

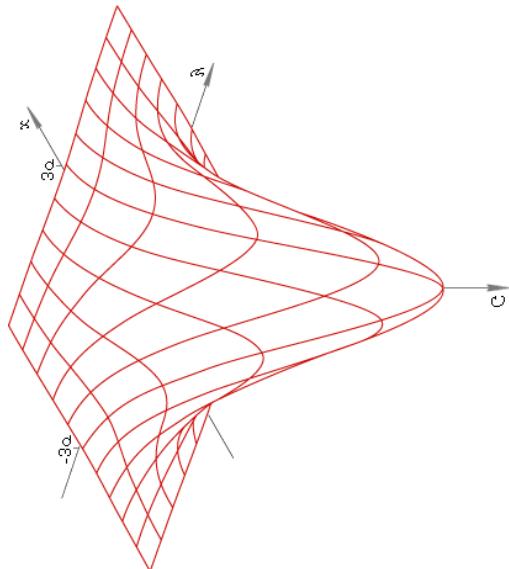
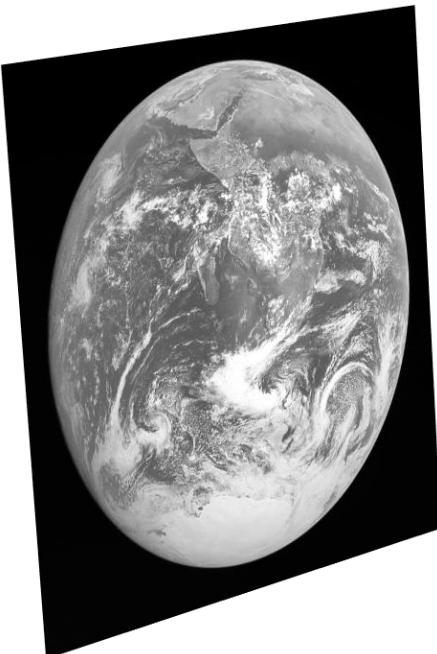
Digital Image Processing and Computer Vision

Introduction

Signal

Definition

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



Signals can be

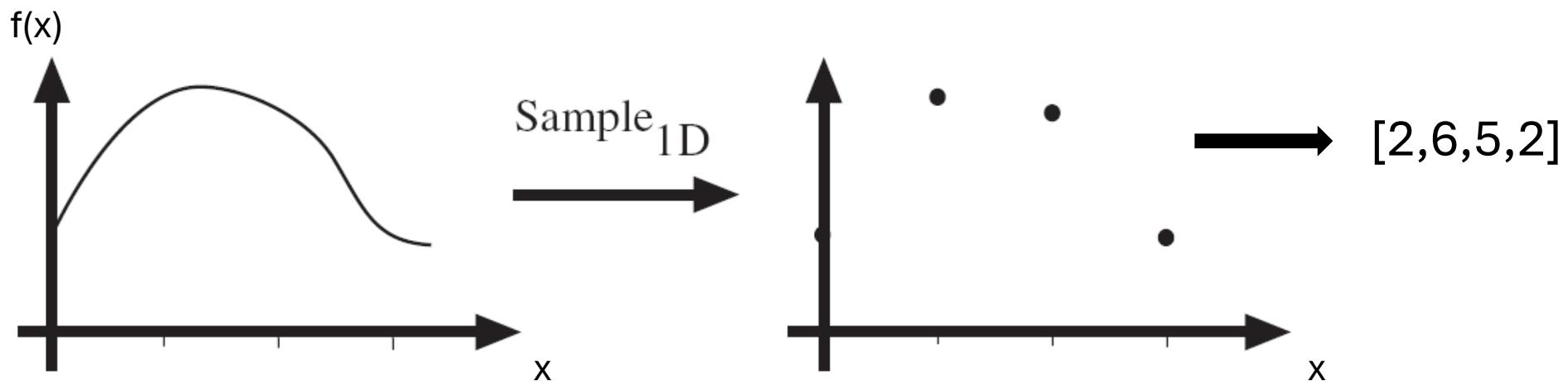
- Continuous: light
- Discrete: measurement of a light
- Sampling: reduction of continuous signal to a discrete signal

Any phenomenon

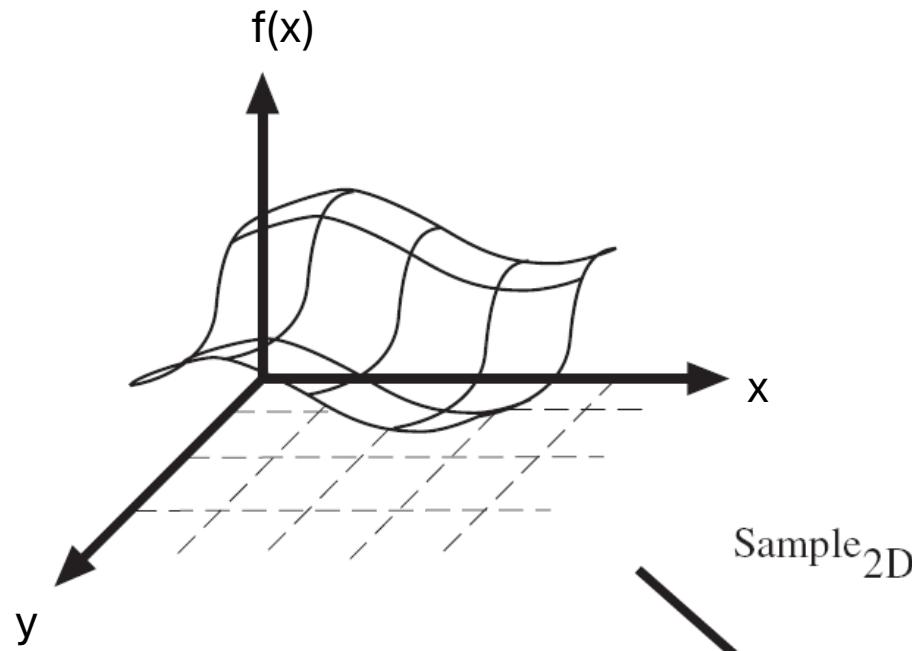
- Light
- Heat
- Gravity

Sampling

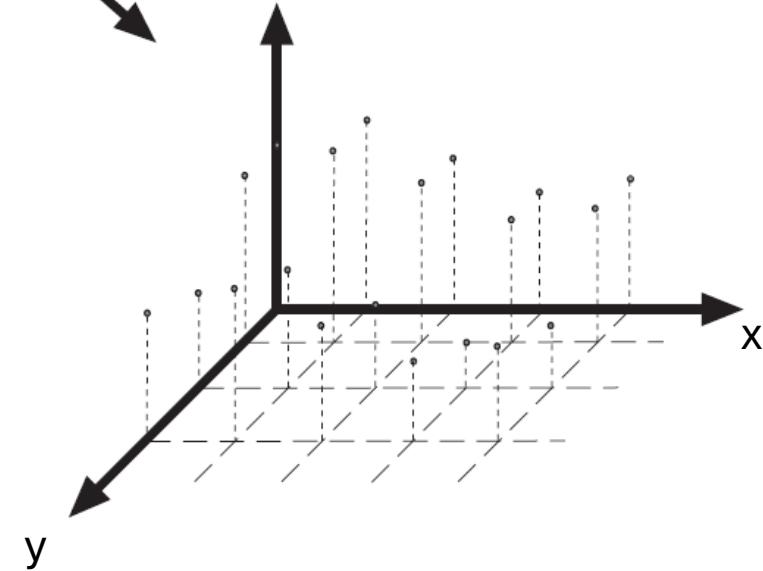
Sampling in 1D takes a function and returns a vector whose elements are values of that function at the sample points.



Sampling



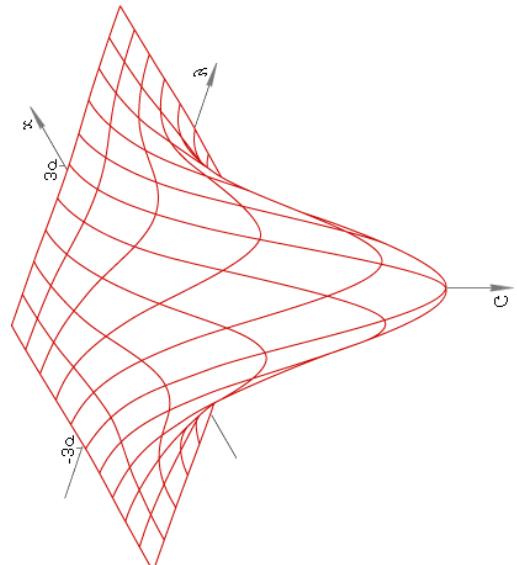
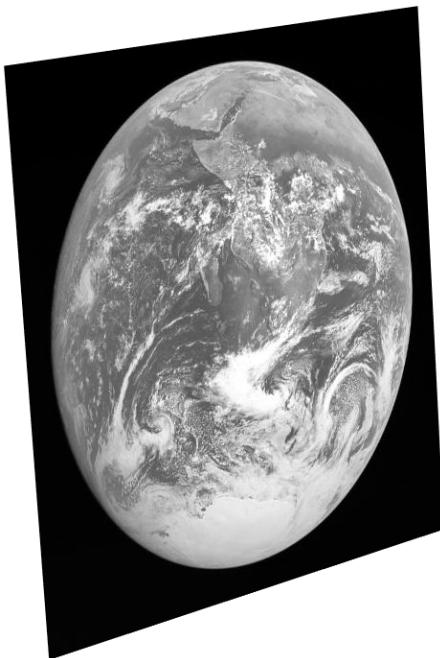
Sampling in 2D takes a function and returns a matrix.



2D Image

Definition

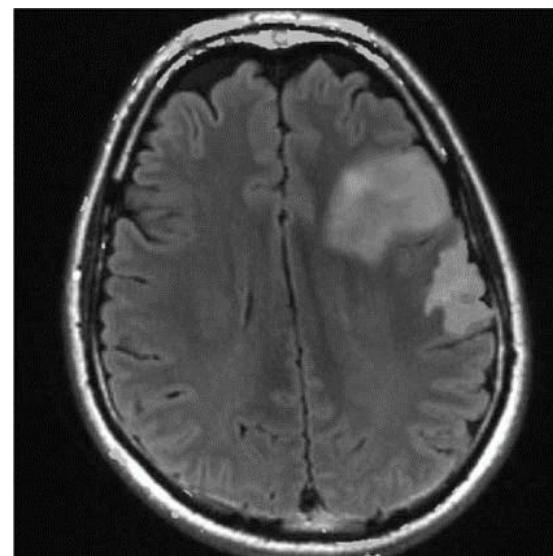
A sampling of a function that contains information about a 2D* signal.



- 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

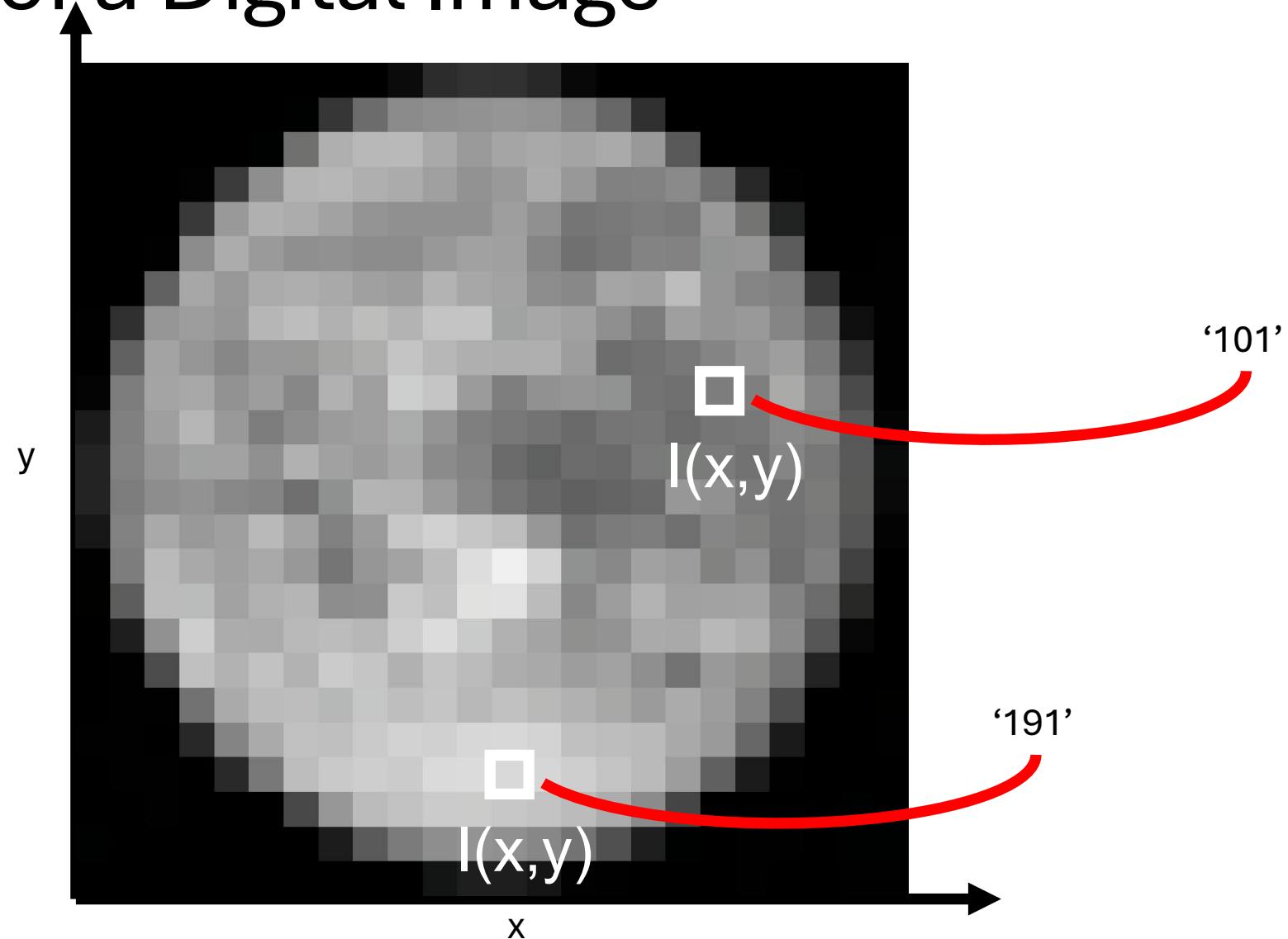
* or a 2D projection of a multi-dimensional signal

