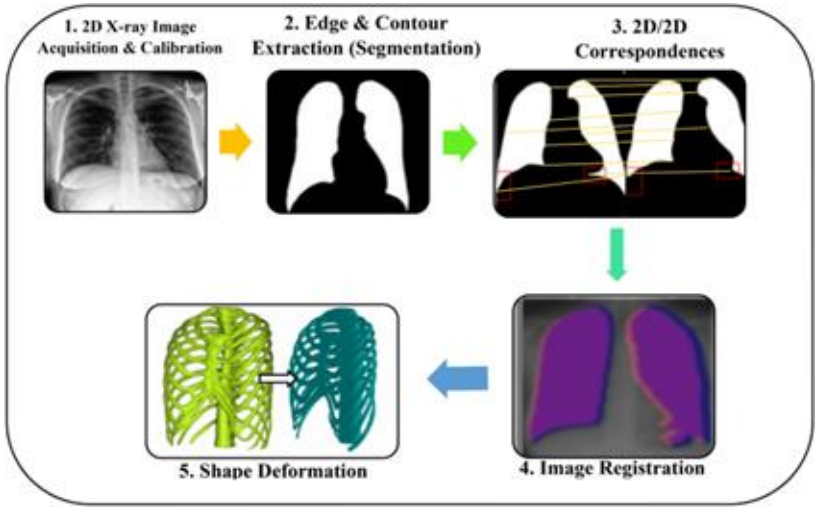
# COMP322: Computer Vision & Image Processing

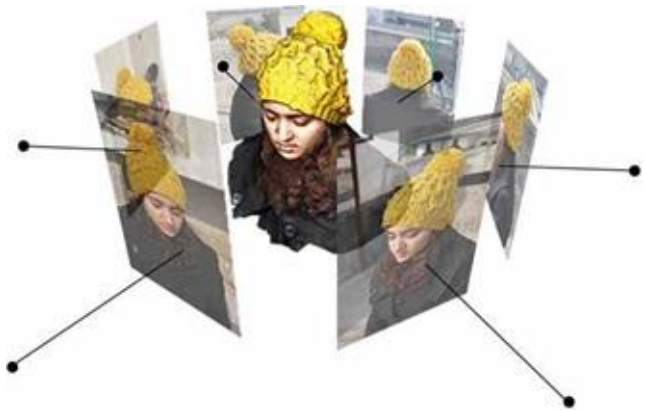
- What is Computer Vision and why?
- What is Image Processing and why?
- Image
- Digital image representation
- Pixels and pixel manipulation
- Images in practice

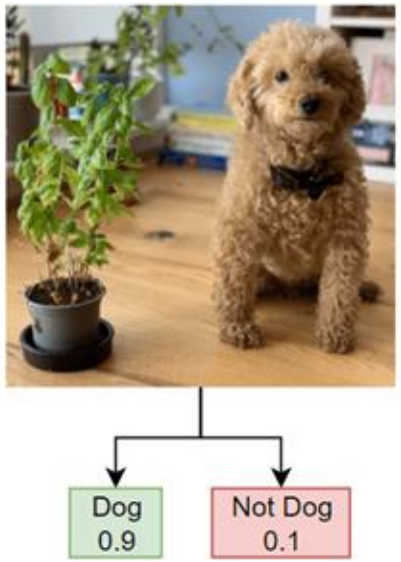
*Modified by Hong Xu, mostly taken from slides by Betsy Sanders, in turn modified from James Tompkin's class.*

*Image Credit: Pixar Animation*

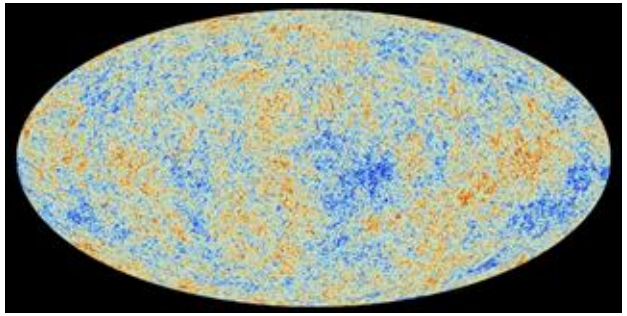



What is IP and why?

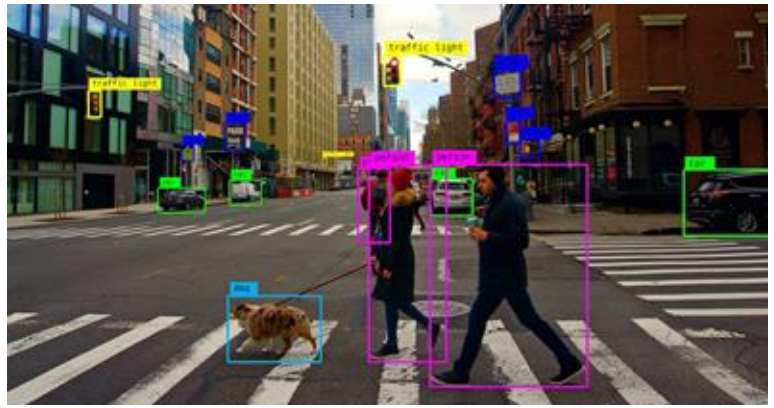






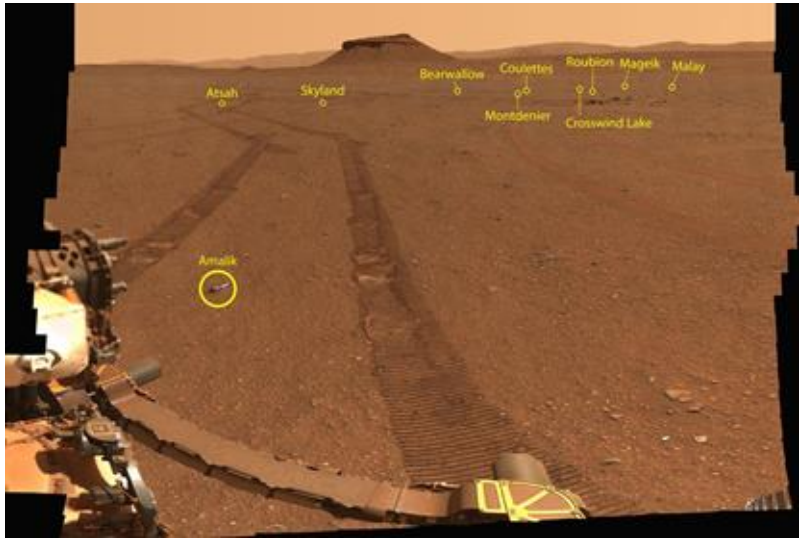


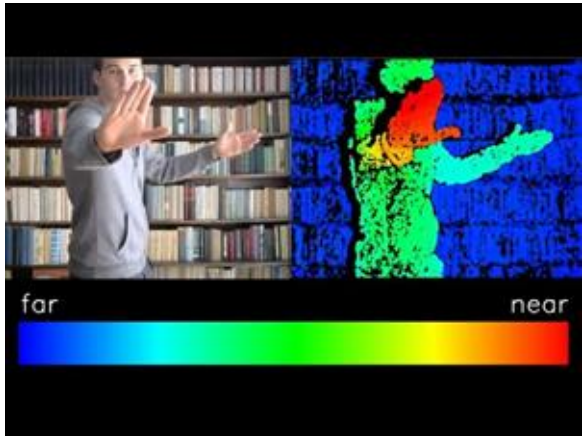
Lots more!





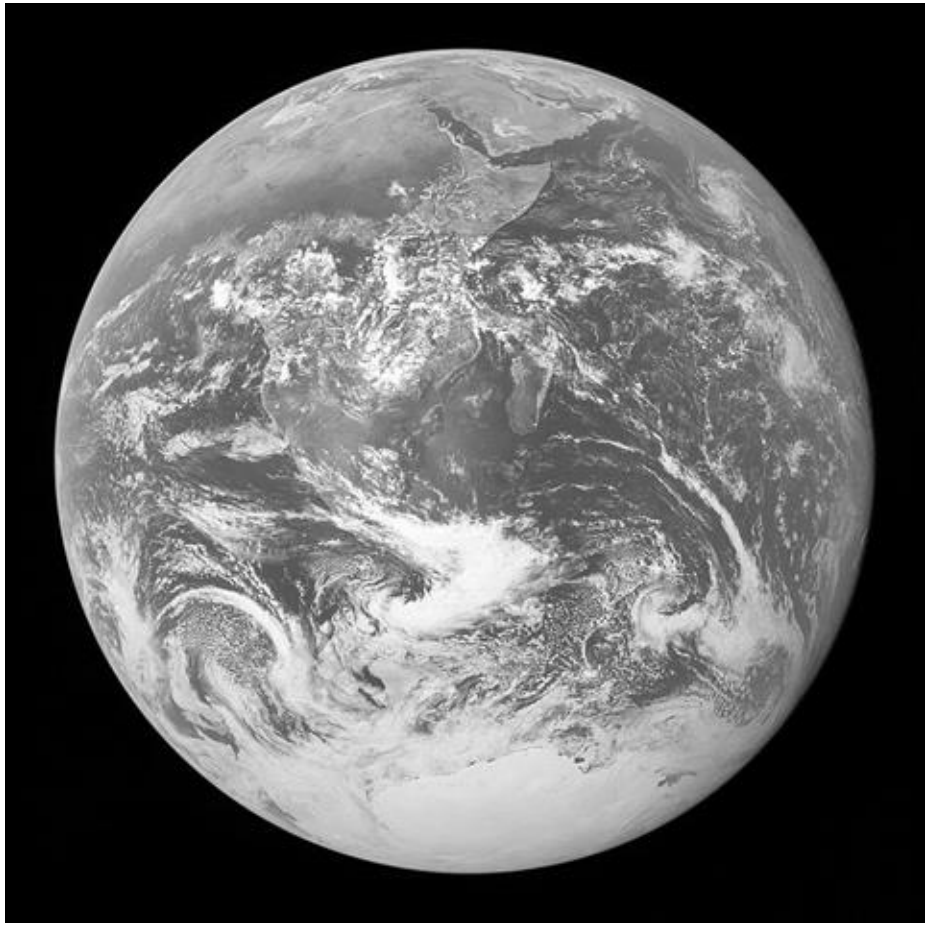
What is CV and why?




Lots more!

Blue Marble. NASA | Apollo 17



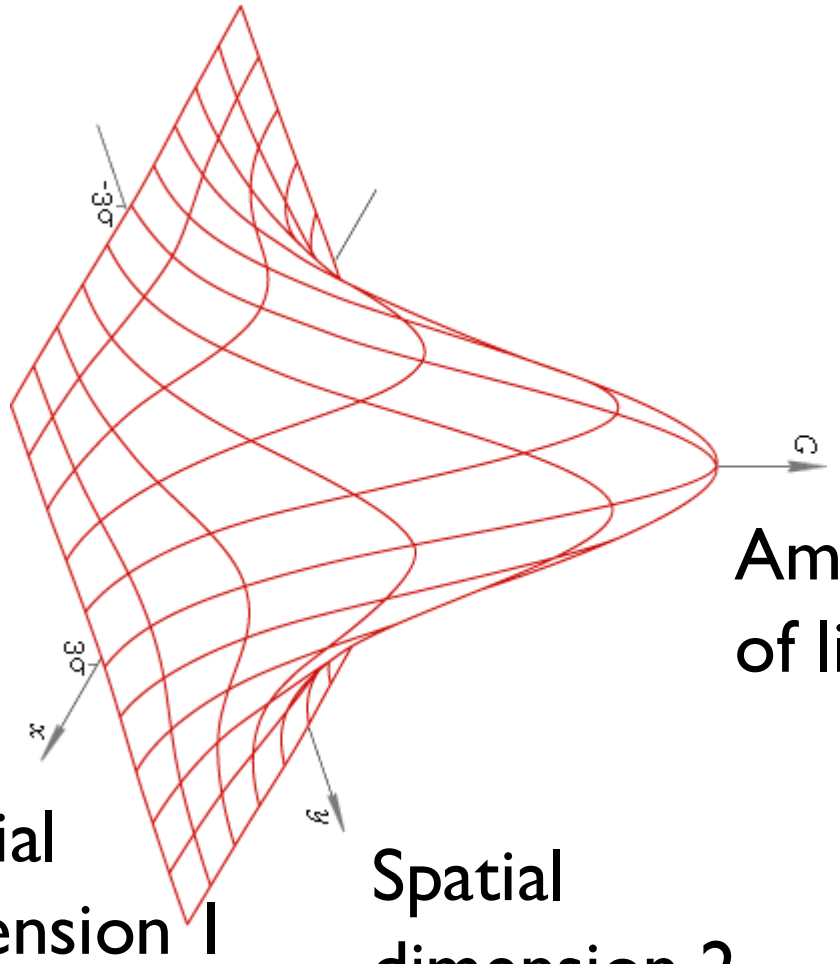
Blue Marble. NASA | Apollo 17

## Light received by camera


[Hasselblad 500 EL - NASA variant – 'Data Camera' (HDC)]

Blue Marble. NASA | Apollo 17

Approximate figurative distribution of light




Amount of light

Light received by camera

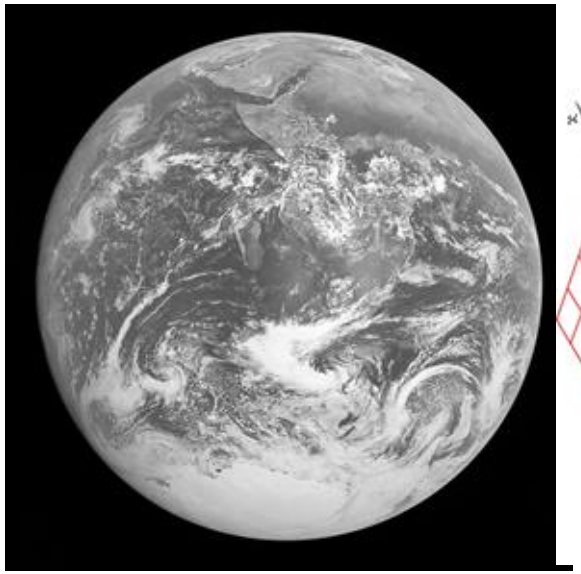

tial dimension 1

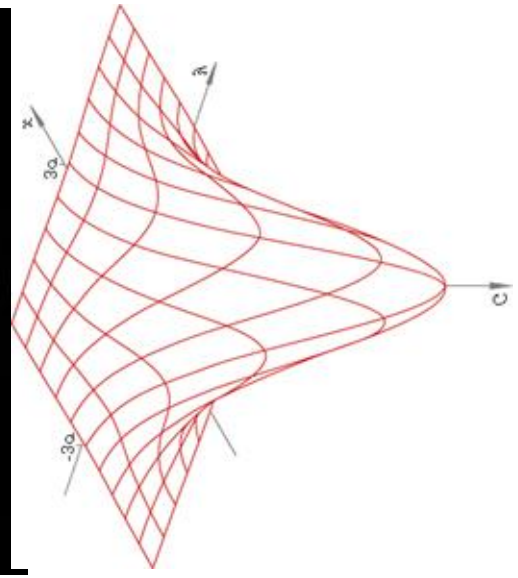
Spatial dimension 2

# Signal

**Definition**

A (multi-dimensional) function that contains information about a phenomenon.

