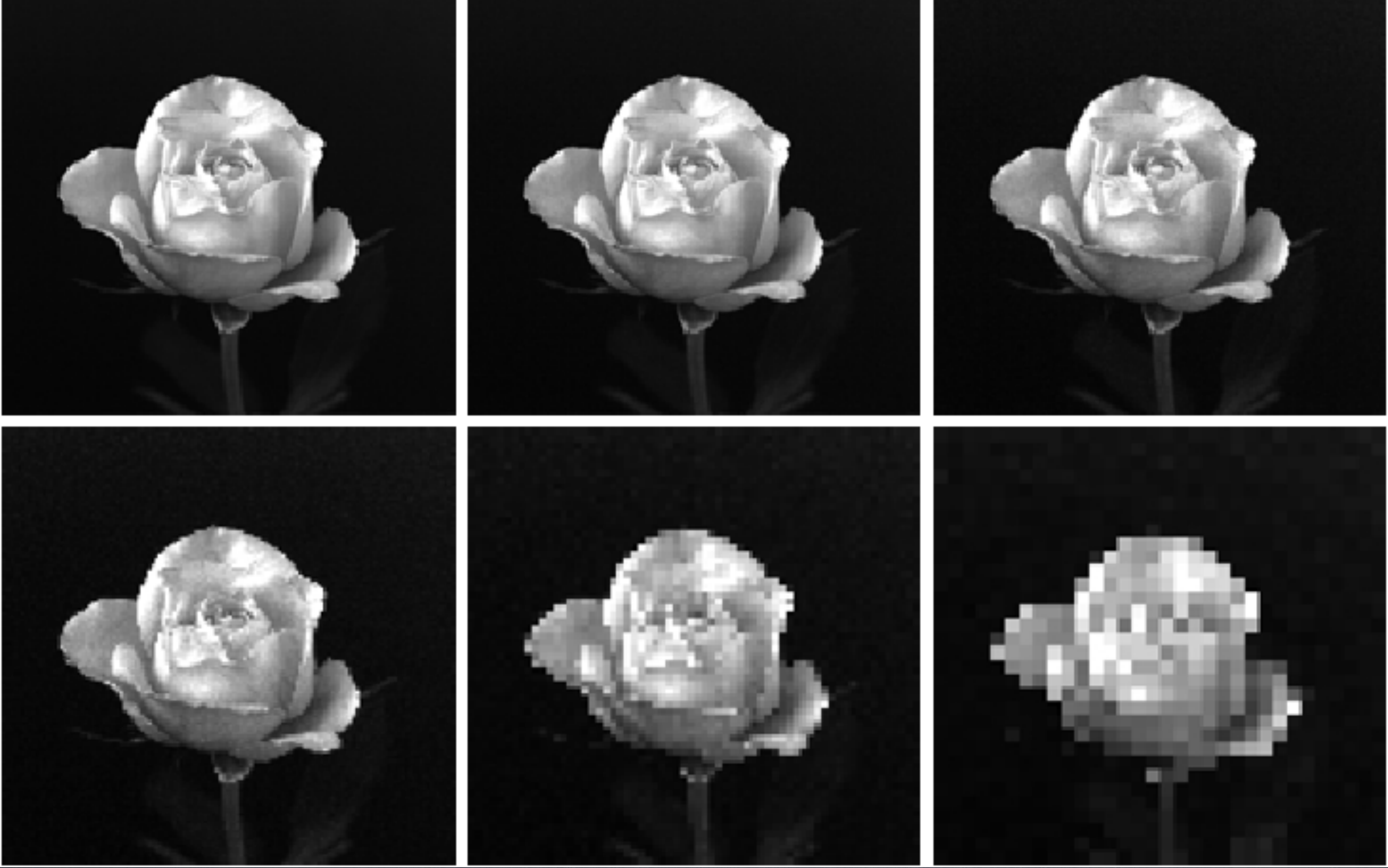
Resolution

- Spatial Resolution: Spatial resolution is the smallest detectable detail in an image.

- Grey level Resolution: *Gray-level resolution* similarly refers to the smallest detectable change in gray level.



A 1024*1024, 8-bit image subsampled down to size 32*32 pixels.The number of allowable gray levels was kept at 256



(a) 1024*1024, 8-bit image. (b) 512*512 image resampled into 1024*1024 pixels by row and column duplication. (c) through (f) 256*256, 128*128, 64*64, and 32*32 images resampled into 1024*1024 pixels.

MP, PPI, DPI ...