Example 2D Images



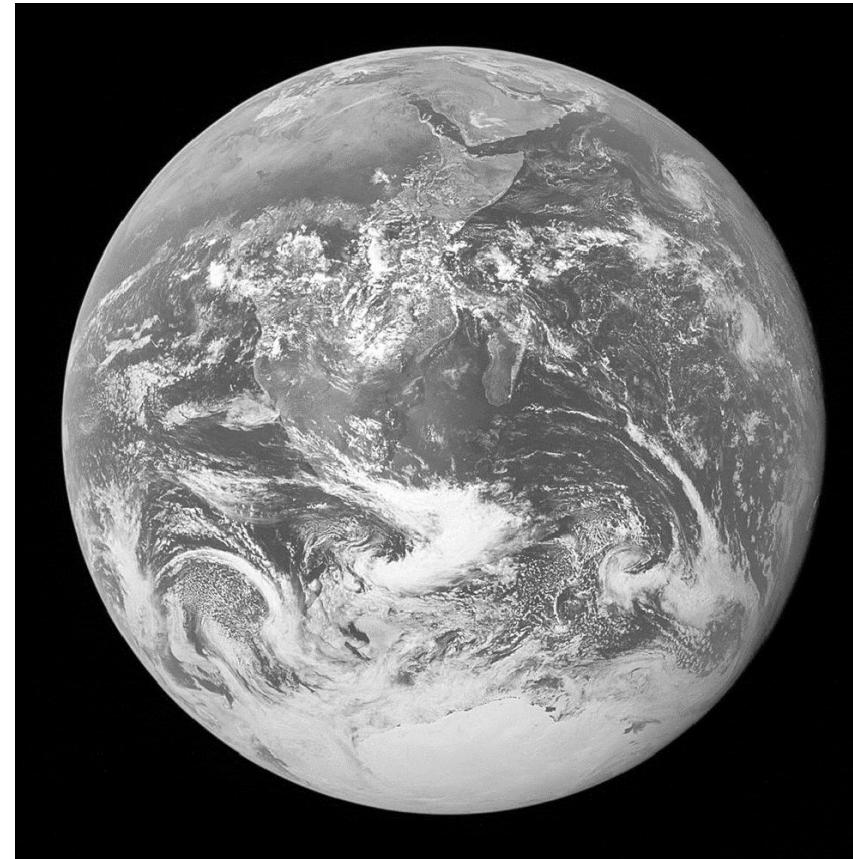
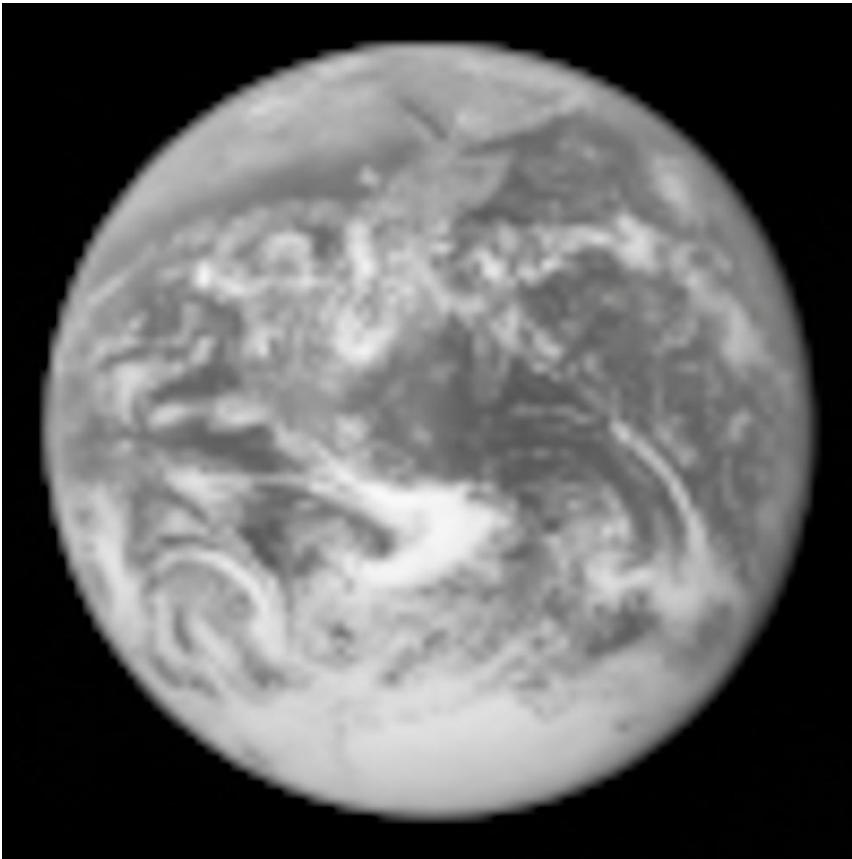
Elements of a Digital Image

