Lecture 2

Pixels and Filters

Slide credit: Ranjay Krishna

Administrative

A0 is out.

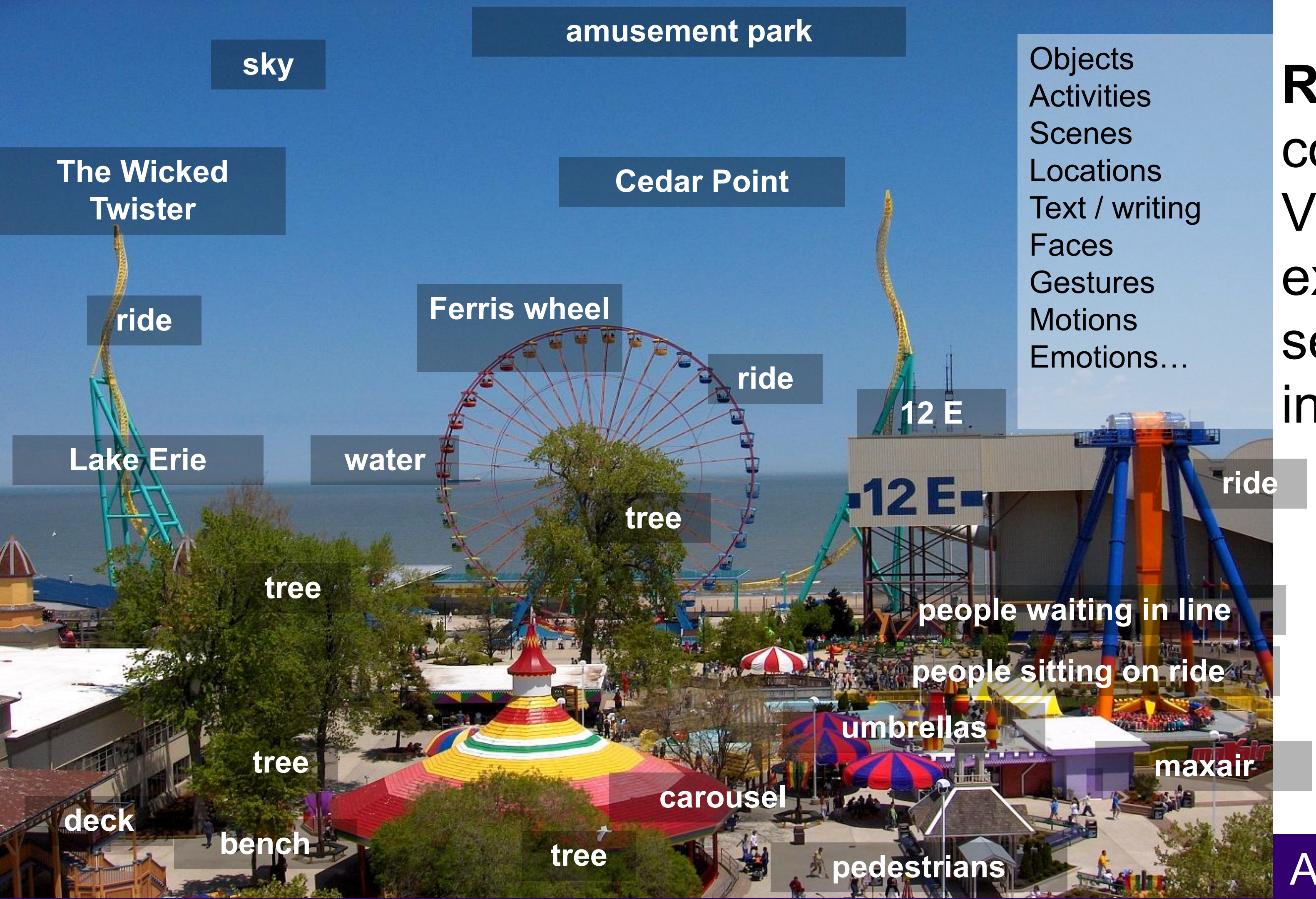
- Due on 4/7, but it is ungraded
- Meant to help you with python and numpy basics
- Learn how to do homeworks and submit them on gradescope.

Administrative

- Recording
    - Hopefully the microphone will fix the recording from today's lecture
- Section
    - Will go over Linear algebra basics this week in recitation
- TA hours
    - Start from next week

Final exam

- ~~Monday Jun 11th 10:30am - 12:20pm @ **TBD**~~
- Monday Jun 9th 2:30 - 4:20 (in person) @ BAG 154
  - We will send out form for students to apply to take the make up
- Will contain written questions from the concept covered in class or any questions in the homeworks.
- Can require you to solve technical math problems.
- Will contain a lot of multiple choice and true-false questions. We will release a practice final towards the end of the quarter.

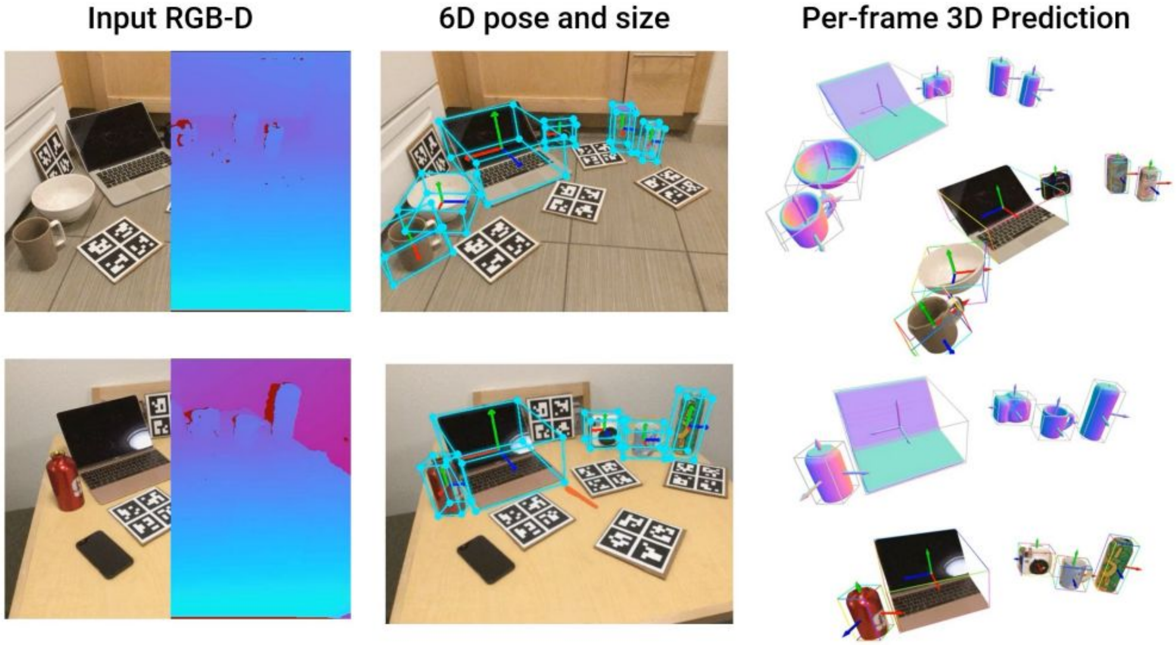


Objects
Activities
Scenes
Locations
Text / writing
Faces
Gestures
Motions
Emotions…

Recap: computer Vision extracts semantic information

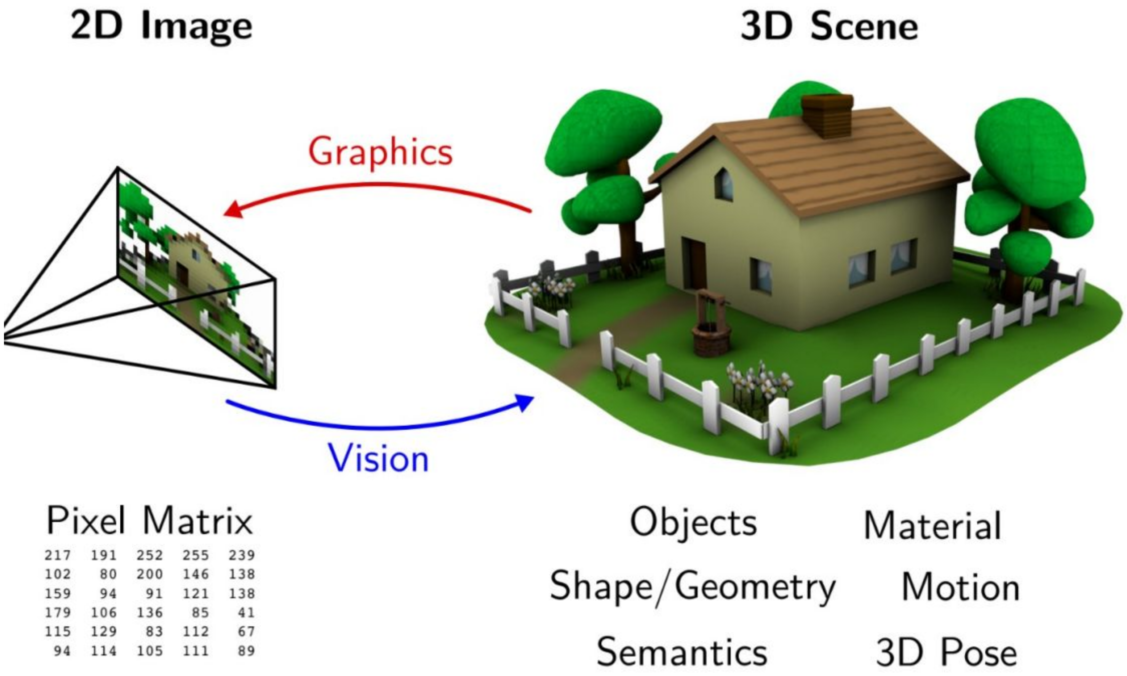
April 02, 2025

Recap: Computer vision extracts geometric 3D information from 2D images



TRI & GATech’s ShaPO (ECCV’22): https://zubair-irshad.github.io/projects/ShAPO.html

So far: why is computer vision hard?



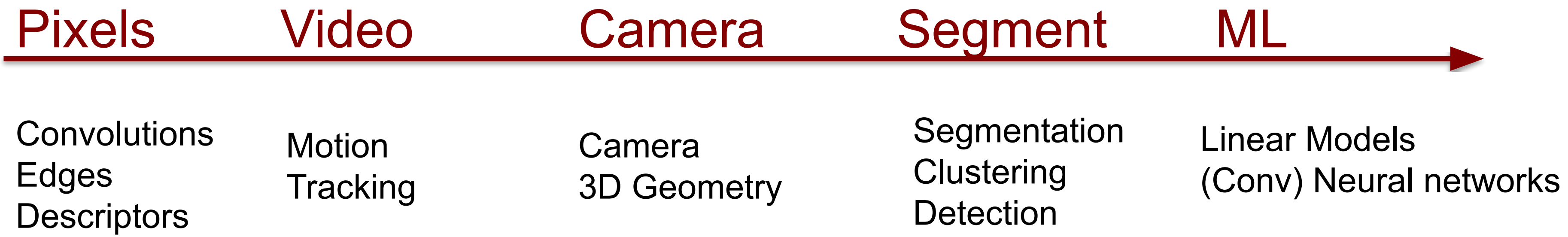
Pixel Matrix

217	191	252	255	239
102	80	200	146	138
159	94	91	121	138
179	106	136	85	41
115	129	83	112	67
94	114	105	111	89

Objects Material
Shape/Geometry Motion
Semantics 3D Pose

It is an ill posed problem

CSE 455 Roadmap



Pixels

Convolutions
Edges
Descriptors

Video

Motion
Tracking

Camera

Camera
3D Geometry

Segment

Segmentation
Clustering
Detection

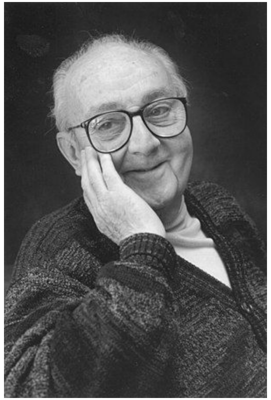
ML

Linear Models
(Conv) Neural networks

From Convolutions to Convolutions

“Every model is wrong, but some are useful”

George E.P. Box



Today's agenda

- Color spaces
- Image sampling and quantization
- Image histograms
- Images as functions
- Filters
- Properties of systems

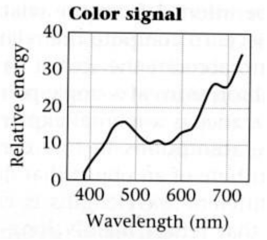
Some background reading:
Forsyth and Ponce, Computer Vision, Chapter 7

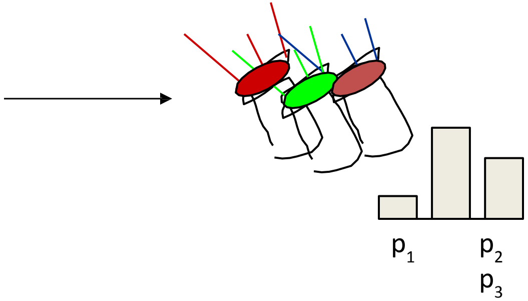
Today’s agenda

- Color spaces
- Image sampling and quantization
- Image histograms
- Images as functions
- Filters
- Properties of systems

Some background reading:
Forsyth and Ponce, Computer Vision, Chapter 7

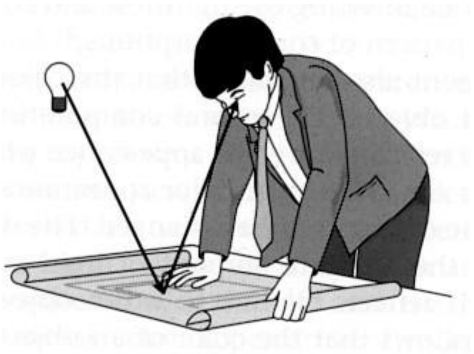
How to compute the weights of the primaries to match any spectral signal

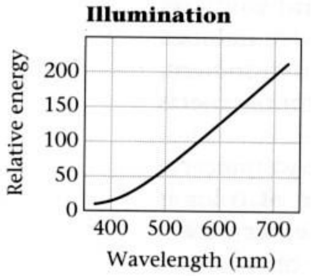




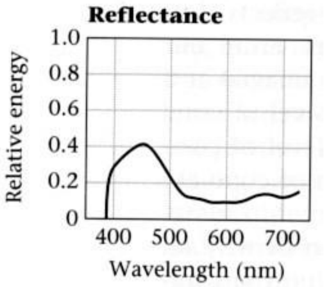
Matching functions: the amount of each primary needed to match a monochromatic light source at each wavelength

Explaining Color - A Simplified “Model”

