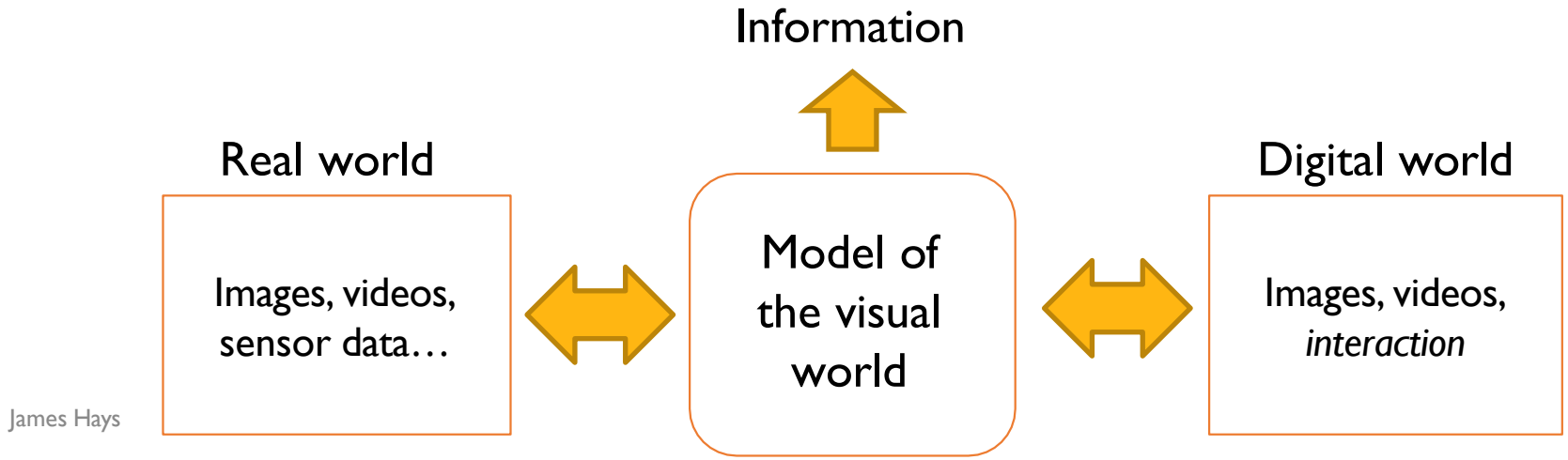
Introduction to Computer Vision

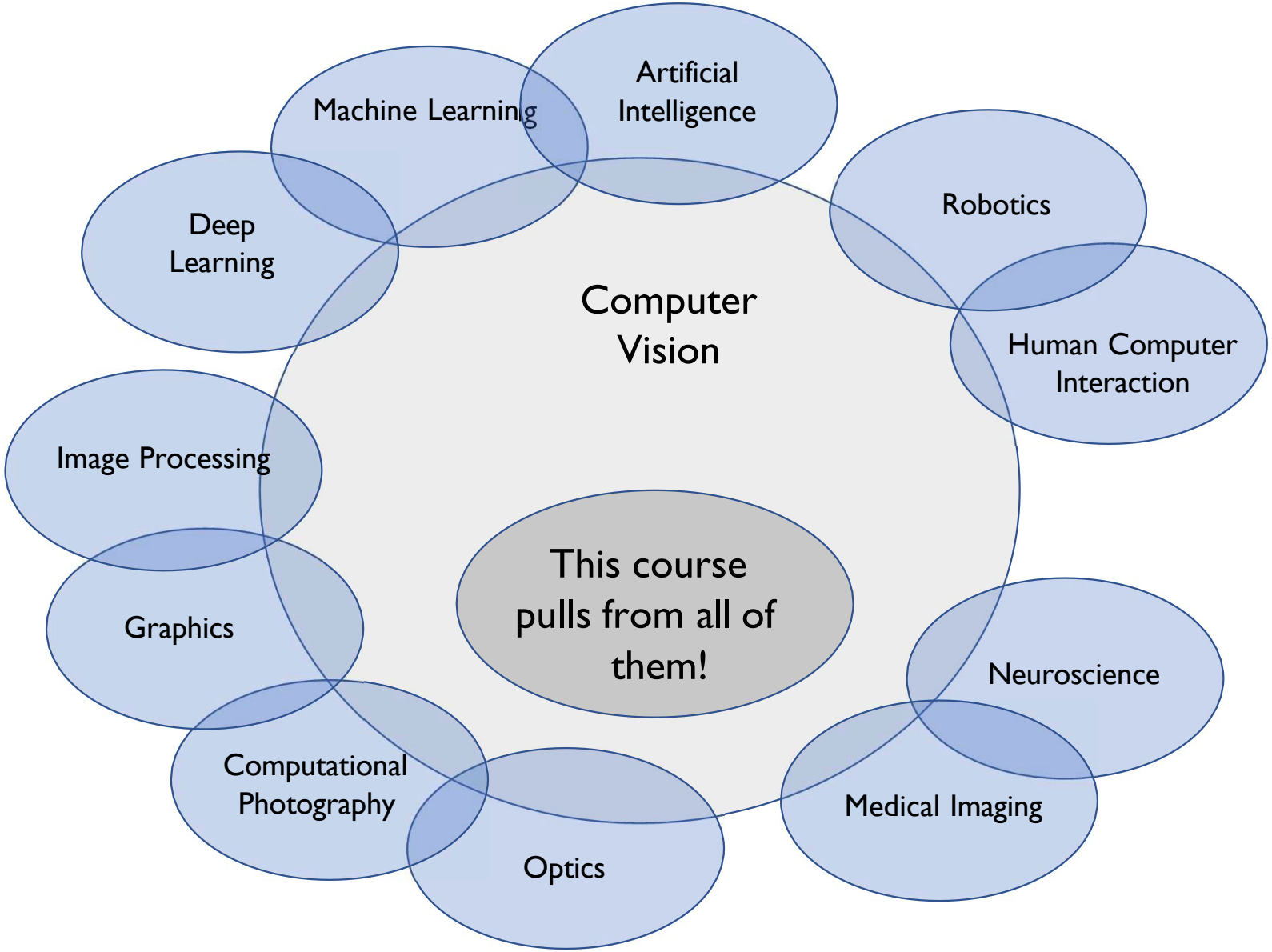
Kaveh Fathian
Assistant Professor
Computer Science Department
Colorado School of Mines