Pixel: picture element

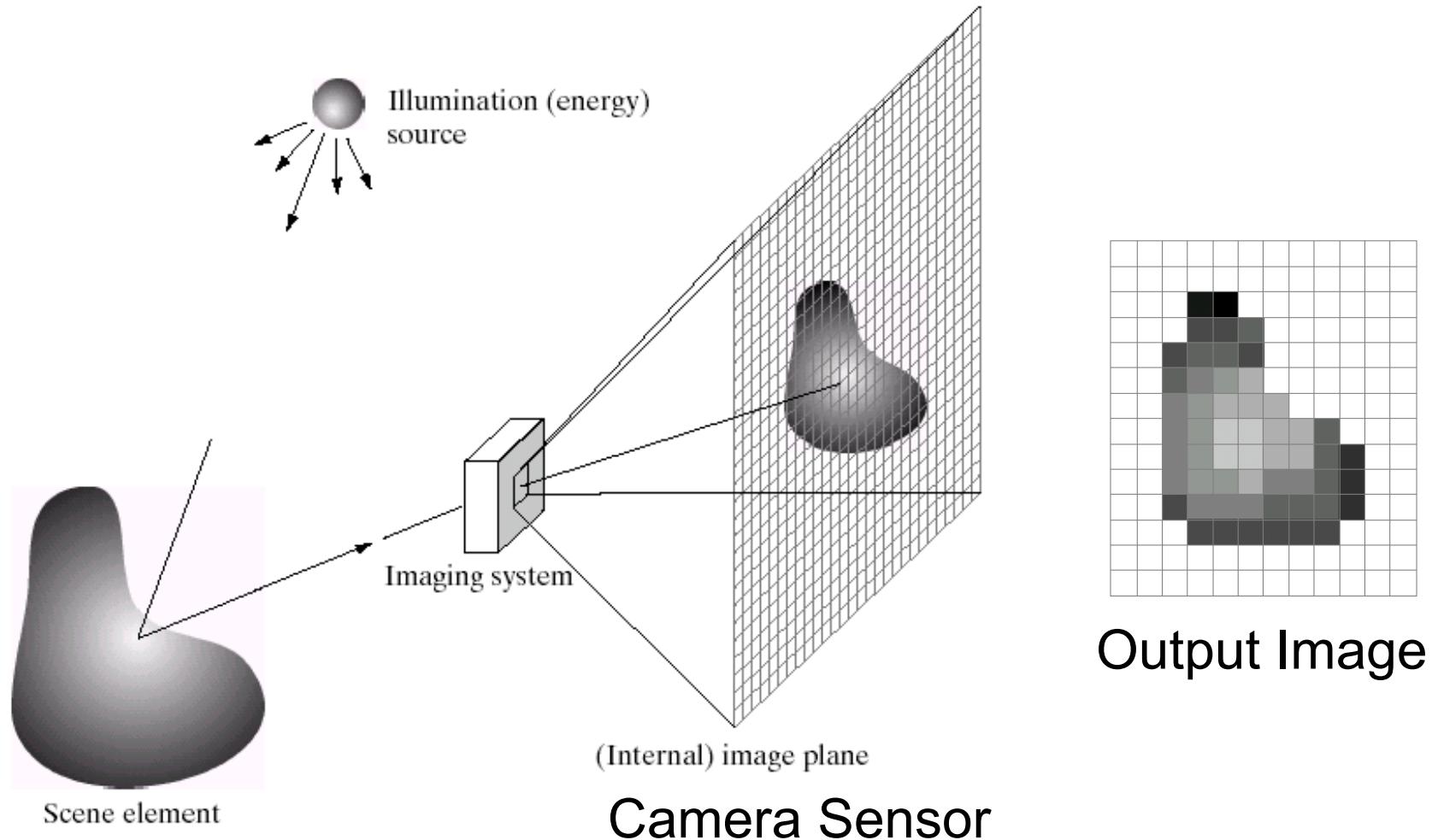


Resolution: geometric vs. spatial

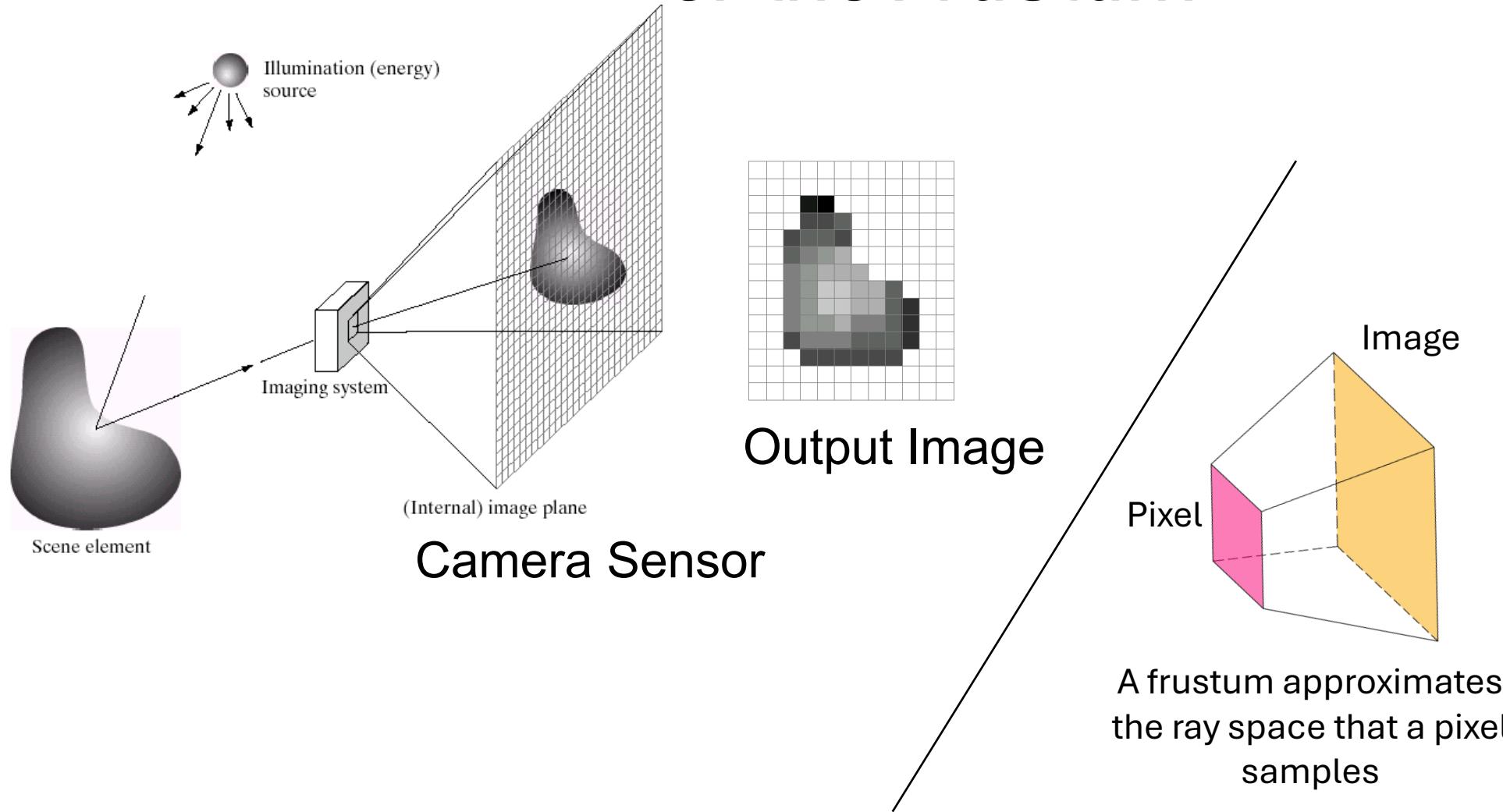
Both images are 1000x1000 pixels



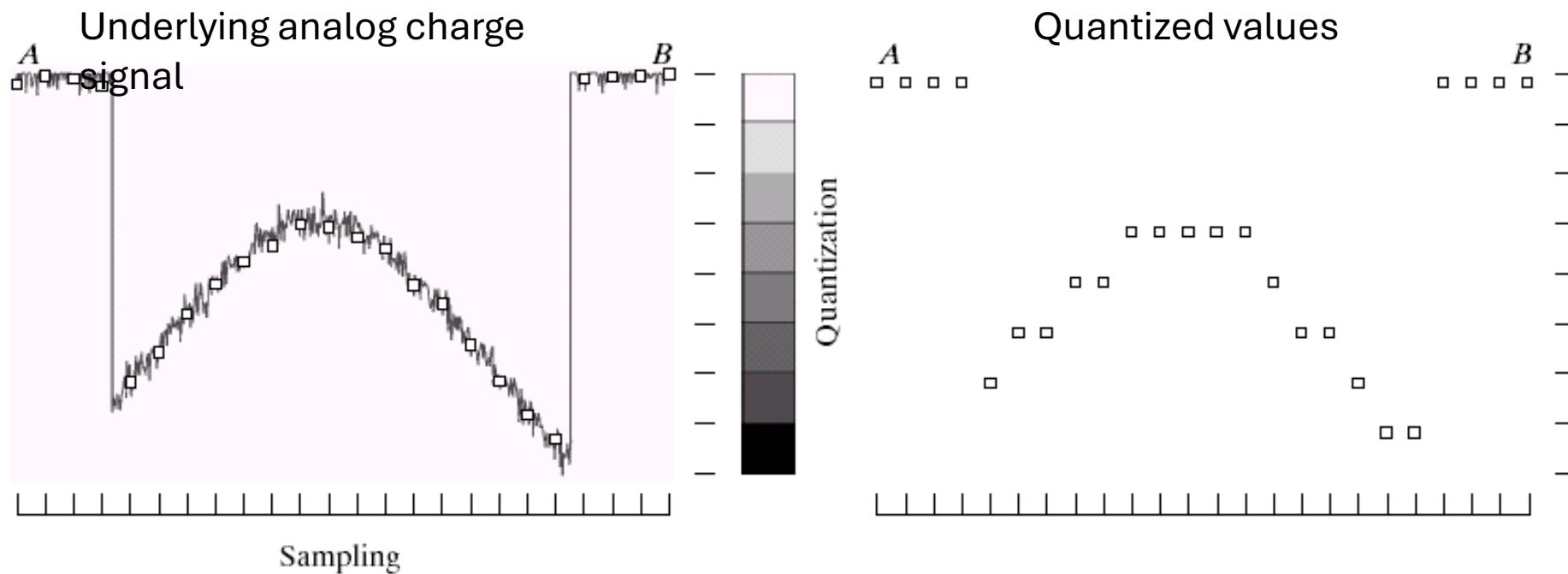
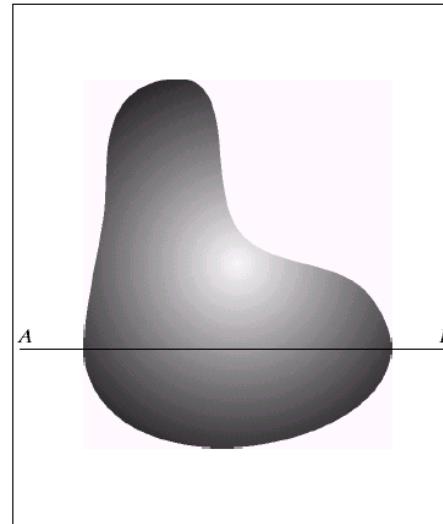
Light Integration Over the *Frustum*



Light Integration Over the *Frustum*



Quantization



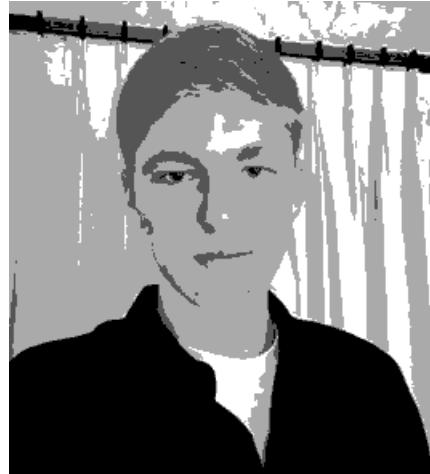
Quantization Effects – Radiometric Resolution



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

- $= 256 \text{ values}^{\wedge} (1000 \times 1000)$



Input
 256^{10^6}

Decimal approximation

$9.2323412683466475285638791370210766478370544416421428078\dots \times 10^{2408239}$

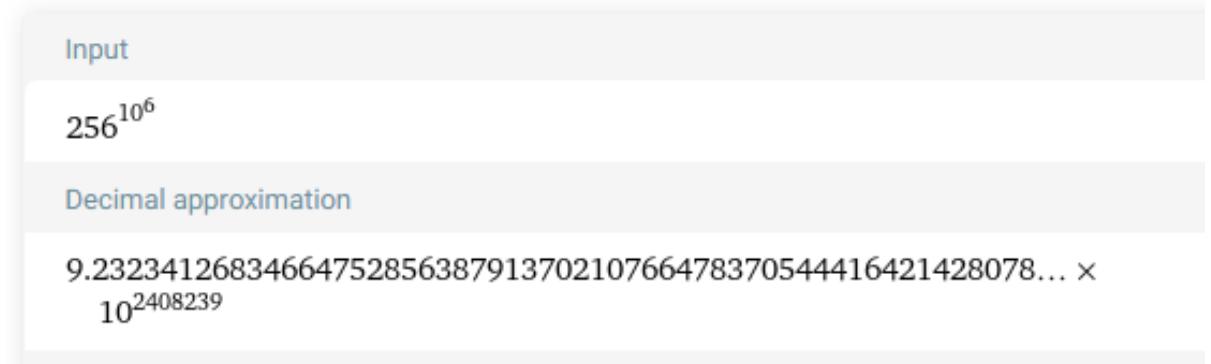
Dimensionality of an Image

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

- $= 256 \text{ values}^{\wedge} (1000 \times 1000)$



All scenes ever depicted



A screenshot of the WolframAlpha interface. The input field shows 256^{10^6} . Below it, a "Decimal approximation" section displays a very long decimal number starting with 9.2323412683466475285638791370210766478370544416421428078... \times 10^{2408239}.

Dimensionality of an Image

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

- $= 256^{\text{values}} \wedge (1000 \times 1000)$



All scenes ever depicted

- In size 1000x1000

A screenshot of the WolframAlpha interface. The input field shows 256^{10^6} . Below it, under "Decimal approximation", is the result: $9.2323412683466475285638791370210766478370544416421428078\dots \times 10^{2408239}$.

Dimensionality of an Image

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

- $= 256 \text{ values}^{\wedge} (1000 \times 1000)$



All scenes ever depicted

- In size 1000x1000

A screenshot of the WolframAlpha interface. The input field shows "256^{10⁶}". Below it, a "Decimal approximation" section shows the result as "9.2323412683466475285638791370210766478370544416421428078... × 10²⁴⁰⁸²³⁹".

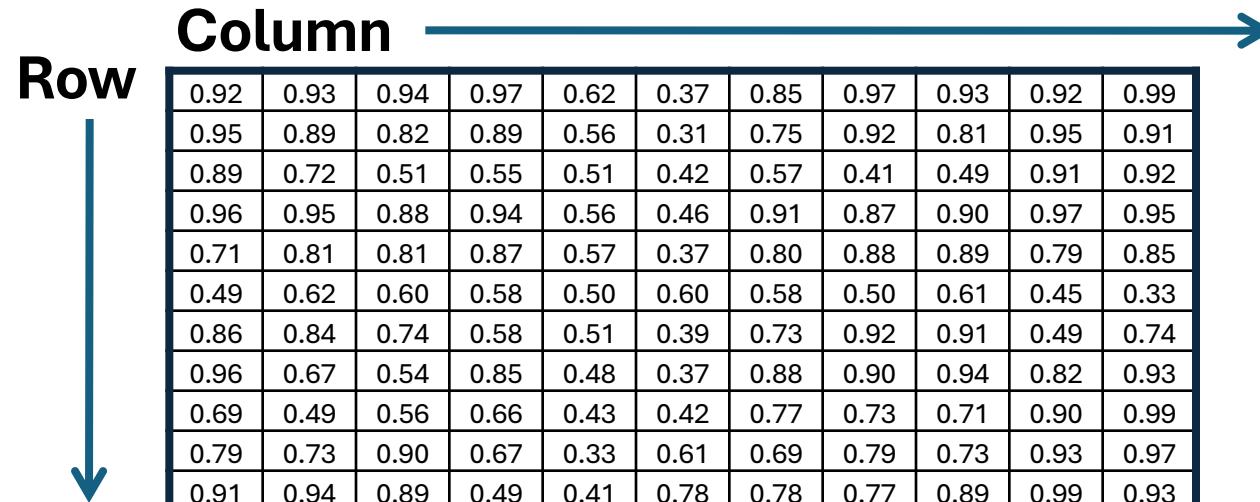
Computer vision as making sense of multi-dimensional data.

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

Images in Python (import numpy)

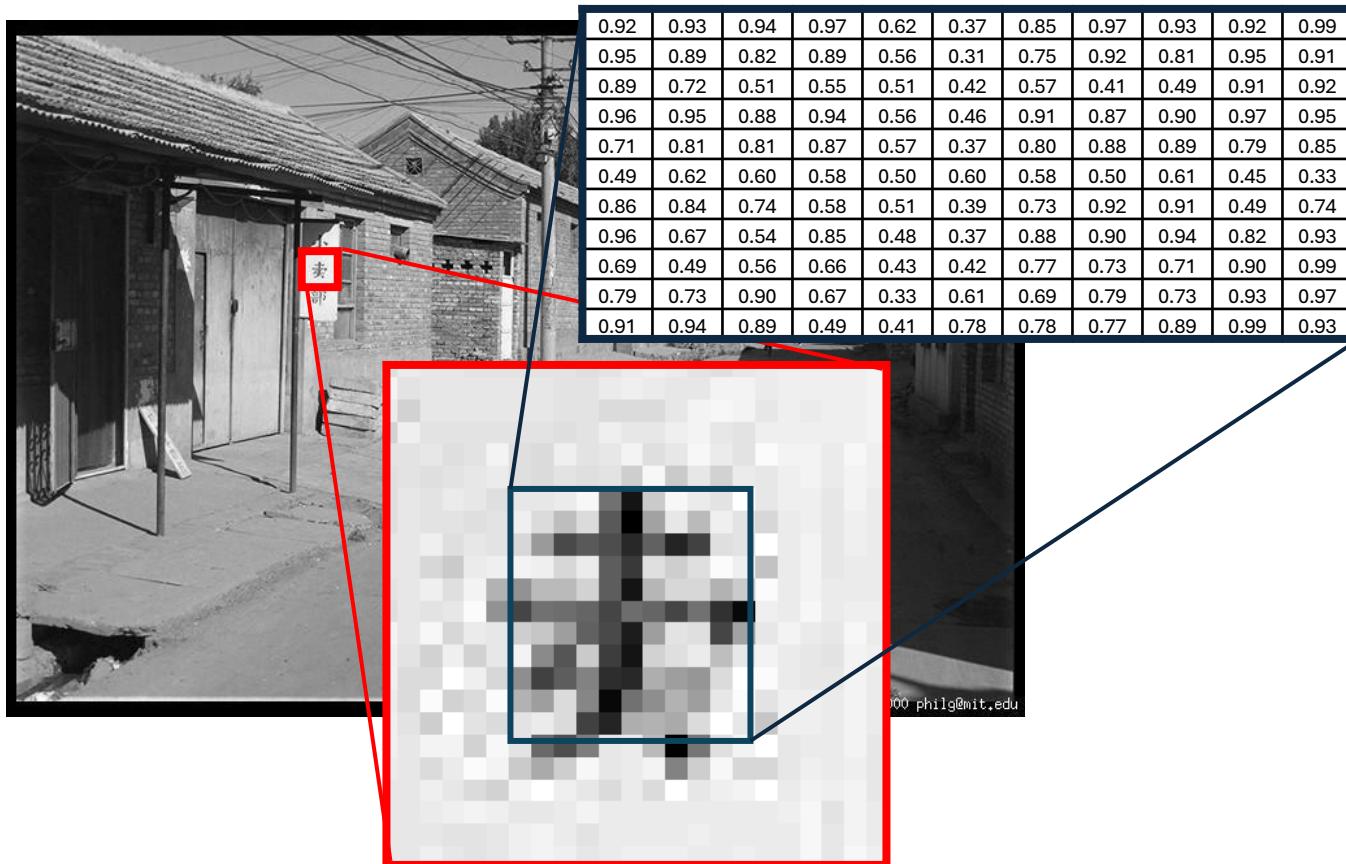
$N \times M$ grayscale image “im”

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

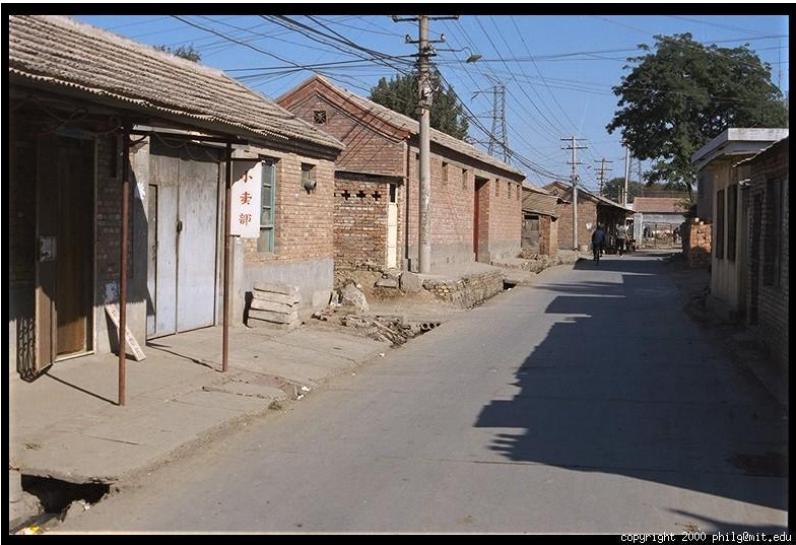


Row	Column	0.92	0.93	0.94	0.97	0.62	0.37	0.85	0.97	0.93	0.92	0.99
		0.95	0.89	0.82	0.89	0.56	0.31	0.75	0.92	0.81	0.95	0.91
		0.89	0.72	0.51	0.55	0.51	0.42	0.57	0.41	0.49	0.91	0.92
		0.96	0.95	0.88	0.94	0.56	0.46	0.91	0.87	0.90	0.97	0.95
		0.71	0.81	0.81	0.87	0.57	0.37	0.80	0.88	0.89	0.79	0.85
		0.49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