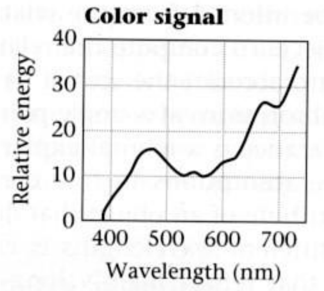




.*

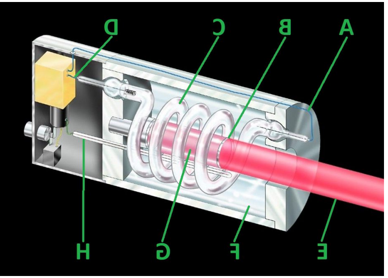


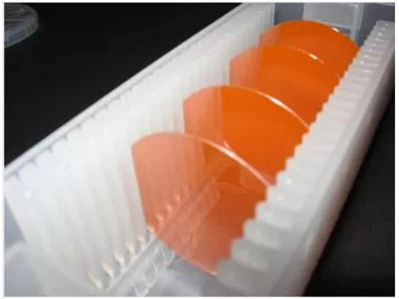
=



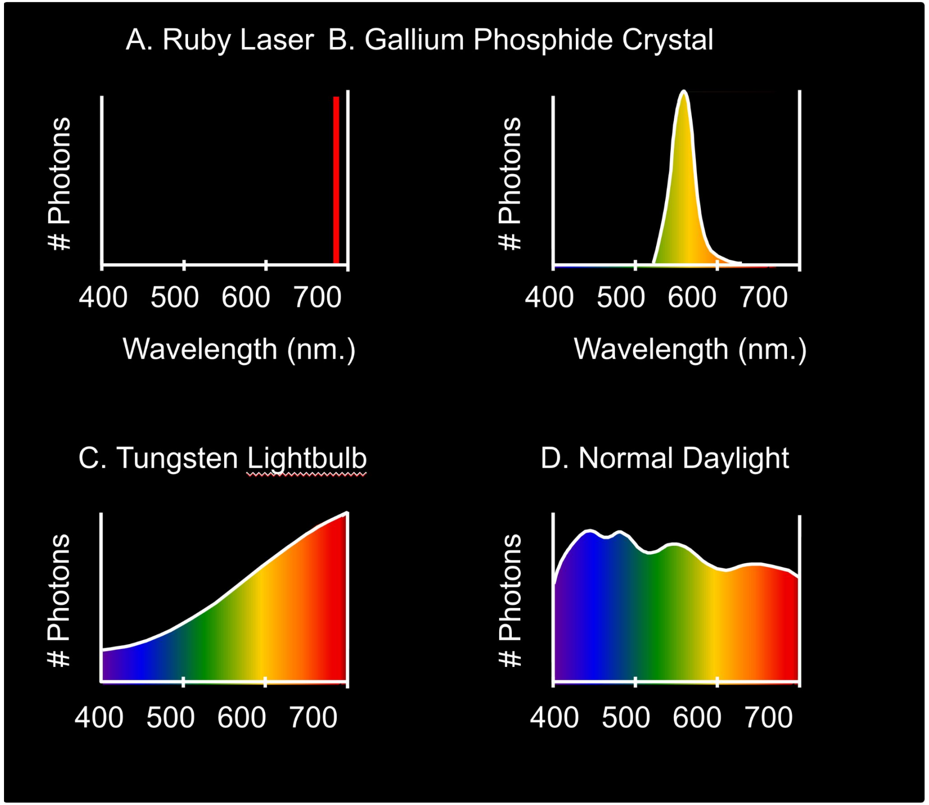
Foundations of Vision, by Brian Wandell, Sinauer Assoc., 1995

The Physics of Light Sources



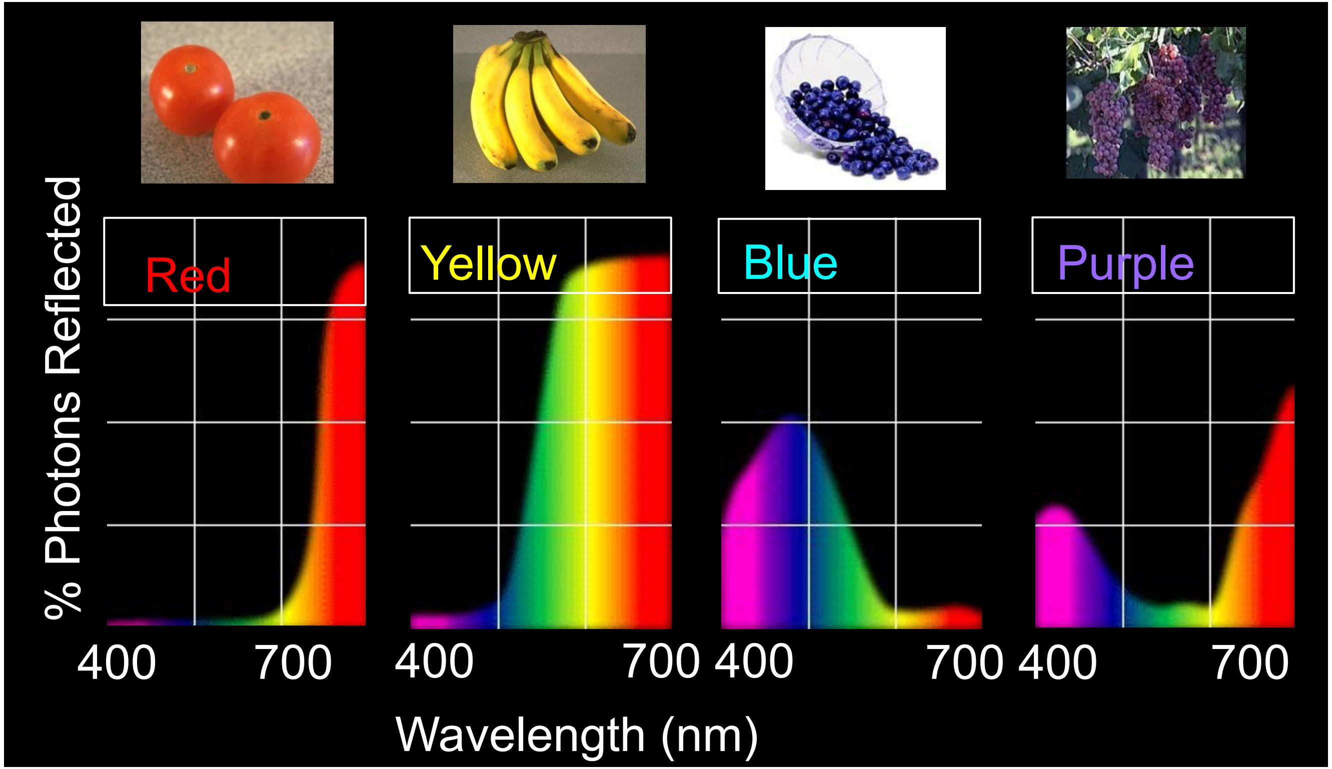


Some examples of the spectra of light sources



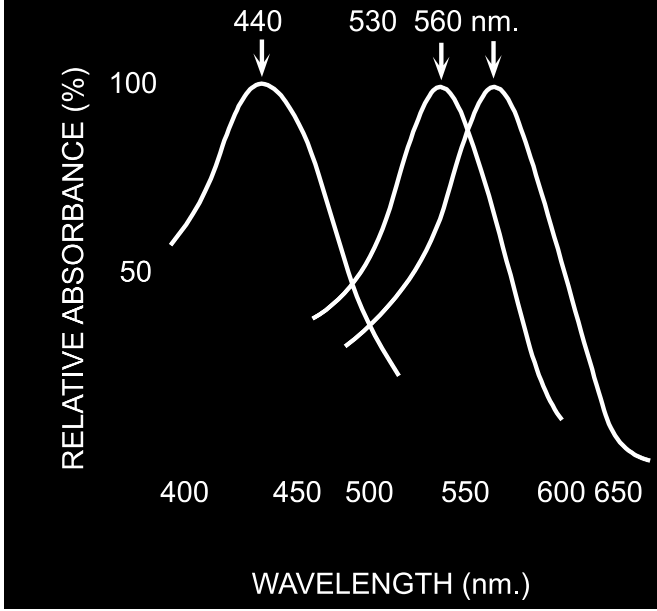
The Physics of Reflectance

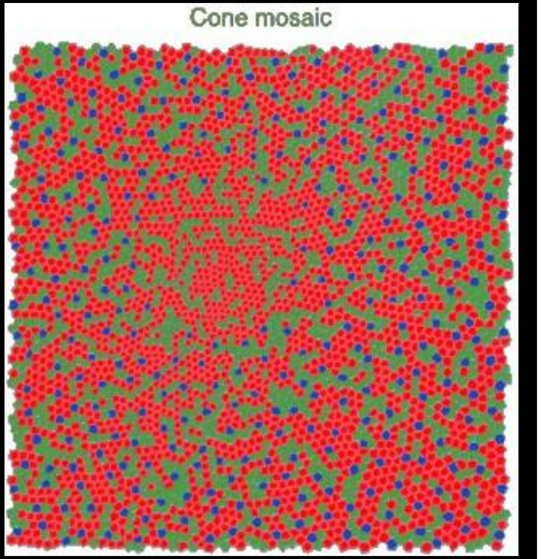
Some examples of the reflectance spectra of surfaces



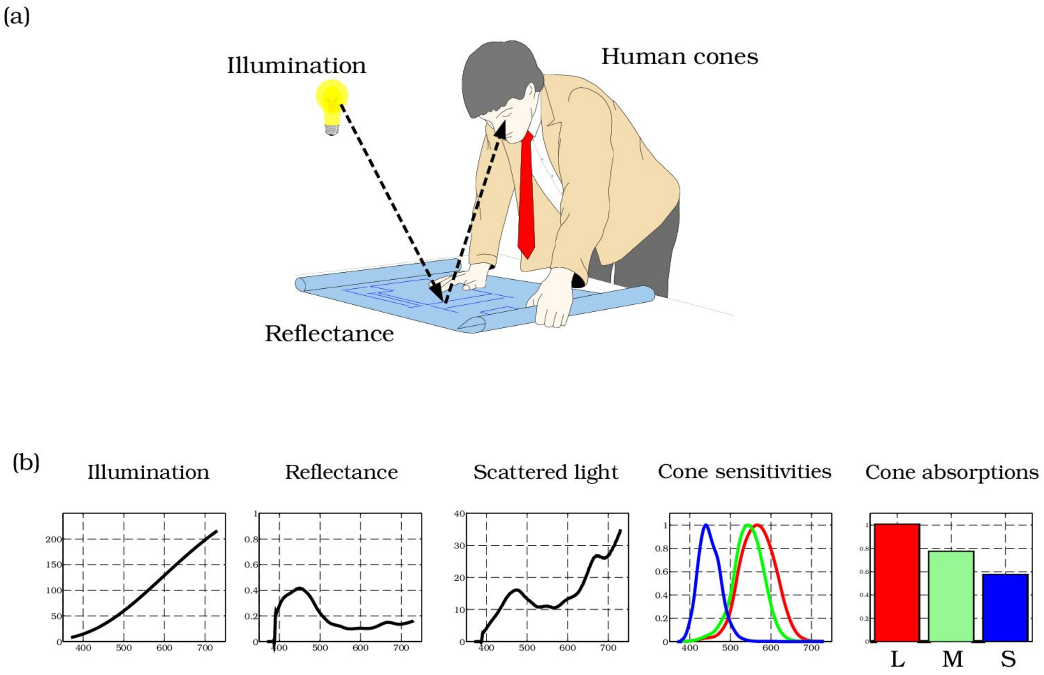
Physiology of Human Vision

Three kinds of cones:





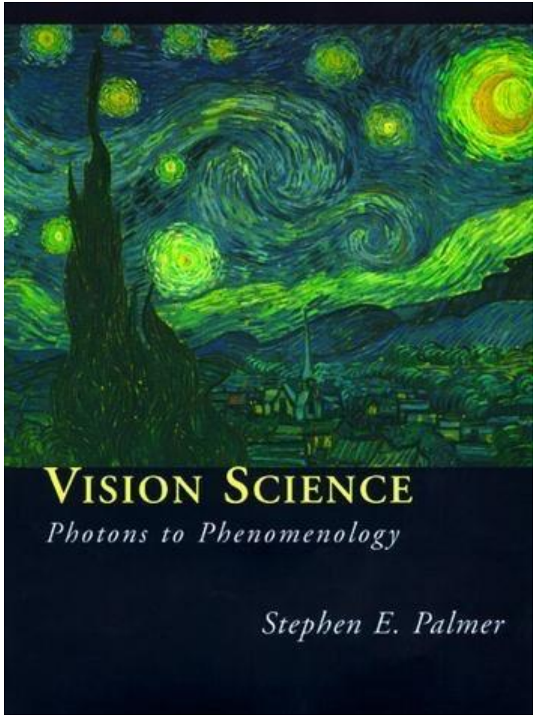
A Slightly Complex “Model”



Color is a psychological phenomenon

- Do we really see the same color?
- The result of interaction between physical light in the environment and our visual system.

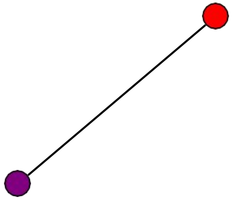
- A *psychological property* of our visual experiences when we look at objects and lights, *not a physical property* of those objects or lights.



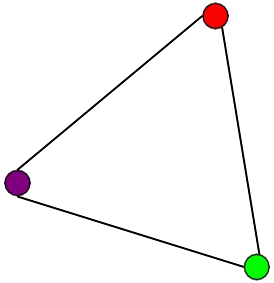
Slide credit: Lana Lazebnik

Linear color spaces

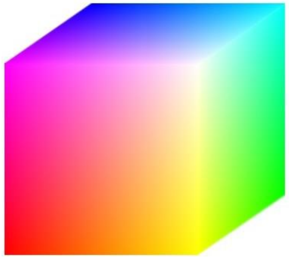
- Defined by a choice of three *primaries*
- The coordinates of a color are given by the weights of the primaries used to match it



mixing two lights produces colors that lie along a straight line in color space



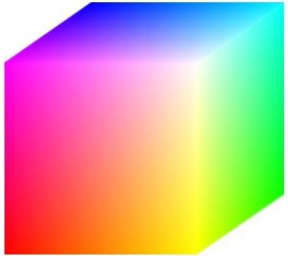
mixing three lights produces colors that lie within the triangle they define in color space



RGB space

Primaries are monochromatic lights (for monitors, they correspond to the three types of phosphors)

RGB primaries

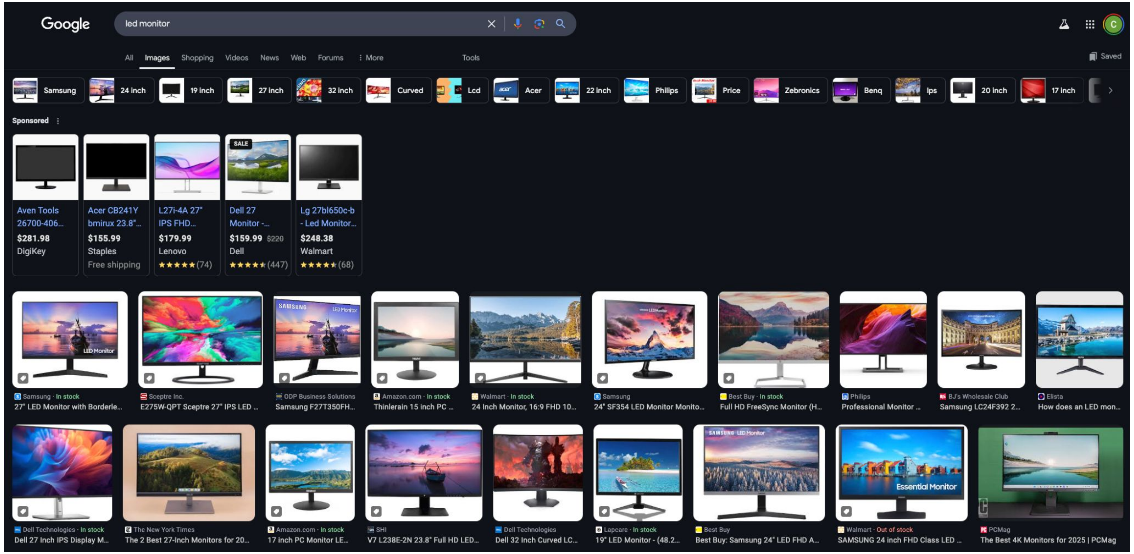


🟥 $p_1 = 645.2$ nm

🟩 $p_2 = 525.3$ nm

🟦 $p_3 = 444.4$ nm

Blue Light LED (90’s)



led monitor

Aven Tools 26700-406...	Acer CB241Y bmirux 23.8"...	L27i-4A 27" IPS FHD...	Dell 27 Monitor -...	Lg 27bl650c-b - Led Monitor...
$281.98	$155.99	$179.99	$159.99 ~~$220~~	$248.38
DigiKey	Staples	Lenovo	Dell	Walmart
	Free shipping	★★★★★ (74)	★★★★★ (447)	★★★★★ (68)

Samsung · In stock
27" LED Monitor with Borderle...

Sceptre Inc.
E275W-QPT Sceptre 27" IPS LED ...

ODP Business Solutions
Samsung F27T350FH...

Amazon.com · In stock
Thinlerain 15 inch PC ...

Walmart · In stock
24 Inch Monitor, 16:9 FHD 10...

Samsung
24" SF354 LED Monitor Monito...

Best Buy · In stock
Full HD FreeSync Monitor (H...

Philips
Professional Monitor ...

BJ's Wholesale Club
Samsung LC24F392 2...

Elista
How does an LED mon...

Dell Technologies · In stock
Dell 27 Inch IPS Display M...

The New York Times
The 2 Best 27-Inch Monitors for 20...

Amazon.com · In stock
17 inch PC Monitor LE...

SHI
V7 L238E-2N 23.8" Full HD LED...