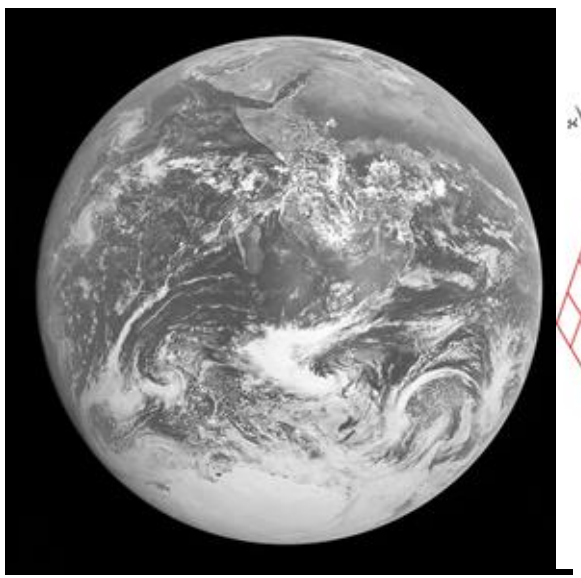


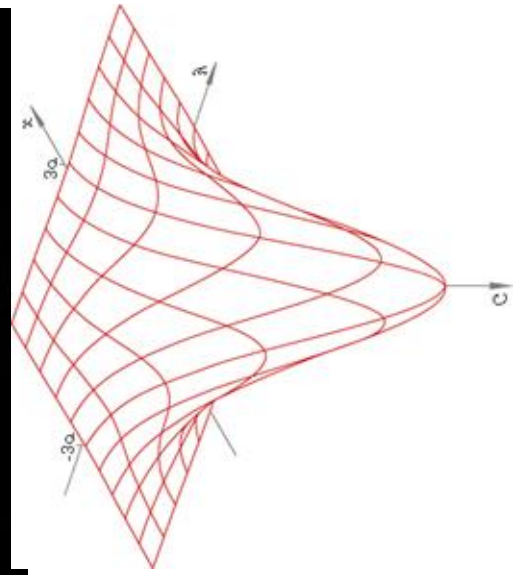


# Signal

**Definition**

A (multi-dimensional) function that contains information about a phenomenon.



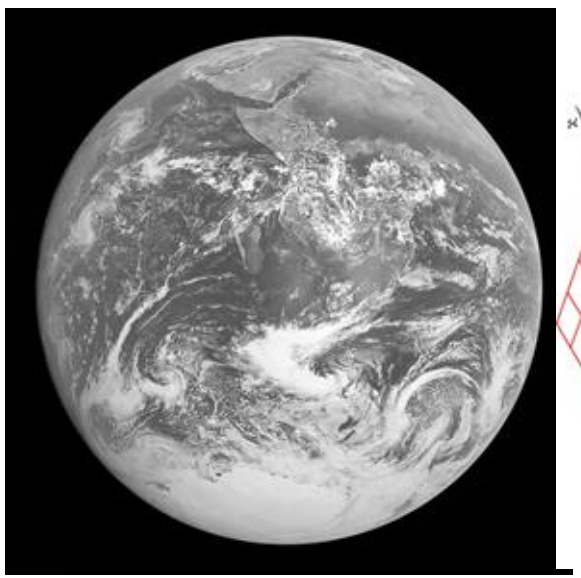


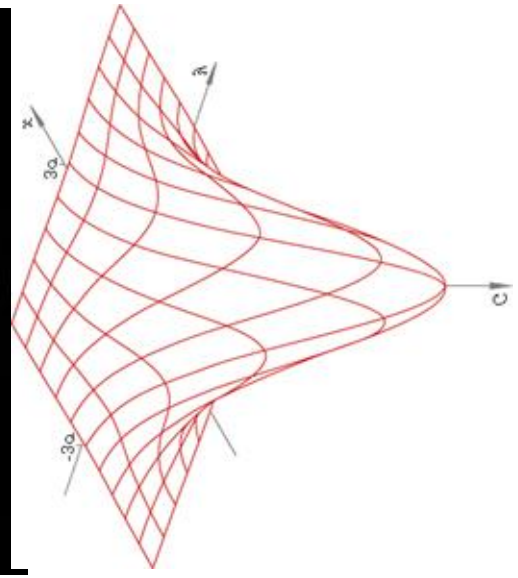

- Signals can be
  - Continuous: light
  - Discrete: measurement of a light
  - Sampling: reduction of continuous signal to a discrete signal

# Signal

**Definition**

A (multi-dimensional) function that contains information about a phenomenon.






Signals can be

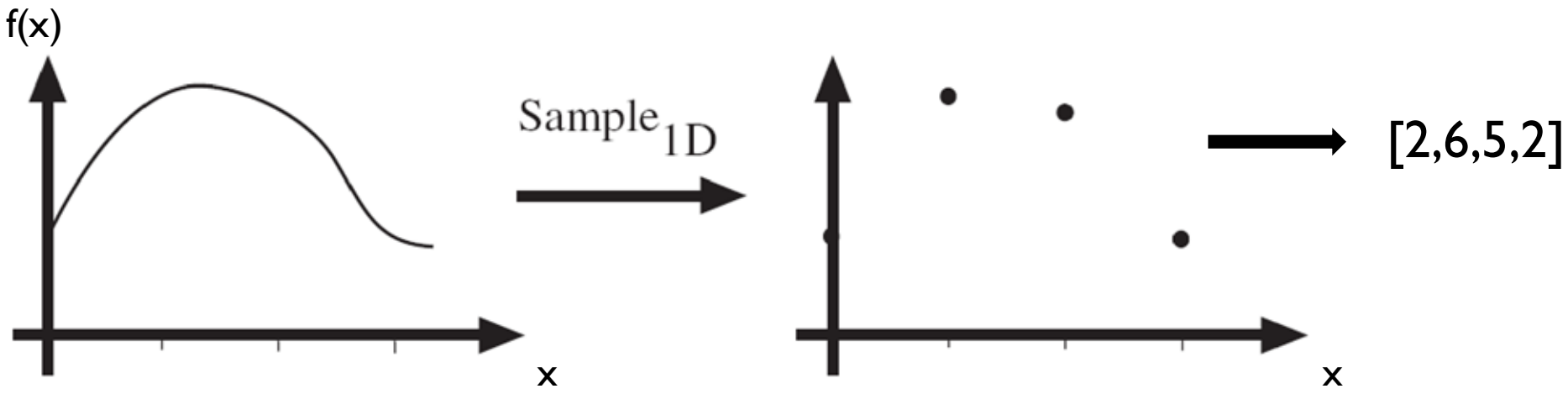
- Continuous: light
- Discrete: measurement of a light
- Sampling: reduction of continuous signal to a discrete signal

Any phenomenon

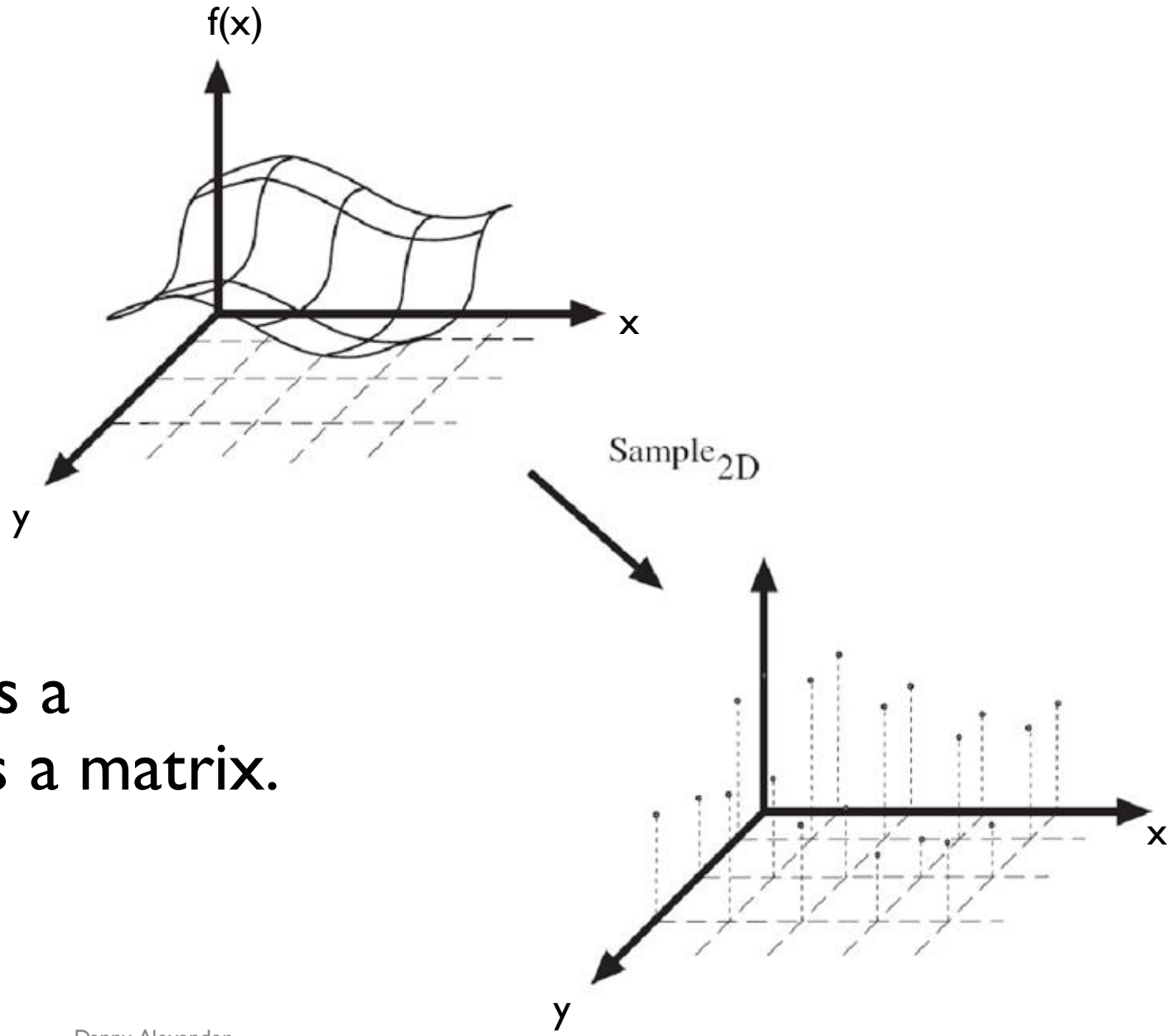
- Light
- Heat
- Gravity

# Sampling

Sampling in 1D takes a function and returns a vector whose elements are values of that function at the sample points.



# Sampling

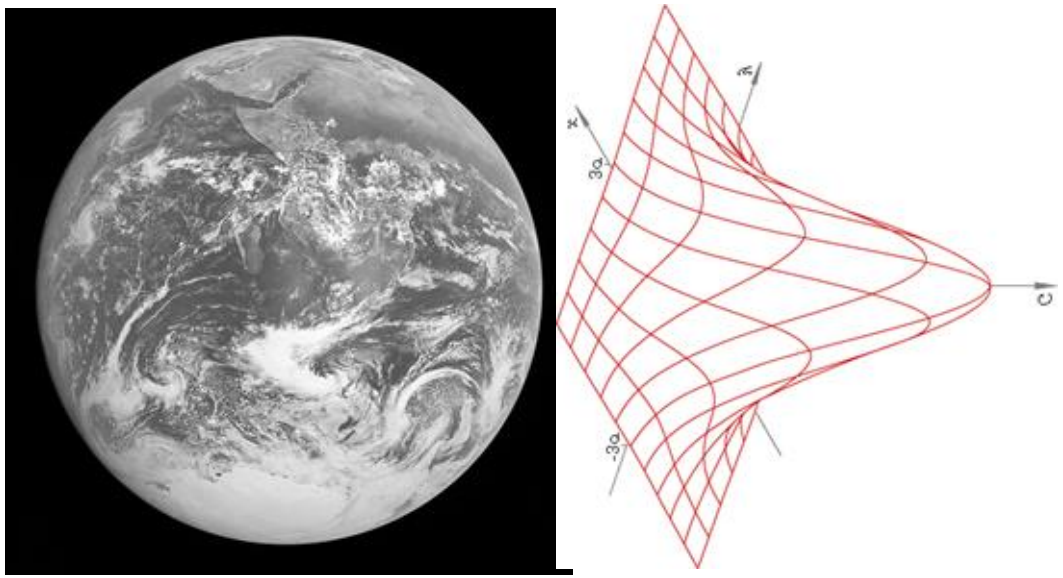



Sampling in 2D takes a function and returns a matrix.