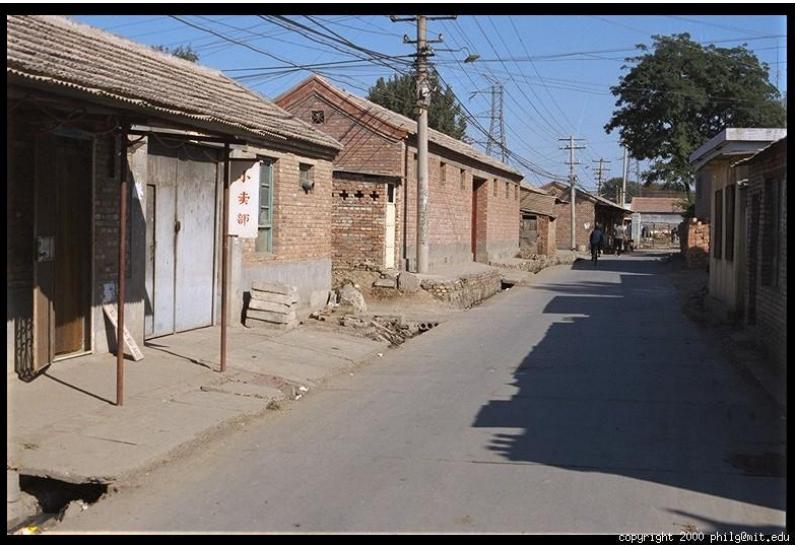
Grayscale Intensity



Color



Color



Red intensity



Green

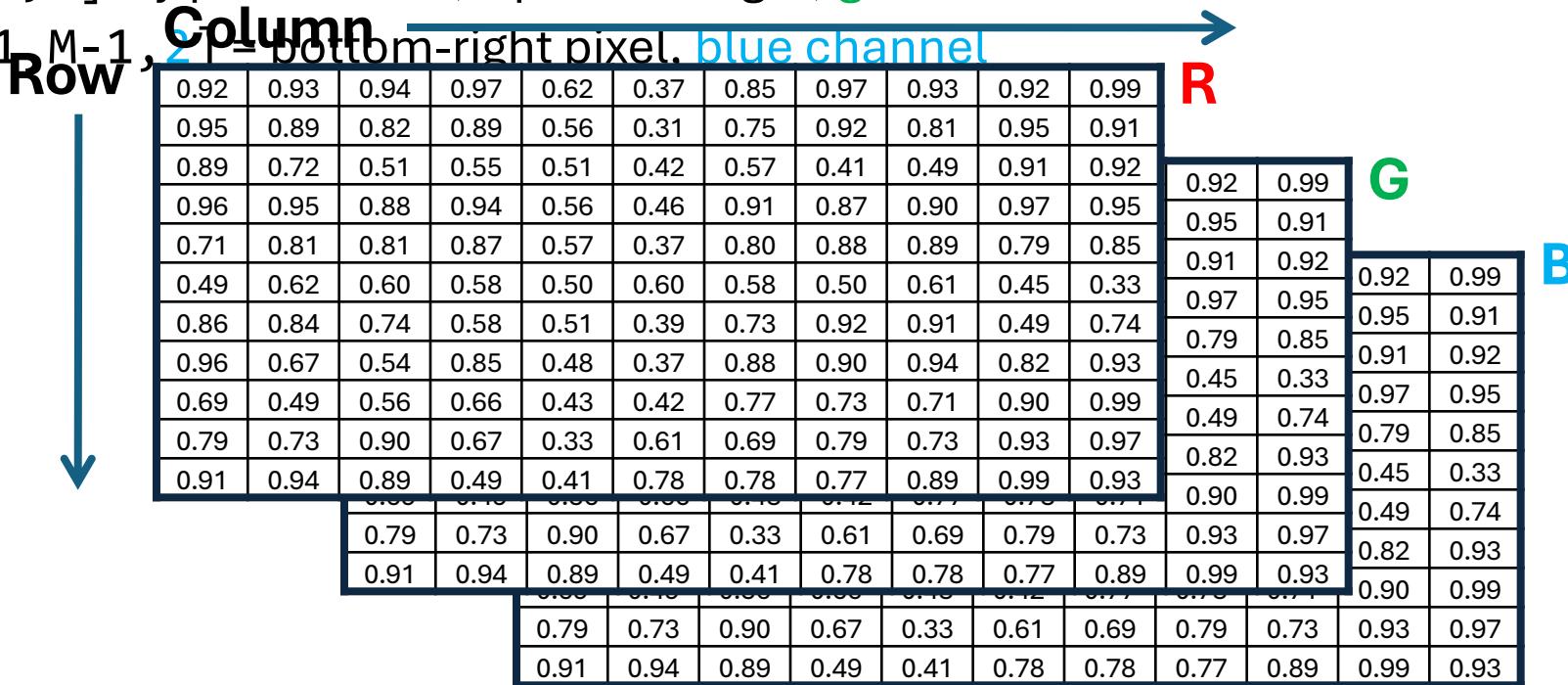


Blue

Images in Python (import numpy)

$N \times M$ grayscale image “im”

- $\text{im}[0, 0, 0]$ = top-left pixel value, red channel
 - $\text{im}[y, x, 1]$ = y pixels down, x pixels to right, green channel
 - $\text{im}[N-1, M-1, 2]$ = 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"))`

Row ↓ Column →

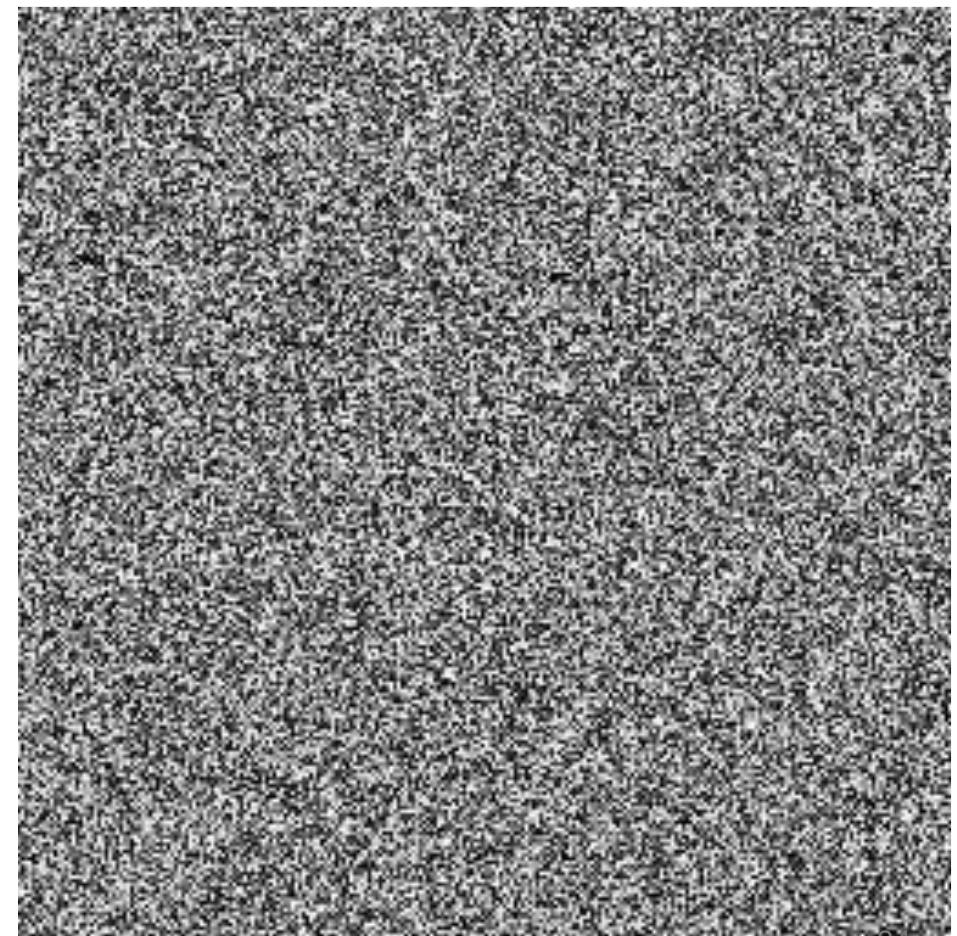
0.92	0.93	0.94	0.97	0.62	0.37	0.85	0.97	0.93	0.92	0.99
0.95	0.89	0.82	0.89	0.56	0.31	0.75	0.92	0.81	0.95	0.91
0.89	0.72	0.51	0.55	0.51	0.42	0.57	0.41	0.49	0.91	0.92
0.96	0.95	0.88	0.94	0.56	0.46	0.91	0.87	0.90	0.97	0.95
0.71	0.81	0.81	0.87	0.57	0.37	0.80	0.88	0.89	0.79	0.85
0.49	0.62	0.60	0.58	0.50	0.60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

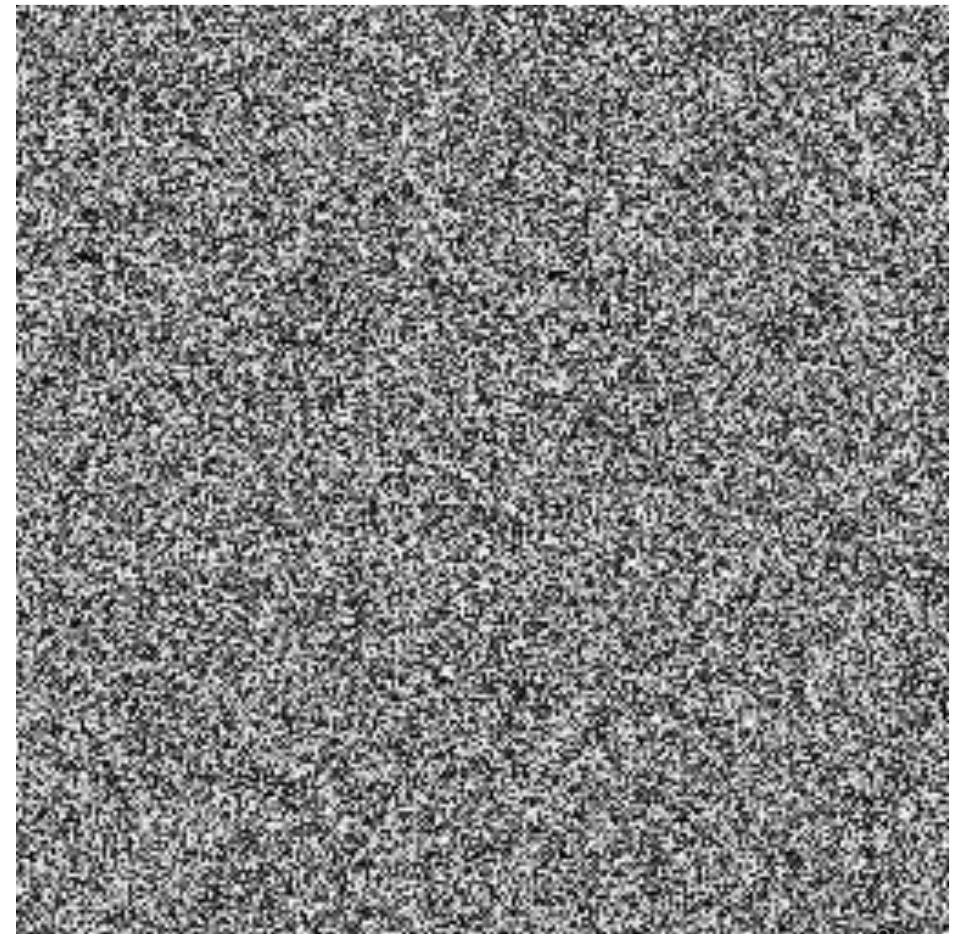
- What is this? What does it look like?
- Which values does it have?
- How many are there?

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

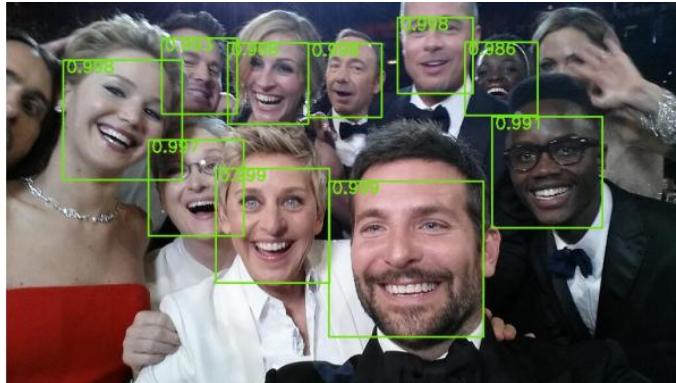


```
>>> from matplotlib import pyplot as p  
>>> I = r.rand(256,256)  
>>> p.imshow(I, cmap='gray')  
>>> p.show()
```

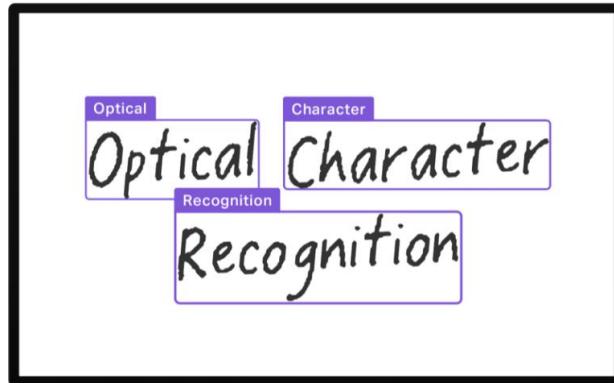
Is it an image?



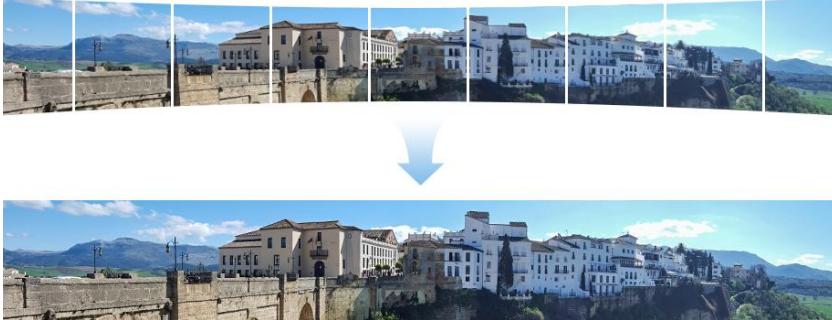
What is Computer Vision?