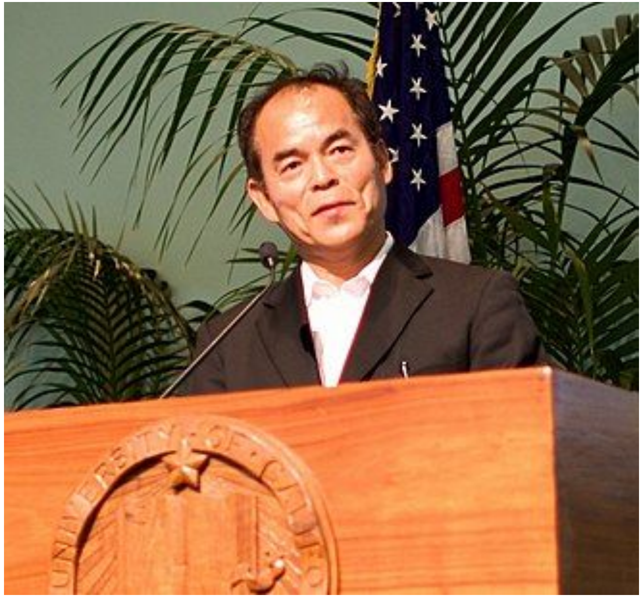
Dell Technologies
Dell 32 Inch Curved LC...

Lapcare · In stock
19" LED Monitor - (48.2...

Best Buy
Best Buy: Samsung 24" LED FHD A...

Walmart · Out of stock
SAMSUNG 24 inch FHD Class LED ...

PCMag
The Best 4K Monitors for 2025 | PCMag

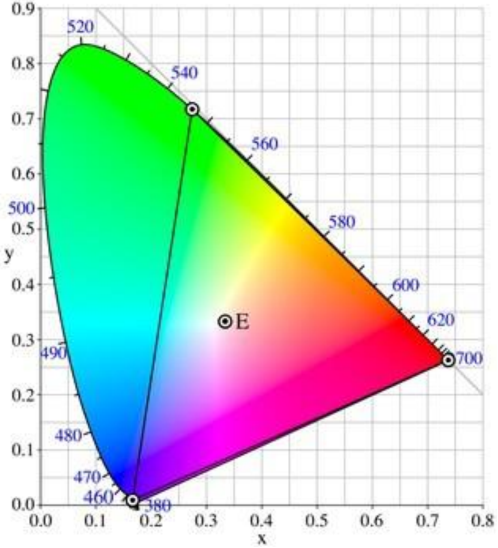


Shuji Nakamura,
Nobel Prize in Physics 2014

Other spaces: CIE XYZ

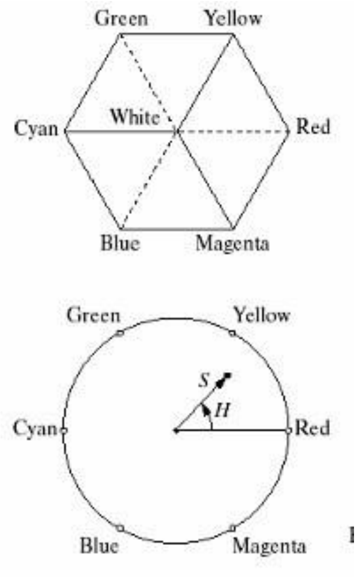
- Primaries (X, Y and Z) are imaginary
- X: Represents a mix of red and green.
- Y: Represents luminance (brightness).
- Z: Represents a mix of blue and green.

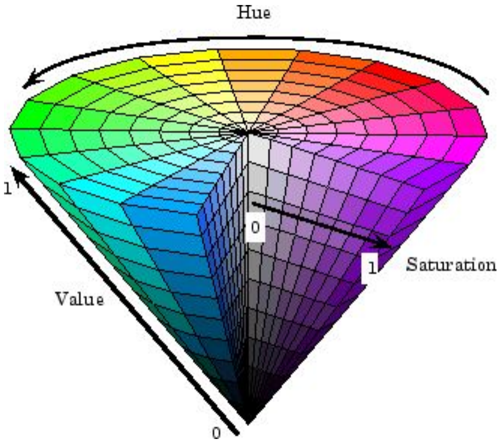
- 2D visualization: draw (*x*,*y*), where
  $x = X/(X+Y+Z)$, $y = Y/(X+Y+Z)$

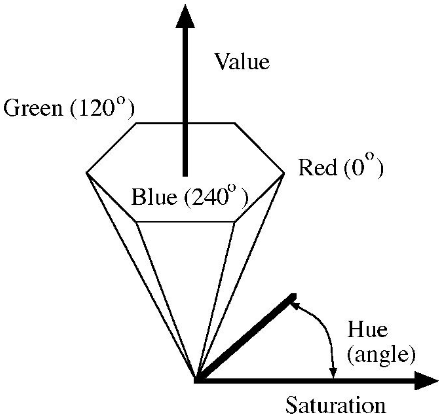


http://en.wikipedia.org/wiki/CIE_1931_color_space

Other color spaces: HSV

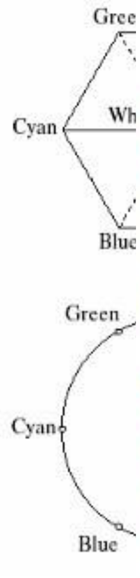


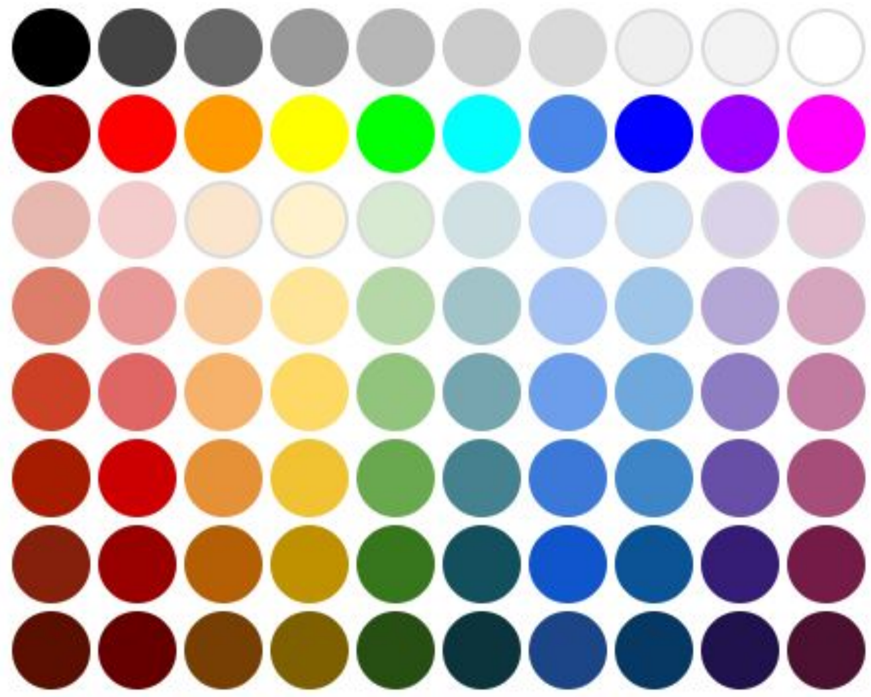


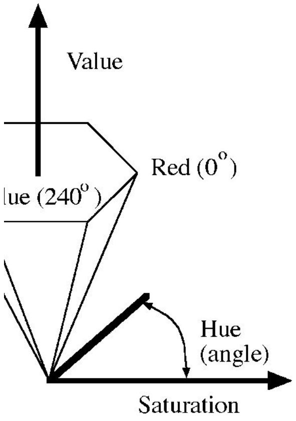


- Perceptually meaningful dimensions: Hue, Saturation (Brightness), Value (Intensity)
- Useful in data augmentation for training large models

Other color spaces: HSV







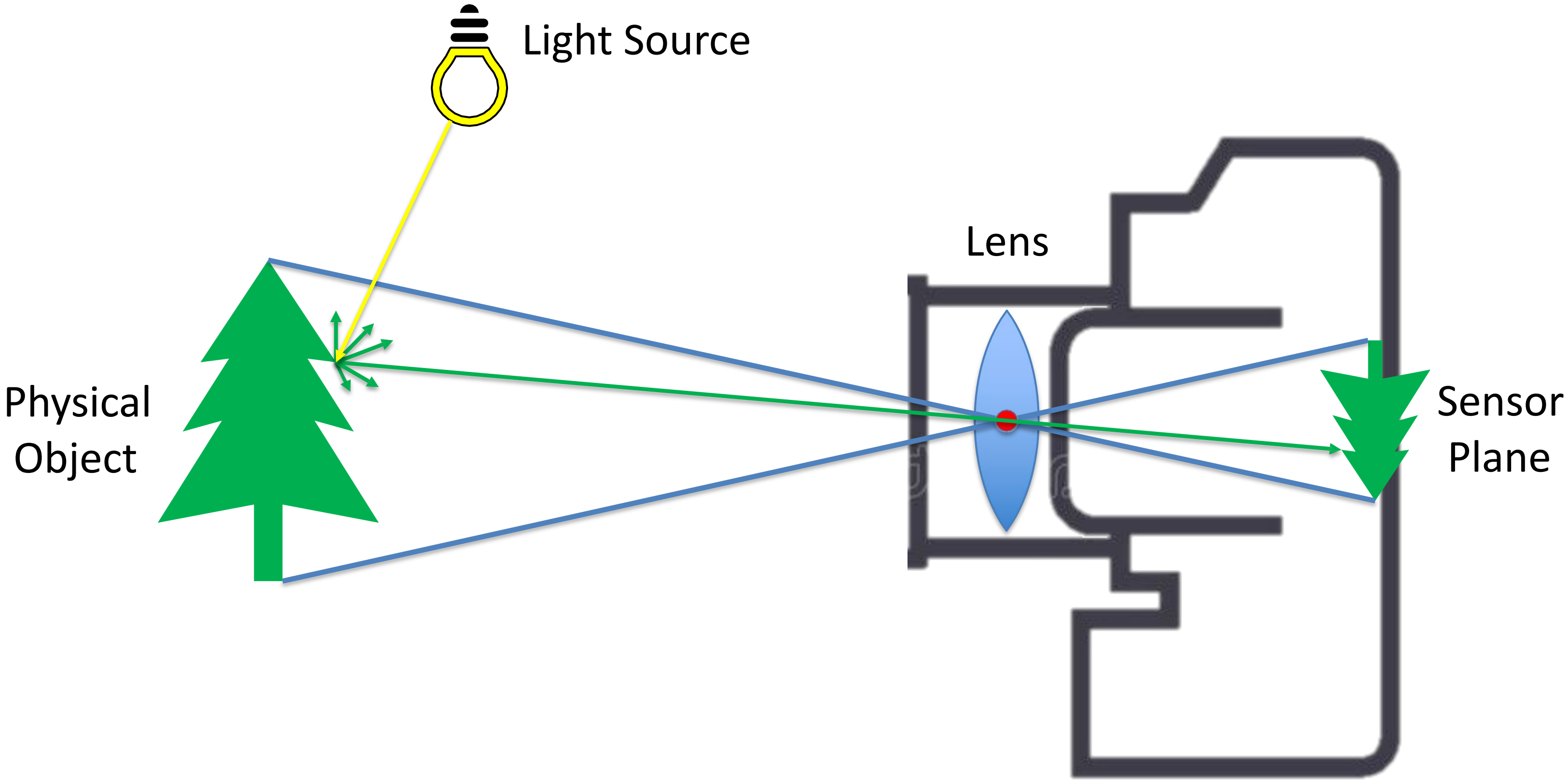
- Perceptually me tensity)

Today’s agenda

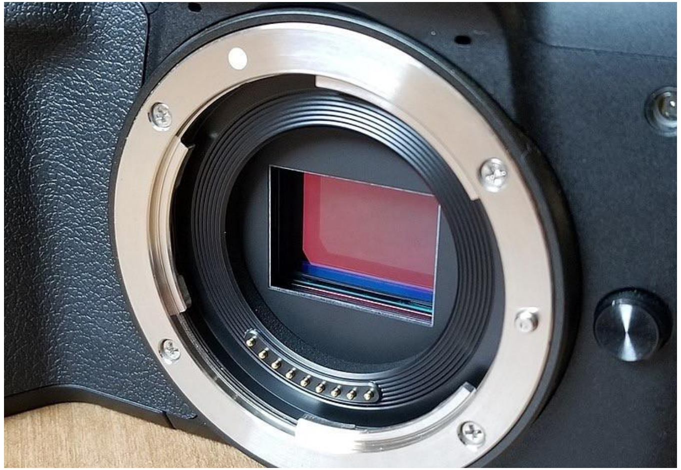
- Color spaces
- Image sampling and quantization
- Image histograms
- Images as functions
- Filters
- Properties of systems

Some background reading:
Forsyth and Ponce, Computer Vision, Chapter 7

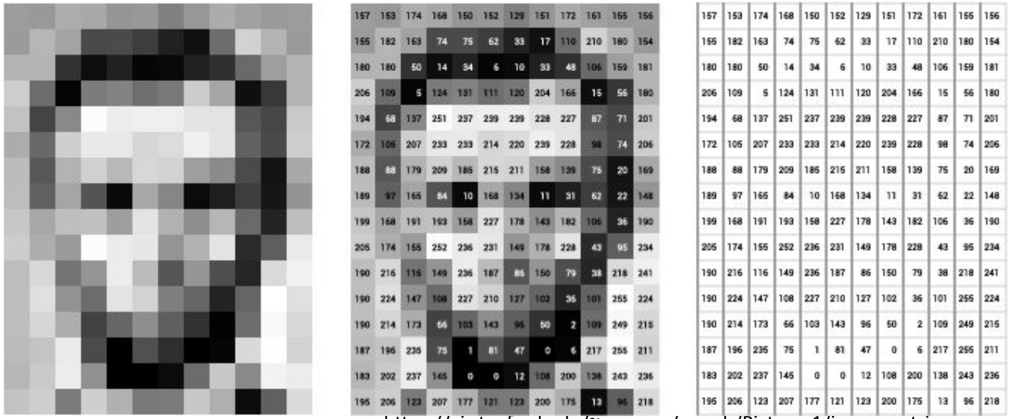
Image Formation



Camera sensors produce discrete outputs



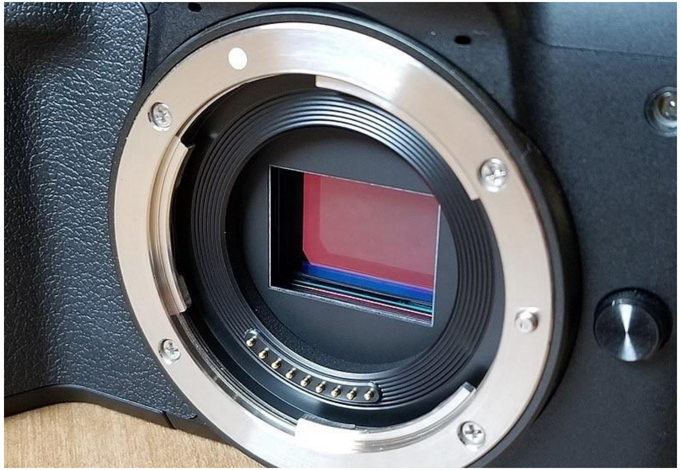
https://commons.wikimedia.org/wiki/File:Mirrorless_Camera_Sensor.jpg



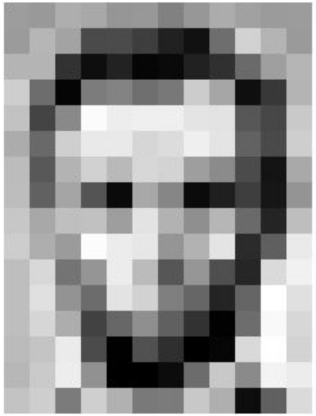
157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

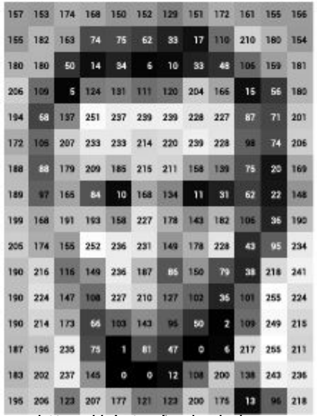
https://ai.stanford.edu/~syyeung/cvweb/Pictures1/imagematrix.png