Lecture 2

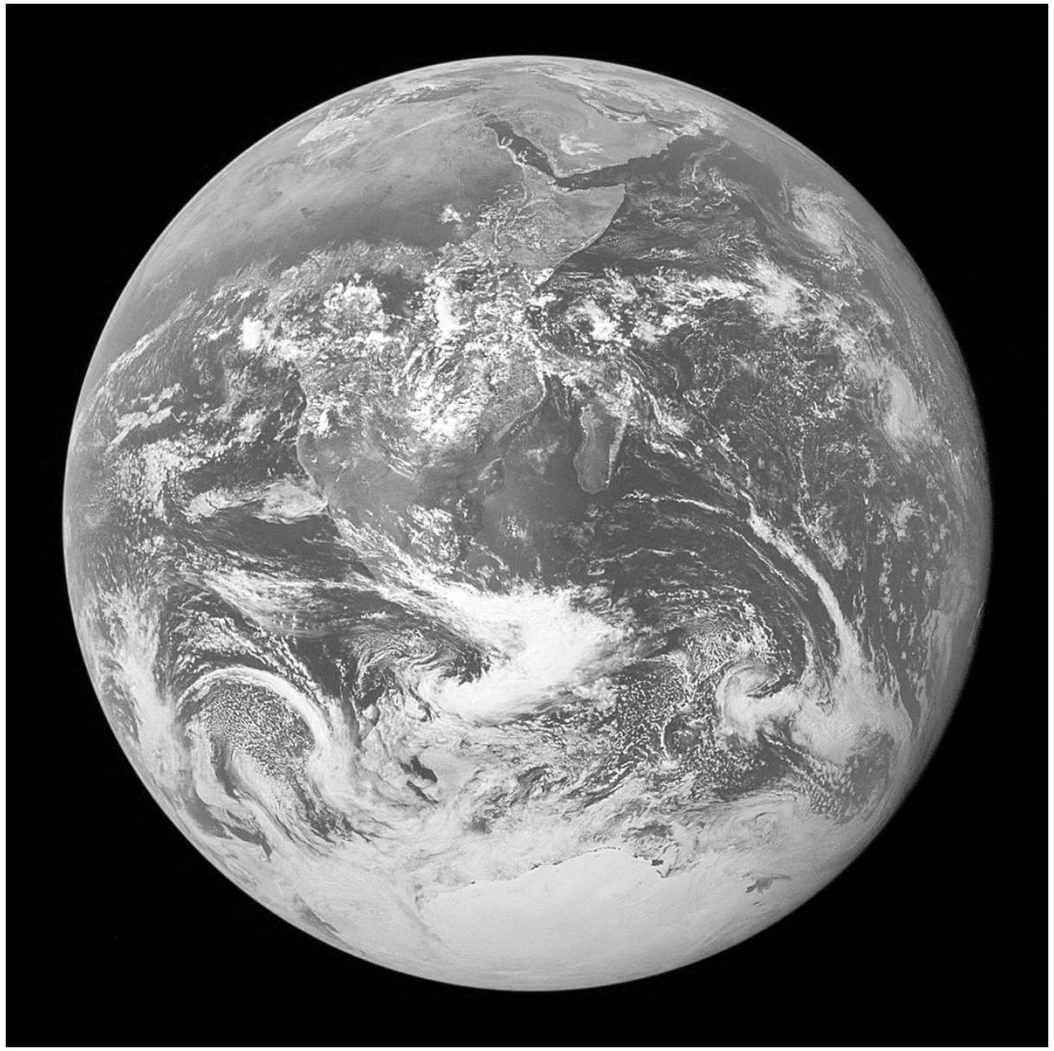
Computer Vision and Nearby Fields



Scope of Computer Vision

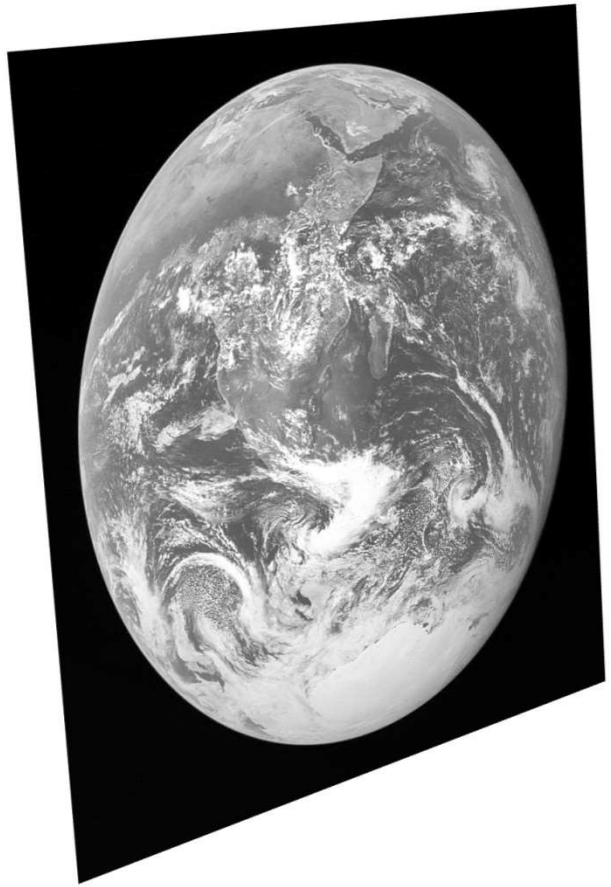


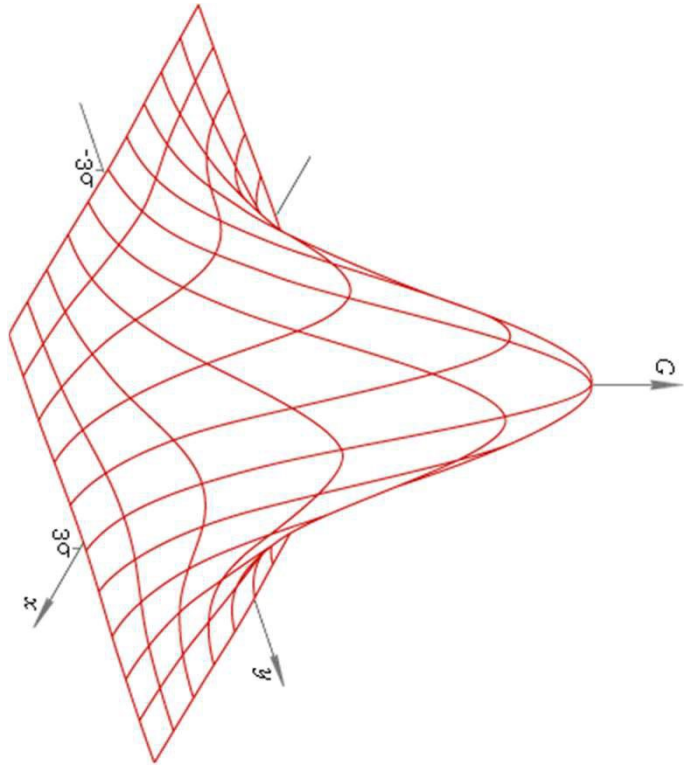
Image



Blue Marble. NASA | Apollo 17

Image



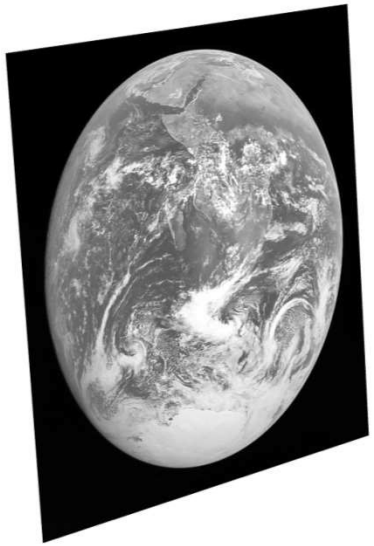


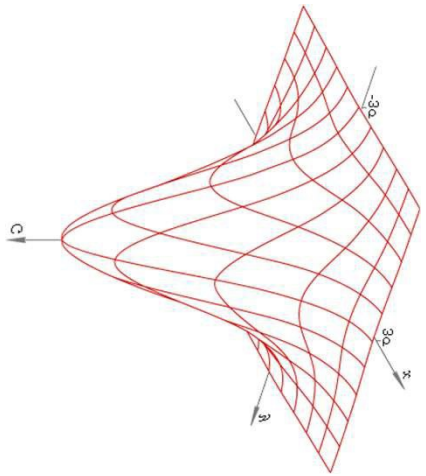
Blue Marble. NASA | Apollo 17

Signal

Definition

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

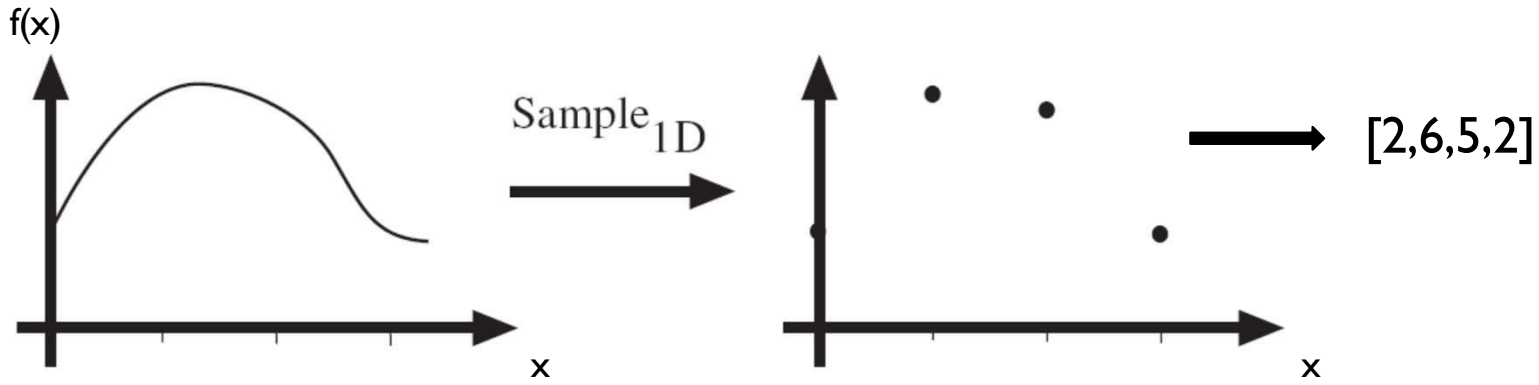




- Signals can be
  - Continuous: light
  - Discrete: measurement of a light
  - Sampling: reduction of continuous signal to a discrete signal
- Any phenomenon
  - Light
  - Heat
  - Gravity

Sampling

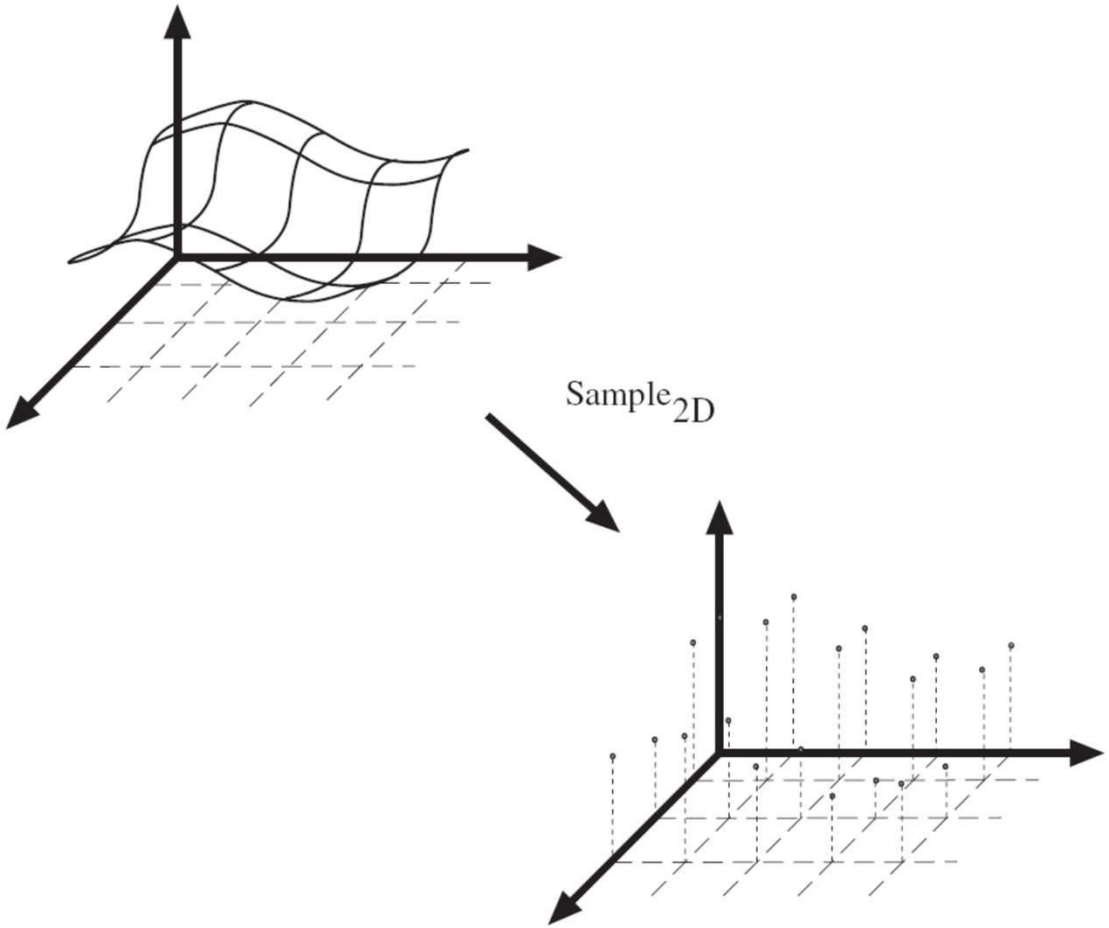
- Sampling in 1D takes a function and returns a vector whose elements are values of that function at the sample points.