# 2D Image

**Definition**

A sampling of a function that contains information about a 2D* signal.

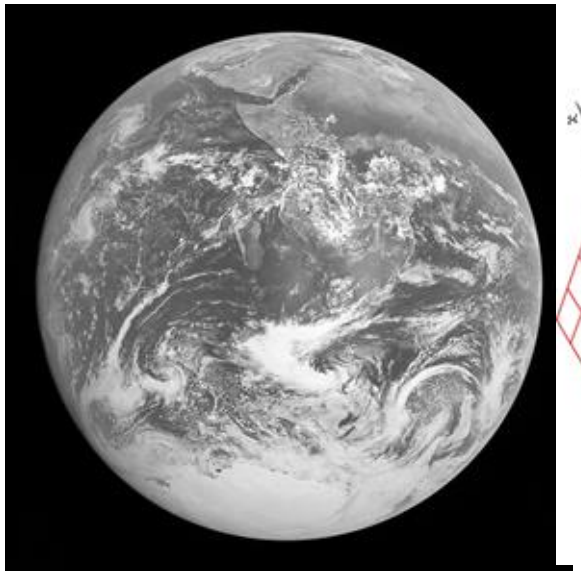


* or a 2D projection of a multi-dimensional signal

# 2D Image

**Definition**

A sampling of a function that contains information about a 2D* signal.



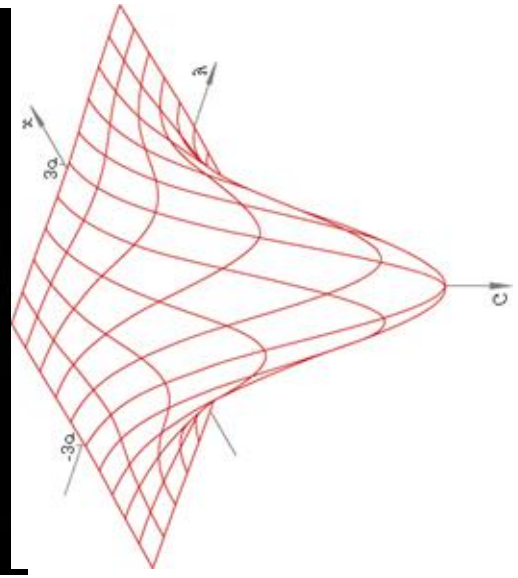



Image stores intensity or ‘brightness’

* or a 2D projection of a multi-dimensional signal

# 2D Image

**Definition**

A sampling of a function that contains information about a 2D* signal.

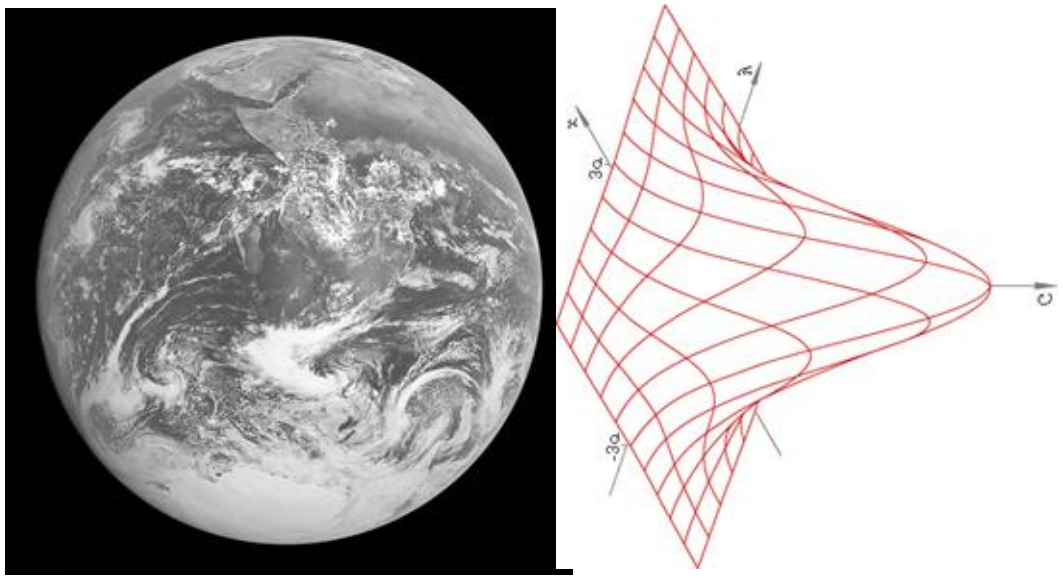


Image stores intensity or ‘brightness’

2D signals are special for us

- Brightness along x and y dimensions

* or a 2D projection of a multi-dimensional signal

# 2D Image

**Definition**

A sampling of a function that contains information about a 2D* signal.

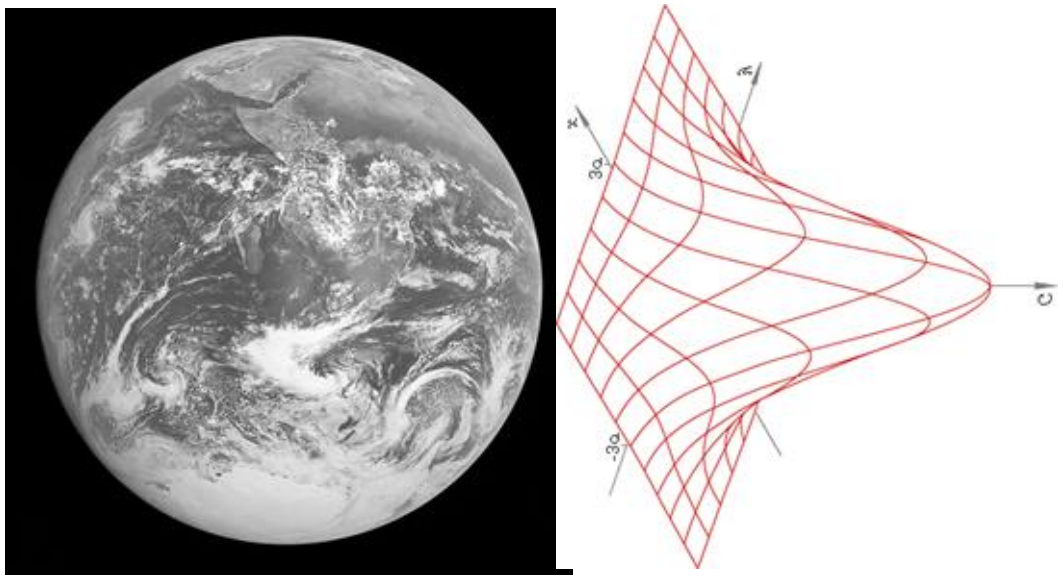


Image stores intensity or ‘brightness’