Camera sensors produce discrete outputs

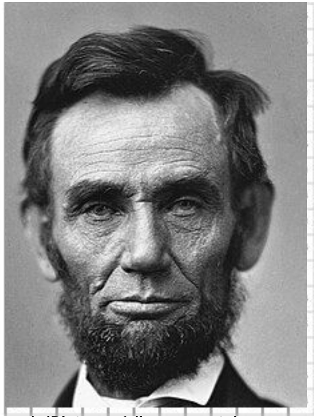


https://commons.wikimedia.org/wiki/File:Mirrorless_Camera_Sensor.jpg





157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	106	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	85	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	95	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218



https://ai.stanford.edu/~syyeung/cvweb/Pictures1/imagematrix.png

Types of Images

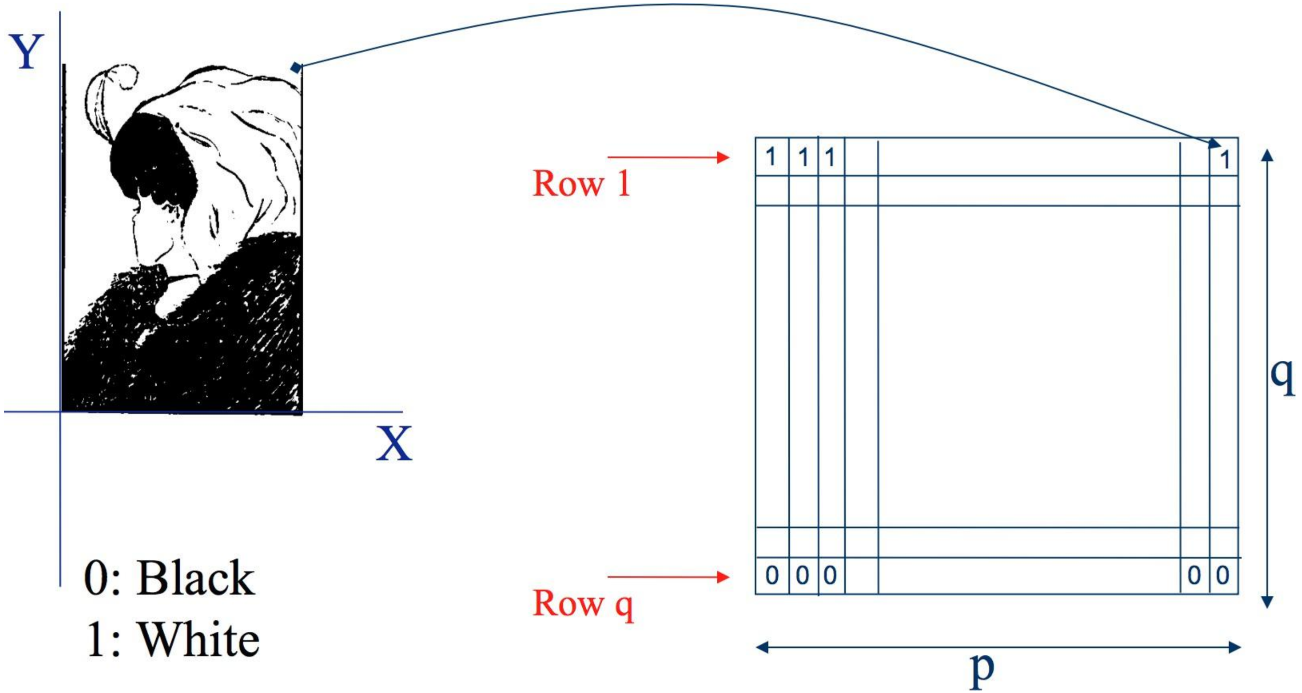
Binary

Grayscale

Color

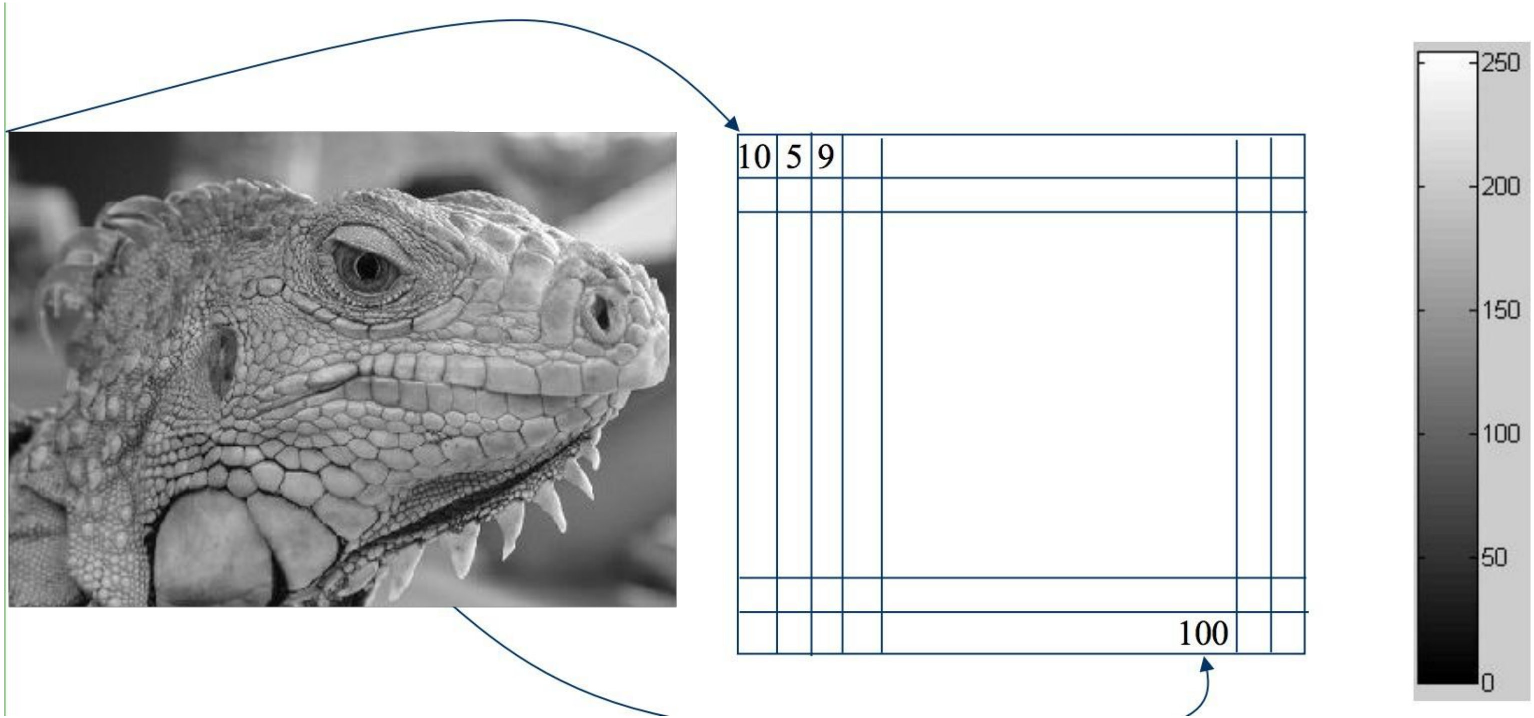
Phil Noble / AP

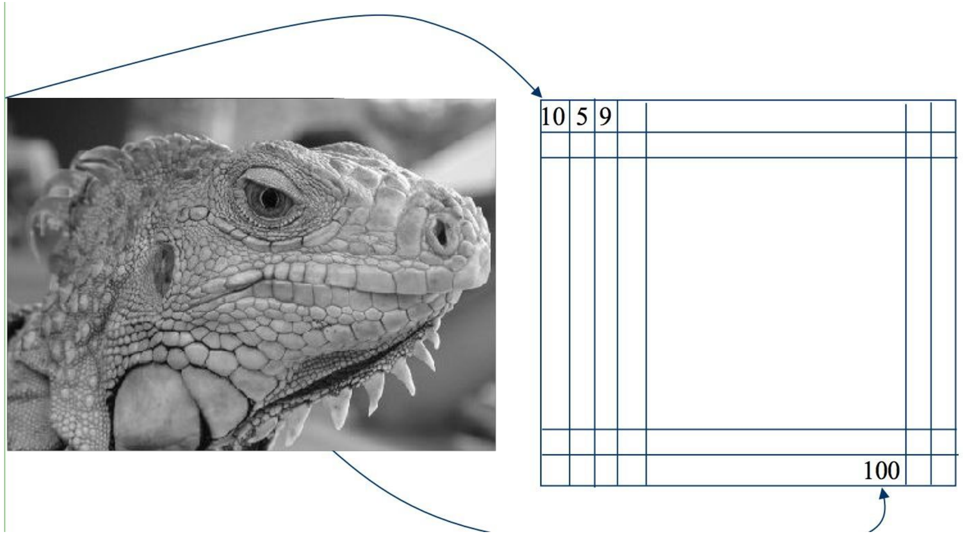
Binary image representation

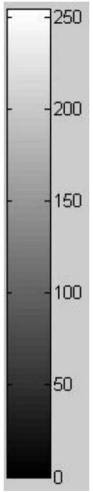


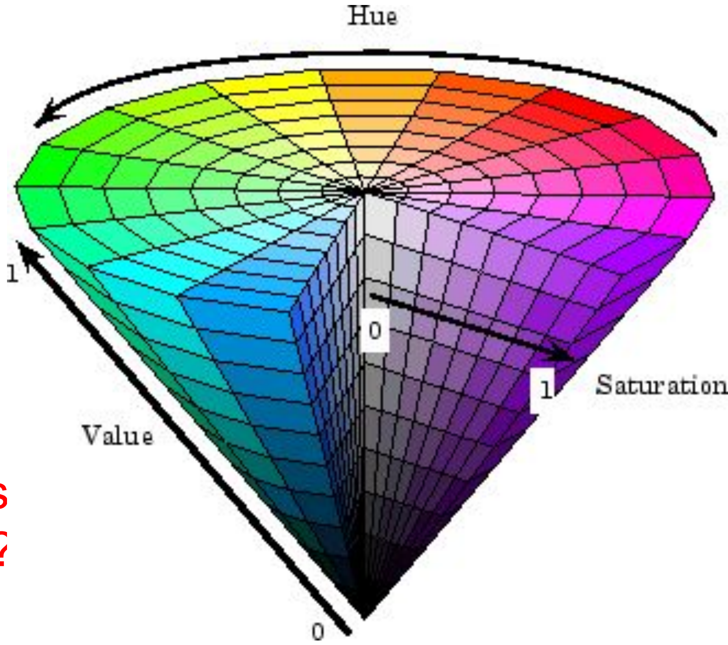
0: Black
1: White

Grayscale image representation





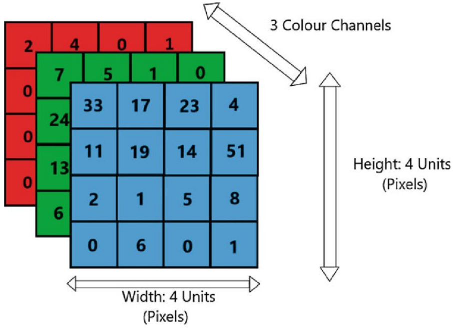




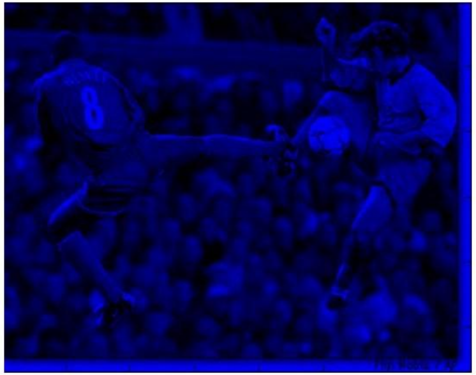
Q. If you used HSV to represent grayscale images, is the slider representing hue? Or saturation? Or value?

Color image representation

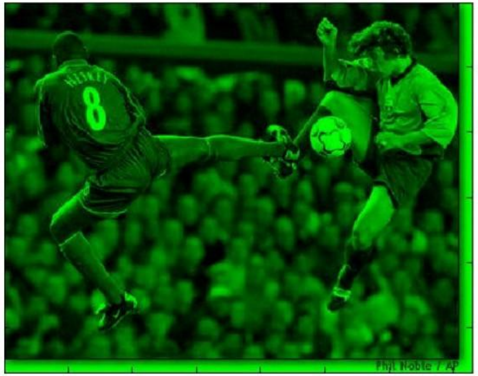
Phil Noble / AP



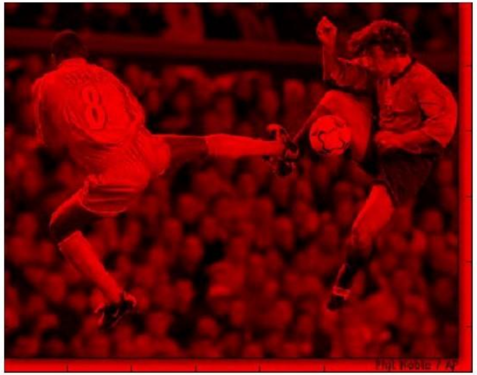
33	17	23	4
11	19	14	51
2	1	5	8
0	6	0	1



B channel



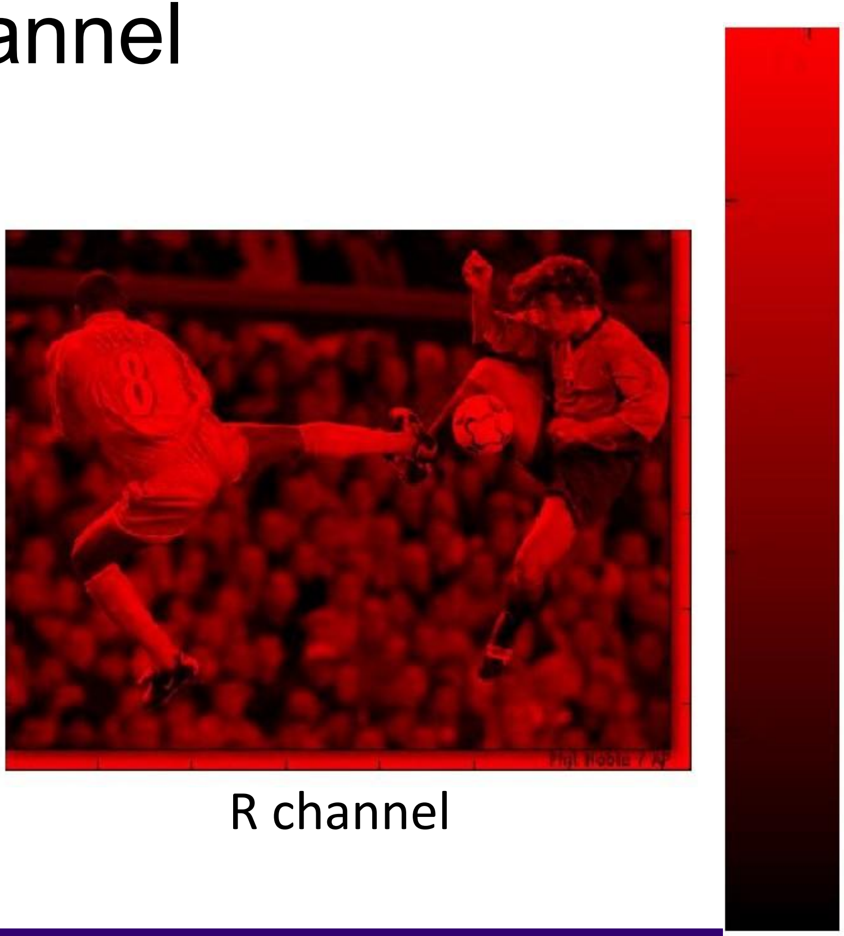
G channel



R channel

Color image - one channel

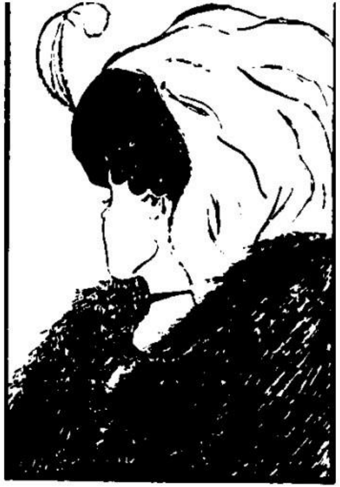
Phil Noble / AP



R channel

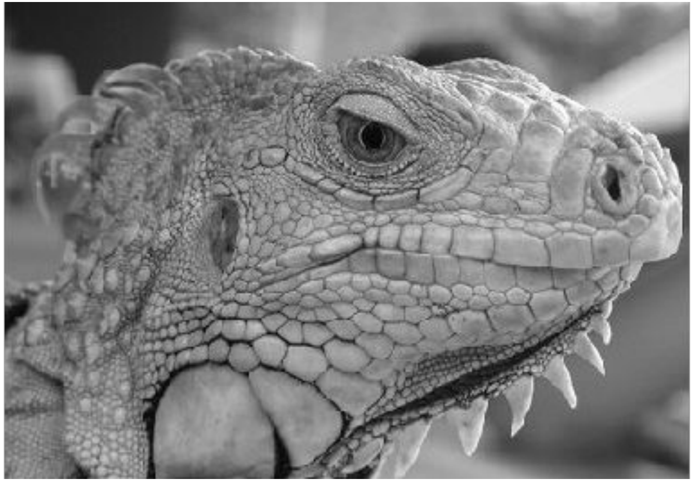
Types of Images

Binary



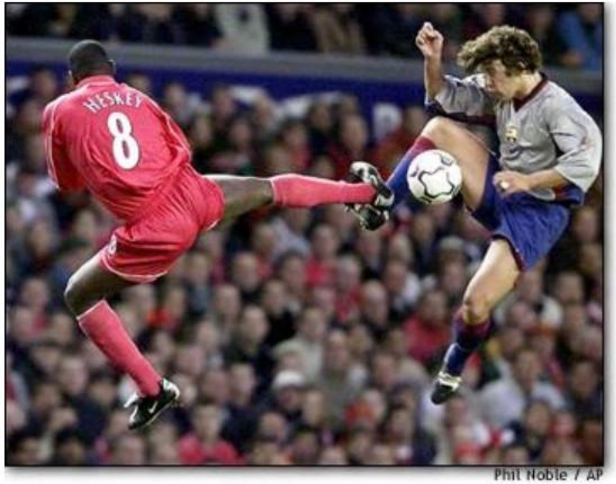
[0, 1]