Face Detection



Optical Character Recognition (OCR)



Panorama Stitching



Surveillanc

e
Computer vision is much more beyond image classification and processing

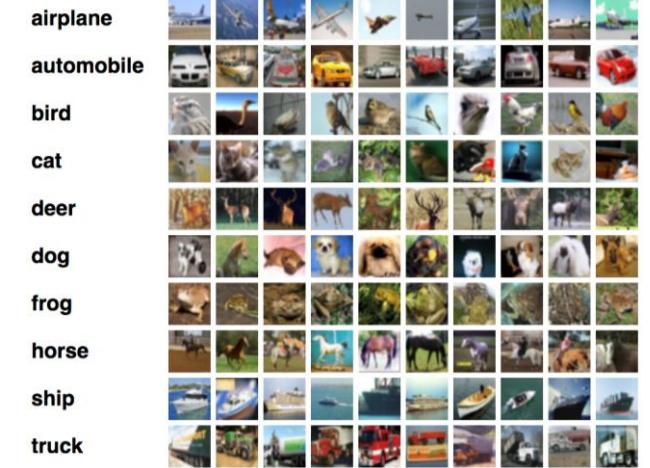
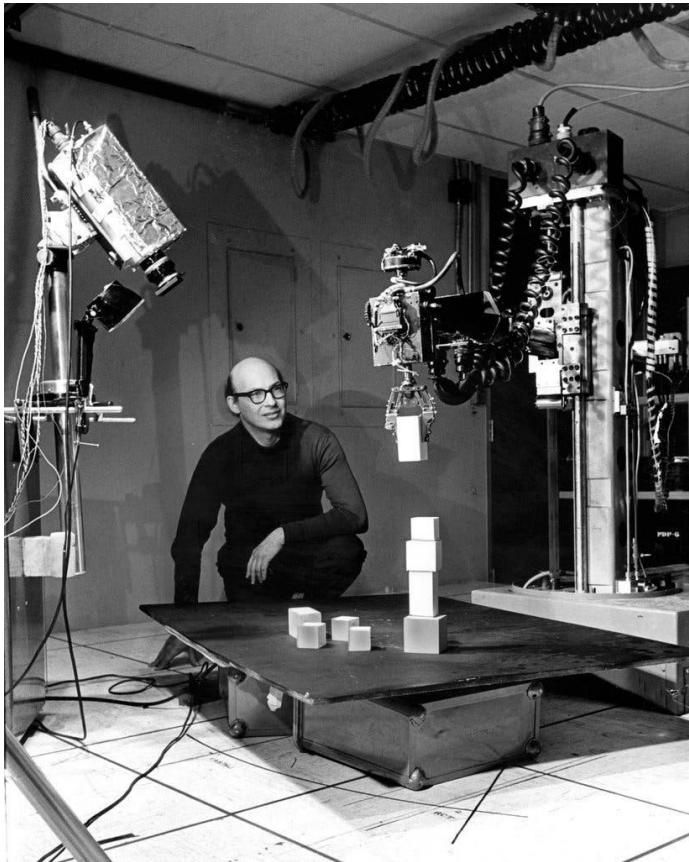


Image Classification



Semantic Segmentation

The Origin of Computer Vision



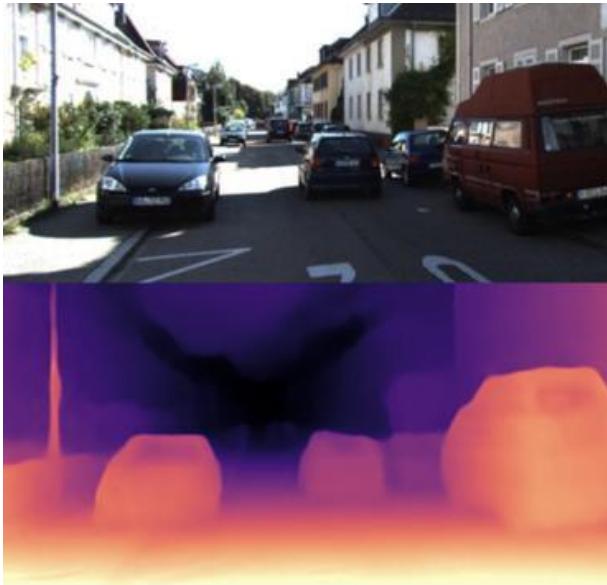
Marvin Minsky in a lab at MIT in 1968

An undergraduate project assigned by Marvin Minsky in 1966

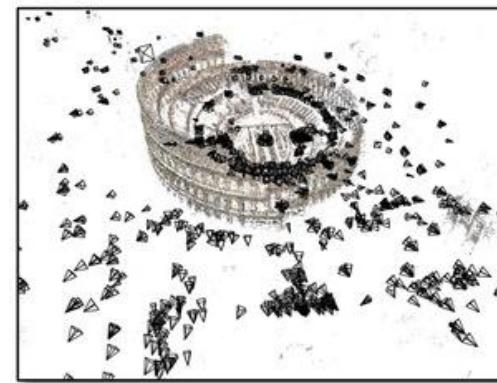
“spend the summer linking a camera to a computer and getting the computer to describe what it saw”

Understand the 3D world from 2D images like humans

What is Computer Vision?

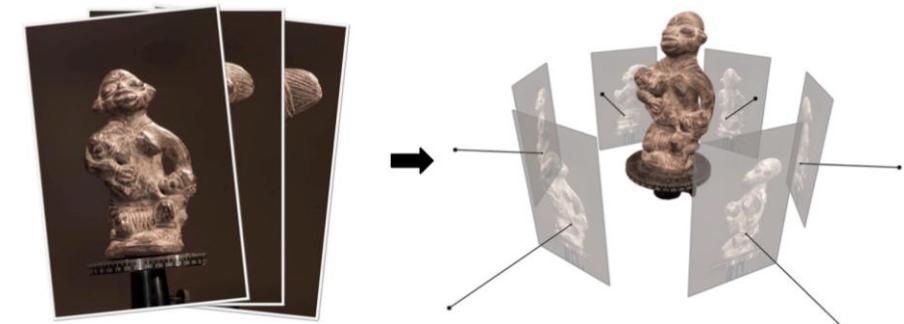


Depth Estimation

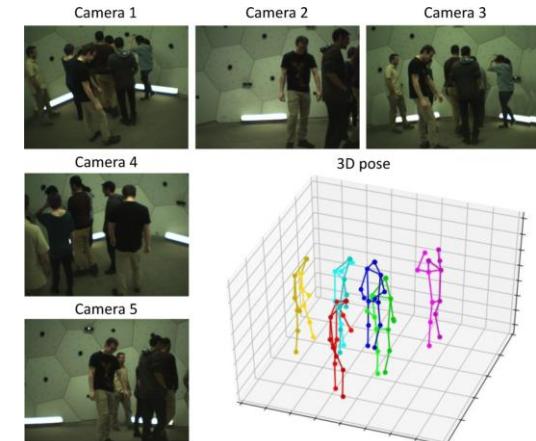


Structure from Motion

Understand the 3D world from 2D images



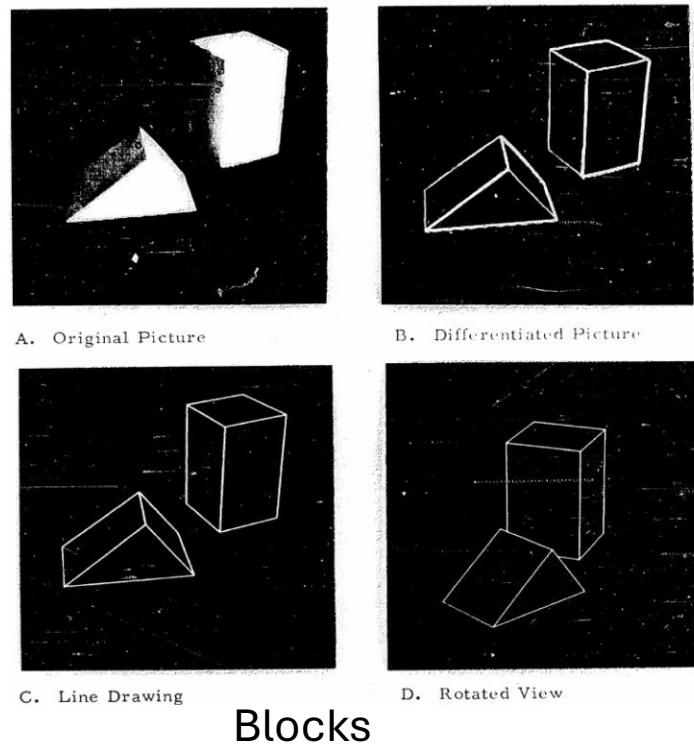
3D Reconstruction



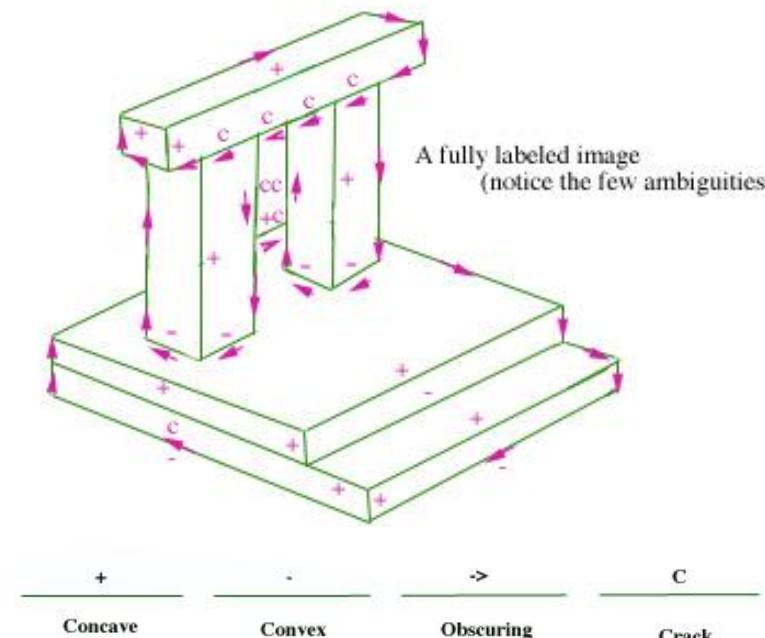
3D Human Pose Estimation
Dong et al. CVPR'19

A Brief CV History and My Chosen Milestones

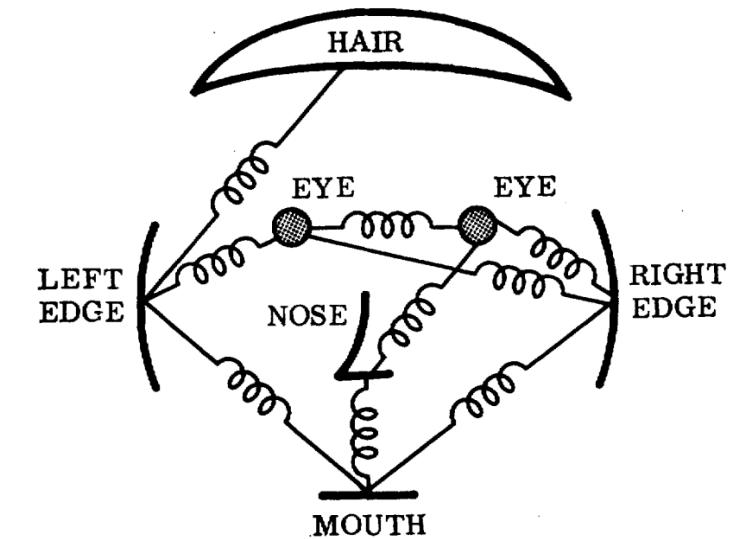
- 1970s
 - Recover 3D structure of the world from images



Roberts: Machine perception of three-dimensional solids. PhD Thesis, 1963



Line Labeling



Pictorial Structure

Fischler and Elschlager 1973

A Brief CV History and My Chosen Milestones

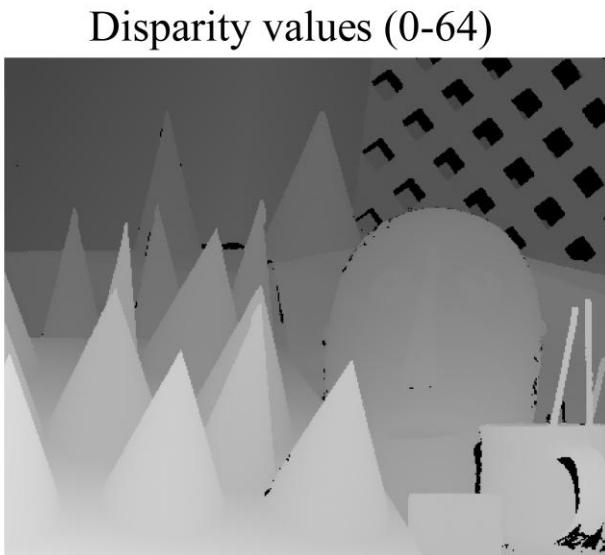
- 1980s
 - Stereo correspondence algorithms and optical flow algorithms



Left image

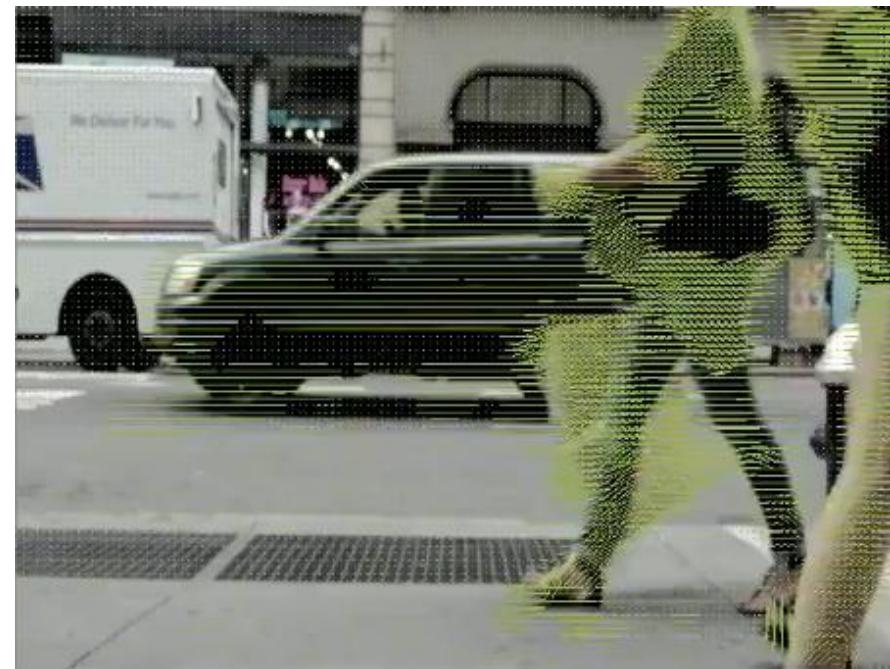


Right image



Note how disparity is larger
(brighter) for closer surfaces.