Danny Alexander

Sampling

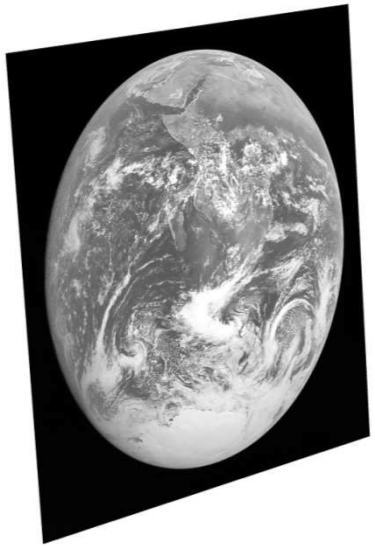
- Sampling in 2D takes a function and returns a matrix

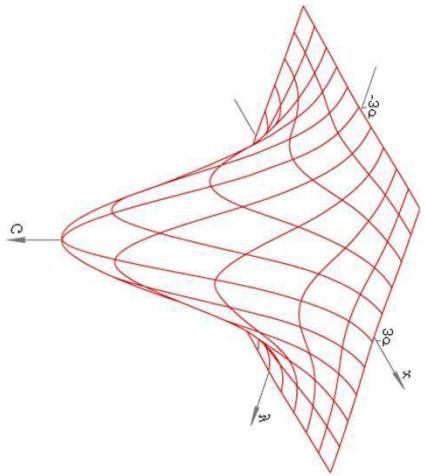


2D Image

Definition

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

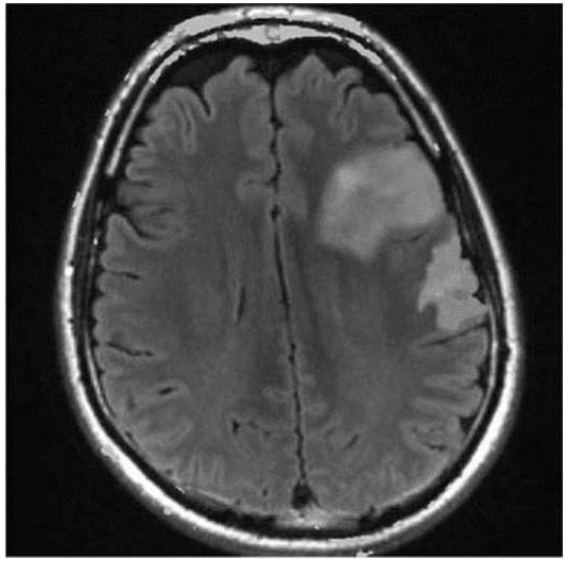




* or a 2D projection of a multi-dimensional signal

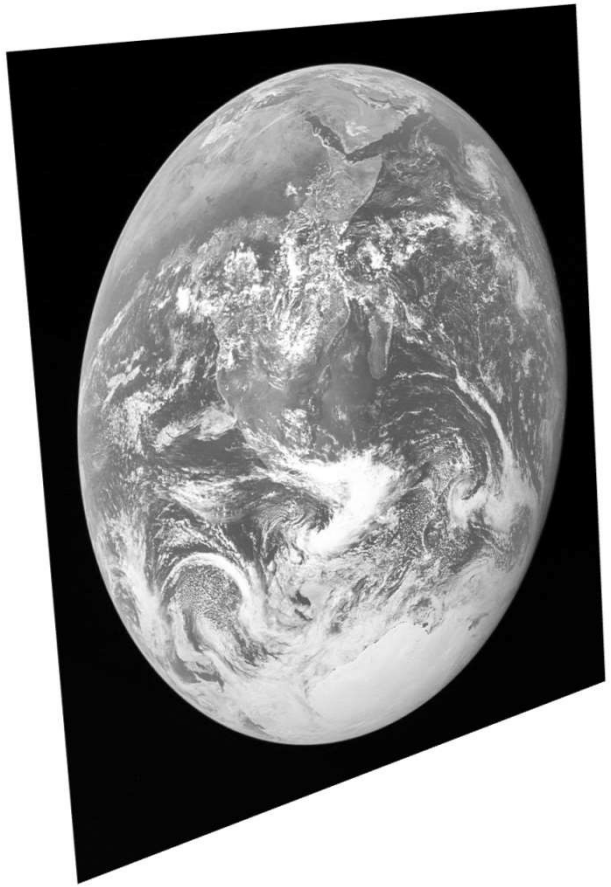
- Function stores 'brightness'
- 2D signals are special for us
  - Brightness along x and y dimensions
- Video: xy-coordinates + time
  - Time-varying 2D signal

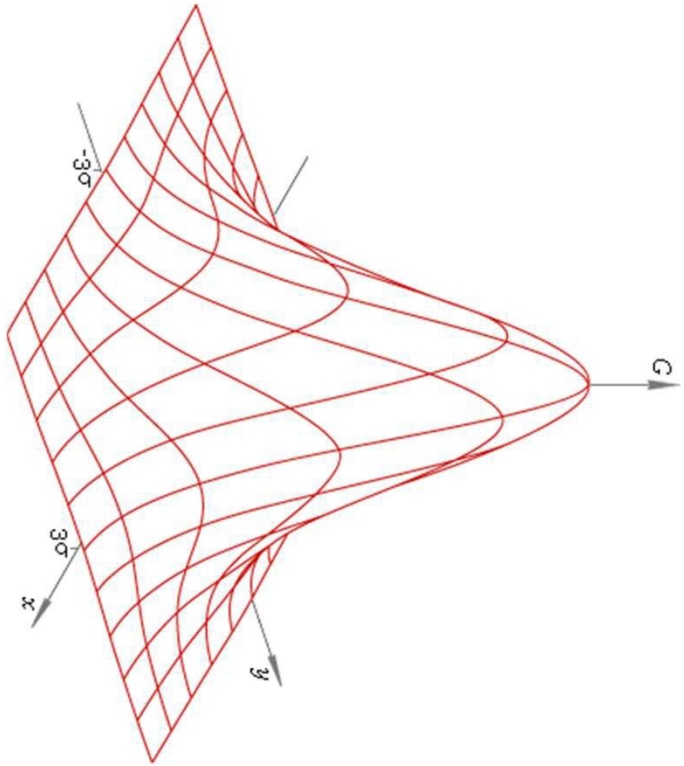
Example of 2D Images



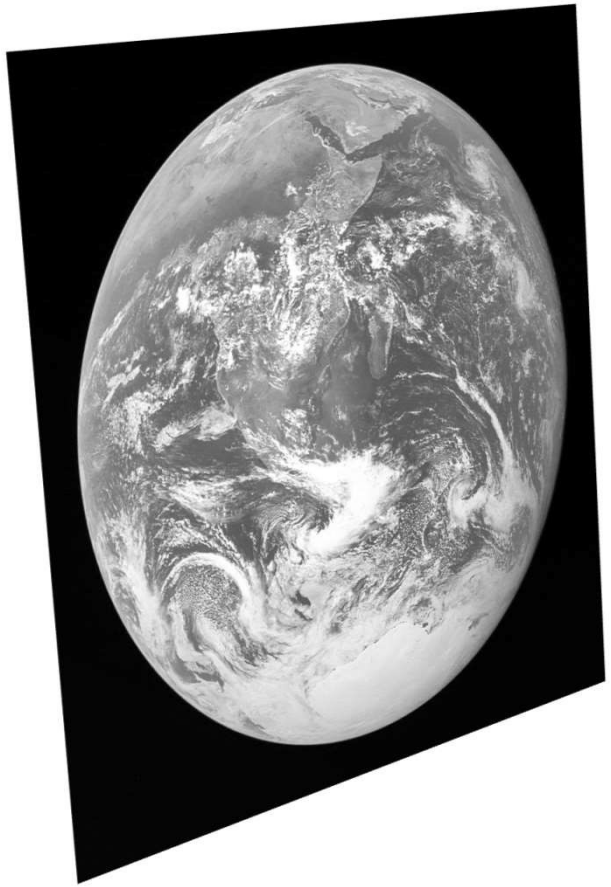
http://radiologykey.com/wp-content/uploads/2016/01/9781604063325_c014_f004.jpg