- Megapixel:
  - 2048×1536 = 3,145,728 pixels
  - 3.1 megapixels
- PPI/DPI:
  - 15 inch (38 cm) display
    - 12 inches (30.48 cm) wide by 9 inches (22.86 cm) high,

IP CV

- An image processing and computer vision operation typically defines a new image g in terms of an existing image f.
- We can transform either the range of f.

$$g(x,y) = t(f(x,y))$$

- Or the domain of f:

$$g(x,y) = f(t_x(x,y), t_y(x,y))$$

IP - CV

- In reality, image processing and CV is nothing but playing with matrices operation with 256 grayscale images-

  - Absolute difference of two images
  - Add two images or add constant to image
  - Complement image
  - Divide one image into another or divide image by constant
  - Linear combination of images
  - Multiply two images or multiply image by constant
  - Subtract one image from another or subtract constant from image

Absolute difference of two images

- X = [255 10 75; 44 225 100];
- Y = [50 50 50; 50 50 50];
- Z = adi(X,Y) → we need to write a function adi
- Z = [205 40 25; 6 175 50]

Absolute difference of two images

Add two images or add constant to image

- X = [255 0 75; 44 225 100];
- Y = [50 50 50; 50 50 50];
- Z = add2im(X,Y)
- Z =
- [255 50 125; 94 255 150]

Subtract two images or add constant to image

- X = [255 10 75; 44 225 100];
- Y = [50 50 50; 50 50 50];
- Z = sub2im(X,Y)
- Z =
- [205 0 25; 0 175 50]

Complement image

- X = [255 10 75; 44 225 100];
- X2 = comp (X)
- X2 =
- [0 245 180; 211 30 155]

Divide one image into another or divide image by constant

- X = [255 10 75; 44 225 100];
- Y = [50 20 50; 50 50 50];
- Z = div (X,Y)
- Z =
- [5 1 2; 1 5 2]

Transpose

SampleArray = 1 3 5
2 4 6

SampleArray' = 1 2
3 4
5 6

Nested Operations

- X = [255 10 75; 44 225 100];
- Y = [50 20 50; 50 50 50];
- Z = div (add2im(X,Y),2)
- Z =
- [128 15 63; 47 128 75]

IP CV

- An image processing - computer vision operation typically defines a new image g in terms of an existing image f.

- We can transform either the range of f.

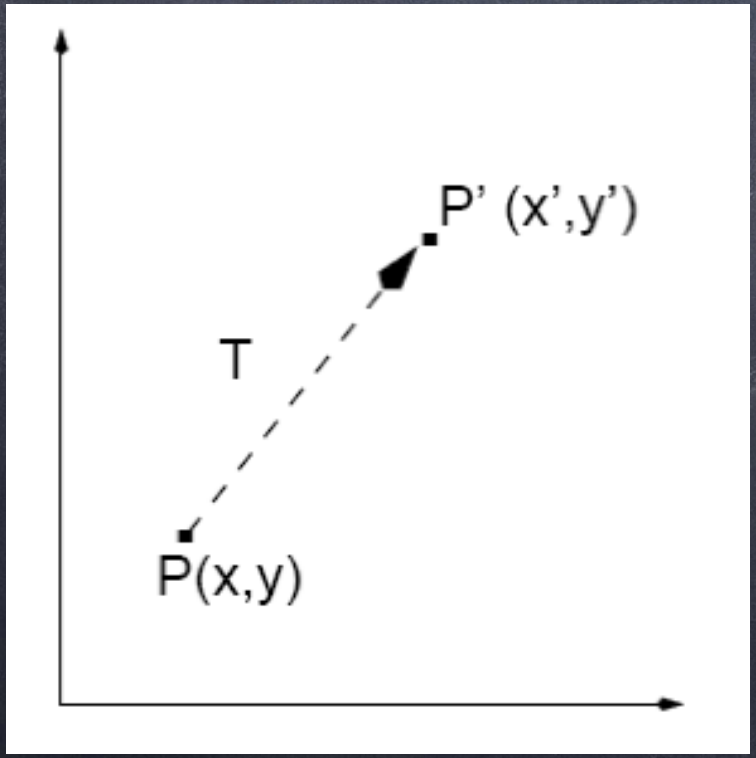
$$g(x, y) = t(f(x, y))$$

- Or the domain of f:

$$g(x, y) = f(t_x(x, y), t_y(x, y))$$

2D Translation

- Moves a point to a new location by adding translation amounts to the coordinates of the point.