Grayscale



[0, 1, …, 255]

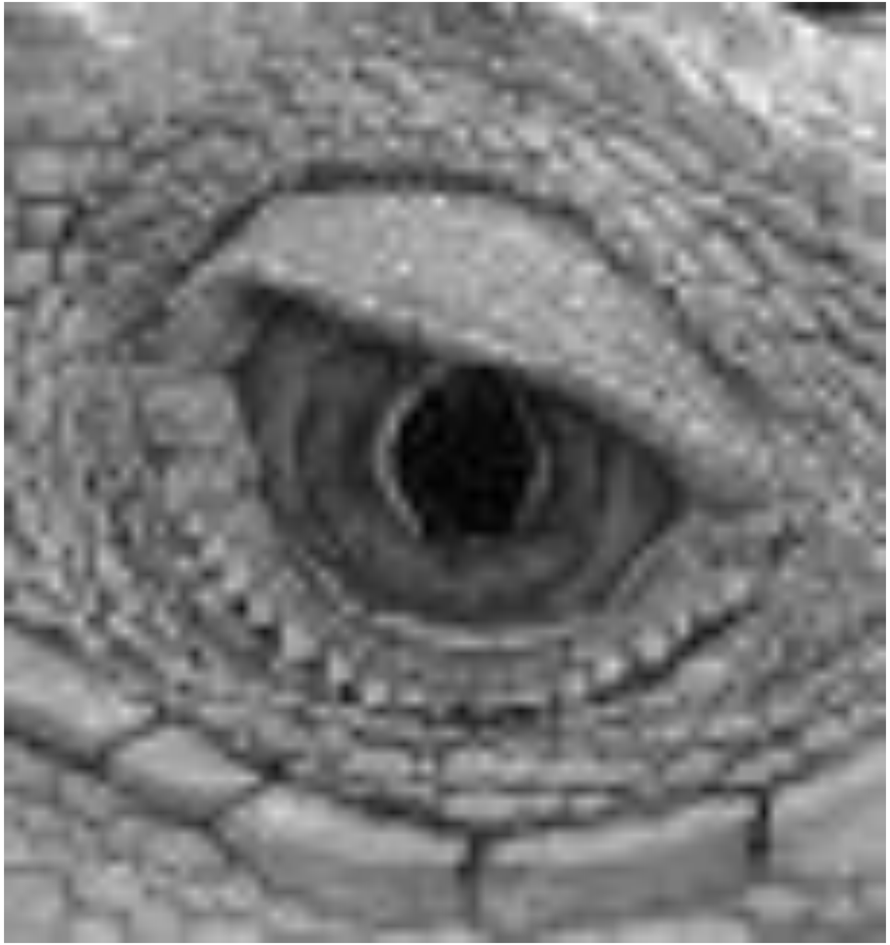
Color



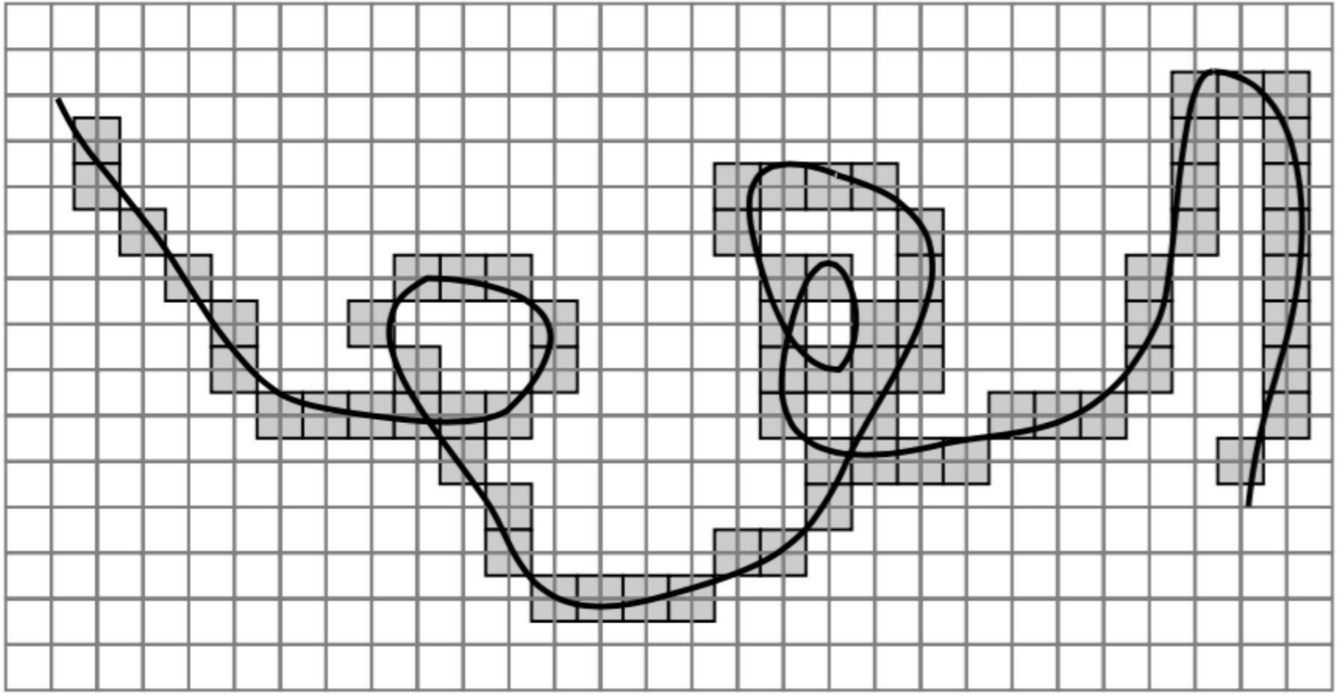
[0, 1, …, 255]^3

Digital Images are sampled

What happens when we zoom into the images we capture?



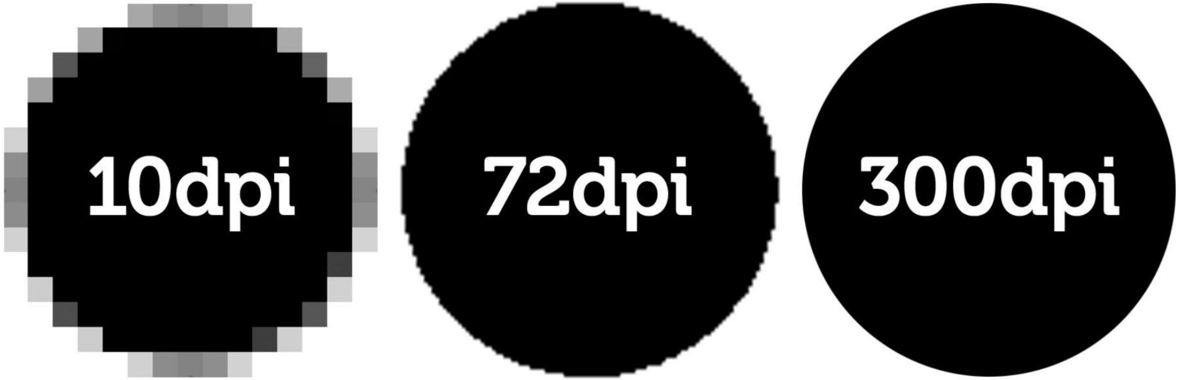
Errors due to Sampling



Q: How to compensate the error?

Resolution

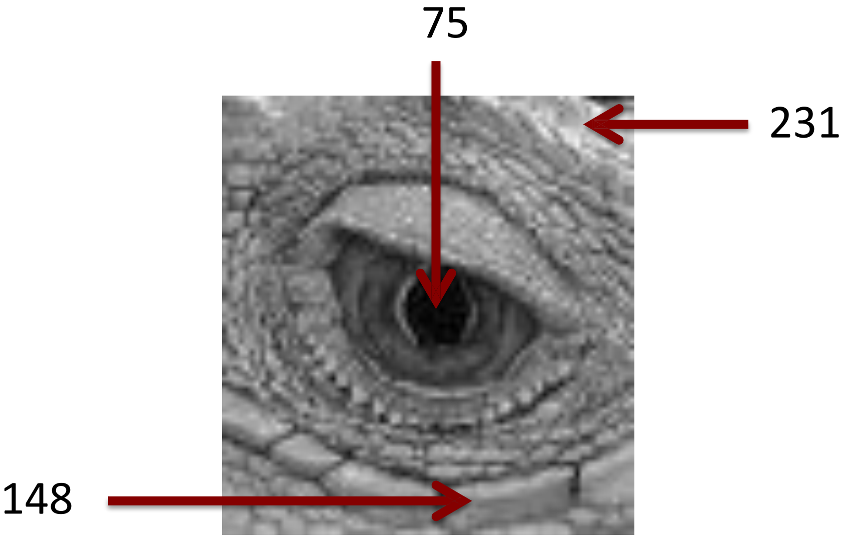
is a **sampling** parameter, defined in dots per inch (DPI) or equivalent measures of spatial pixel density



Resolution

Images are Sampled and Quantized

- An image contains discrete number of pixels
  - –Pixel value:
    - •“grayscale”
      (or “intensity”): [0,255]



Images are Sampled and Quantized

- An image contains discrete number of pixels
  - –Pixel value:
    - •“grayscale”

      (or “intensity”): [0,255]
    - •“color”
      - –RGB: [R, G, B]

Q: Why [0, 255] but not [0, 1]?

[90, 0, 53]

[249, 215, 203]

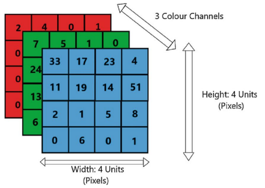
[213, 60, 67]

Phil Noble / AP

With this loss of information (from sampling and quantization),

How many possible 256x256x3 images do we have?

$$256^{256x256x3} = 2^{1572864}$$



How many images can a person perceive in the whole life?

1 (img/sec) x 86,400 (sec/day) x 365 (day/year) x 80 (years) ≈ 2,500,000,000

Q: What's the implication?

With this loss of information (from sampling and quantization),

Can we still use images for useful tasks?

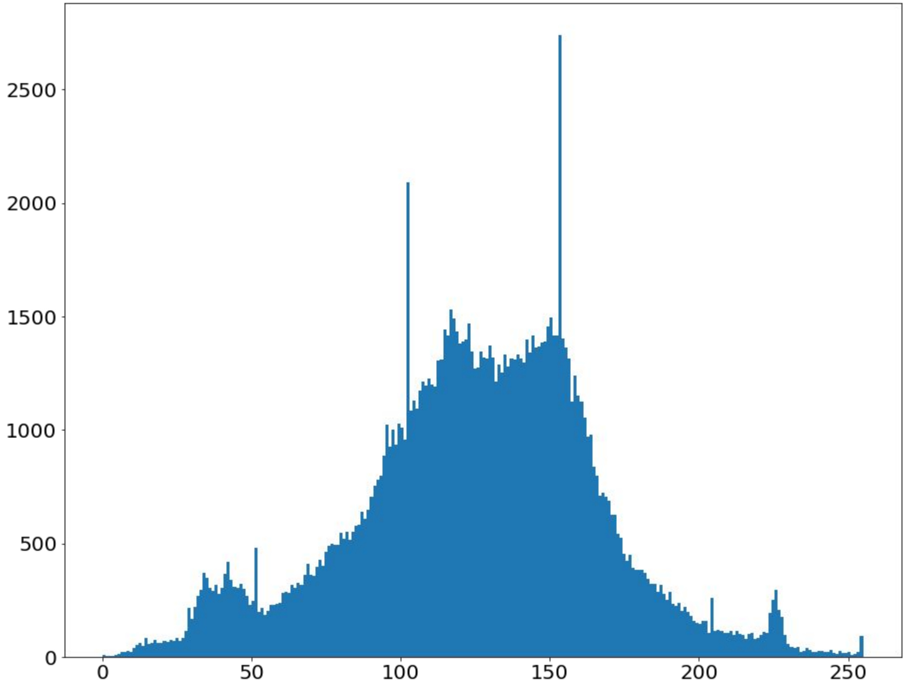
Today's agenda

- Color spaces
- Image sampling and quantization
- **Image histograms**
- Images as functions
- Filters
- Properties of systems

Some background reading:
Forsyth and Ponce, Computer Vision, Chapter 7

Starting with grayscale images:

- Histogram captures the distribution of gray levels in the image.
- How frequently each gray level occurs in the image



Grayscale histograms in code

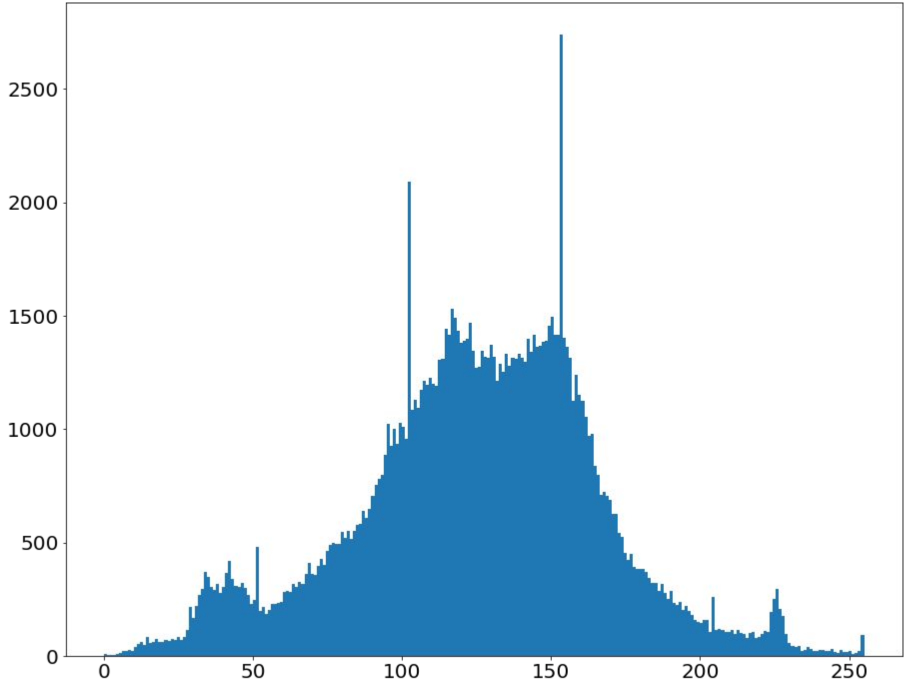
- Histogram of an image provides the frequency of the brightness (intensity) value in the image.