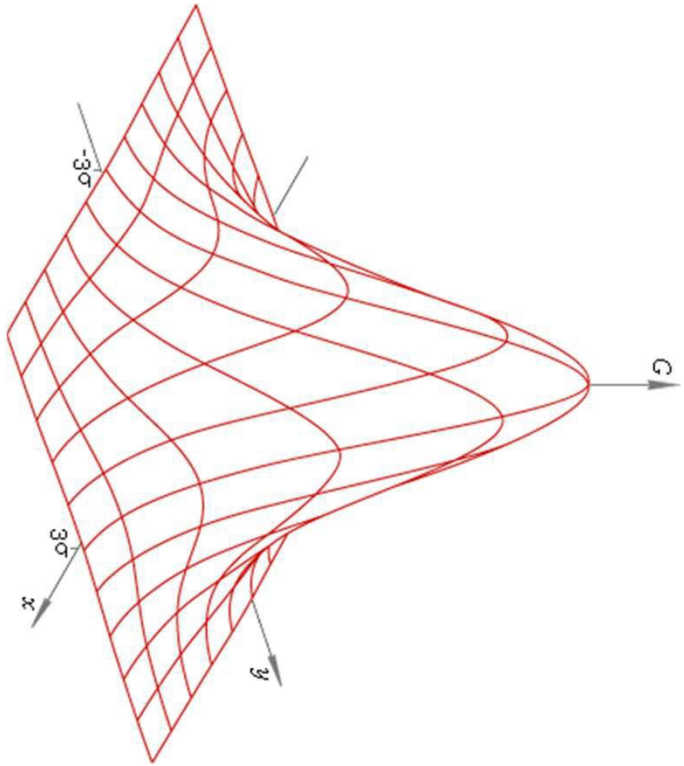
Sampling in Practice

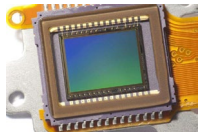




Sampling in Practice: Digital Image

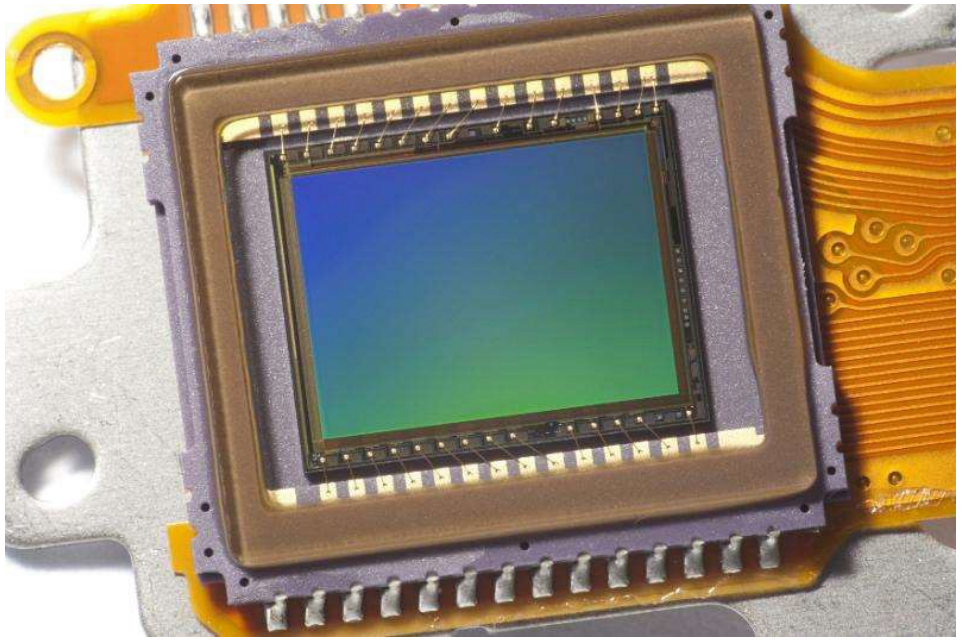




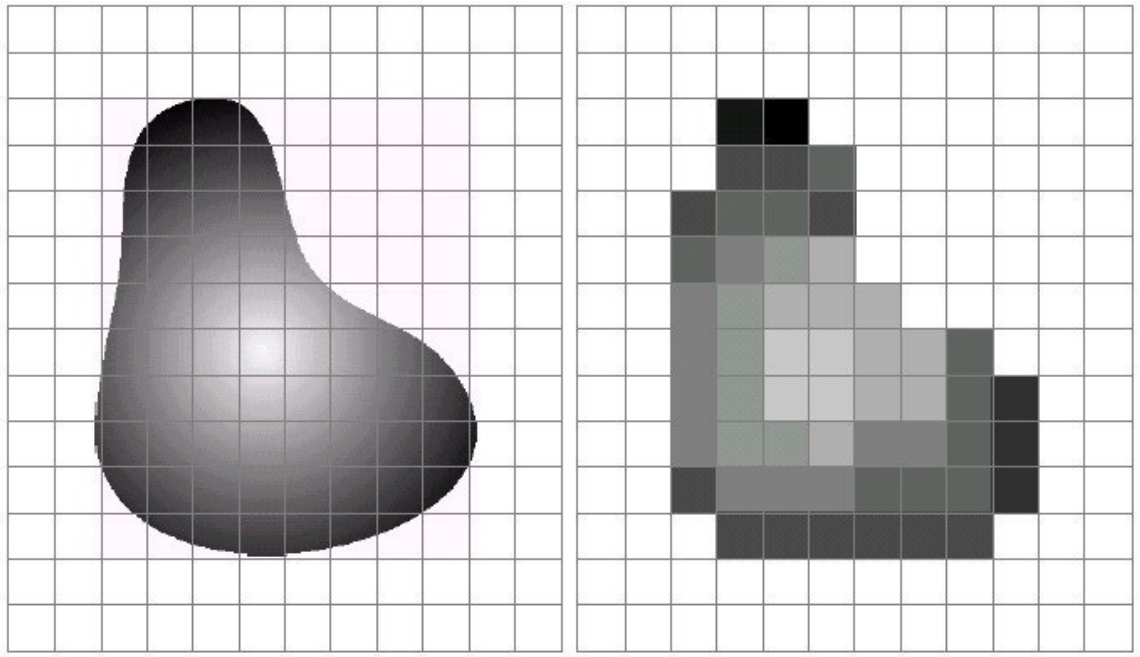


CMOS / CCD

Sensor Array



CMOS sensor



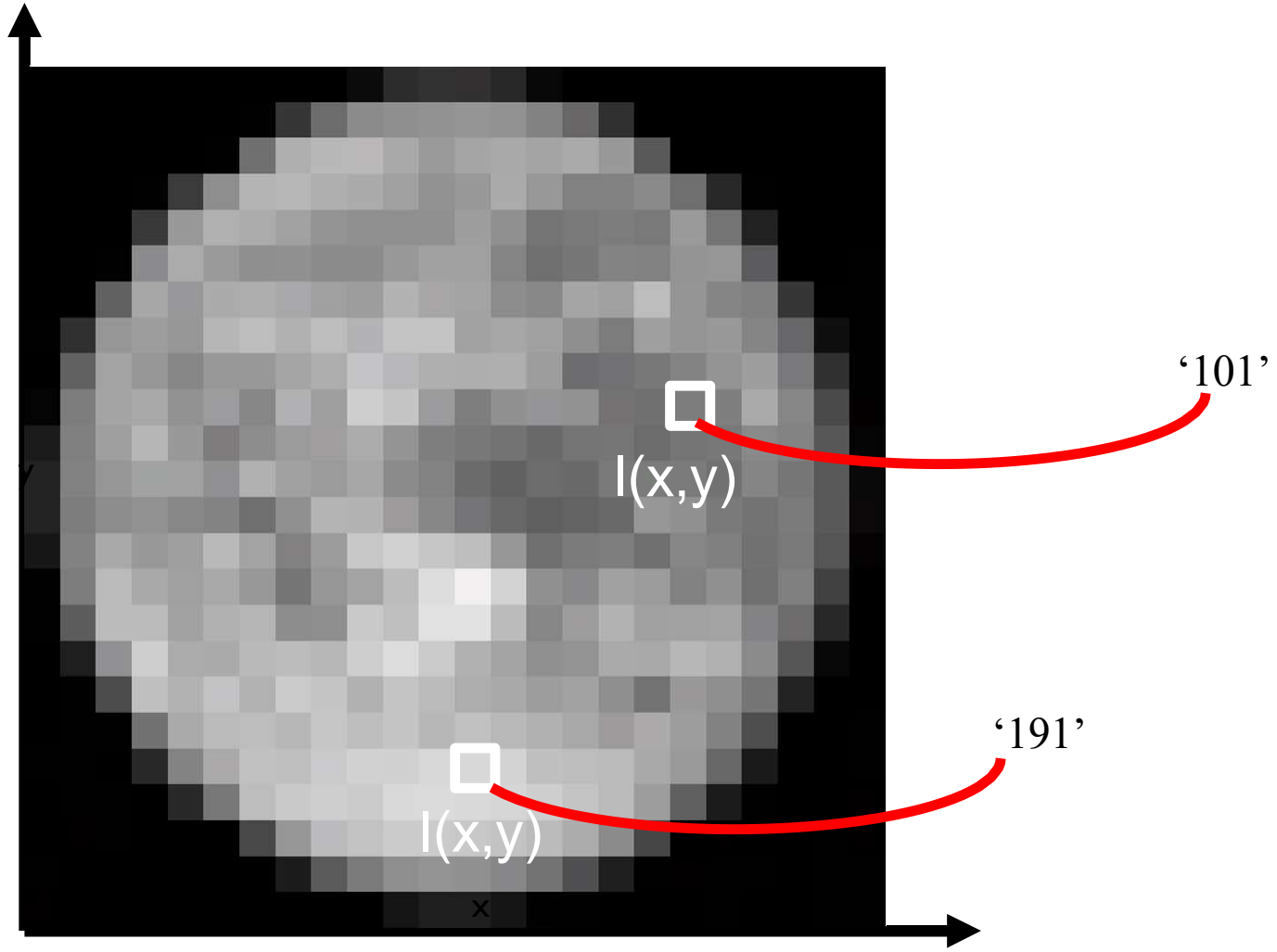
a b

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

James Hays

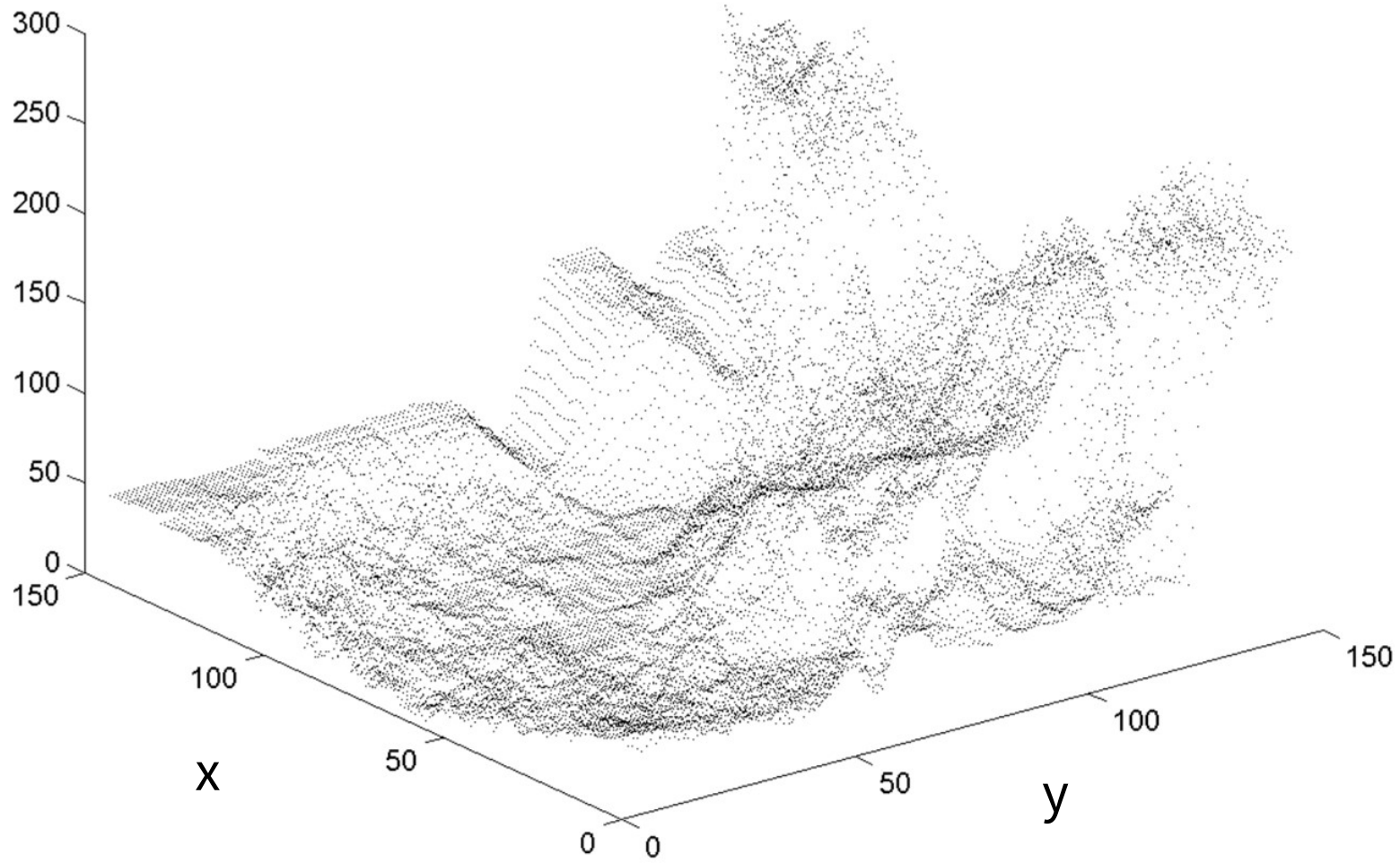