2D signals are special for us

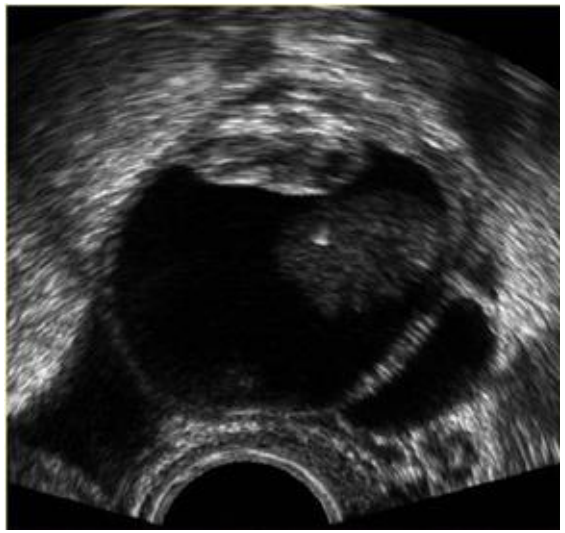
- Brightness along x and y dimensions

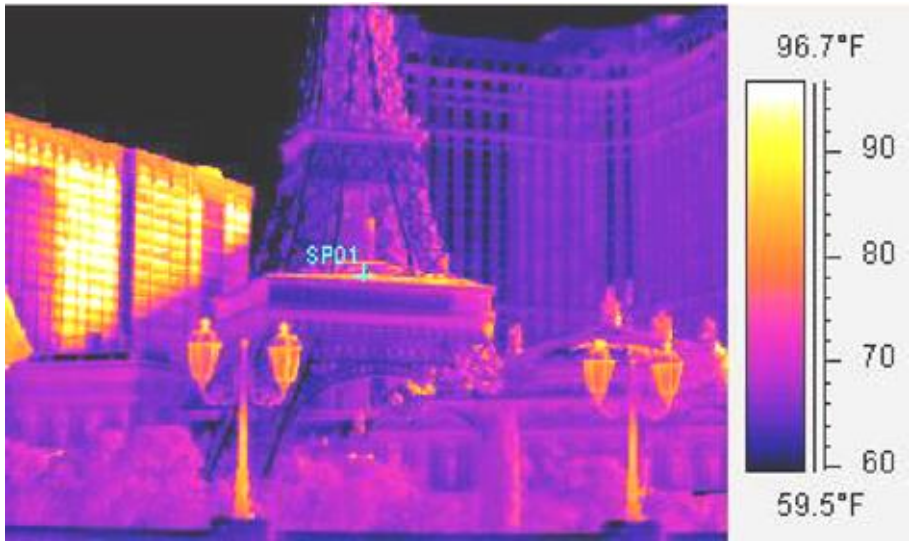
Video: xy-coordinates + time

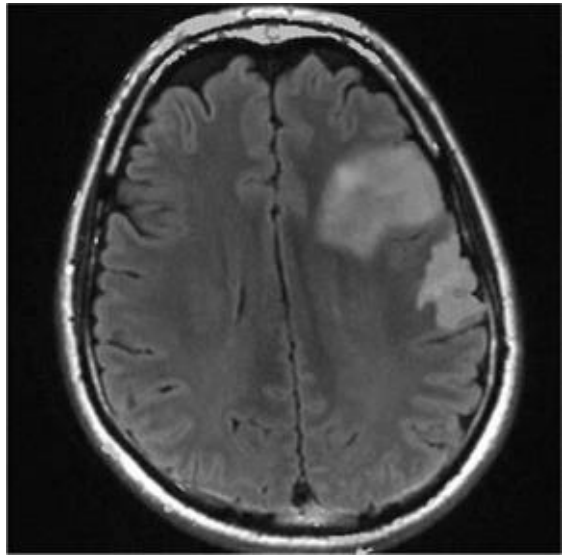
- Time-varying 2D signal

* or a 2D projection of a multi-dimensional signal

# Example 2D Images

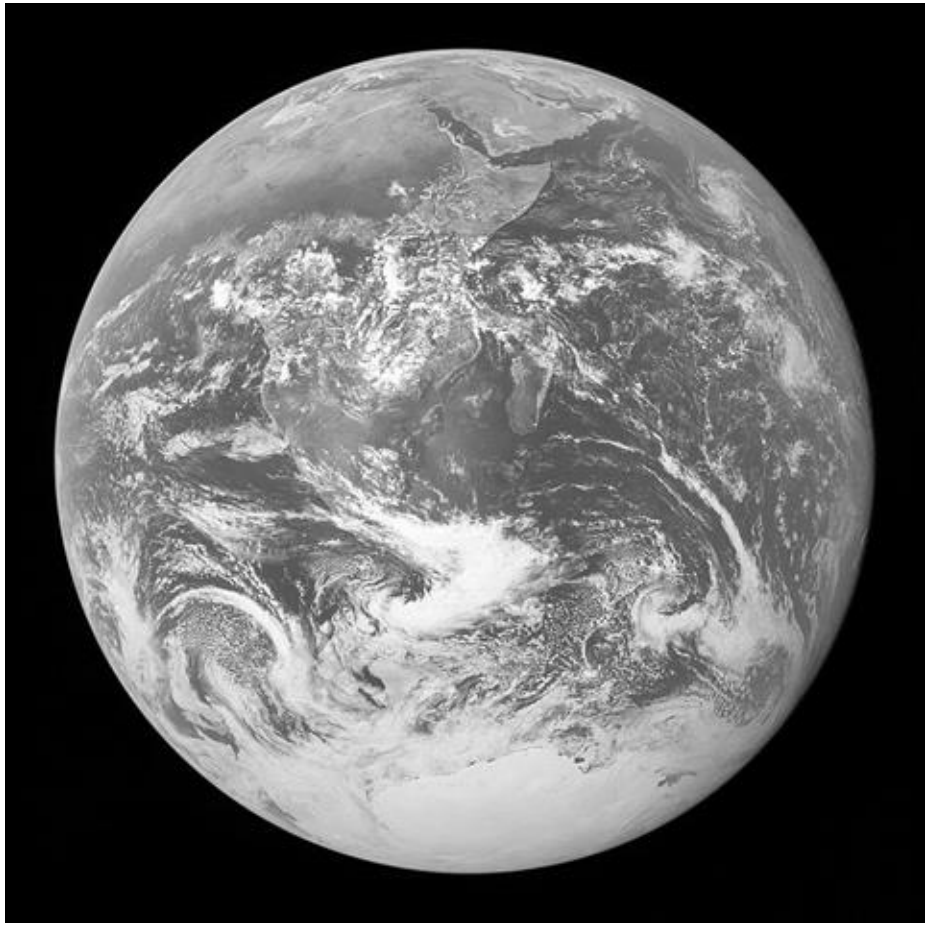


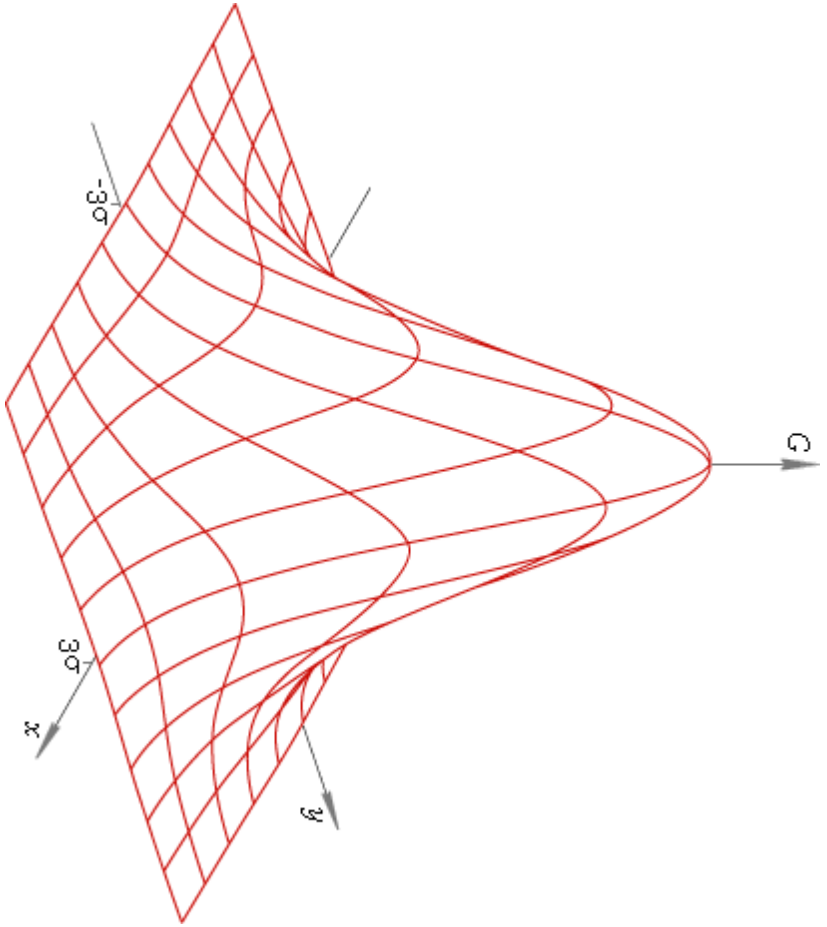


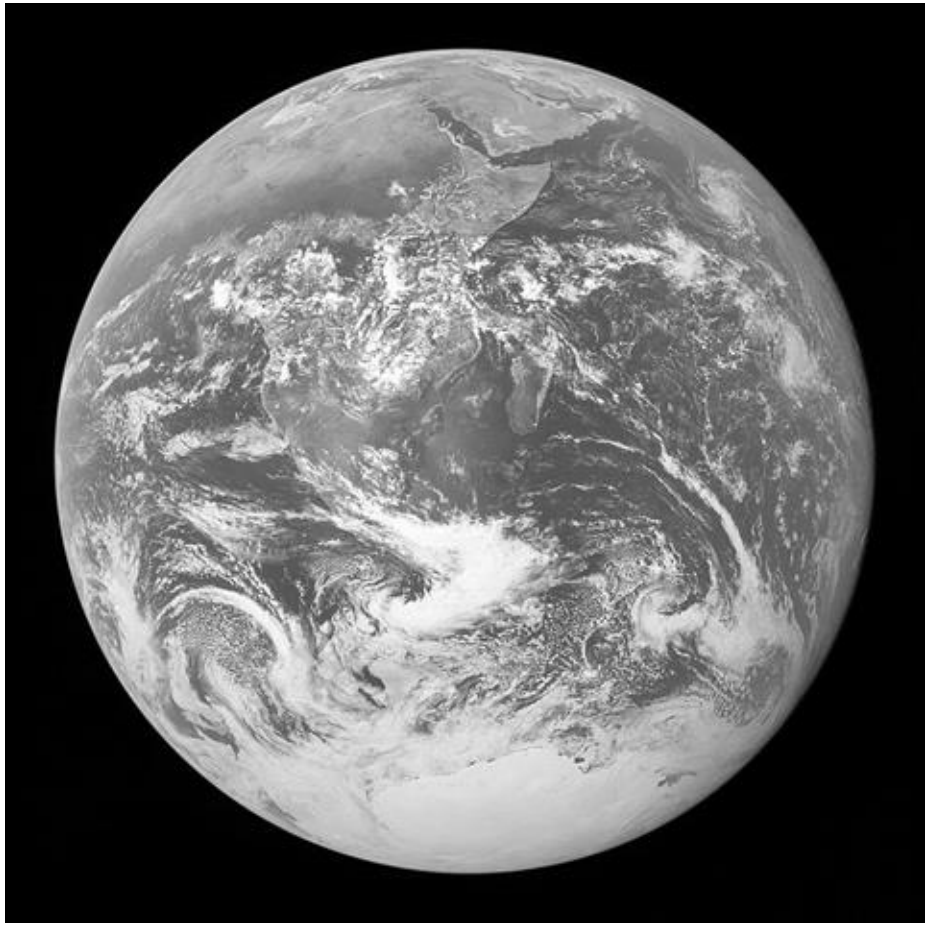
http://radiologykey.com/wp-content/uploads/2016/01/9781604063325_c014_f004.jpg