Here is a simple implementation of calculating histograms:

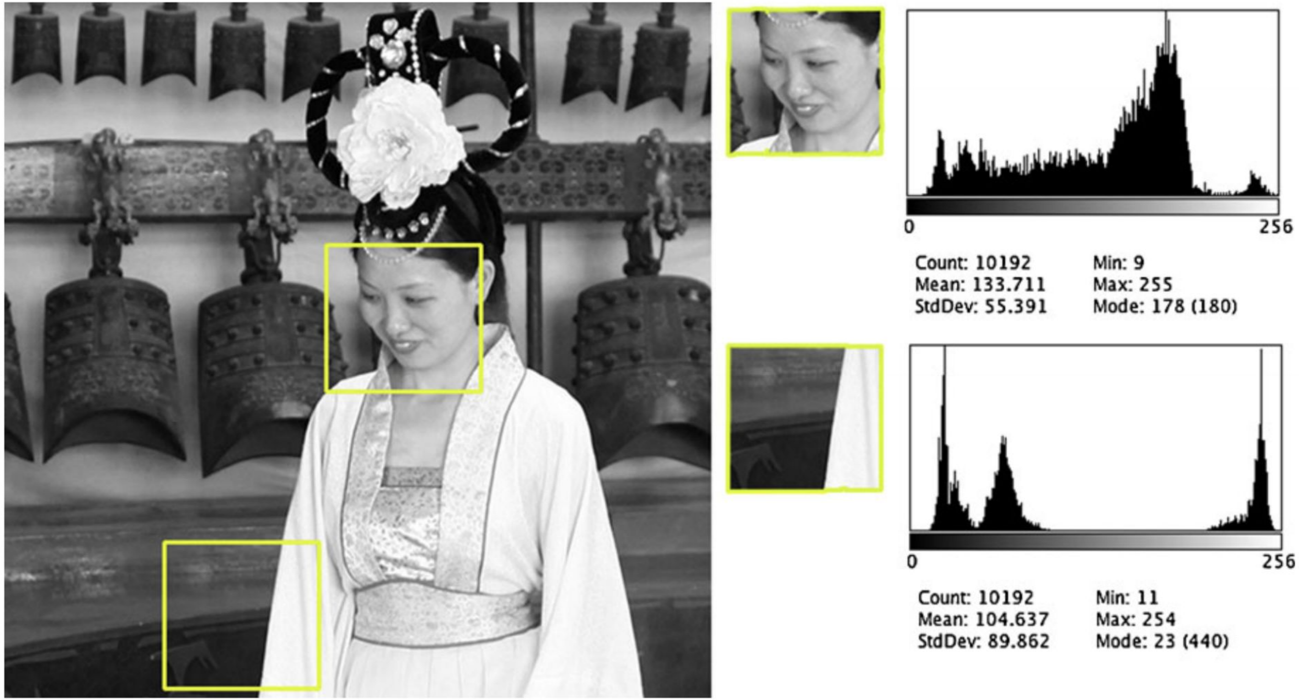
```
def histogram(im):
    h = np.zeros(256)
    for row in im.shape[0]:
        for col in im.shape[1]:
            val = im[row, col]
            h[val] += 1
```

Grayscale histograms in code

```
def histogram(im):
    h = np.zeros(256)
    for row in im.shape[0]:
        for col in im.shape[1]:
            val = im[row, col]
            h[val] += 1
```



Visualizing Histograms for patches



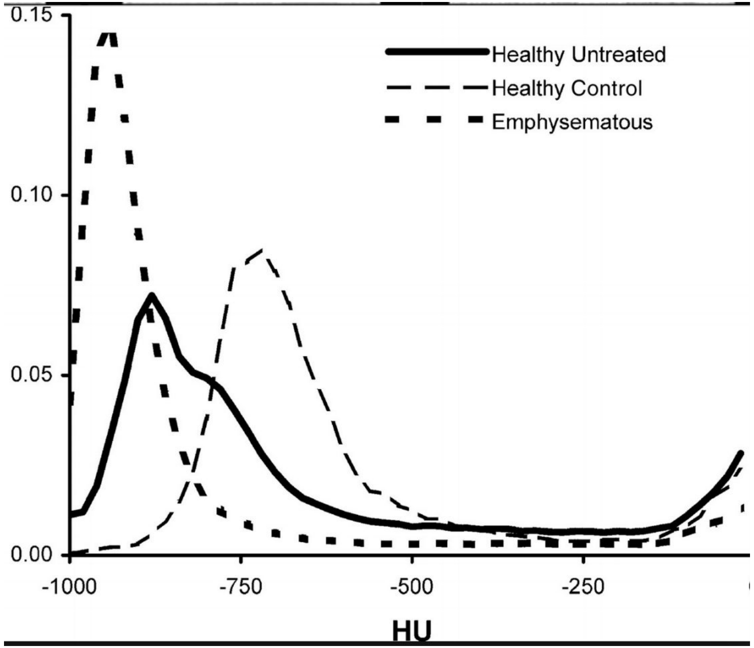
Q: How to use histogram for retrieval?

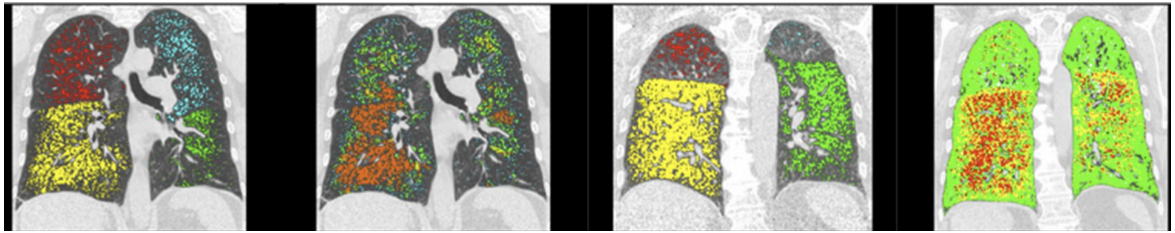
Slide credit: Dr. Mubarak Shah

Histogram – use case

In emphysema, the inner walls of the lungs' air sacs called alveoli are damaged, causing them to eventually rupture.

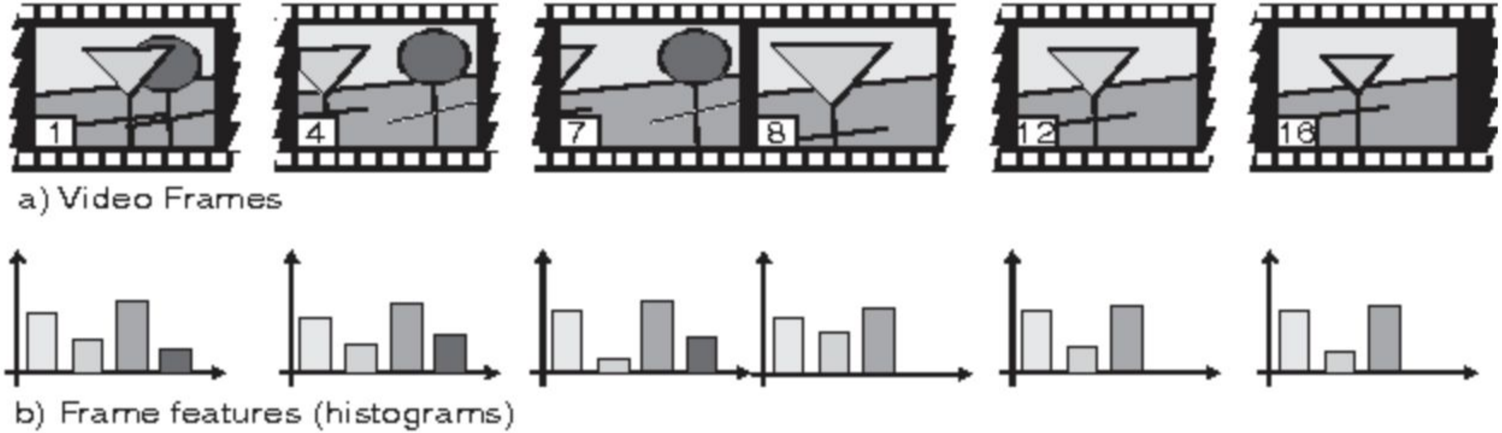
You can take a picture of the lung with special dye to mark the alveoli





Slide credit: Dr. Mubarak Shah

Histogram – use case



Video Shot Boundary Detection and Condensed Representation : A Review

Histograms are a convenient representation to extract information

- Commonly used before deep learning or low-power devices
- A very cheap “**representation**”
- Still useful even in deep learning era (really?!?!)

- Q: Is image/histogram an one-to-one mapping transformation?
- Can we develop better transformations than histograms?

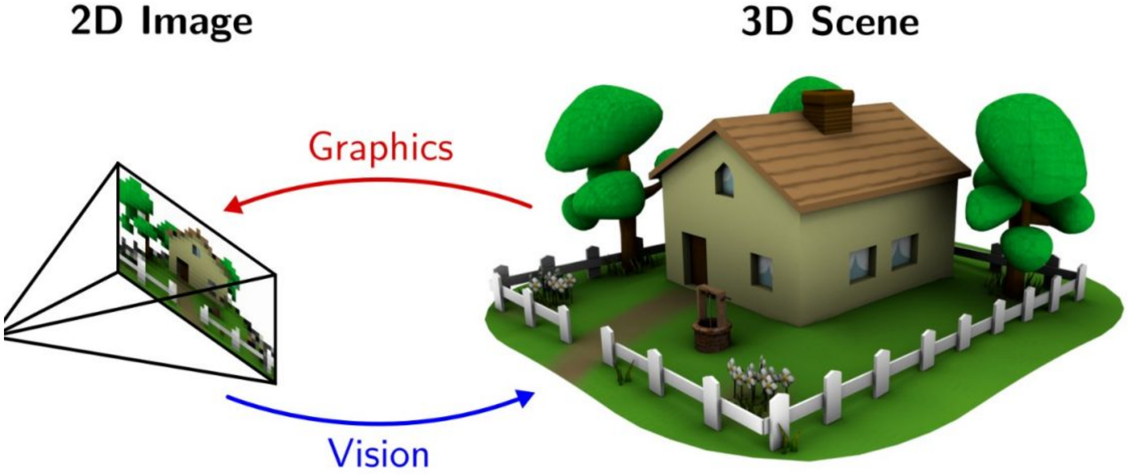
Today’s agenda

- Color spaces
- Image sampling and quantization
- Image histograms
- Images as functions
- Filters
- Properties of systems

Some background reading:
Forsyth and Ponce, Computer Vision, Chapter 7

Images are a function!!!

This is a new formalism that will allow us to borrow ideas from signal processing to extract meaningful information.

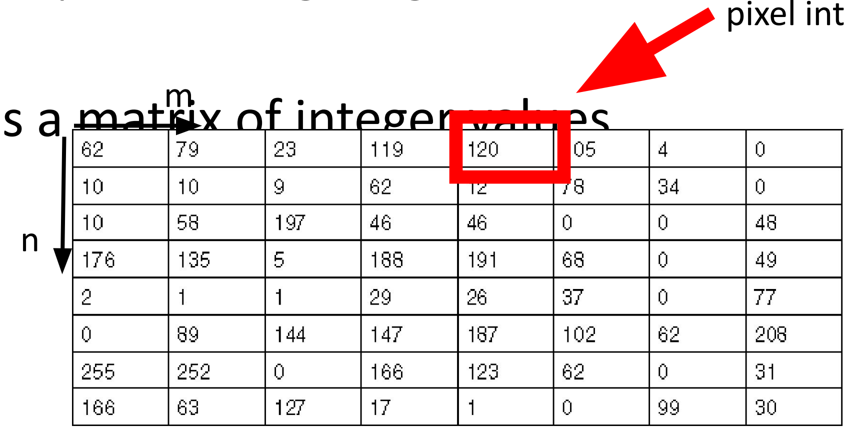


At every pixel location, we get an intensity value for that pixel.

The world captured by the image continues beyond the confines of the image

Images as discrete functions

- Also popular in high-dimensional statistics and machine learning
  - function v.s. vector
- Digital images are usually discrete:
  - **Sample** the 2D space on a regular grid
- Represented as a matrix of integer values

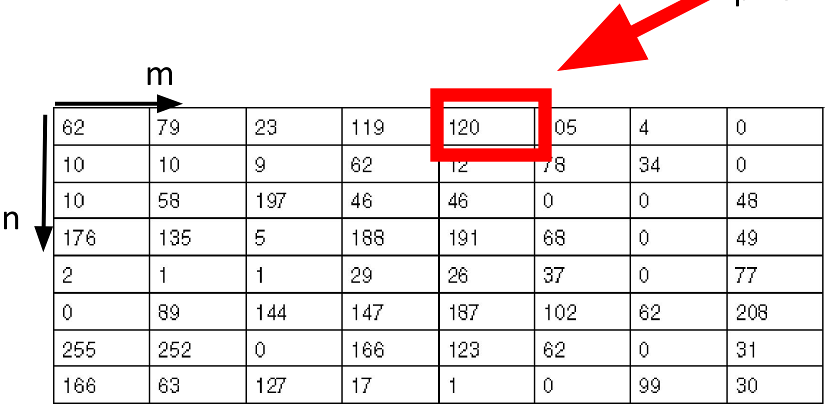


pixel intensity

62	79	23	119	120	105	4	0
10	10	9	62	12	78	34	0
10	58	197	46	46	0	0	48
176	135	5	188	191	68	0	49
2	1	1	29	26	37	0	77
0	89	144	147	187	102	62	208
255	252	0	166	123	62	0	31
166	63	127	17	1	0	99	30

Images as discrete function f

- The input to the image function is a pixel location, [n m]
- The output to the image function is the pixel intensity



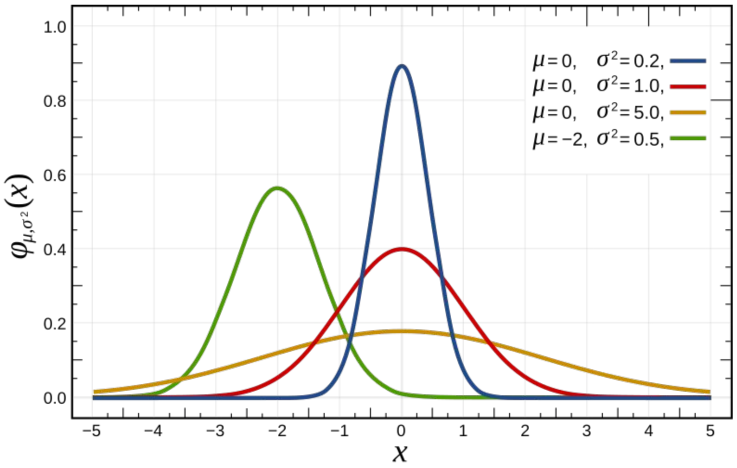
pixel intensity

62	79	23	119	120	05	4	0
10	10	9	62	12	78	34	0
10	58	197	46	46	0	0	48
176	135	5	188	191	68	0	49
2	1	1	29	26	37	0	77
0	89	144	147	187	102	62	208
255	252	0	166	123	62	0	31
166	63	127	17	1	0	99	30

~~f(0, 5)~~ f[0, 5] = 120

Images as discrete function f

- Also popular in high-dimensional statistics and machine learning
  - function v.s. vector



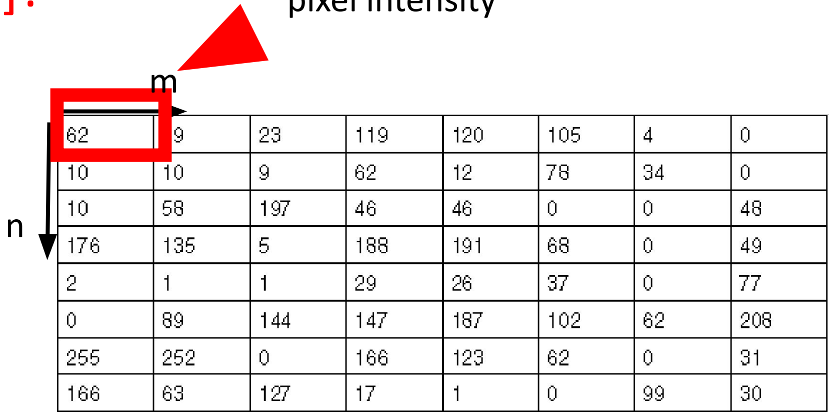
$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} .$$

Images as discrete function f

- The input to the image function is a pixel location, [n m]
- The output to the image function is the pixel intensity

Q1. What is f[0, 0]?

pixel intensity

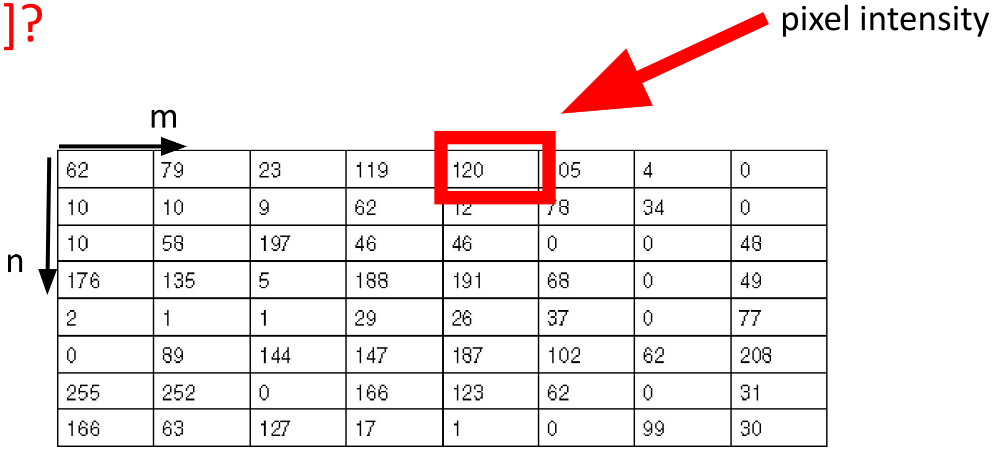


62	9	23	119	120	105	4	0
10	10	9	62	12	78	34	0
10	58	197	46	46	0	0	48
176	135	5	188	191	68	0	49
2	1	1	29	26	37	0	77
0	89	144	147	187	102	62	208
255	252	0	166	123	62	0	31
166	63	127	17	1	0	99	30

Images as discrete function f

- The input to the image function is a pixel location, [n m]
- The output to the image function is the pixel intensity

Q2. What is f[0, 4]?