Elements of a Digital Image

Pixel: picture element

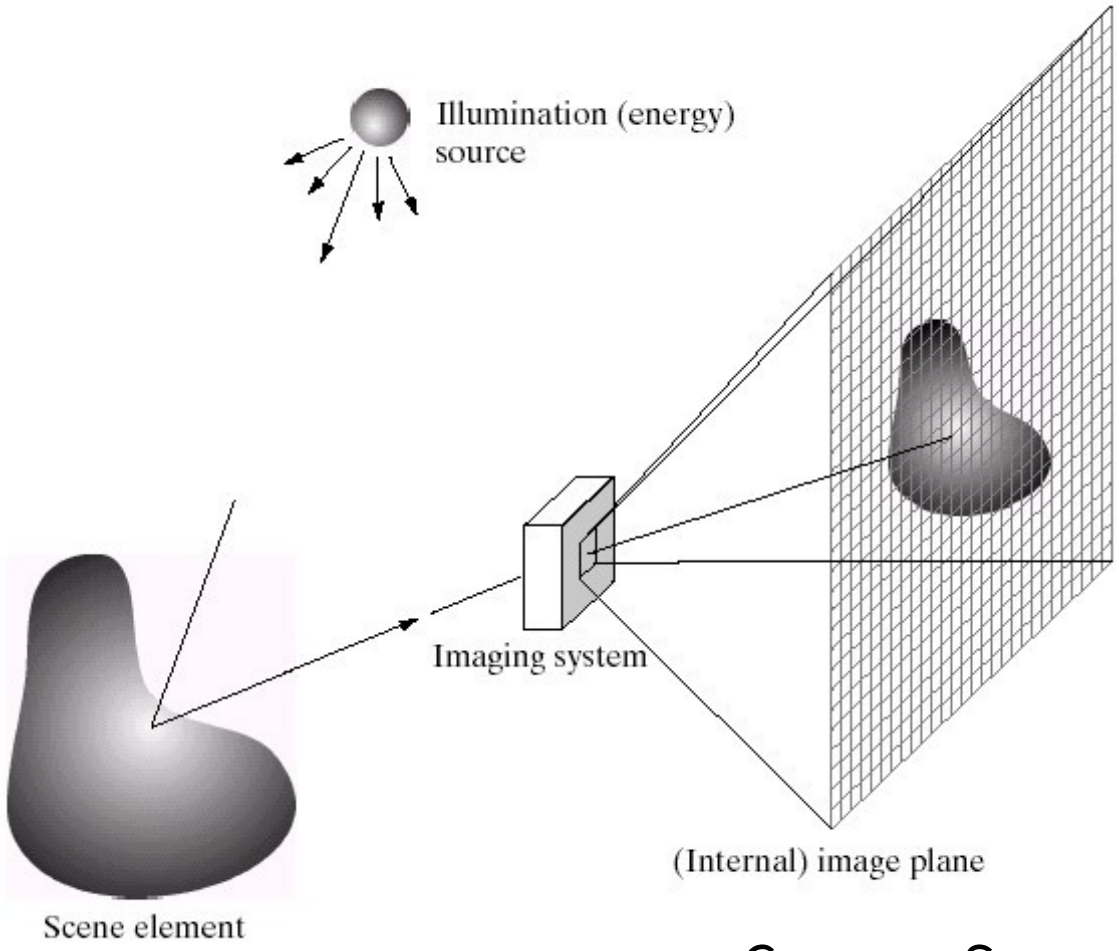


Digital Image is a 2D Signal

Brightness or intensity

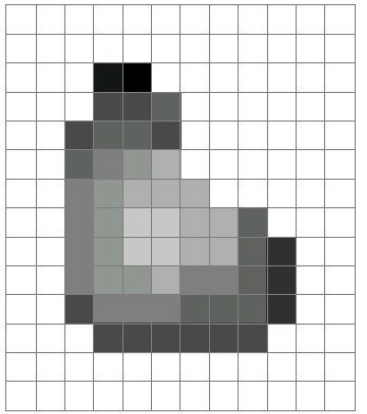


Light Integration Over the “Frustum”



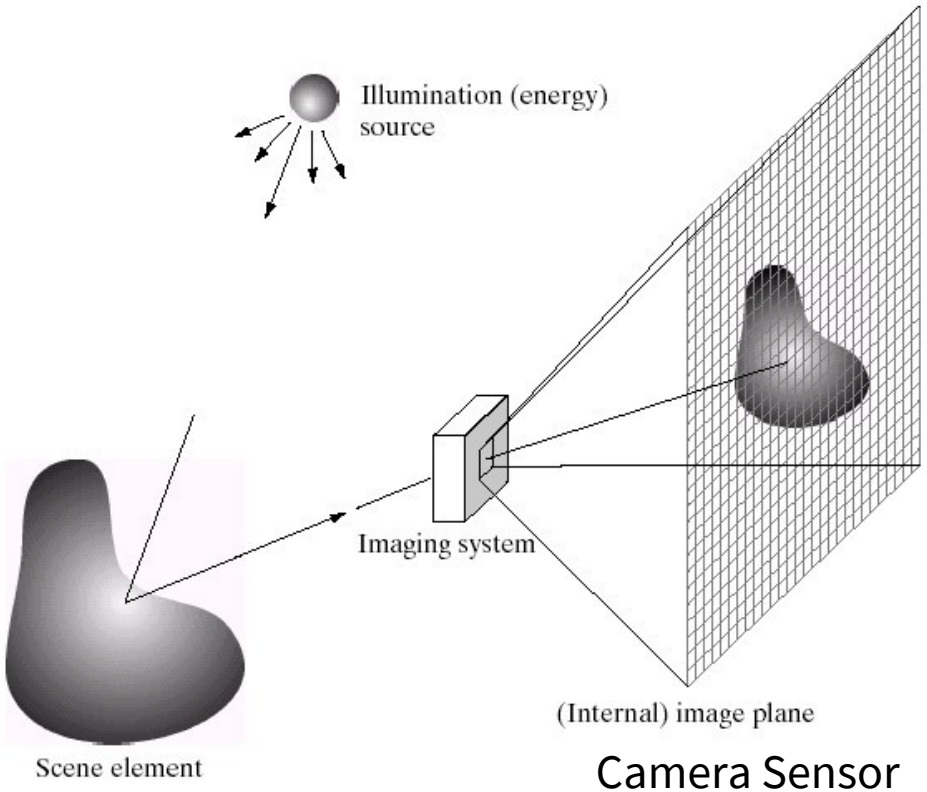
James Hays

Camera Sensor



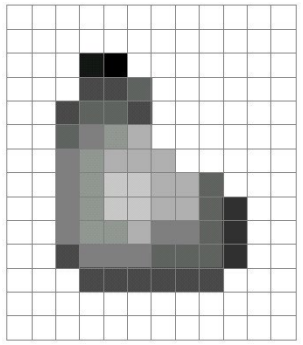
Output Image

Light Integration Over the “Frustum”