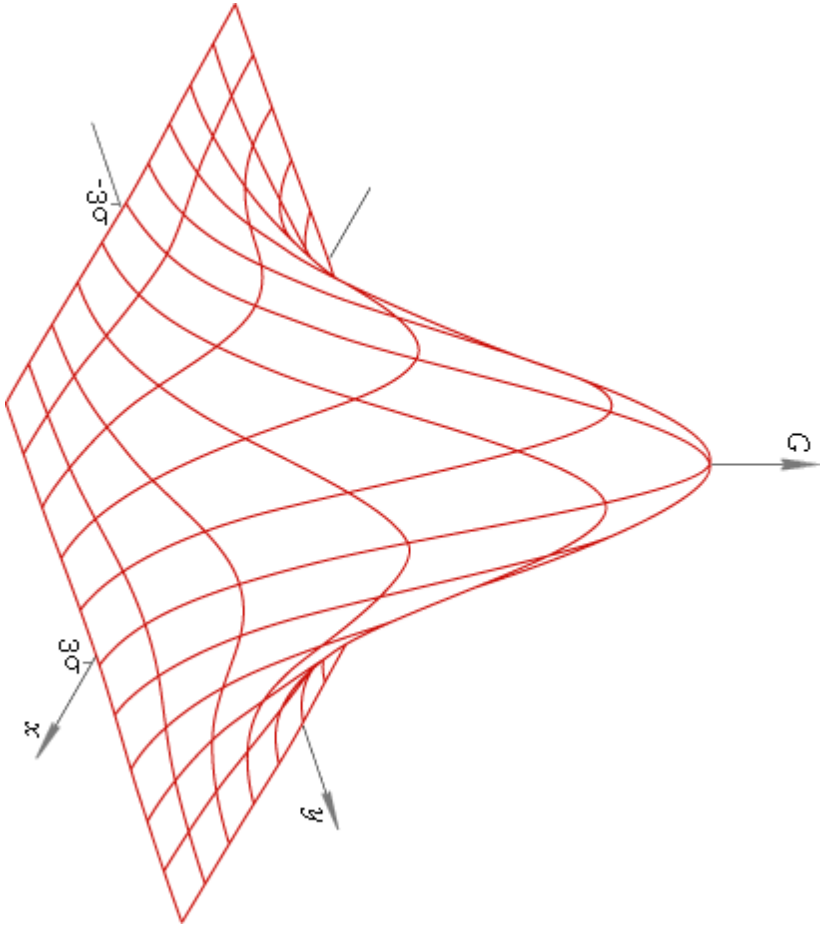
# Sampling in Practice





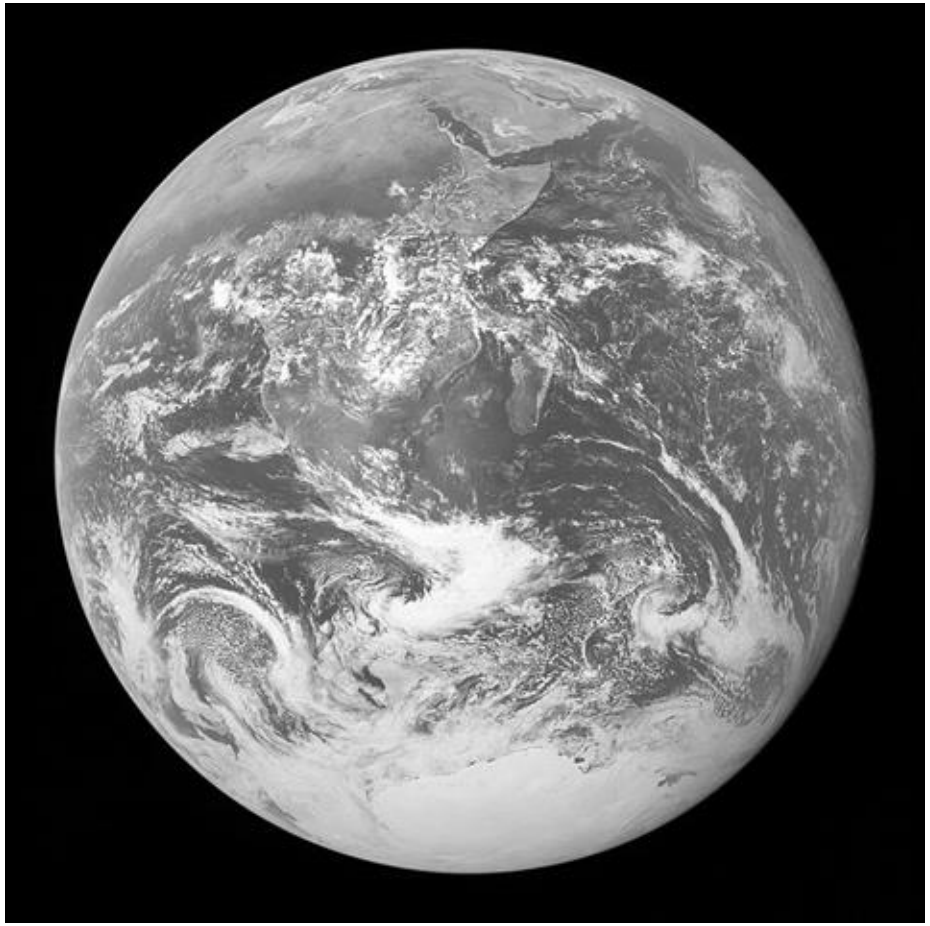

# Sampling in Practice

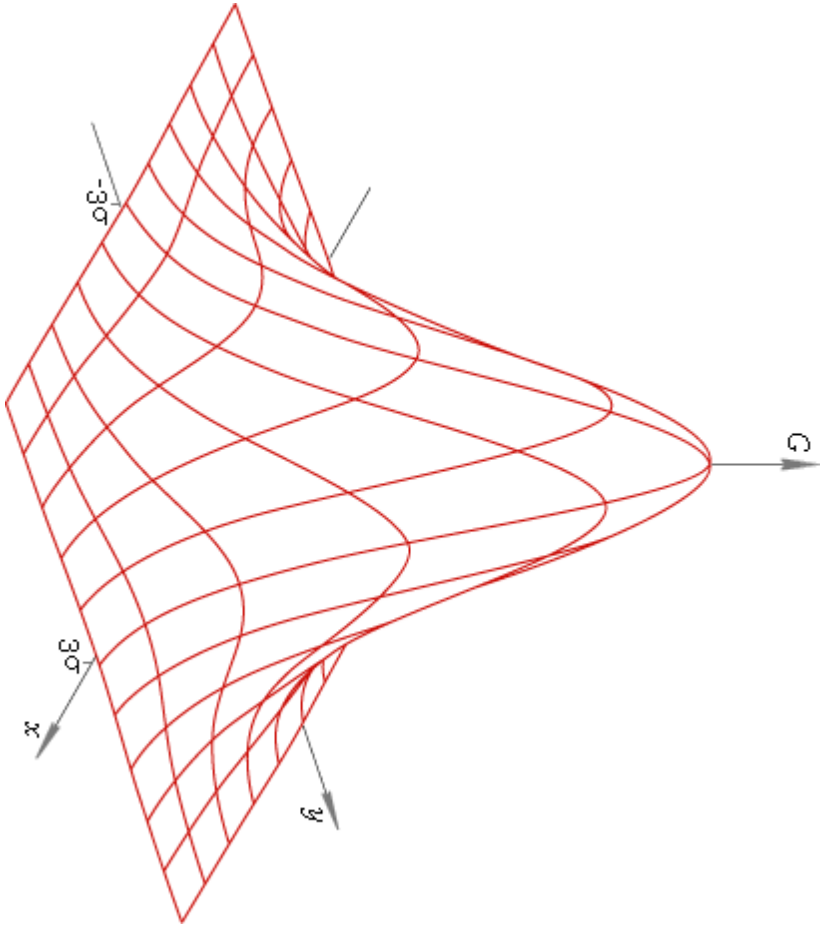




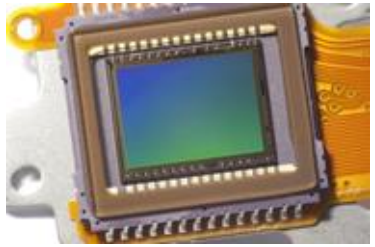


# Sampling in Practice: Digital Image





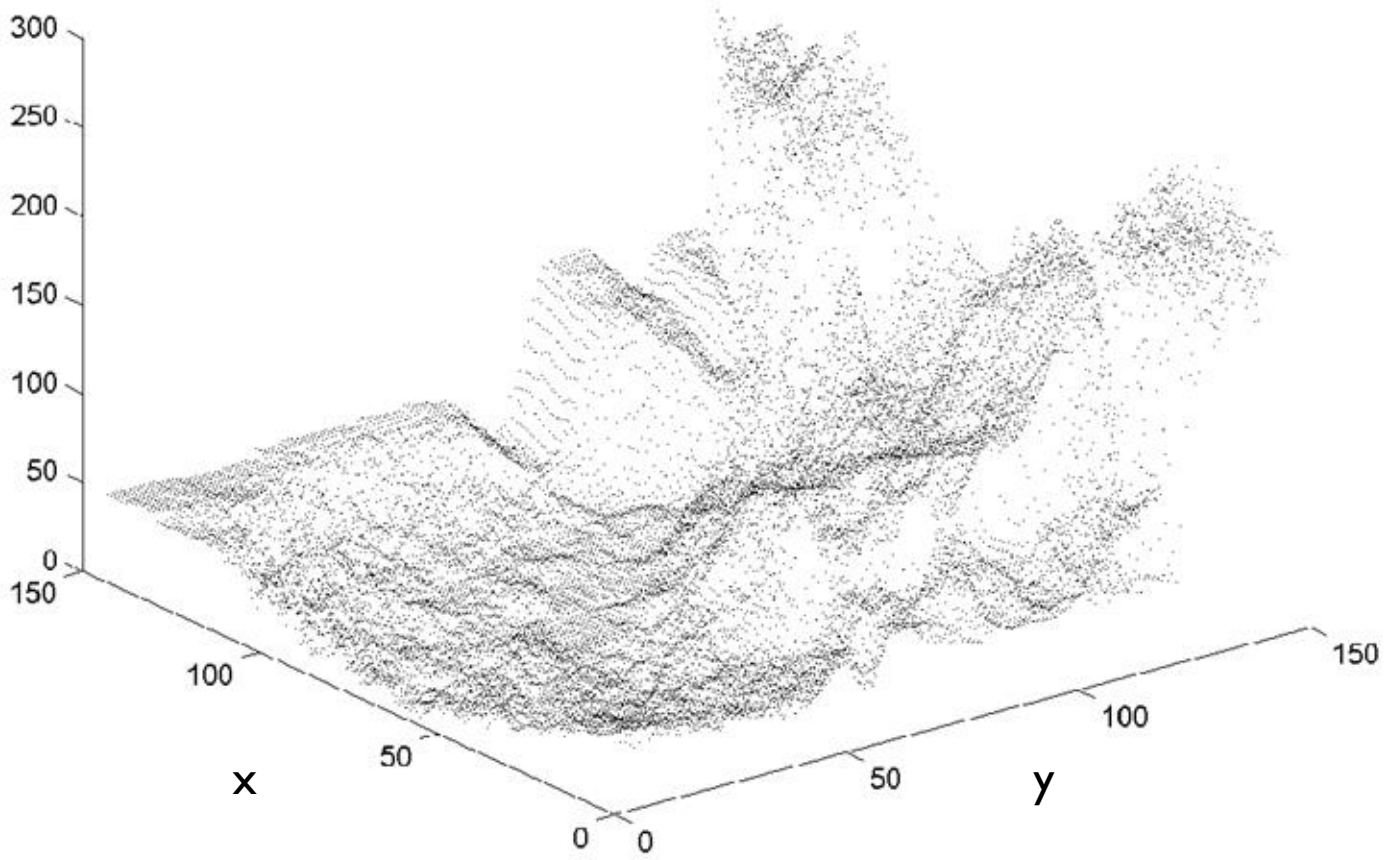




CMOS / CCD

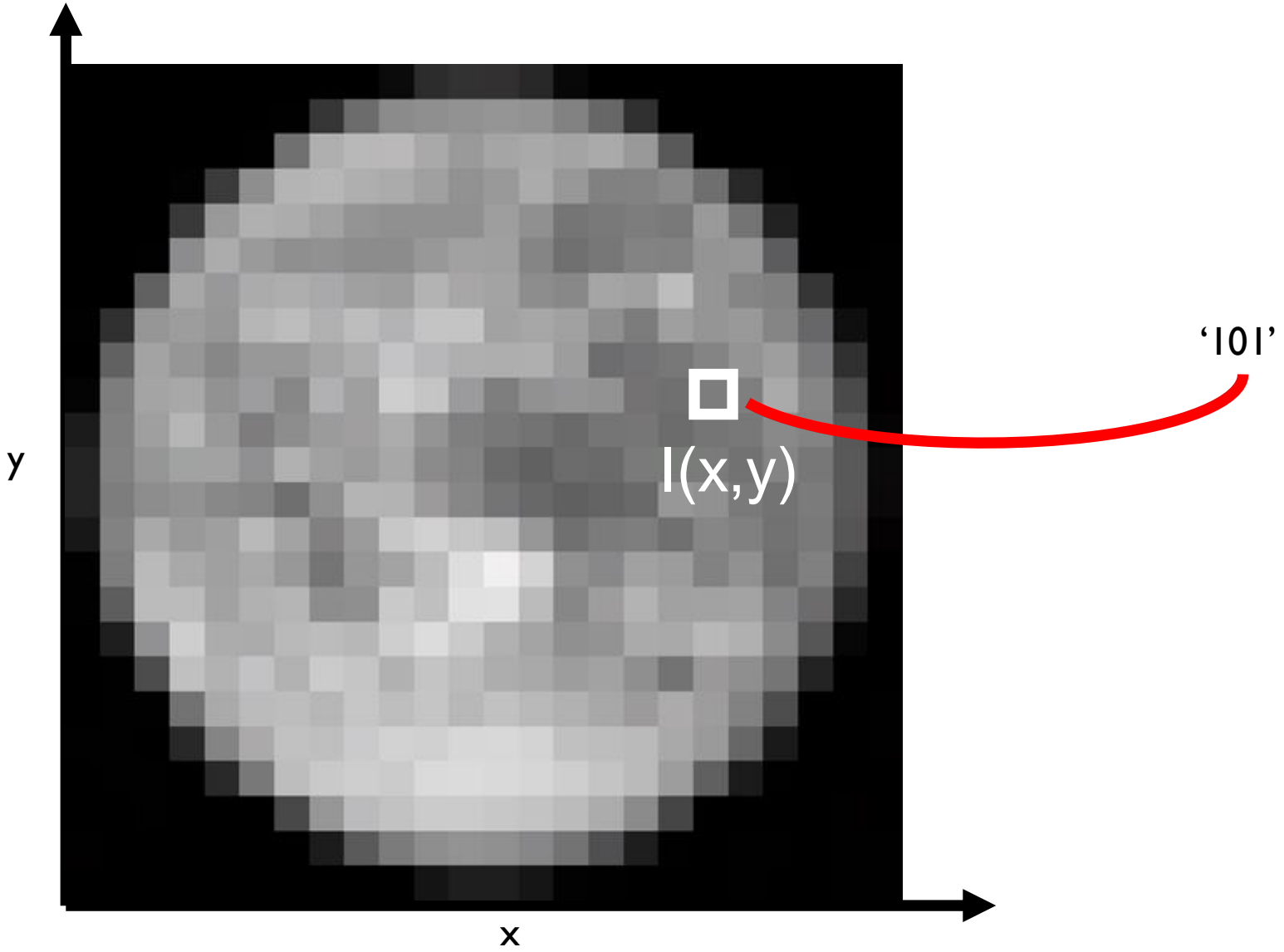
# Digital Image as 'Height Map'