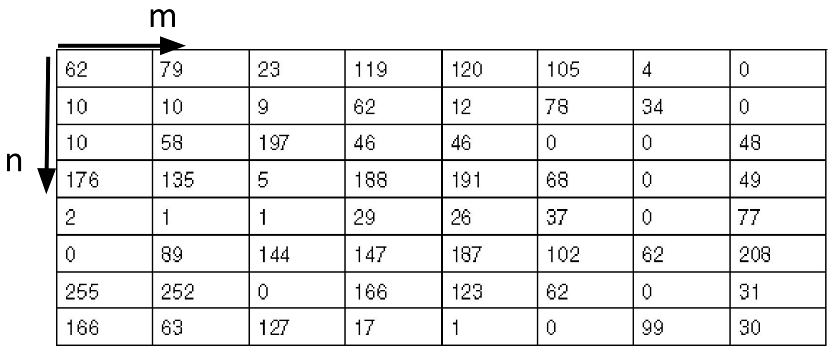


62	79	23	119	120	05	4	0
10	10	9	62	12	78	34	0
10	58	197	46	46	0	0	48
176	135	5	188	191	68	0	49
2	1	1	29	26	37	0	77
0	89	144	147	187	102	62	208
255	252	0	166	123	62	0	31
166	63	127	17	1	0	99	30

Images as discrete function f

- The input to the image function is a pixel location, [n m]
- The output to the image function is the pixel intensity

Q2. What is f[0, -8]?



62	79	23	119	120	105	4	0
10	10	9	62	12	78	34	0
10	58	197	46	46	0	0	48
176	135	5	188	191	68	0	49
2	1	1	29	26	37	0	77
0	89	144	147	187	102	62	208
255	252	0	166	123	62	0	31
166	63	127	17	1	0	99	30

Images as coordinates

We can represent this function as f.

f[n, m] represents the pixel intensity at that value.

$$f[n,m] = \begin{bmatrix} \ddots & & \vdots & & \\ & f[-1,-1] & f[-1,0] & f[-1,1] & \\ \dots & f[0,-1] & \underline{f[0,0]} & f[0,1] & \dots \\ & f[1,-1] & f[1,0] & f[1,1] & \\ & & \vdots & & \ddots \end{bmatrix}$$

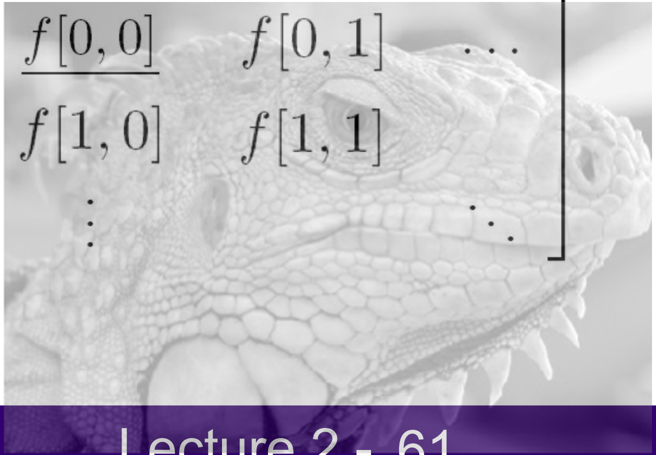
Notation for discrete functions

n and *m* can be any integer

Even negative!!

We don’t have the intensity values for negative indices

$$f[n,m] = \begin{bmatrix} \ddots & & \vdots & & \\ & f[-1,-1] & f[-1,0] & f[-1,1] & \\ \dots & f[0,-1] & \underline{f[0,0]} & f[0,1] & \dots \\ & f[1,-1] & f[1,0] & f[1,1] & \\ & & \vdots & & \ddots \end{bmatrix}$$

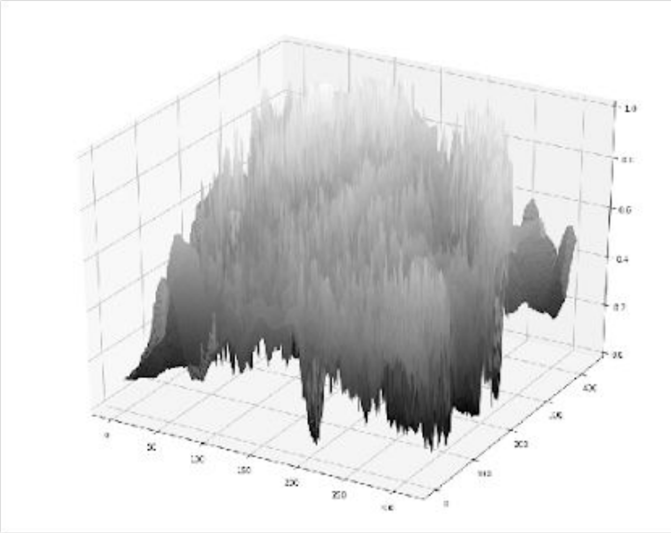


n and *m* can be any integer

Even negative!!

Images as functions

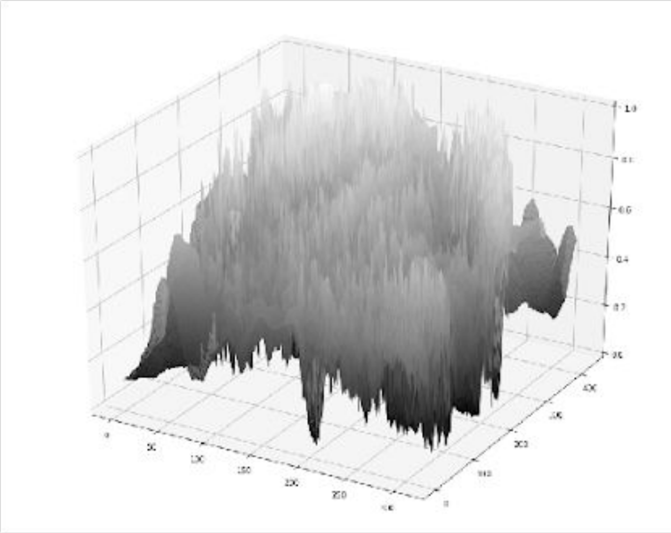
- **An Image** as a function *f* from R^2 to R^C:
  - if grayscale, C=1,
  - if color, C=3



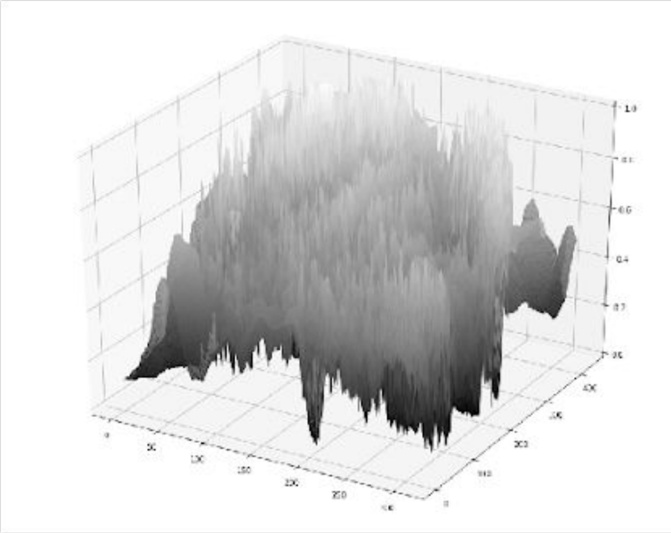
Images as functions

- **An Image** as a function *f* from R^2 to R^C:
  - if grayscale, C=1,
  - if color, C=3
  - f [n, m] gives the intensity at position [n, m]
  - Has values over a rectangle, with a finite range:

$$f\colon \underbrace{[0,H] \text{ x } [0,W]}_{\text{Domain support}} \rightarrow \underbrace{[0,255]}_{\text{range}}$$



Images as functions



- **An Image** as a function *f* from R^2 to R^C:
  - if grayscale, C=1,
  - if color, C=3
  - f [n, m] gives the intensity at position [n, m]
  - Has values over a rectangle, with a finite range:

$$f\colon \underbrace{[0,H] \times [0,W]}_{\text{Domain support}} \rightarrow \underbrace{[0,255]}_{\text{range}}$$

- Doesn't have values outside of the image rectangle

$$f\colon [-inf,inf] \times [-inf,inf] \rightarrow [0,255]$$

- we assume that f[n, m] = 0 outside of the image rectangle

Images as functions

- **An Image** as a function f from R^2 to R^C:
  - f [n, m] gives the intensity at position [n, m]
  - Defined over a rectangle, with a finite range:

$$f\colon \underbrace{[a,b] \times [c,d\,]}_{\text{Domain support}} \rightarrow \underbrace{[0,255]}_{\text{range}}$$

During my PhD Defense

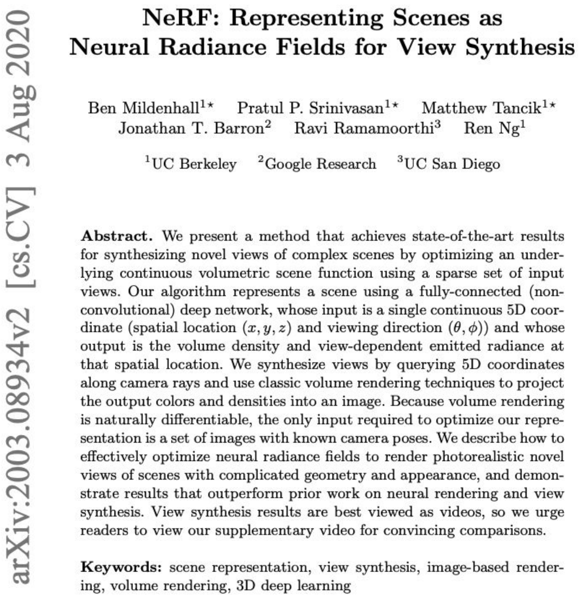
Philips asked me a question about image as a function, and I didn't get it ….

Prof. Philips Isola (MIT)

A year later

“NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis”, ECCV 2020

- Image as a function + neural network
- One of the most important paper in the recent CV development
- >10,000 citations in 5 years



arXiv:2003.08934v2 [cs.CV] 3 Aug 2020

NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis

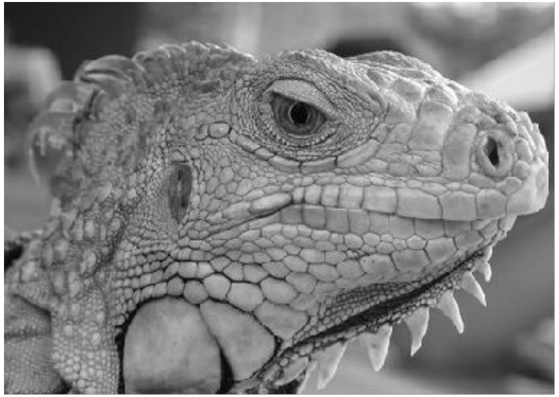
Ben Mildenhall[1*] Pratul P. Srinivasan[1*] Matthew Tancik[1*]
Jonathan T. Barron[2] Ravi Ramamoorthi[3] Ren Ng[1]

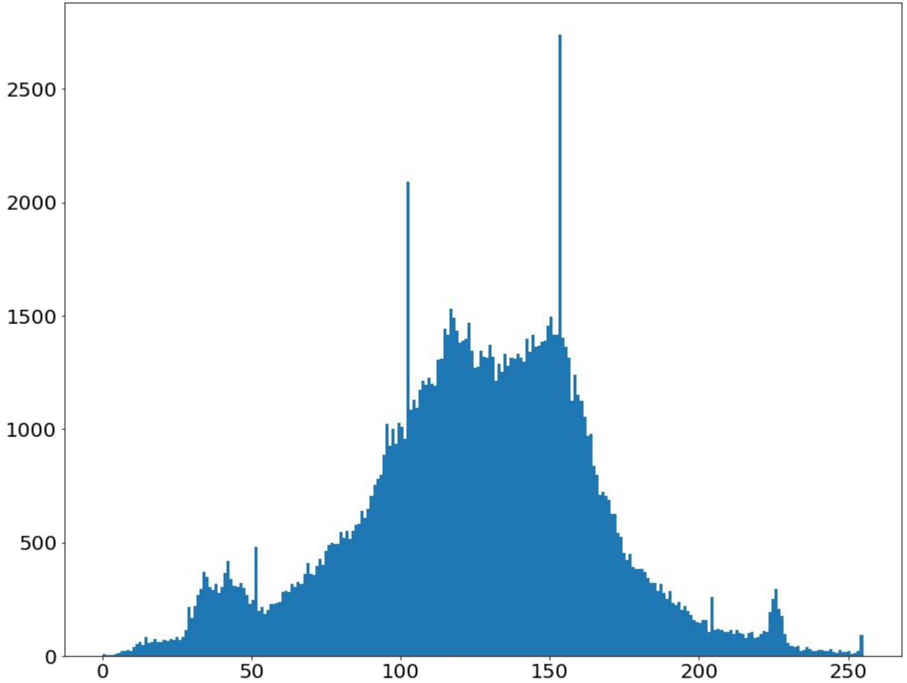
[1]UC Berkeley [2]Google Research [3]UC San Diego

Abstract. We present a method that achieves state-of-the-art results for synthesizing novel views of complex scenes by optimizing an underlying continuous volumetric scene function using a sparse set of input views. Our algorithm represents a scene using a fully-connected (non-convolutional) deep network, whose input is a single continuous 5D coordinate (spatial location (x, y, z) and viewing direction (θ, ϕ)) and whose output is the volume density and view-dependent emitted radiance at that spatial location. We synthesize views by querying 5D coordinates along camera rays and use classic volume rendering techniques to project the output colors and densities into an image. Because volume rendering is naturally differentiable, the only input required to optimize our representation is a set of images with known camera poses. We describe how to effectively optimize neural radiance fields to render photorealistic novel views of scenes with complicated geometry and appearance, and demonstrate results that outperform prior work on neural rendering and view synthesis. View synthesis results are best viewed as videos, so we urge readers to view our supplementary video for convincing comparisons.