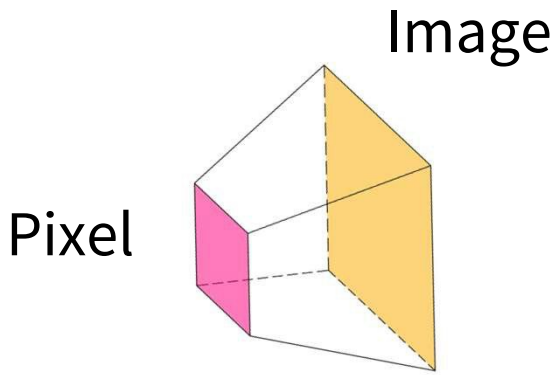


Camera Sensor

James Hays



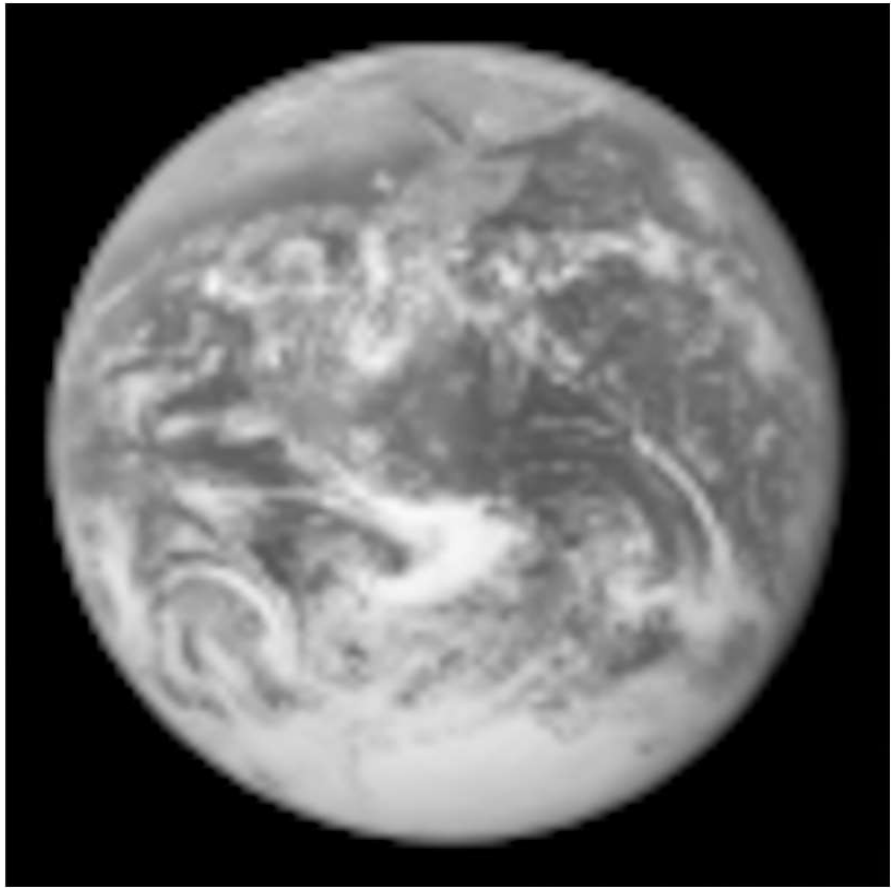
Output Image



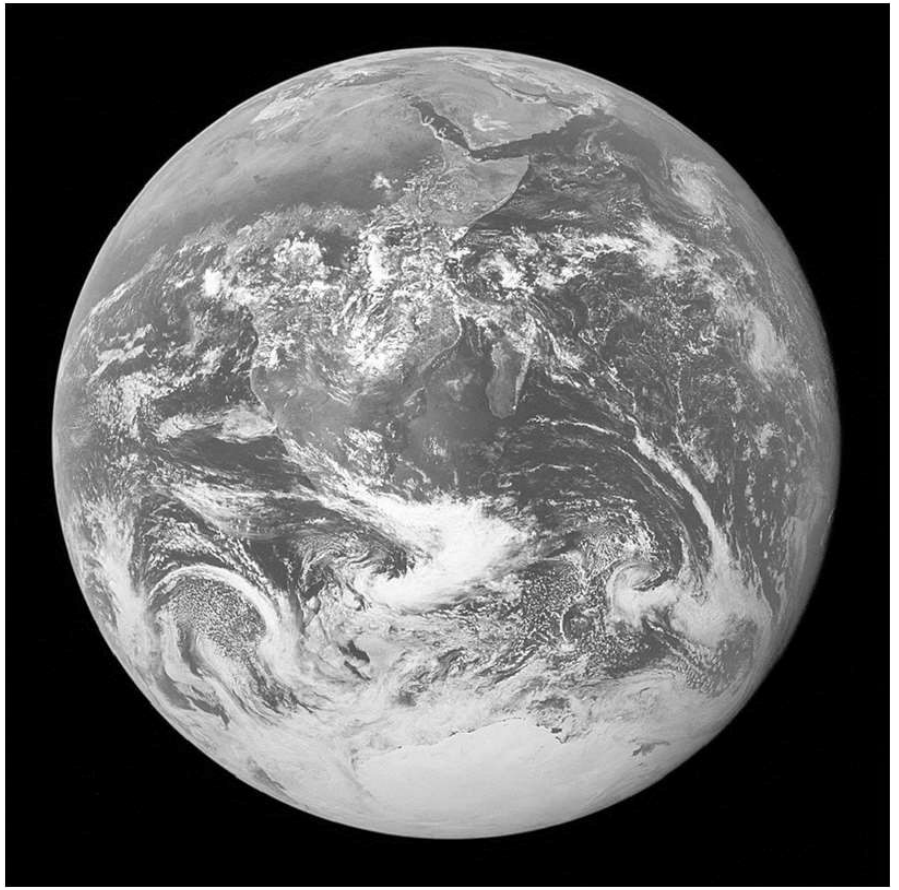
A frustum approximates the ray space that a pixel samples

Resolution: geometric vs. spatial

Both images are 1000x1000 pixels—same spatial resolution:

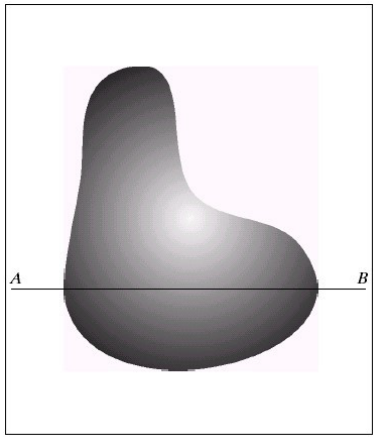


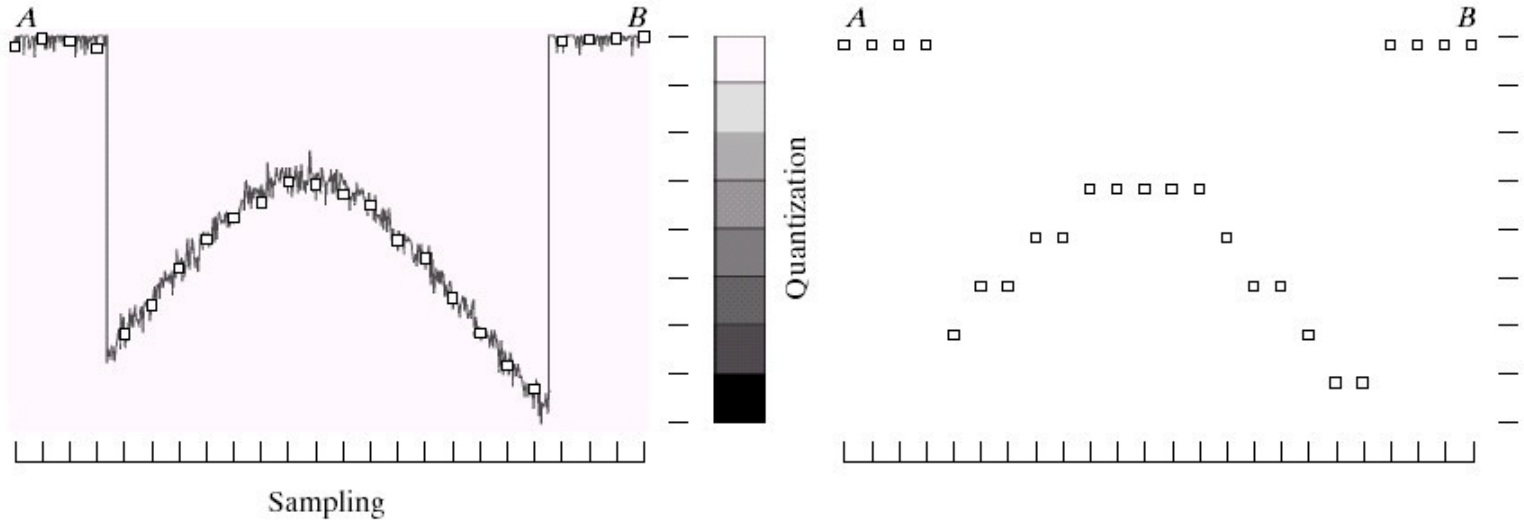
Low geometric resolution



High geometric resolution

Quantization

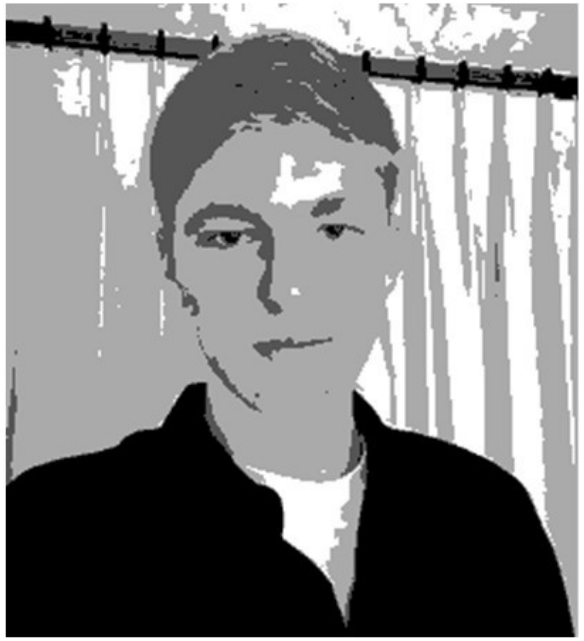




Quantization Effects – Radiometric Resolution

8 bit – 256 levels

4 bit – 16 levels



2 bit – 4 levels

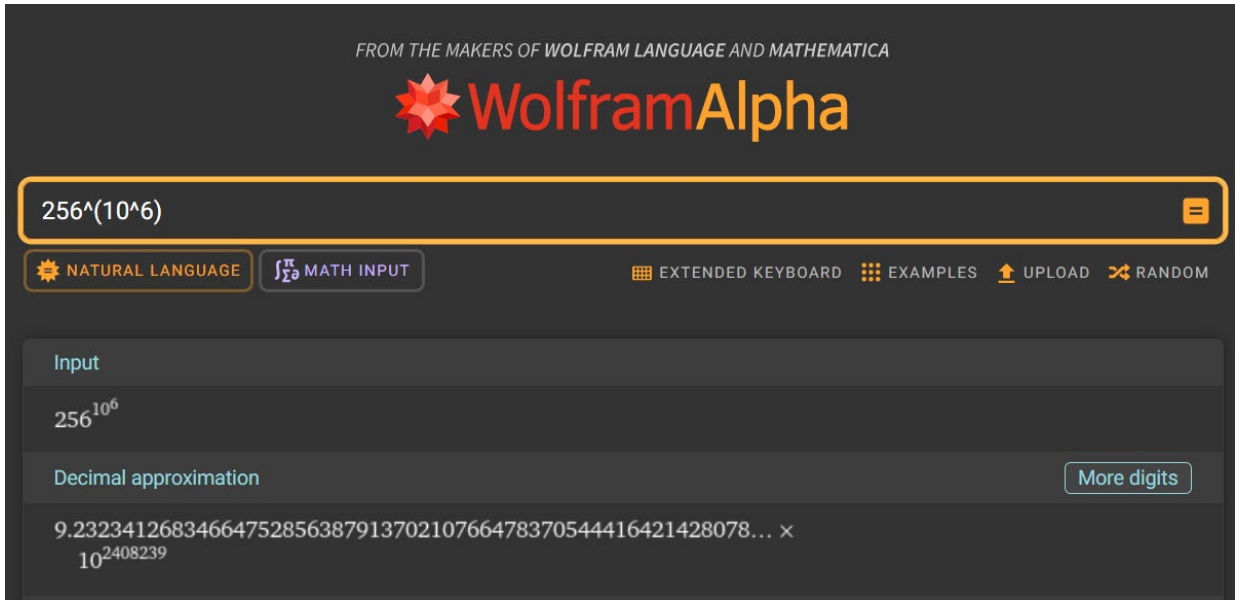
1 bit – 2 levels

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)

- All images ever created
  - of size 1000x1000



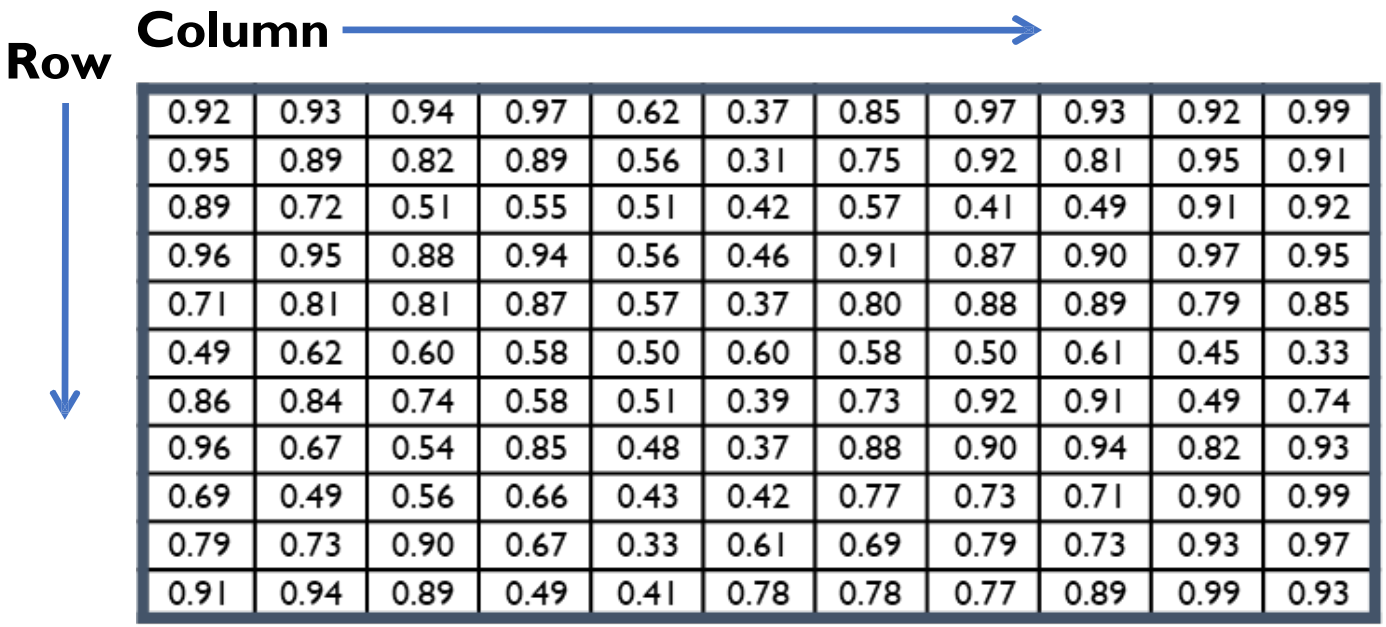
* it is estimated that there may be 10^{78} to 10^{82} atoms in the known universe!

- Computer vision as making sense of an extremely high-dimensional data
  - ‘natural’ images are a subspace

Images in Python (`import numpy`)

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

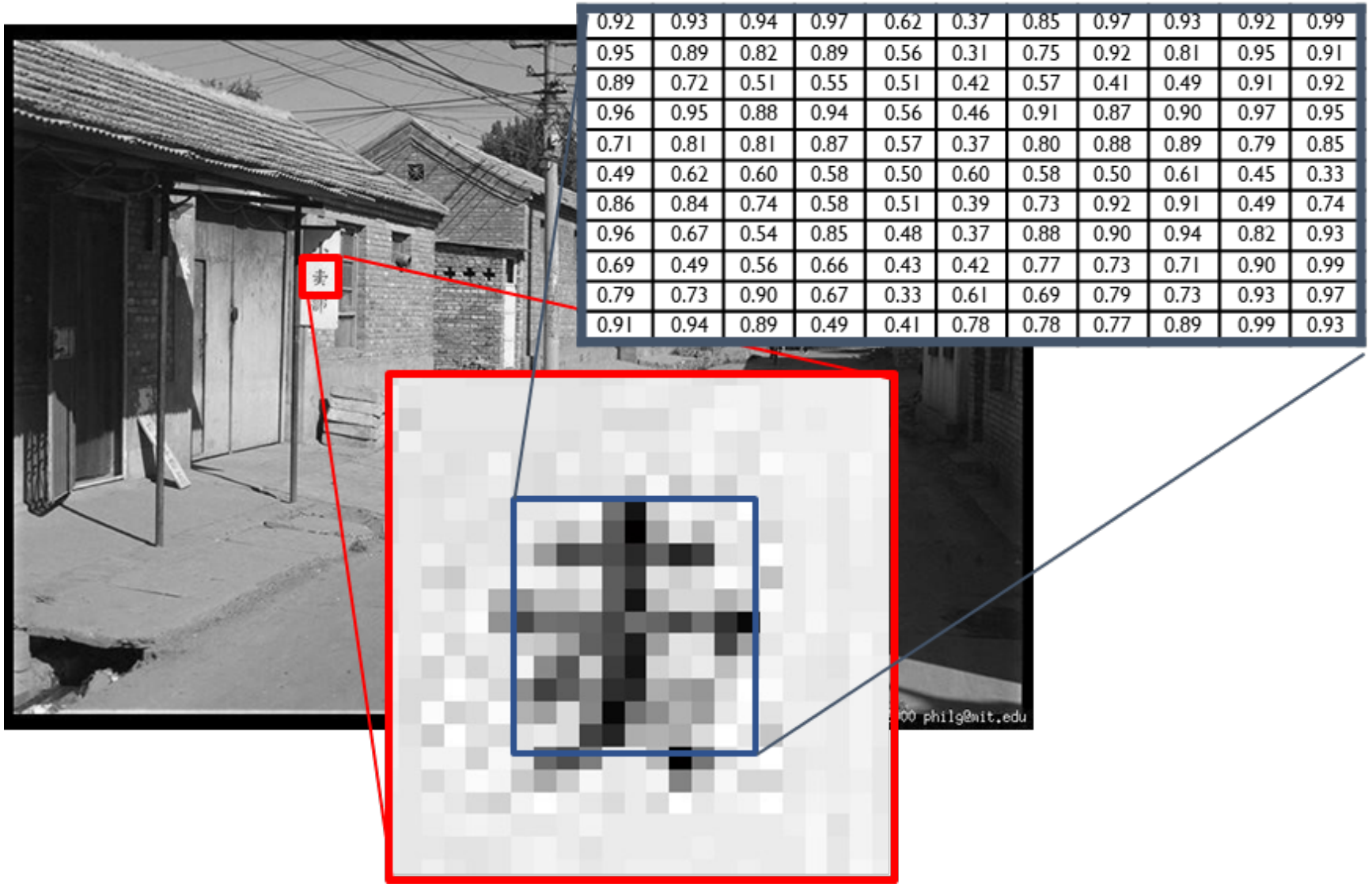
- `im[0,0]` = top-left pixel value
- `im[y,x]` = y pixels down, x pixels to right
- `im[N-1,M-1]` = bottom-right pixel