Brightness or intensity



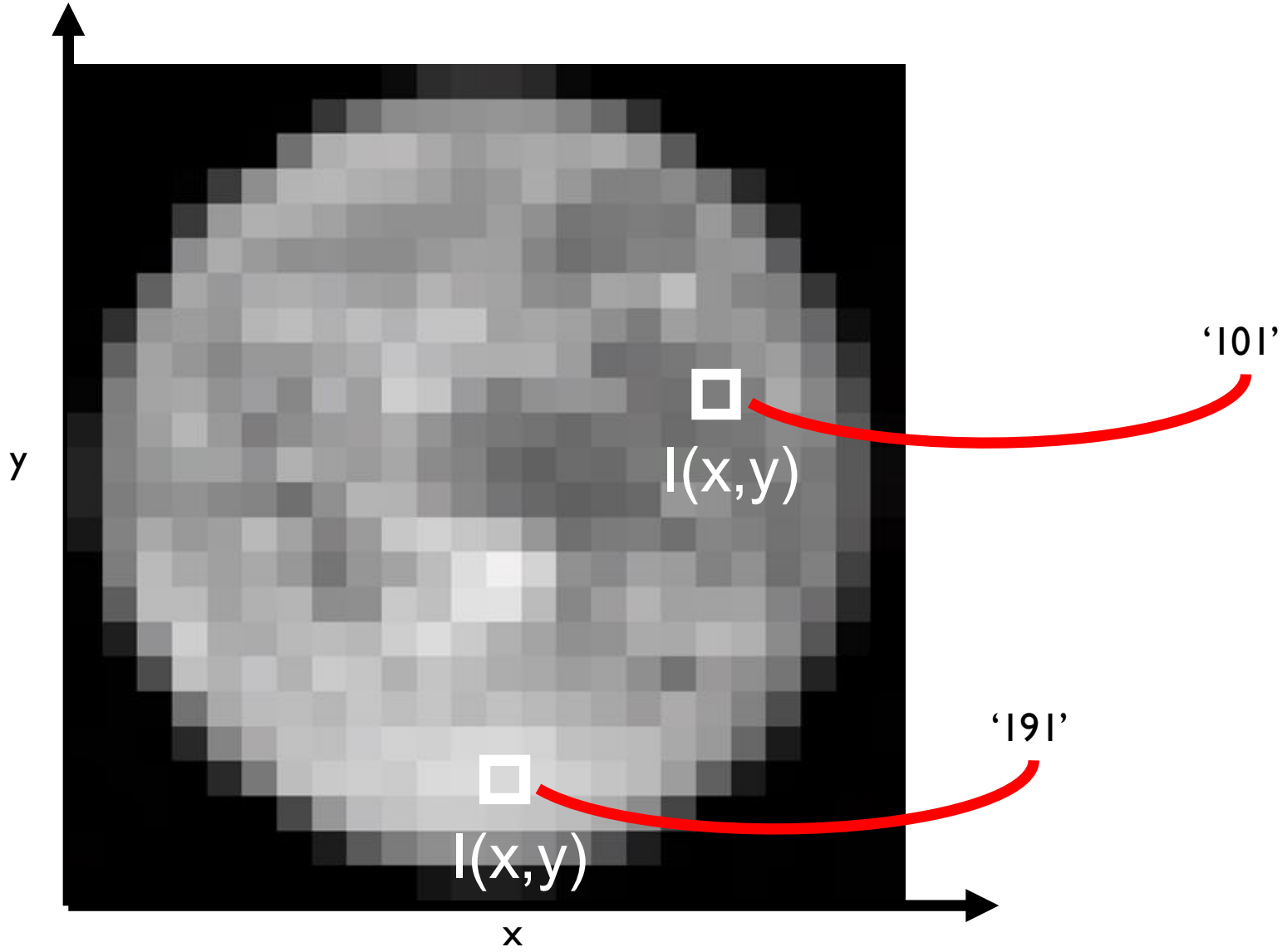
# Elements of a Digital Image

Pixel: picture element



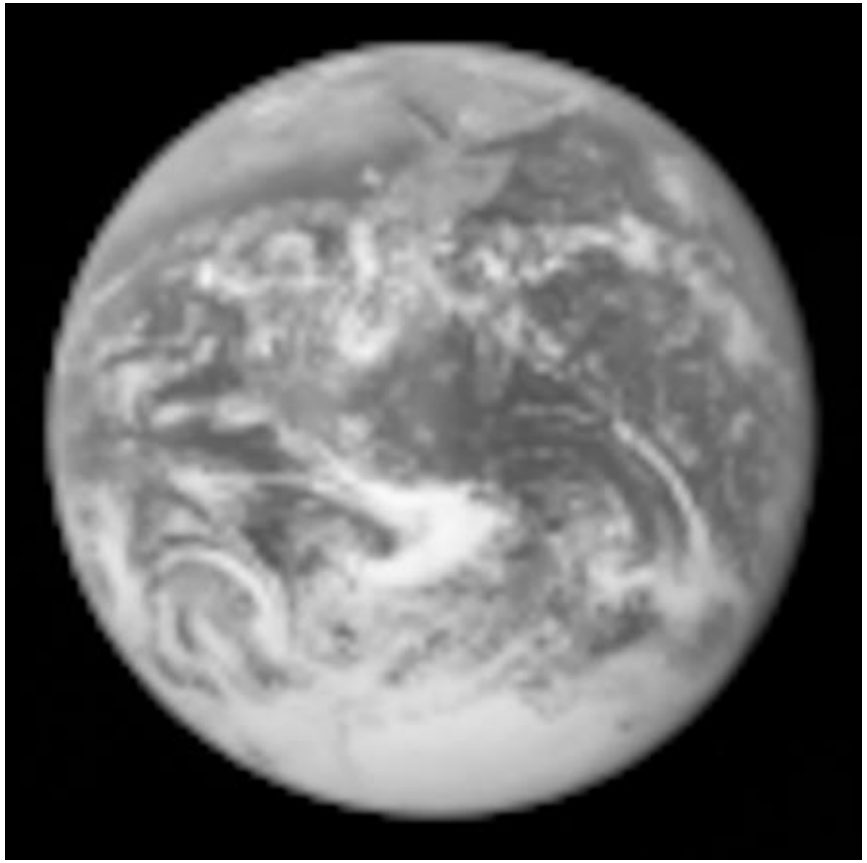
# Elements of a Digital Image

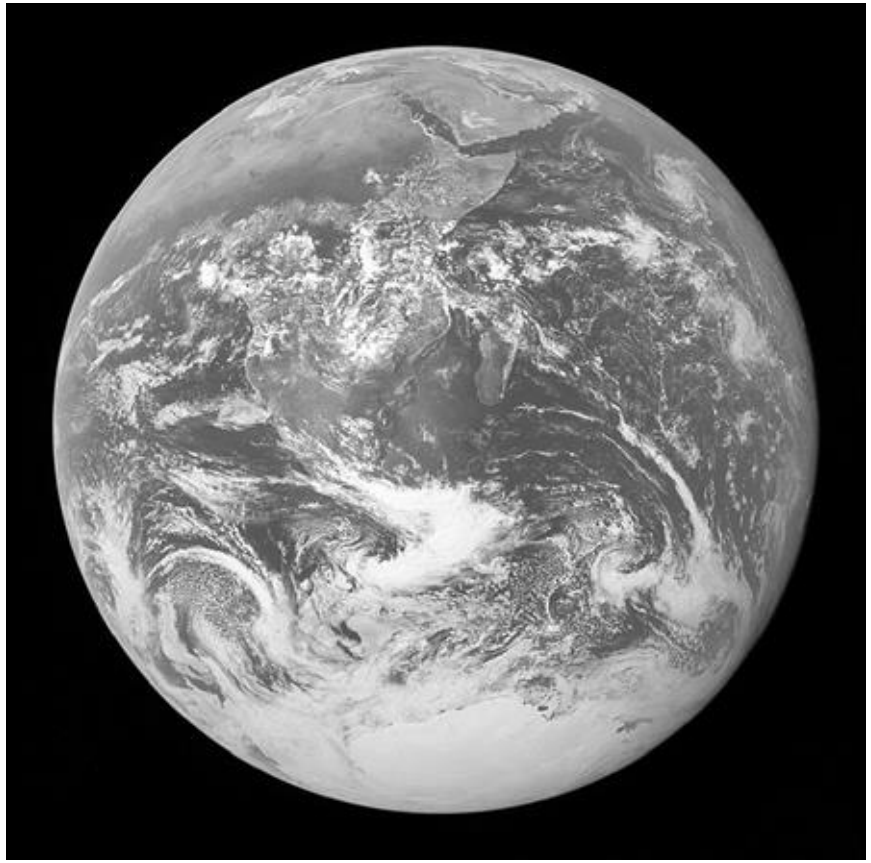
Pixel: picture element



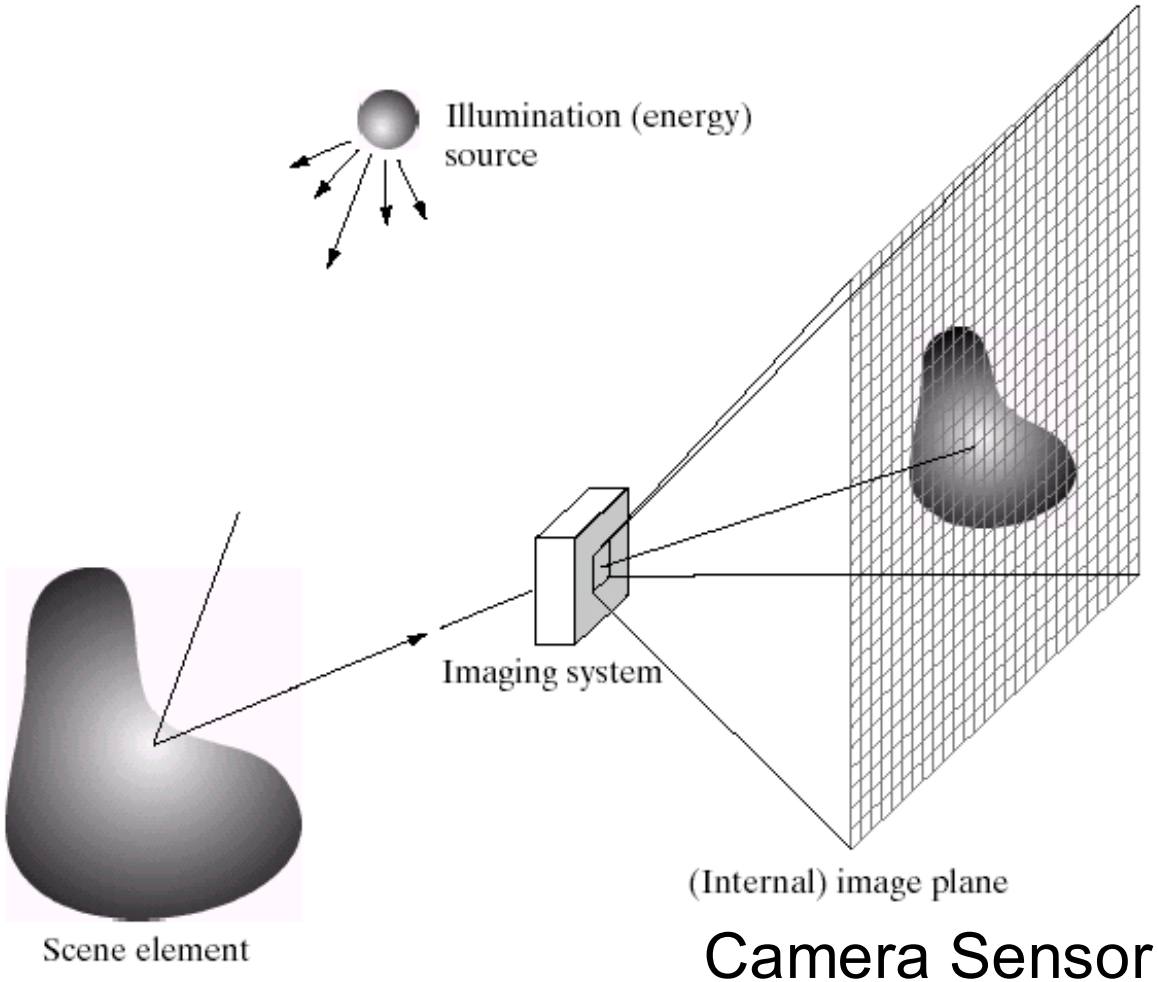
# Resolution: geometric vs. spatial

Both images are 1000x1000 pixels

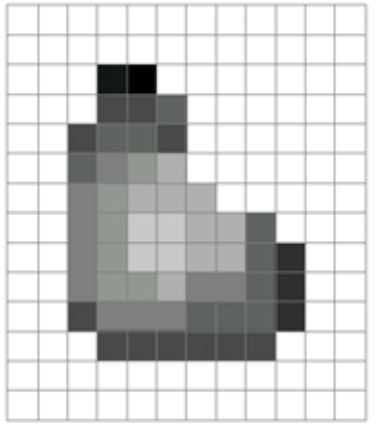




# Light Integration Over the *Frustum*

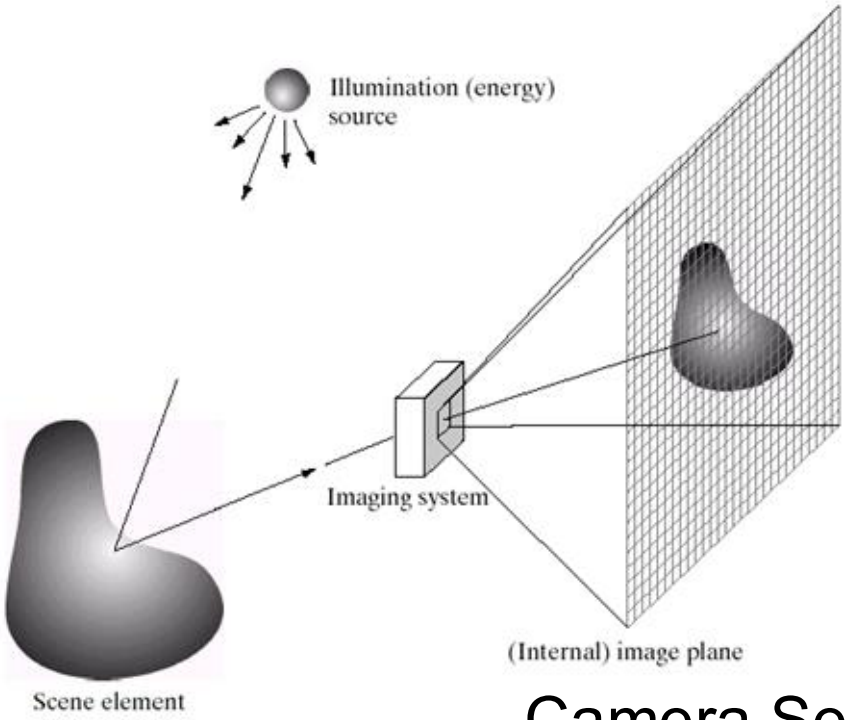



Camera Sensor

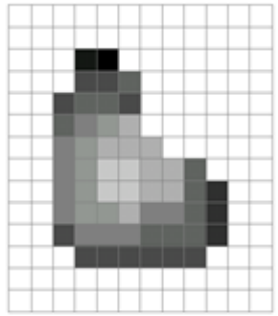


Output Image

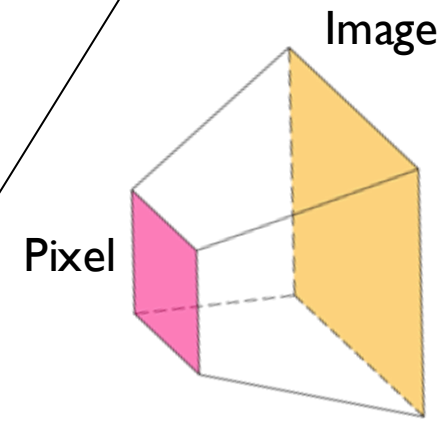
# Light Integration Over the *Frustum*




Camera Sensor



Output Image



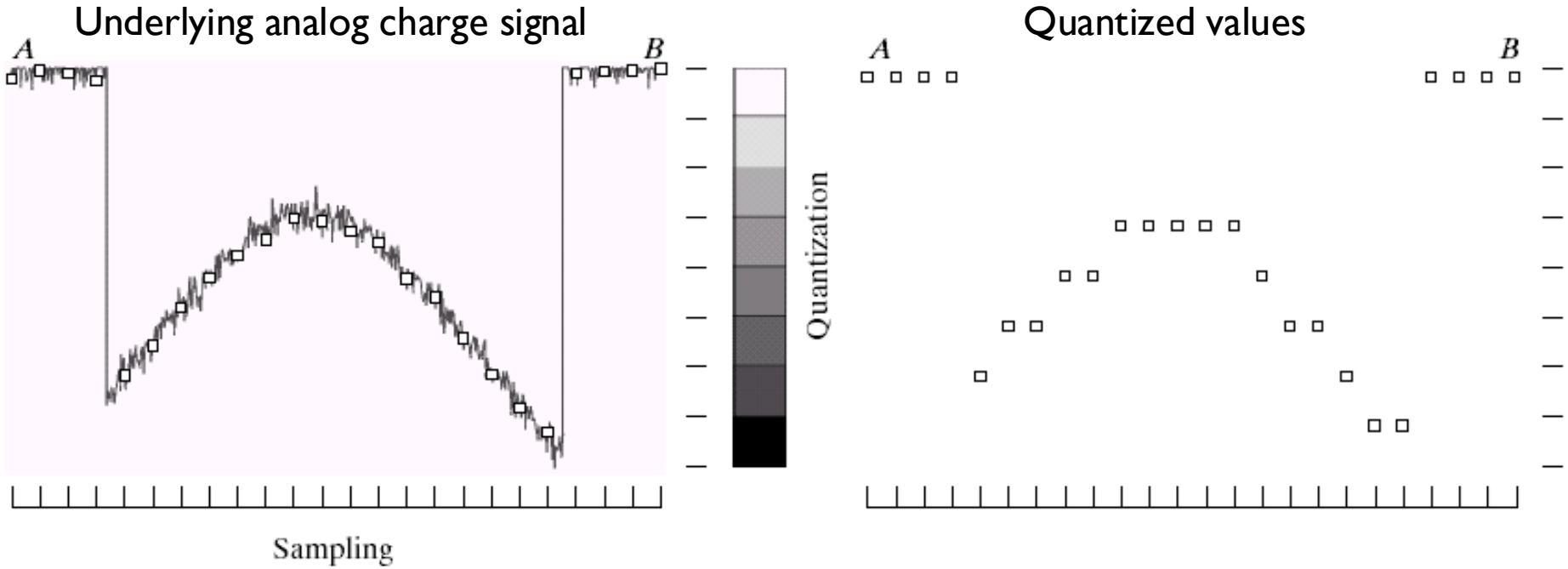

A frustum approximates the ray space that a pixel samples

# Quantization

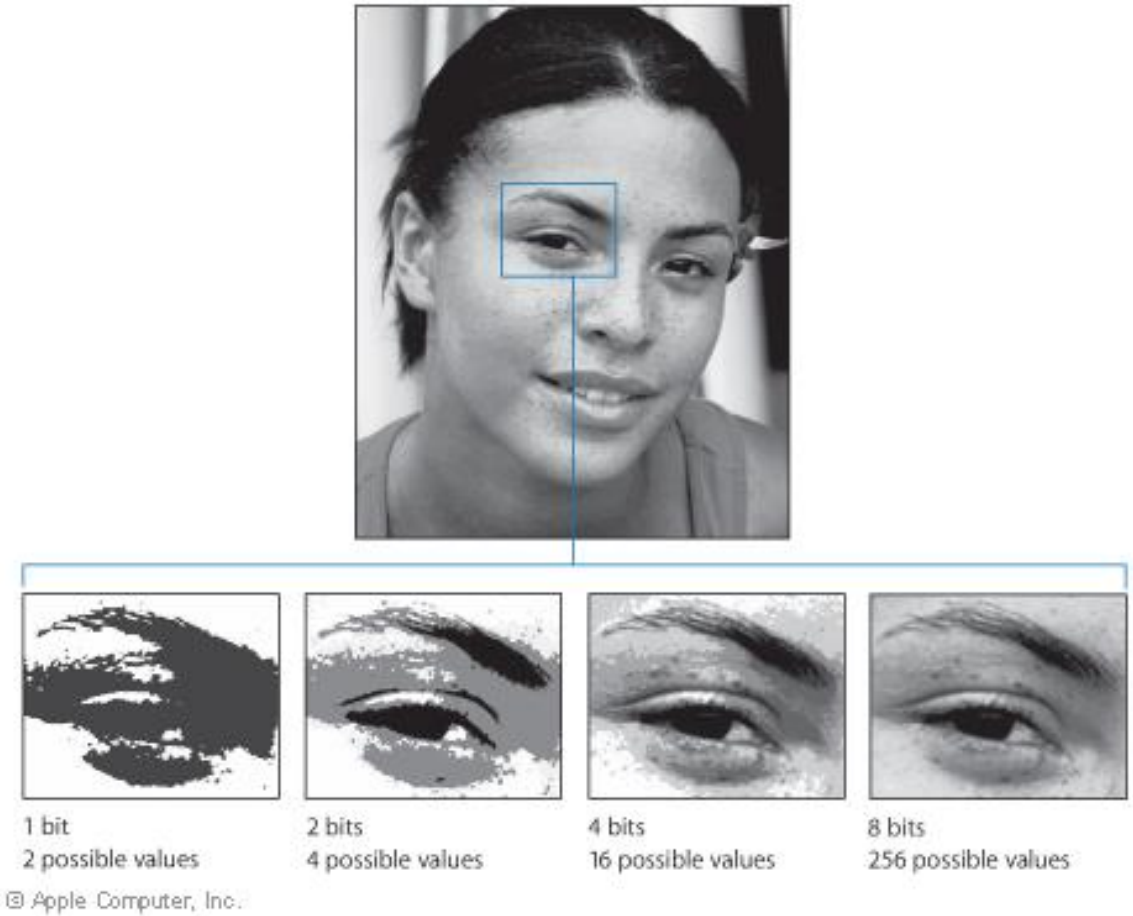
Signals from the digital world need to to translated to digital. The image must be digitized spatially AND in amplitude.

Two STEPS to Digitizing:

1. Sampling
2. Quantization: digitizing the amplitude value



# Quantization Effects – Radiometric Resolution




We often call this *bit depth.*

For photography, this is also related to *dynamic range.*