$$x' = x + dx, \quad y' = y + dy$$

or

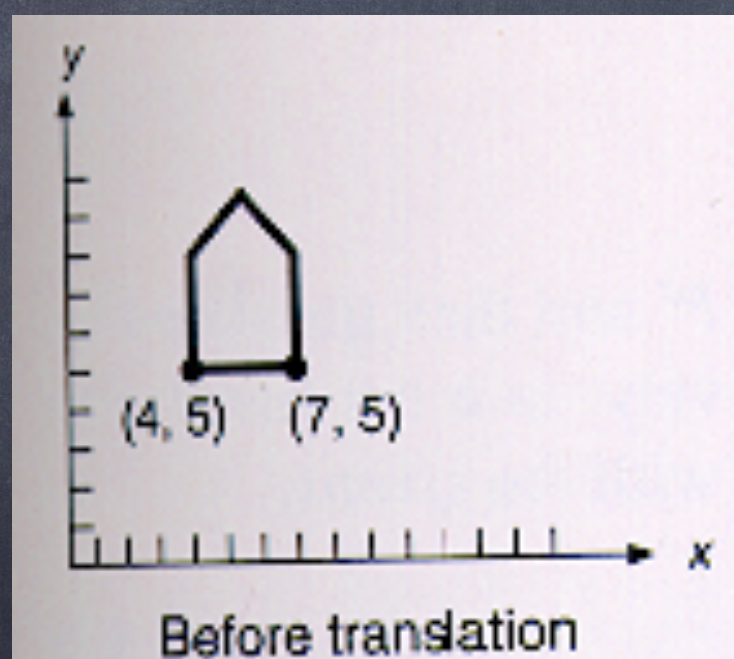
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} dx \\ dy \end{bmatrix}$$

or

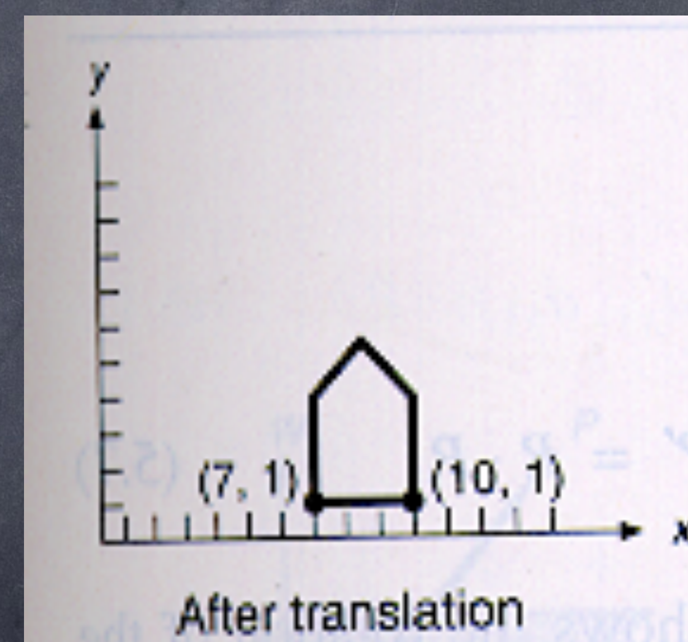
$$\underline{P' = P + T}$$

2D Translation (cont'd)

- To translate an object, translate every point of the object by the same amount.



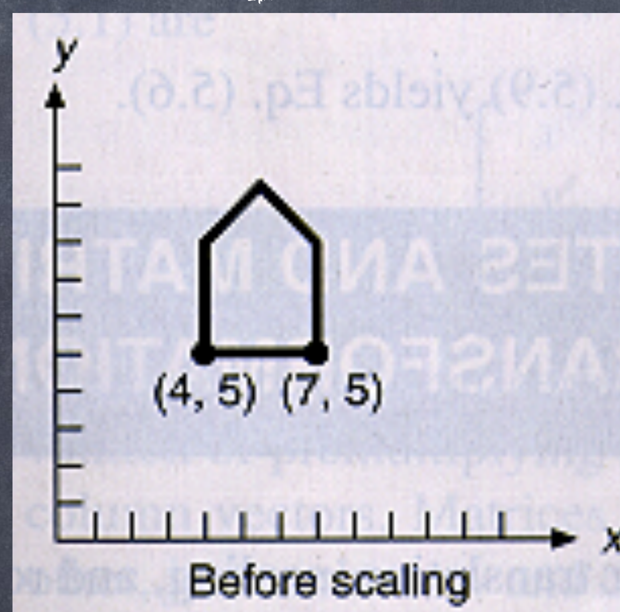
Before translation

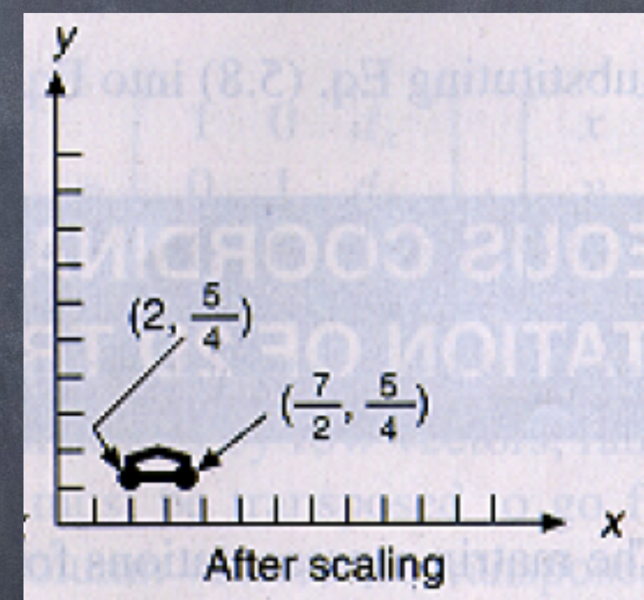


After translation

2D Scaling

- Changes the size of the object by multiplying the coordinates of the points by scaling factors.