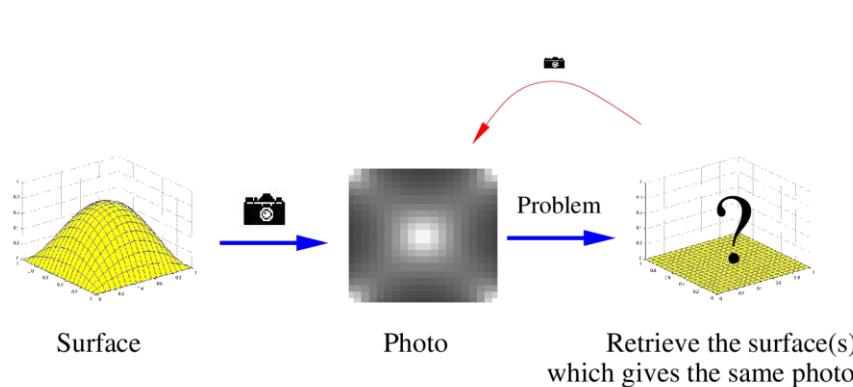
Stereo
Correspondence



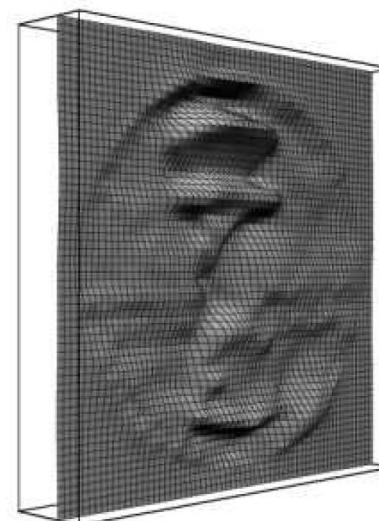
Optical Flow

A Brief CV History and My Chosen Milestones

- 1980s
 - Shape from X techniques (shape from shading, shape from texture, shape from shadows)
 - Edge and contours



Shape from shading



Freeman and Adelson
1991



Canny edge detector. Canny, 1986

A Brief CV History and My Chosen Milestones

- 1980s
 - Markov Random Fields (MRFs)

$$E(x) = \sum_i \underbrace{\Psi_i(x_i)}_{\text{Unary}} + \sum_{i \sim j} \underbrace{\Psi_{i,j}(x_i, x_j)}_{\text{Pairwise}}$$

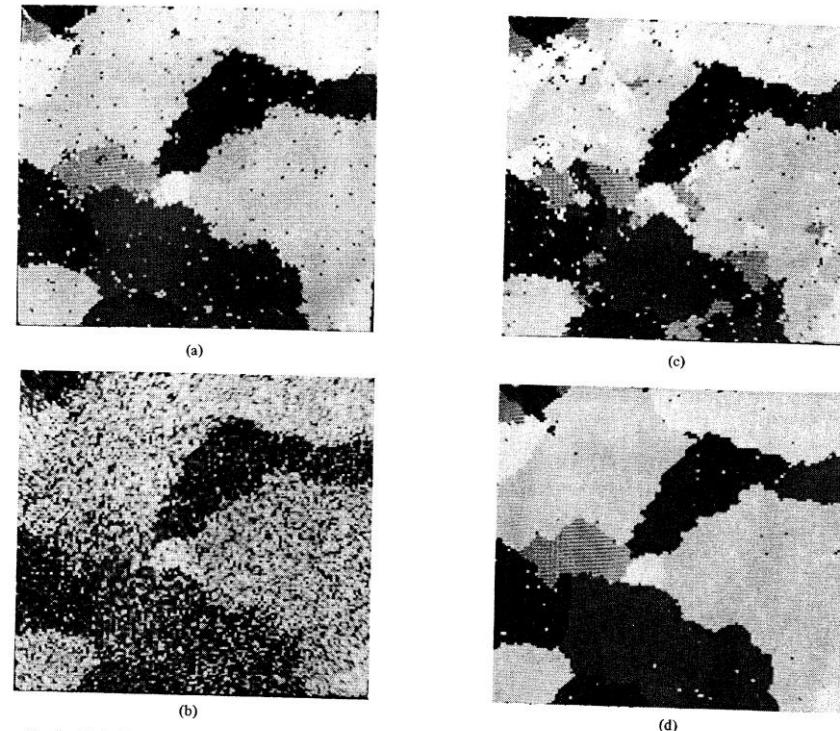
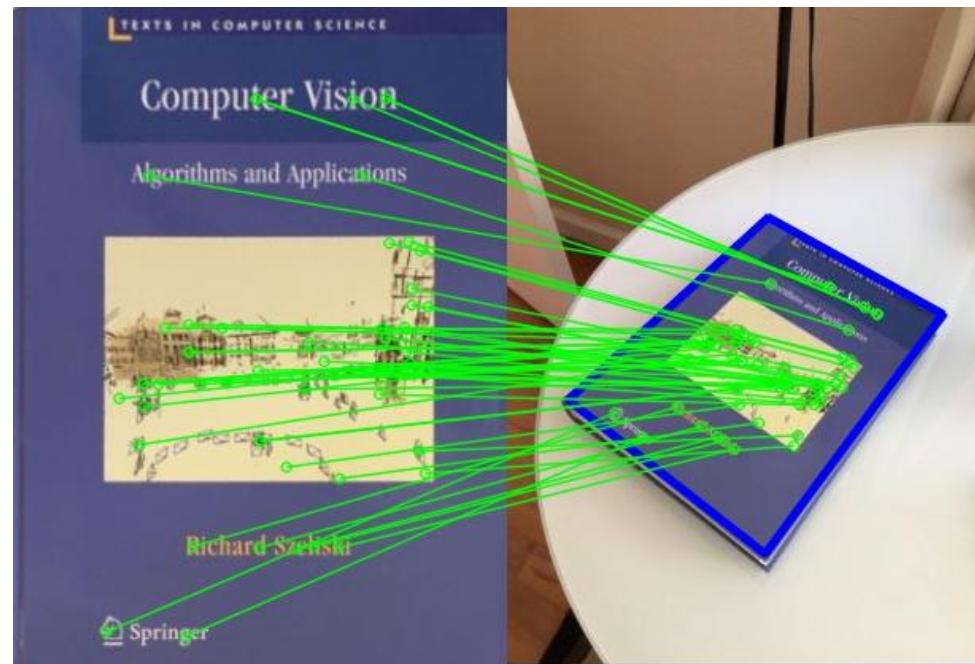


Fig. 2. (a) Original image: Sample from MRF. (b) Degraded image: Additive noise. (c) Restoration: 25 iterations. (d) Restoration: 300 iterations.

Geman and Geman: Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. PAMI, 1984

A Brief CV History and My Chosen Milestones

- 1990s
 - Structure from Motion and Multi-view Reconstruction
 - Scale Invariance Feature Transform (SIFT)



David Lowe: Object recognition from local scale-invariant features. ICCV, 1999.

A Brief CV History and My Chosen Milestones

- 1990s
 - Statistical learning techniques started appearing

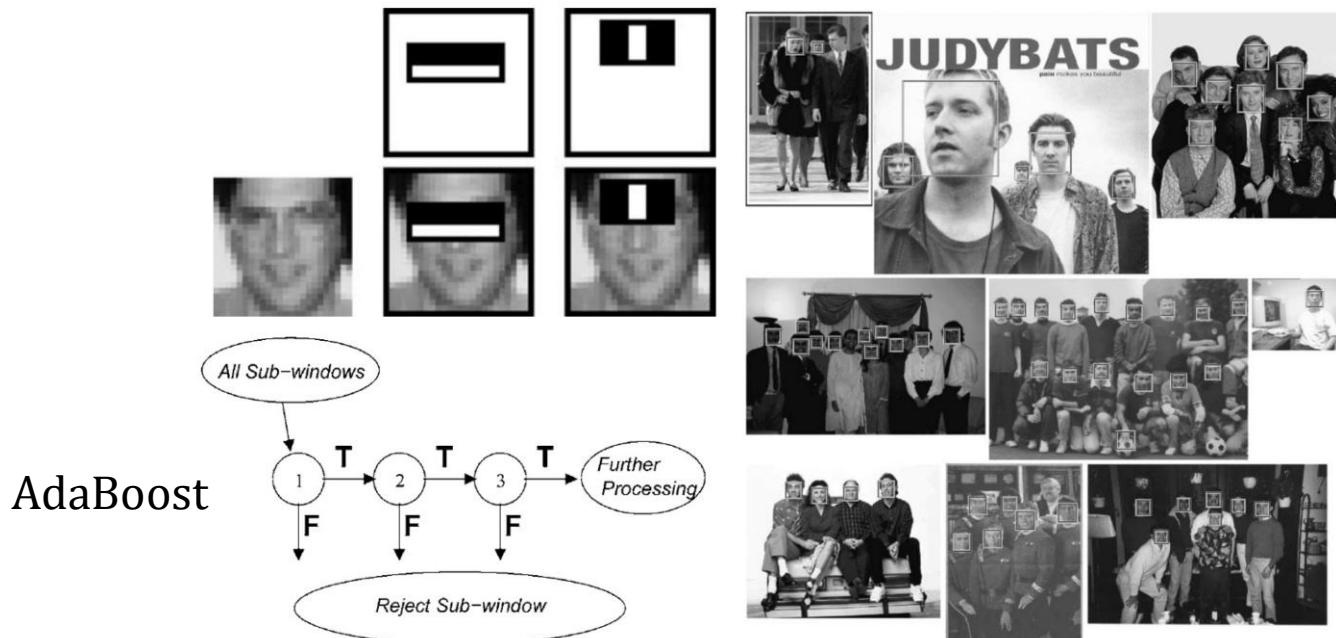


Eigenfaces

Turk and Pentland: Face recognition using Eigenfaces. CVPR, 1991

A Brief CV History and My Chosen Milestones

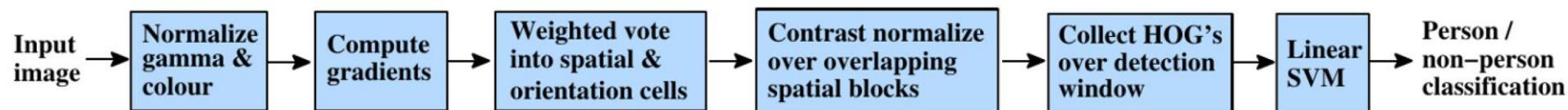
- 2000s
 - Data-driven and learning approaches
 - Cascaded classifiers for object detection



Viola and Jones: Robust Real-time Object Detection. IJCV, 2001.

A Brief CV History and My Chosen Milestones

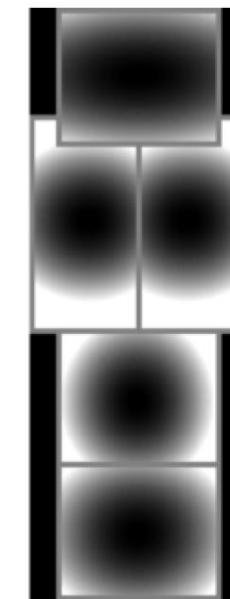
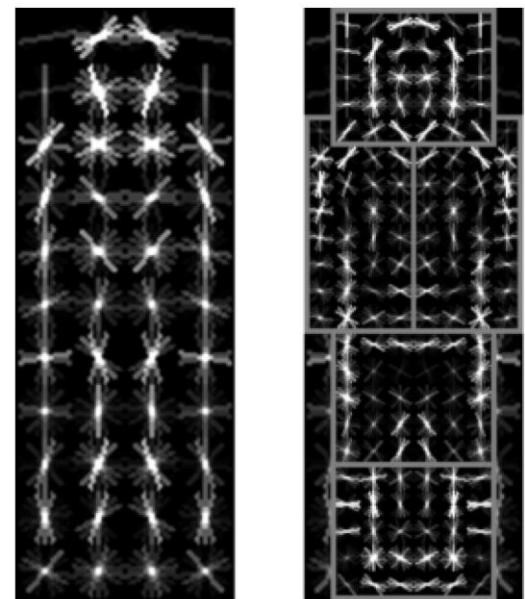
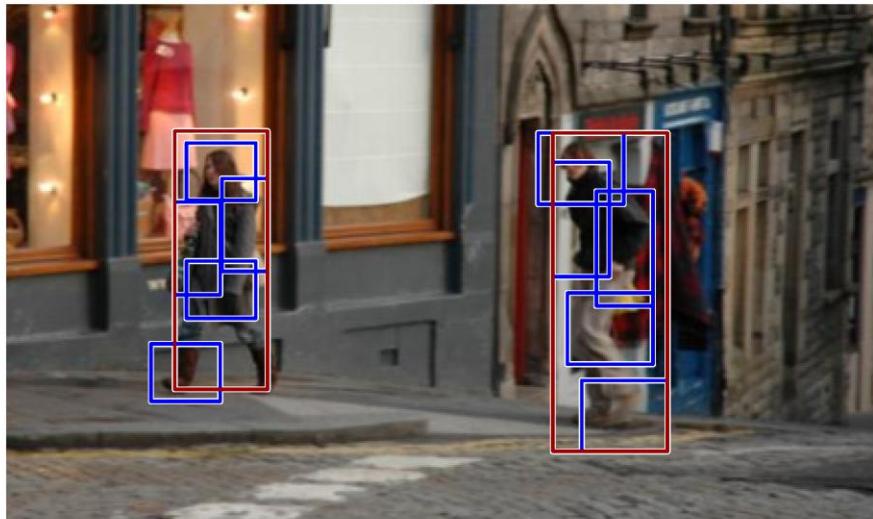
- 2000s
 - Histogram of Oriented Gradients for object detection



Dalal and Triggs: Histograms of Oriented Gradients for Human Detection. CVPR, 2005.