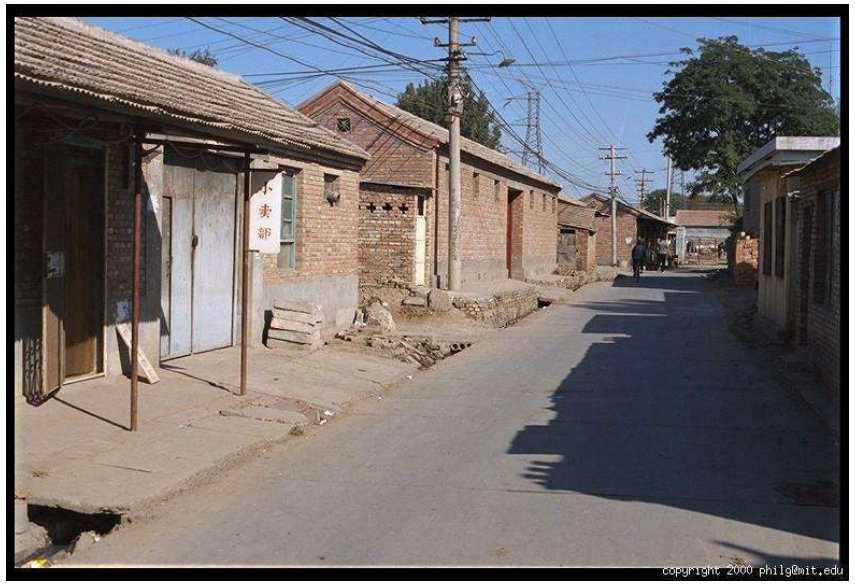
0.92	0.93	0.94	0.97	0.62	0.37	0.85	0.97	0.93	0.92	0.99
0.95	0.89	0.82	0.89	0.56	0.31	0.75	0.92	0.81	0.95	0.91
0.89	0.72	0.51	0.55	0.51	0.42	0.57	0.41	0.49	0.91	0.92
0.96	0.95	0.88	0.94	0.56	0.46	0.91	0.87	0.90	0.97	0.95
0.71	0.81	0.81	0.87	0.57	0.37	0.80	0.88	0.89	0.79	0.85
0.49	0.62	0.60	0.58	0.50	0.60	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

James Hays

Grayscale Intensity



Color



Red intensity

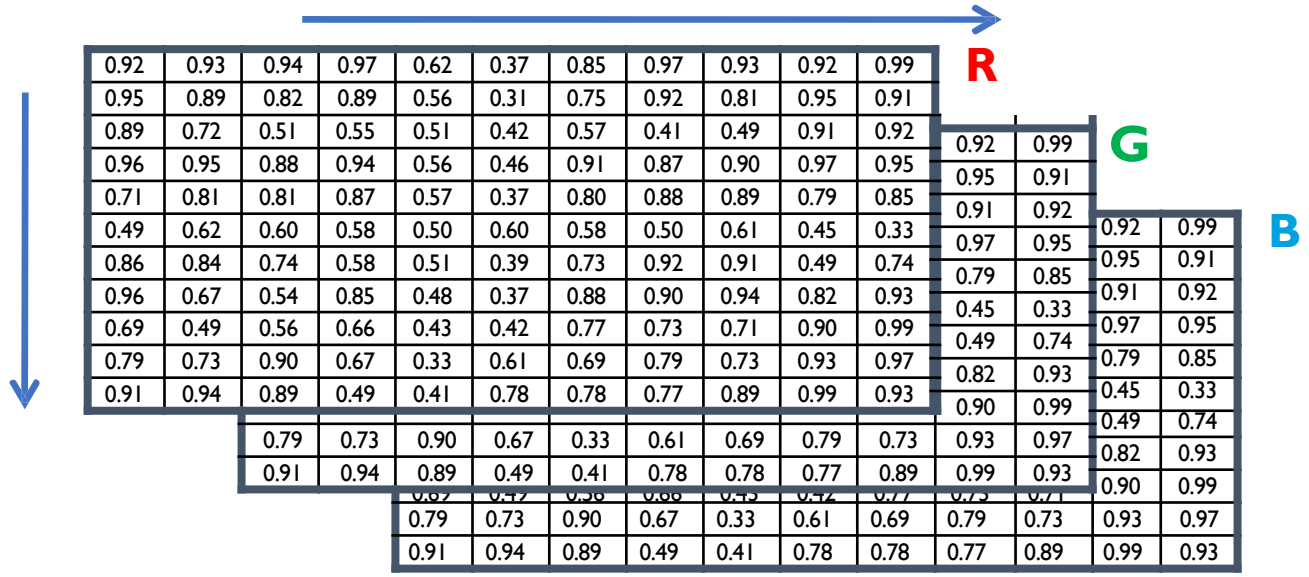
Green

Blue

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

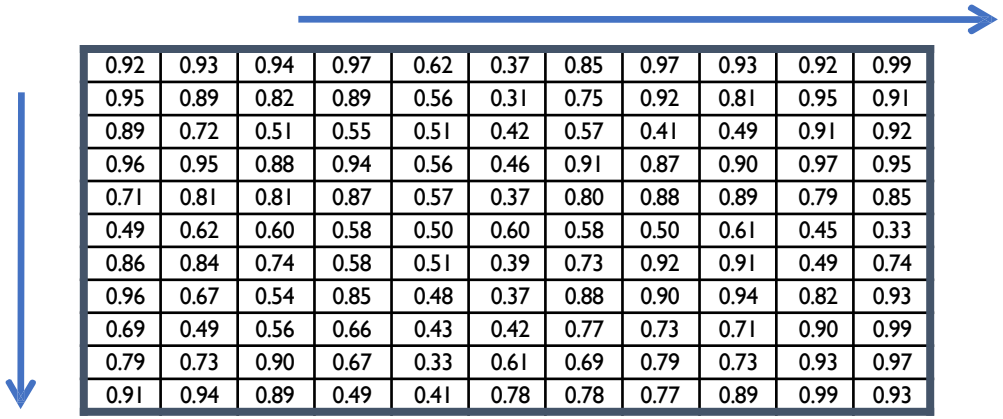


James Hays

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

James Hays

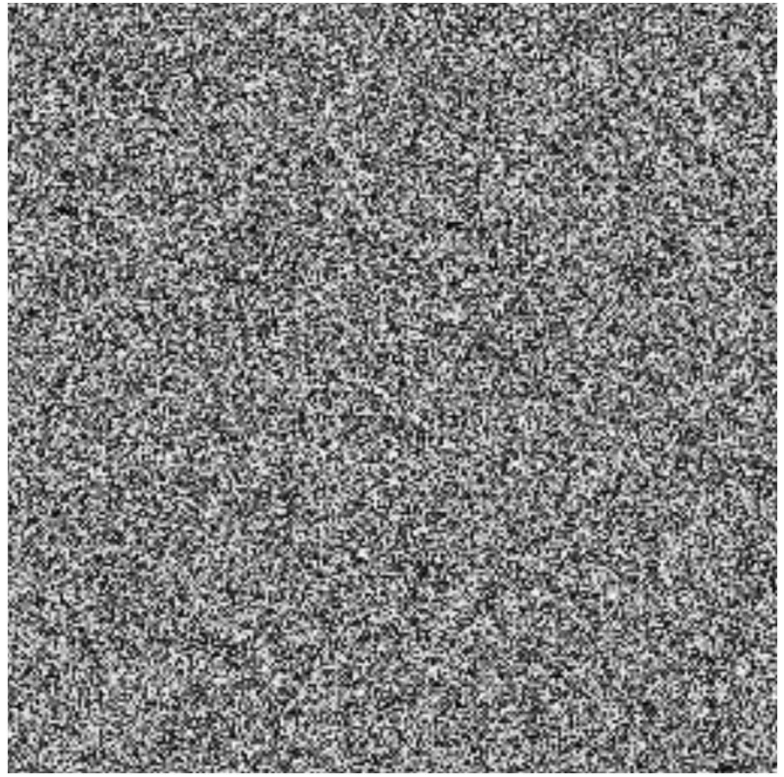
Images in Python

```
>>> 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?

Images in Python

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



Is it an image?

Coding In This Course

We use

- Package manager conda
- Python and its virtual environments
- `numpy` numerics library
- Visual Studio Code (VSCode) debugger

GenAI (e.g. ChatGPT)

- Do ***NOT*** use! Otherwise, you won't learn to code!

Be kind & supportive

- Students have diverse background (CS/EE/ME/…)
- Python proficiency varies among students
- Share your skills & knowledge with others on Ed

Python Refresher

1. Download **tutorials**: https://github.com/ariarobotics/cv/tree/main/code/tutorials

2. **Install Git:** follow instructions in “Github Guide”

3. **Install Python/Conda:** follow instructions in “Python Setup”

4. **Install VSCode:** follow instructions in “Setting up VSCode”

5. Run & review the `numpy` tutorial (on Google Colab)
    1. Finish all 10 exercises at the end of the turorial