# Dimensionality of an Image

An image of size 1000x1000 with 8-bit quantization per pixel

- = 256 values ^ (1000 x 1000)

# Dimensionality of an Image

An image of size 1000x1000 with 8-bit quantization per pixel

- = 256 values ^ (1000 x 1000)

All scenes ever depicted

# Dimensionality of an Image

An image of size 1000x1000 with 8-bit quantization per pixel

- = 256 values ^ (1000 x 1000)

All scenes ever depicted

- In size 1000x1000

# Dimensionality of an Image

An image of size 1000x1000 with 8-bit quantization per pixel

- = 256 values ^ (1000 x 1000)


All scenes ever depicted

- In size 1000x1000

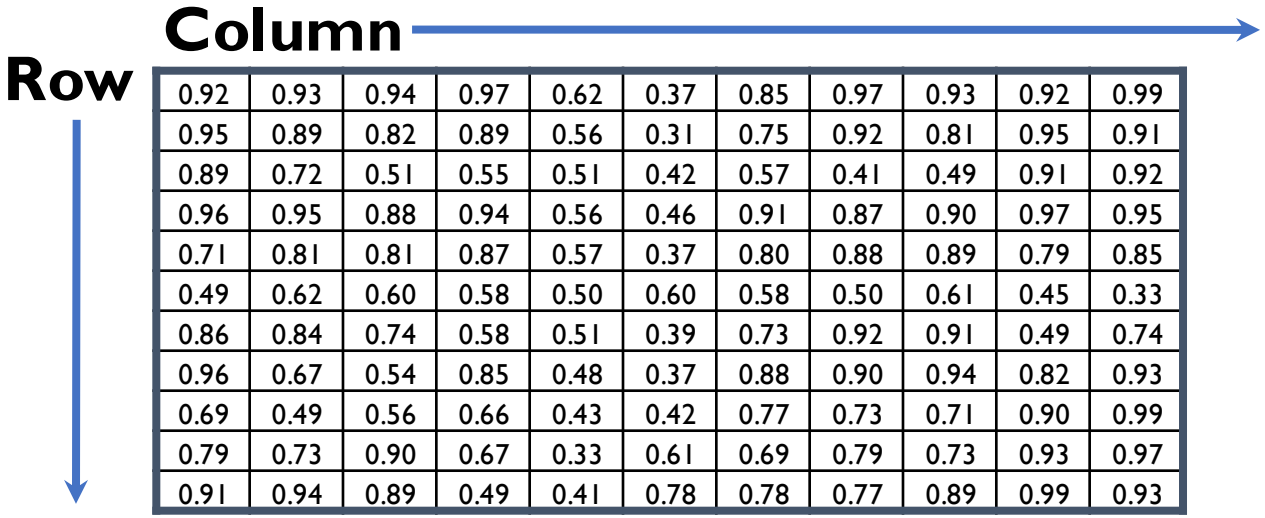
Computer vision as making sense of high-dimensional data.

- ‘Natural’ images are a subspace
- CV as deriving low-dimensional, explainable models.

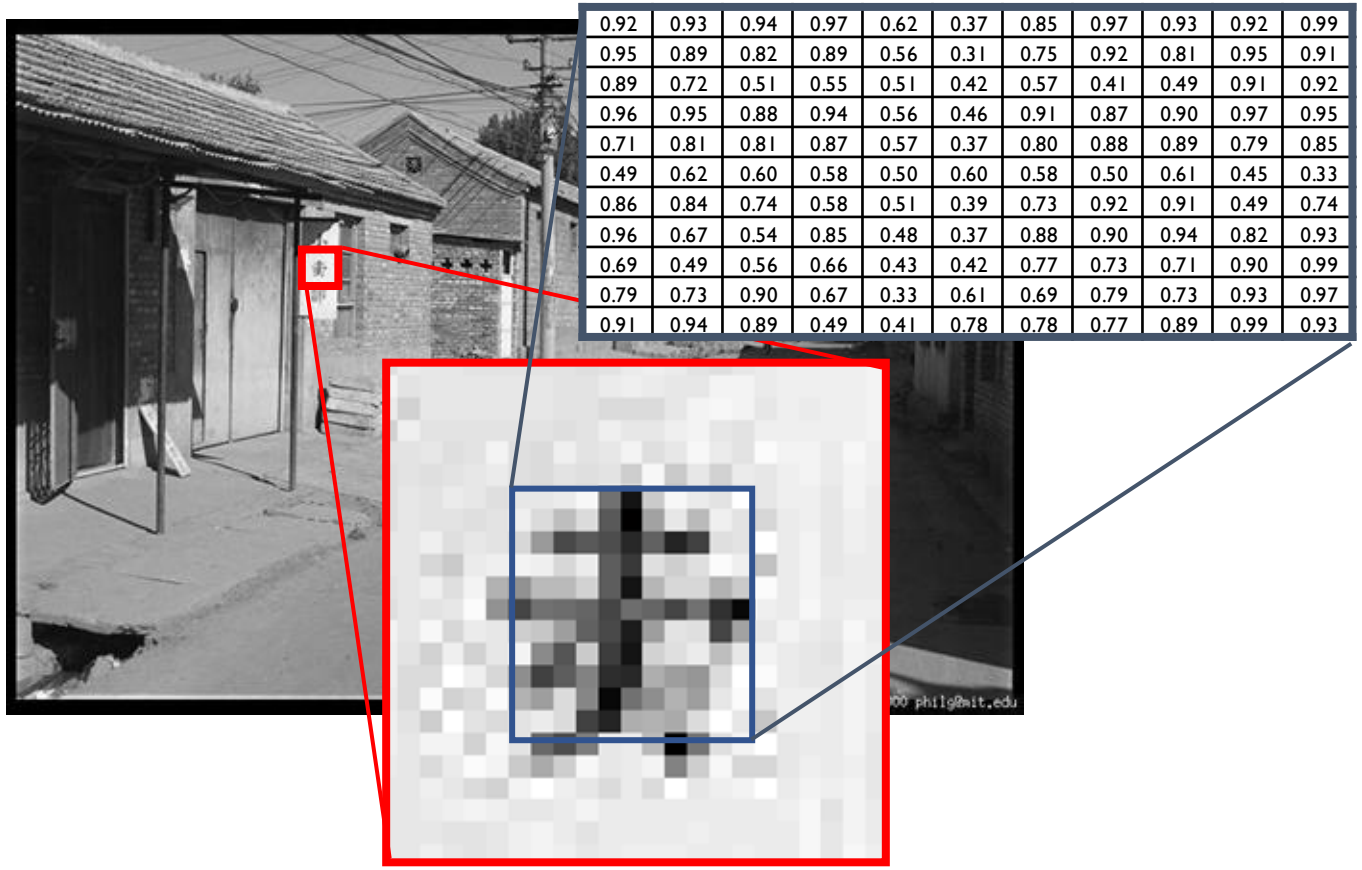
# Images in Python (import numpy)

`N x M` grayscale image "`im`" (N is number of rows and M is number of columns)

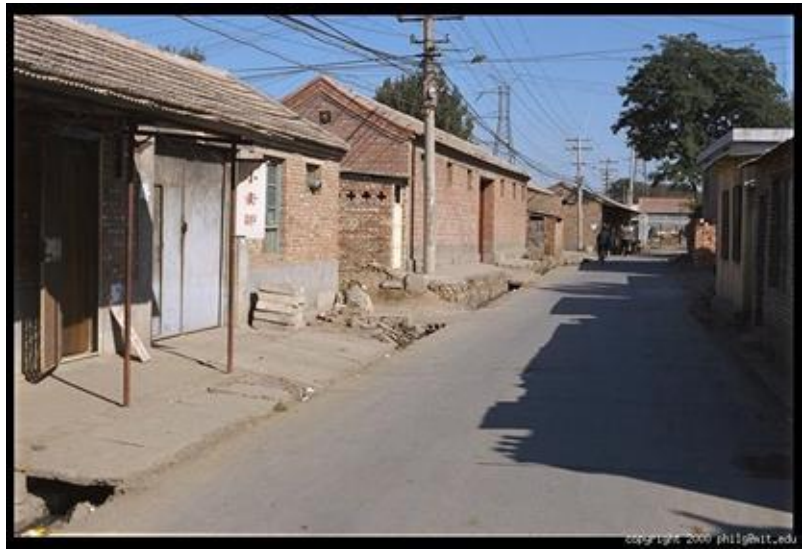
- `im[0,0]` = top-left pixel value
- `im[row,col]` = row pixels down, col pixels to right
- `im[N-1,M-1]` = bottom-right pixel



# Grayscale Intensity



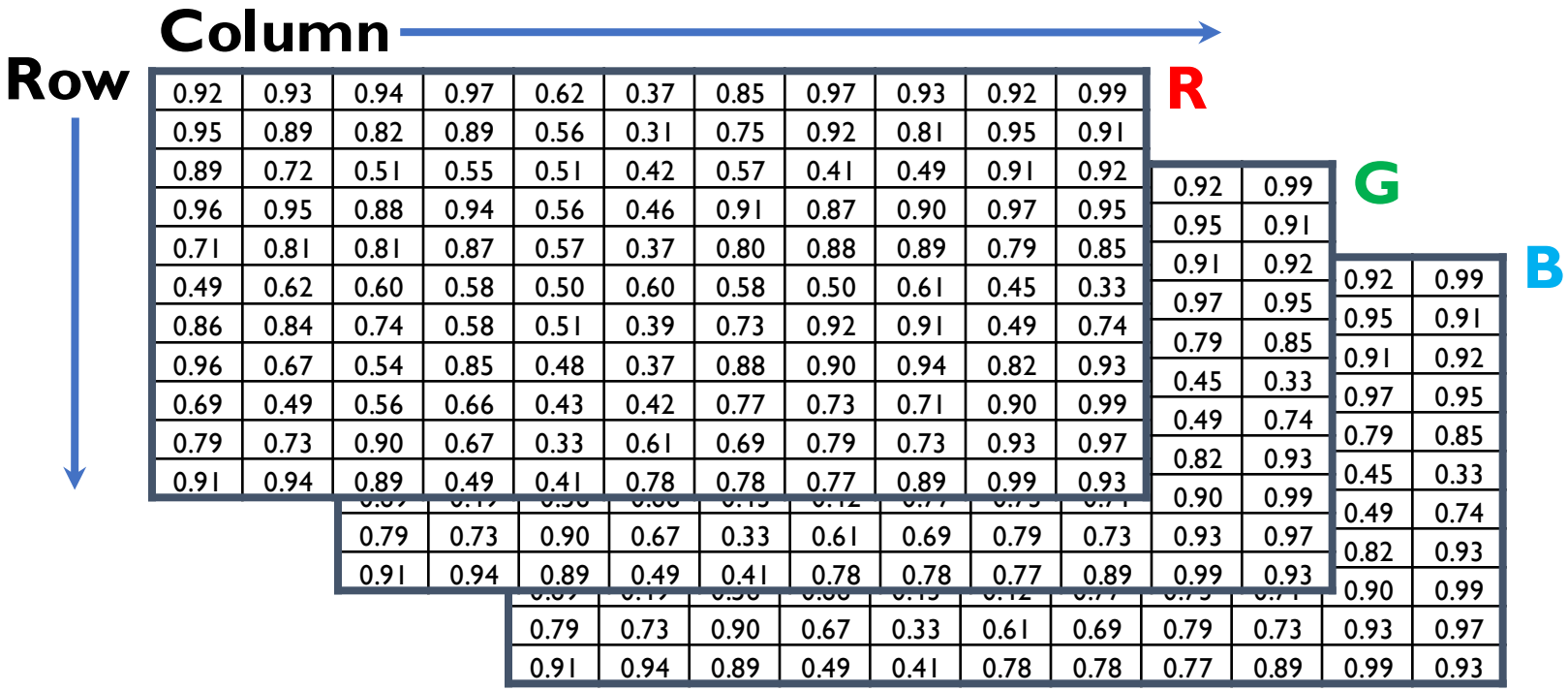
# Color



# Images in Python (import numpy)

`N x M` grayscale image "`im`"