A Brief CV History and My Chosen Milestones

- 2000s
 - Deformable parts models for object detection



Felzenszwalb et al. Object detection with discriminatively trained part-based models . TPAMI, 2009.

A Brief CV History and My Chosen Milestones

- 2000s
 - Datasets



The PASCAL Visual Object Classes Challenge 2007



PASCAL VOC, Everingham et al., 2005 - 2012



ImageNet, Deng et al., 2009

A Brief CV History and My Chosen Milestones

- 2000s
 - Large-scale structure from motion

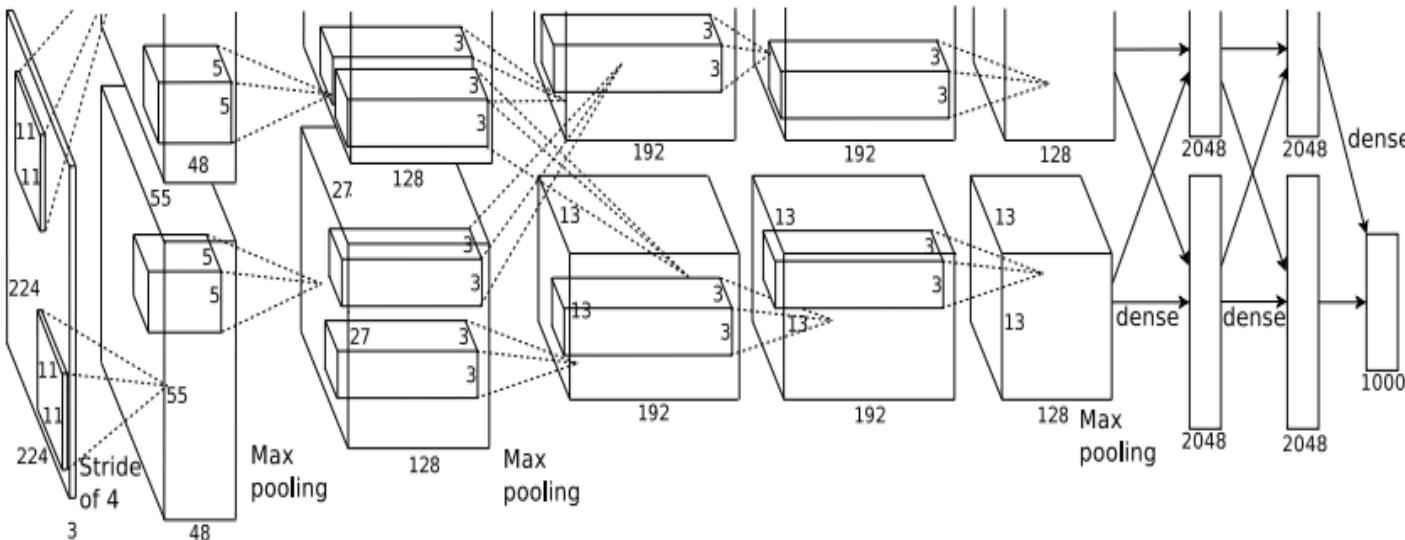


San Marco Square: 13,699 images, 4,515,157 points

Agarwal et al. Building Rome in day. ICCV, 2009.

A Brief CV History and My Chosen Milestones

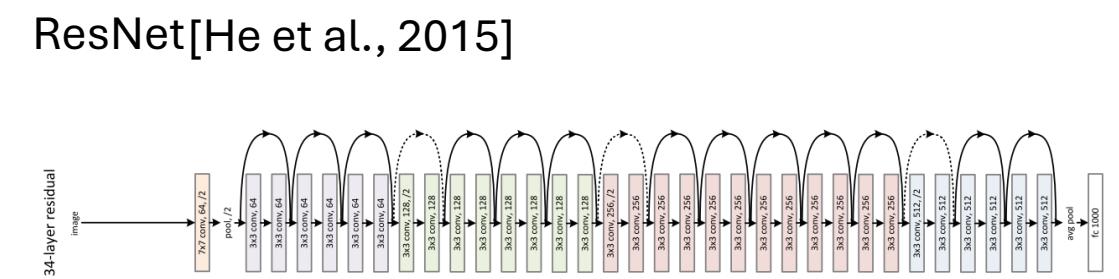
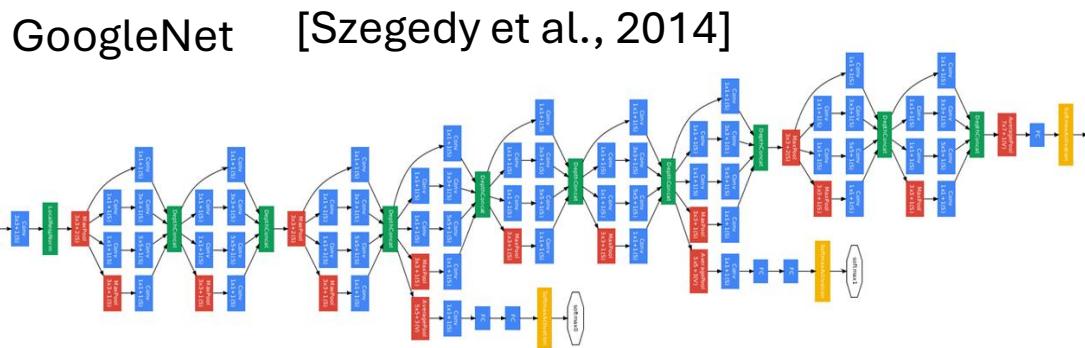
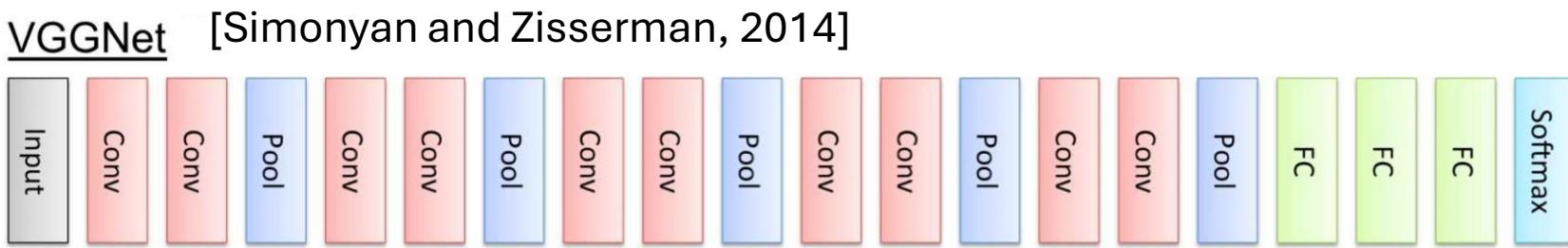
- 2010s
 - Deep Learning in CV



AlexNet. Krizhevsky et al., 2012, designed for ImageNet classification

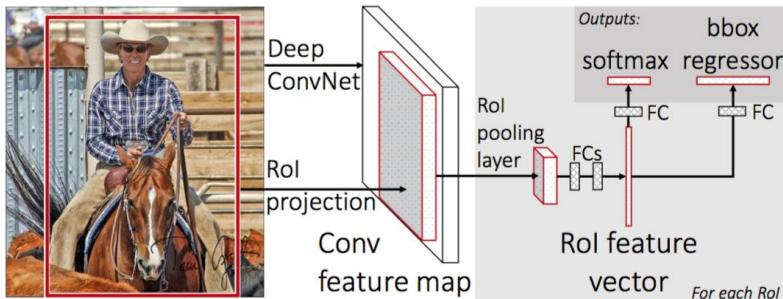
A Brief CV History and My Chosen Milestones

- 2010s
 - Deeper and wider networks



A Brief CV History and My Chosen Milestones

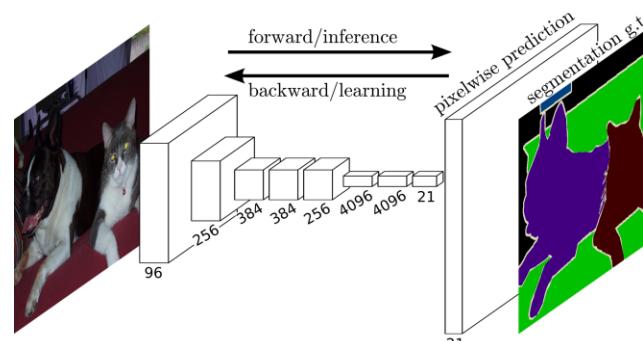
- 2010s
 - Neural networks for recognition



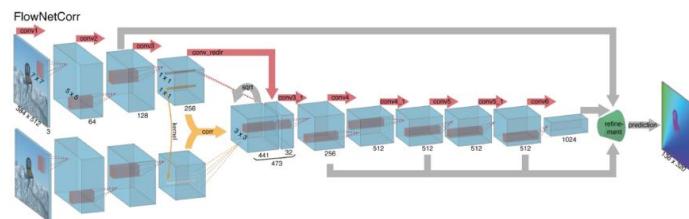
Object Detection (Fast RCNN, Girshick, 2015)



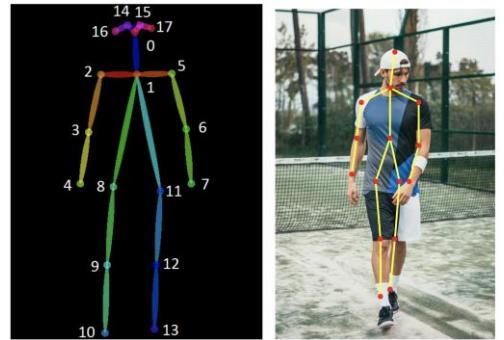
Depth Estimation (Eigen et al. 2014)



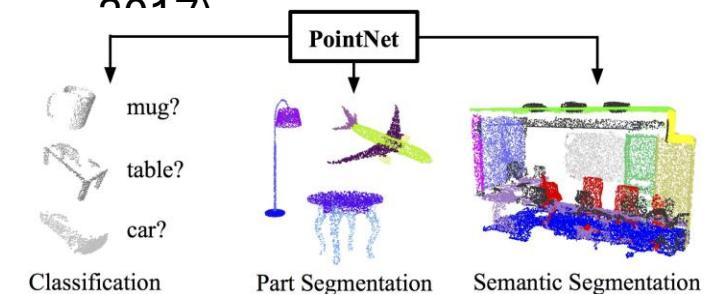
Semantic Segmentation
(FCN, Long et al., 2014)



Optical Flow
(FlowNet Fischer et al.
2015)



Human Pose Estimation
(OpenPose, Cao et al.,
2017)



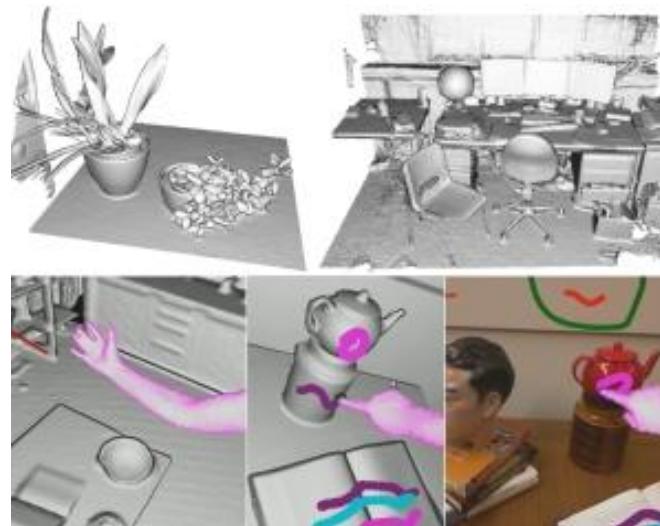
Point Cloud Recognition
(PoinetNet, Qi et al., 2016)

A Brief CV History and My Chosen Milestones

- 2010s
 - Depth sensing and 3D vision



Microsoft Kinect, 2010



KinectFusion, Newcombe et al., 2011



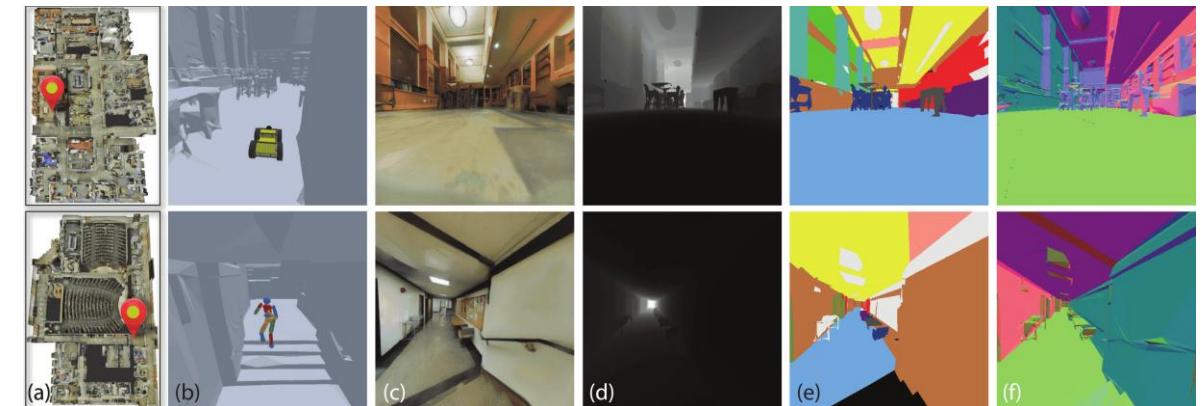
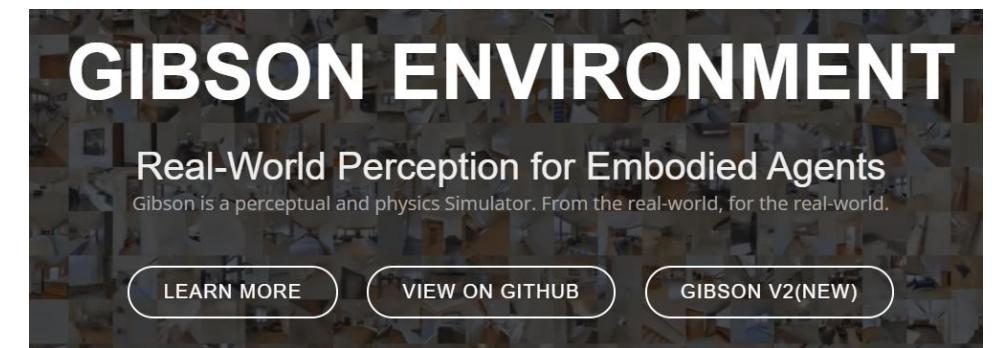
DynamicFusion, Newcombe et al., 2015