$$x' = x\, s_x,\;\; y' = y\, s_y$$

or

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}, \quad s_x, s_y > 0$$

or

$$\underline{P' = S\ P}$$

2D Scaling (cont'd)

If $s_x = s_y$ uniform scaling

If $s_x \neq s_y$ nonuniform scaling

If $s_x, s_y < 1$, size is reduced, object moves closer to origin

If $s_x, s_y > 1$, size is increased, object moves further from origin

If $s_x = s_y = 1$, size does not change

2D Rotation

- Rotates points by an angle θ about origin
- ($\theta > 0$: counterclockwise rotation)

2D Rotation (cont'd)

- We have:

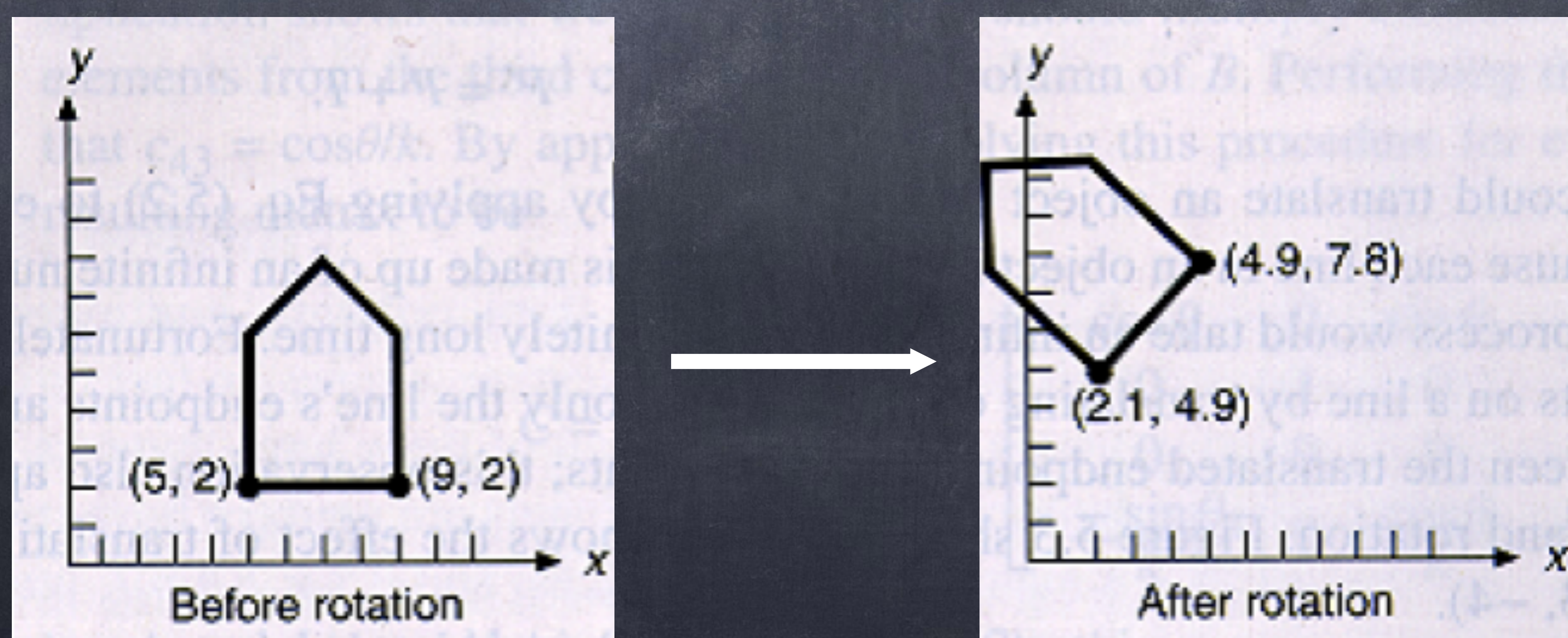
$$x' = x\cos(\theta) - y\sin(\theta), \quad y' = x\sin(\theta) + y\cos(\theta)$$

or

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

or

$$\underline{P' = R\ P}$$



Summary of 2D transformations

Translation: $P' = P + T$

Scale: $P' = S\ P$

Rotation: $P' = R\ P$

Food for though: How can we create an image mosaic?

Affine Transform

- Next lecture