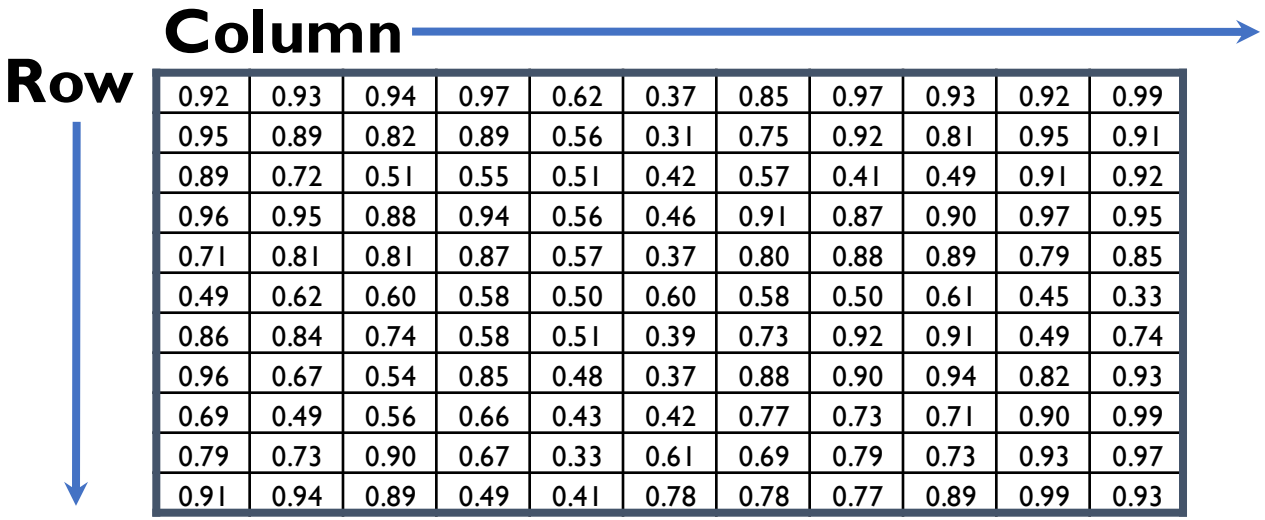
- `im[0,0,0]` = top-left pixel value, red channel
- `im[y,x,1]` = y pixels down, x pixels to right, green channel
- `im[N-1,M-1,2]` = bottom-right pixel, blue channel



# Images in Python (import numpy,scikit)

Take care of types!

- uint8 (values 0 to 255) – io.imread("file.jpg")
- float32 (values 0 to 255) – io.imread("file.jpg").astype(np.float32)
- float32 (values 0 to 1) – img_as_float32(io.imread("file.jpg"))



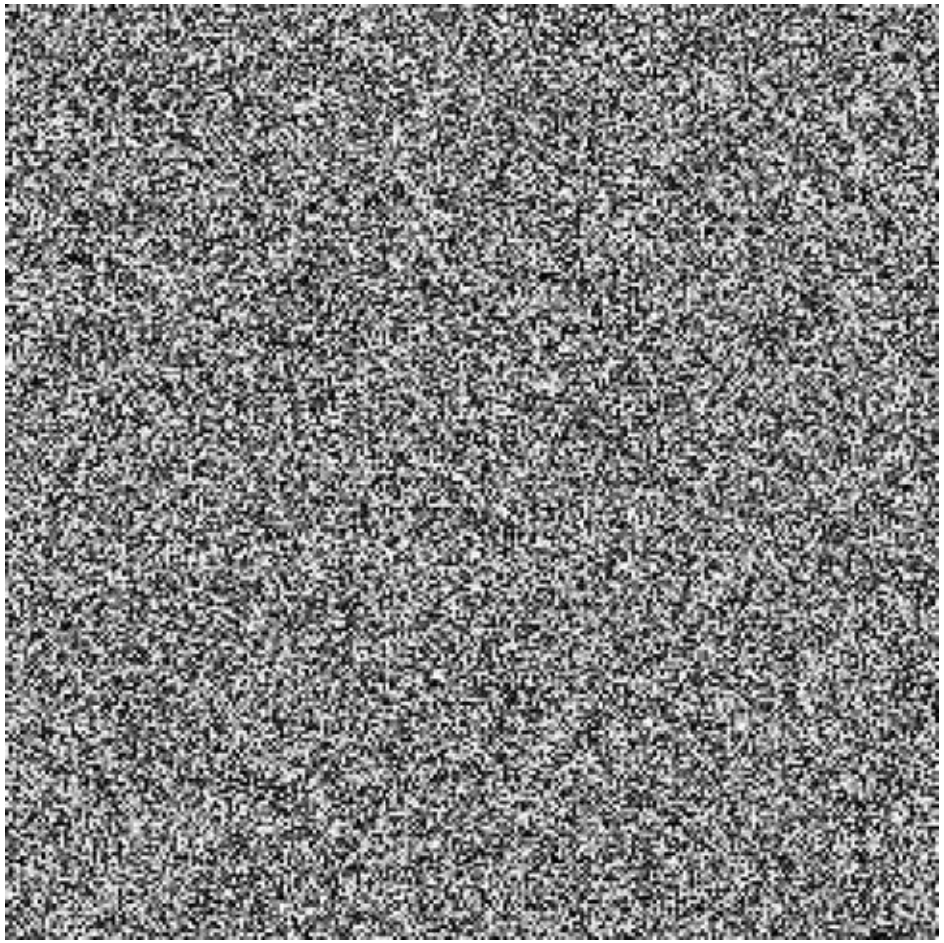

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0.92 | 0.93 | 0.94 | 0.97 | 0.62 | 0.37 | 0.85 | 0.97 | 0.93 | 0.92 | 0.99 |
| 0.95 | 0.89 | 0.82 | 0.89 | 0.56 | 0.31 | 0.75 | 0.92 | 0.81 | 0.95 | 0.91 |
| 0.89 | 0.72 | 0.51 | 0.55 | 0.51 | 0.42 | 0.57 | 0.41 | 0.49 | 0.91 | 0.92 |
| 0.96 | 0.95 | 0.88 | 0.94 | 0.56 | 0.46 | 0.91 | 0.87 | 0.90 | 0.97 | 0.95 |
| 0.71 | 0.81 | 0.81 | 0.87 | 0.57 | 0.37 | 0.80 | 0.88 | 0.89 | 0.79 | 0.85 |
| 0.49 | 0.62 | 0.60 | 0.58 | 0.50 | 0.60 | 0.58 | 0.50 | 0.61 | 0.45 | 0.33 |
| 0.86 | 0.84 | 0.74 | 0.58 | 0.51 | 0.39 | 0.73 | 0.92 | 0.91 | 0.49 | 0.74 |
| 0.96 | 0.67 | 0.54 | 0.85 | 0.48 | 0.37 | 0.88 | 0.90 | 0.94 | 0.82 | 0.93 |
| 0.69 | 0.49 | 0.56 | 0.66 | 0.43 | 0.42 | 0.77 | 0.73 | 0.71 | 0.90 | 0.99 |
| 0.79 | 0.73 | 0.90 | 0.67 | 0.33 | 0.61 | 0.69 | 0.79 | 0.73 | 0.93 | 0.97 |
| 0.91 | 0.94 | 0.89 | 0.49 | 0.41 | 0.78 | 0.78 | 0.77 | 0.89 | 0.99 | 0.93 |

# Think-Pair-Share

```
>>> from numpy import random as r
>>> I = r.rand(256,256)
```

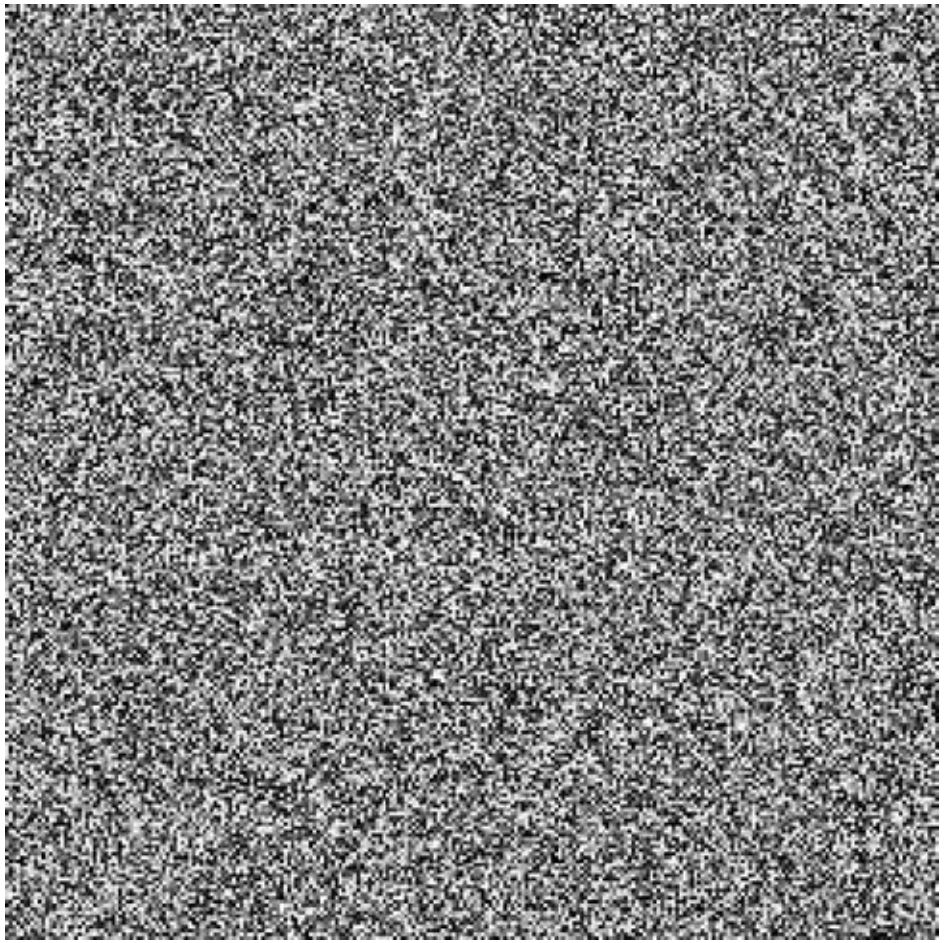
- What is this? What does it look like?
- Which values does it have?
- How many are there?

```
>>> from matplotlib import pyplot as p
>>> I = r.rand(256,256)
>>> p.imshow(I, cmap='gray')
>>> p.show()
```



```
>>> from matplotlib import pyplot as p
>>> I = r.rand(256,256)
>>> p.imshow(I, cmap='gray')
>>> p.show()
```

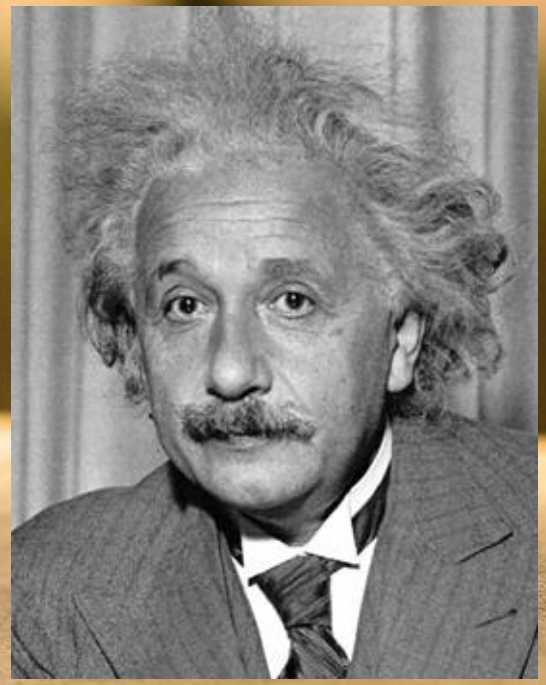
Is it an image?



COMP322: Computer Vision & Image Processing

# Next Episode

- Matrix Operations
- Image filtering



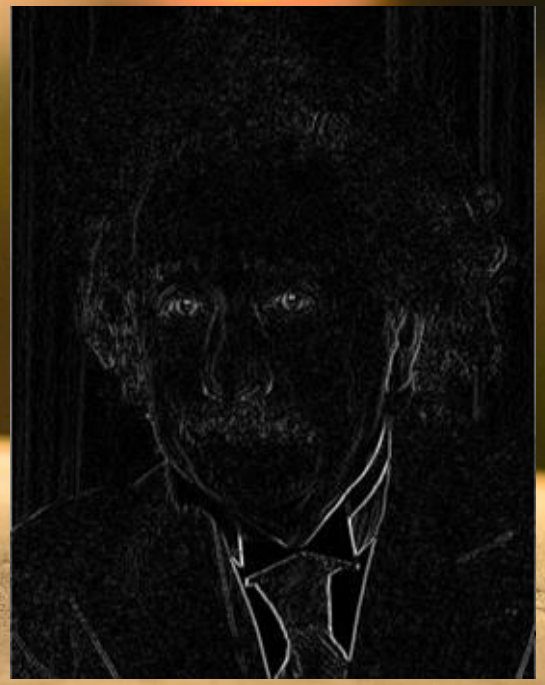


*Image Credit: Wallpapercrafter.com*