A Brief CV History and My Chosen Milestones

- 2010s
 - Autonomous driving and embodied AI



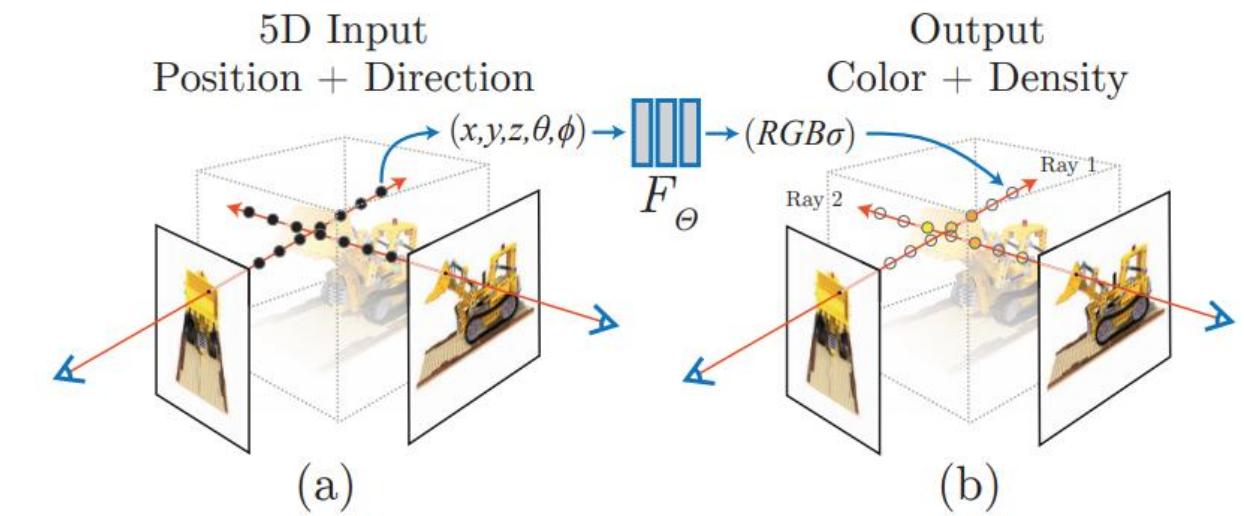
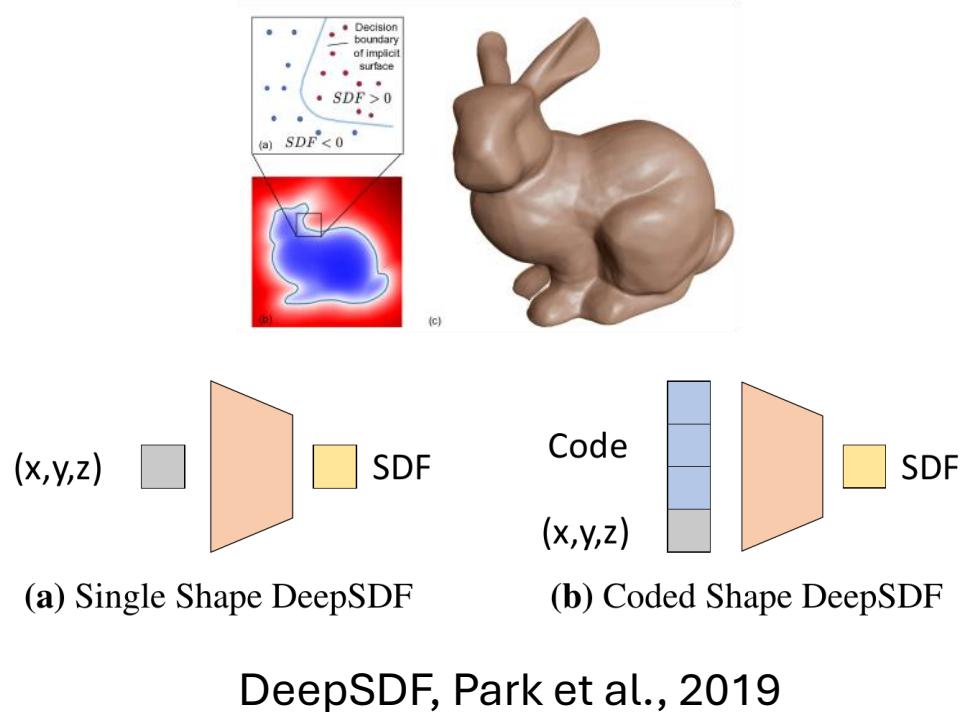
The KITTI dataset, Geiger et al., 2012



The Gibson environment, Xia et al., 2018

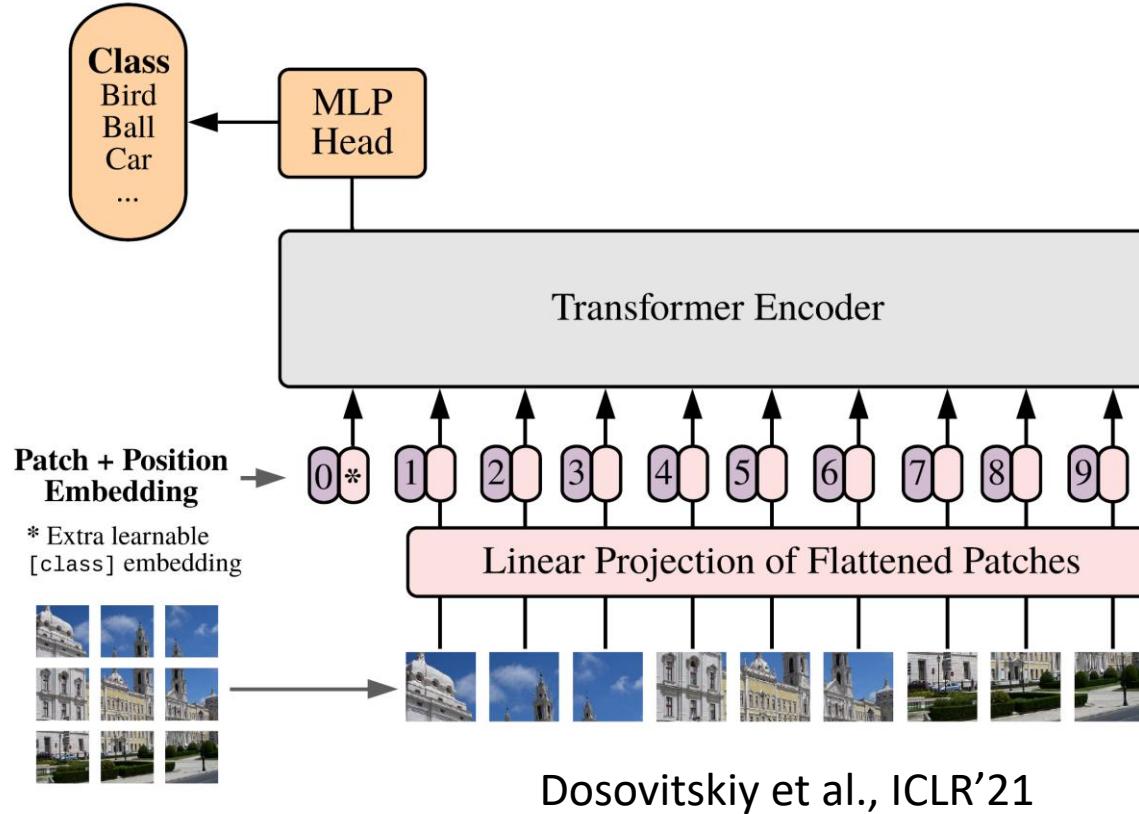
A Brief CV History and My Chosen Milestones

- 2010s
 - Neural implicit representations

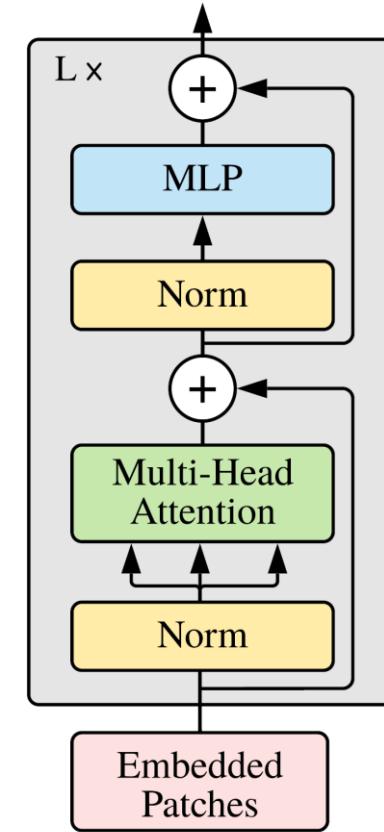


A Brief CV History and My Chosen Milestones

- 2020s



Transformer Encoder



A Brief CV History and My Chosen Milestones

- 2023
 - Large Multi-modal Model
 - GPT-4, etc

User What is funny about this image? Describe it panel by panel.



The image is a composite of three panels. Panel 1 shows a black smartphone lying on a light-colored surface, connected to a blue VGA cable with its white connector inserted into the phone's charging port. Panel 2 shows a product package for an "ELECOM LIGHTNING CABLE" adapter. The package is white with blue and yellow accents, featuring a picture of a VGA connector on it. Panel 3 is a close-up of the blue VGA connector, which has a small Lightning connector attached to its end.

Source: [hmmm \(Reddit\)](#)

GPT-4 The image shows a package for a "Lightning Cable" adapter with three panels.

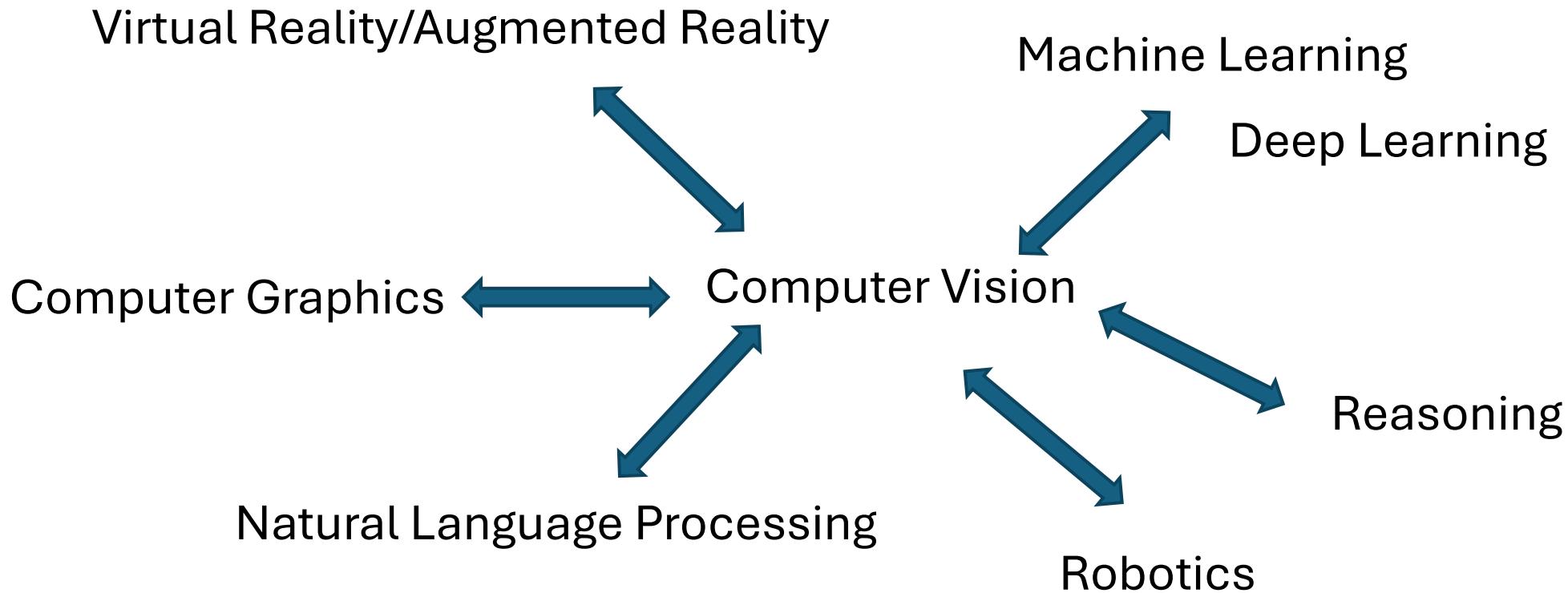
Panel 1: A smartphone with a VGA connector (a large, blue, 15-pin connector typically used for computer monitors) plugged into its charging port.

Panel 2: The package for the "Lightning Cable" adapter with a picture of a VGA connector on it.

Panel 3: A close-up of the VGA connector with a small Lightning connector (used for charging iPhones and other Apple devices) at the end.

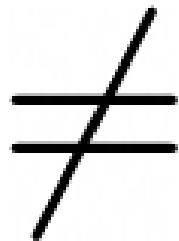
The humor in this image comes from the absurdity of plugging a large, outdated VGA connector into a small, modern smartphone charging port.

Computer Vision in AI



Computer Vision in AI

Datasets



Real World

Test your algorithms in the real world, e.g., with a camera

NEXT:

Spatial Domain Processing