Keywords: scene representation, view synthesis, image-based rendering, volume rendering, 3D deep learning

Histograms are also a type of function





Today’s agenda

- Color spaces
- Image sampling and quantization
- Image histograms
- Images as functions
- Filters
- Properties of systems

Some background reading:
Forsyth and Ponce, Computer Vision, Chapter 7

Systems and Filters

Filering:

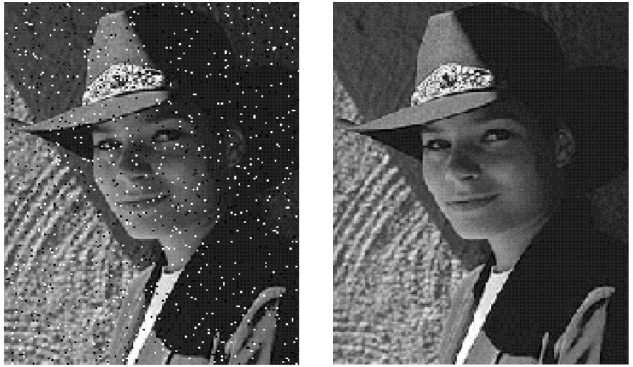
- Forming a new image whose pixel values are transformed from original pixel values

Goals of filters:

- Goal is to extract useful information from images, or transform images into another domain where we can modify/enhance image properties
  - Features (edges, corners, blobs…)
  - super-resolution; in-painting; de-noising

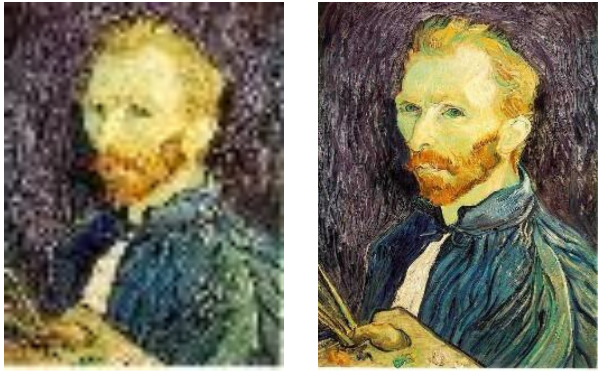
Applications of filters

De-noising

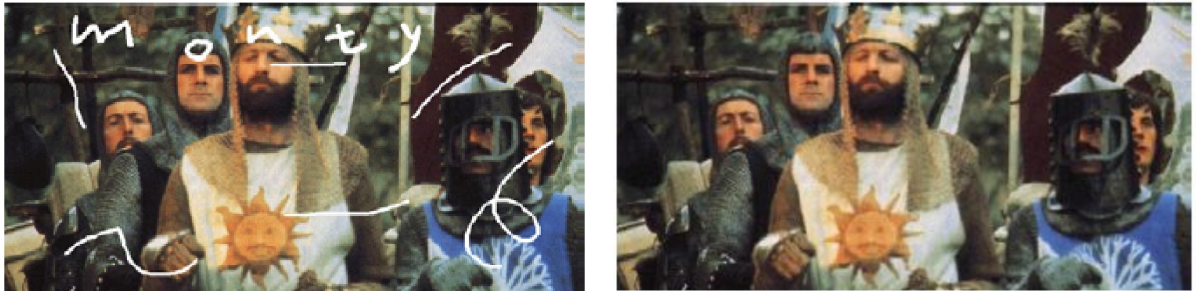


Salt and pepper noise

Super-resolution



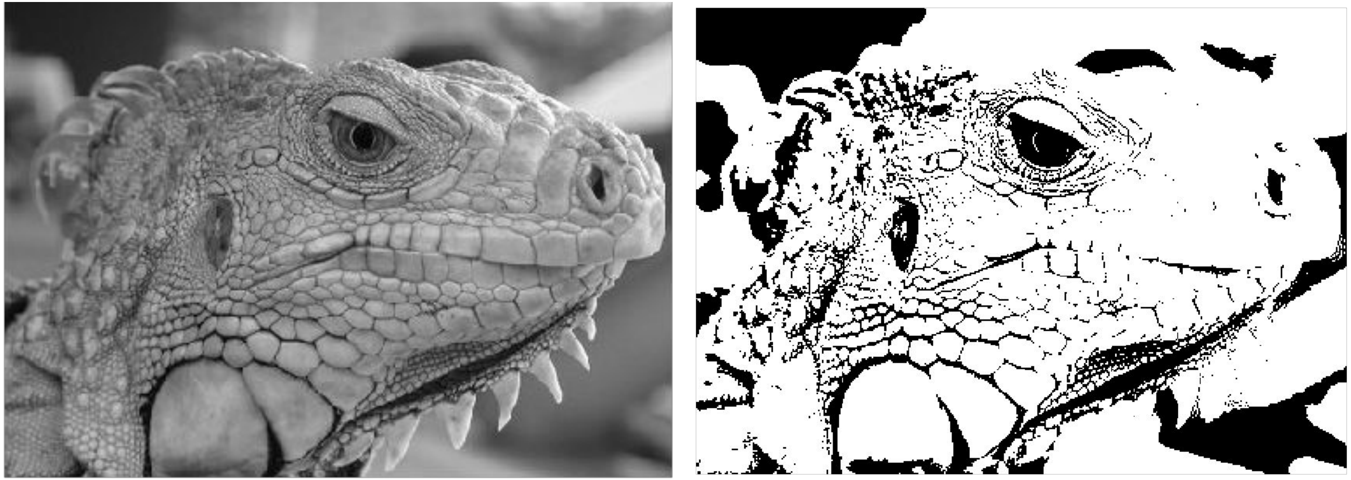
In-painting



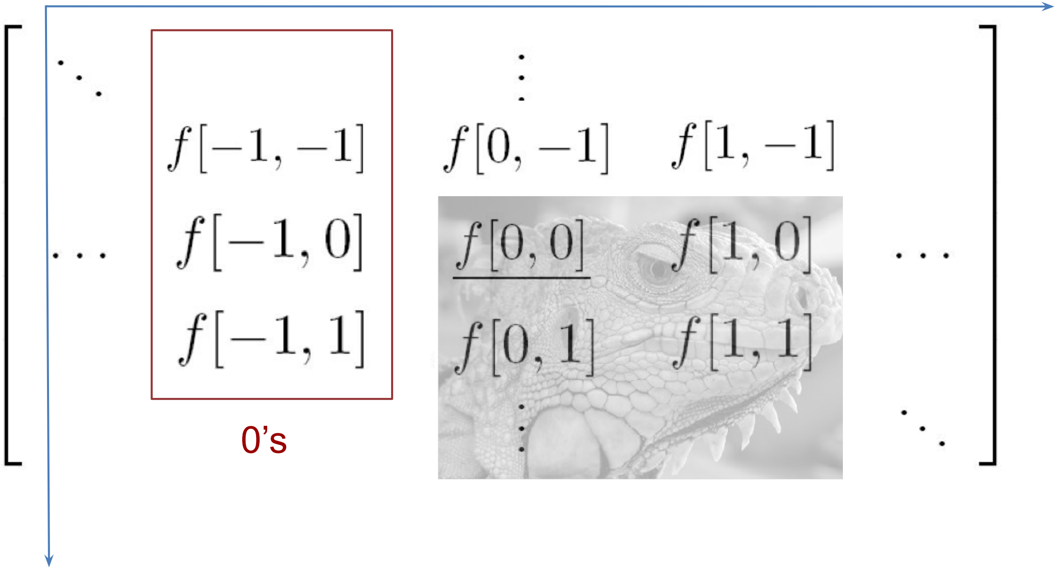
Bertamio et al

Intuition behind systems

- We will view systems as a sequence of filters applied to an image
    - function v.s. functions of functions



Repea: Images produce a 2D matrix with pixel intensities at every location



$$f[n,m] = \begin{bmatrix} \ddots & & \vdots & & \\ & f[-1,-1] & f[0,-1] & f[1,-1] & \\ \ldots & f[-1,0] & \underline{f[0,0]} & f[1,0] & \ldots \\ & f[-1,1] & f[0,1] & f[1,1] & \\ & & \vdots & & \ddots \end{bmatrix}$$

0's

Notation for discrete functions

Systems use Filters

- we define a system as a unit that converts an input function f[n,m] into an output (or response) function g[n,m]
  - where (n,m) index into the function
  - In the case for images, (n,m) represents the **spatial position in the image.**

$$f[n,m] \rightarrow \boxed{\text{System } \mathcal{S}} \rightarrow g[n,m]$$

2D discrete system (system is a sequence of filters)

S is the **system operator**, defined as a mapping or assignment of possible inputs f[n,m] to some possible outputs g[n,m].

$$f[n,m] \rightarrow \boxed{\text{System } \mathcal{S}} \rightarrow g[n,m]$$

Other notations:

$$g = \mathcal{S}[f], \quad g[n,m] = \mathcal{S}\{f[n,m]\}$$

$$f[n,m] \xrightarrow{\mathcal{S}} g[n,m]$$

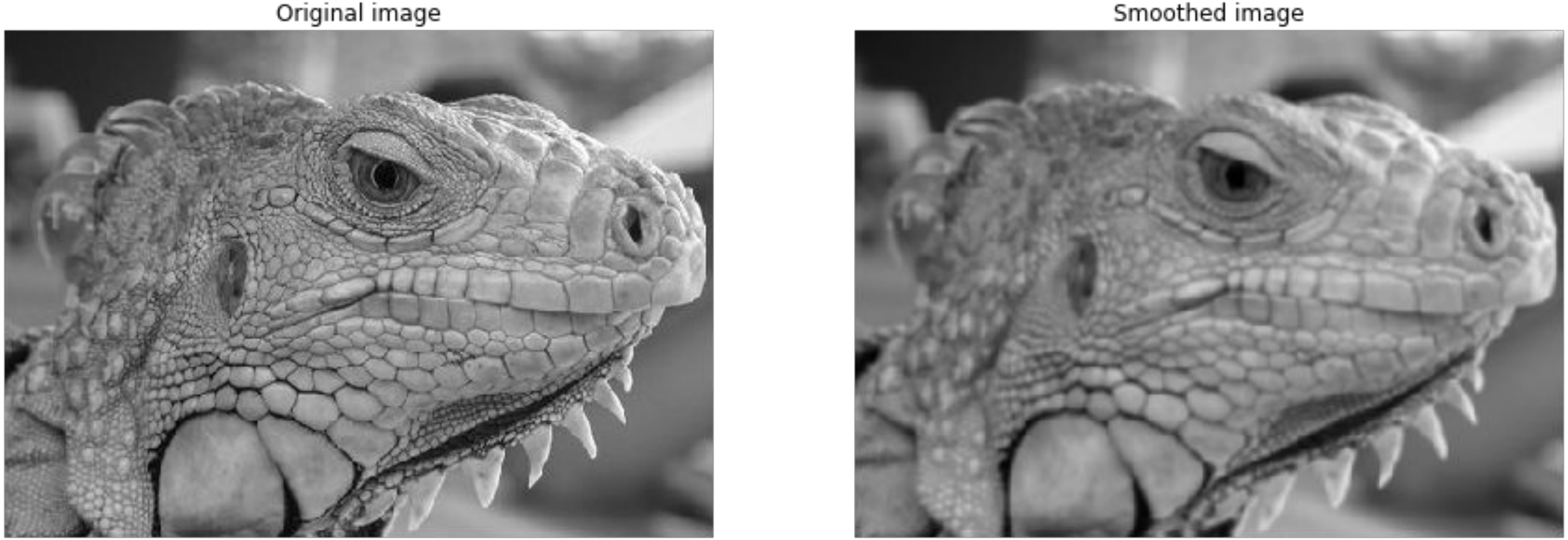
Filter example #1: Moving Average

Original image

Q. What do you think will happen to the photo if we use a moving average filter?

Assume that the moving average replaces each pixel with an average value of itself and all its neighboring pixels.

Filter example #1: Moving Average

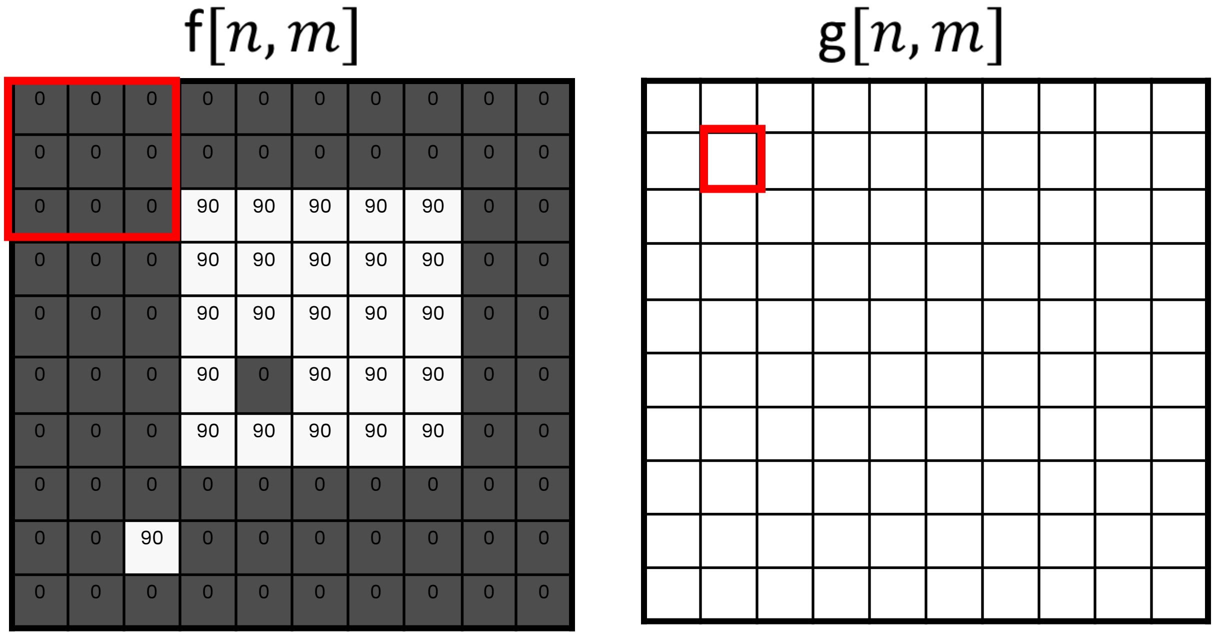


Original image

Smoothed image

Visualizing what happens with a moving average filter

The red box is the **h** matrix



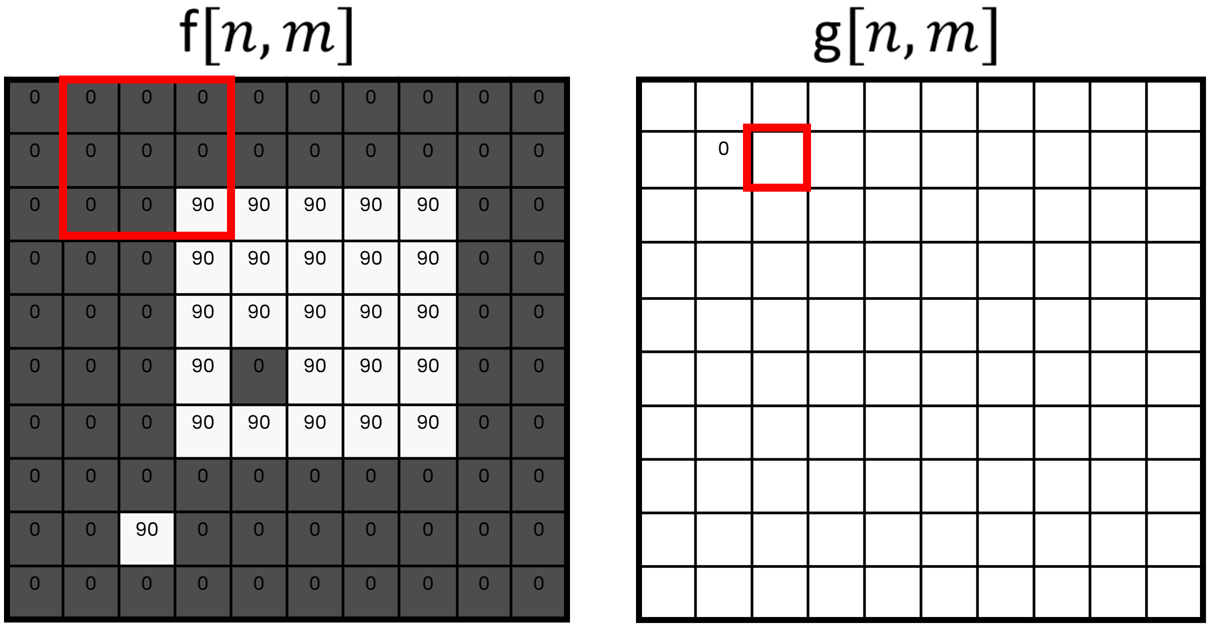
$f[n,m]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$g[n,m]$

Courtesy of S. Seitz

Visualizing what happens with a moving average filter



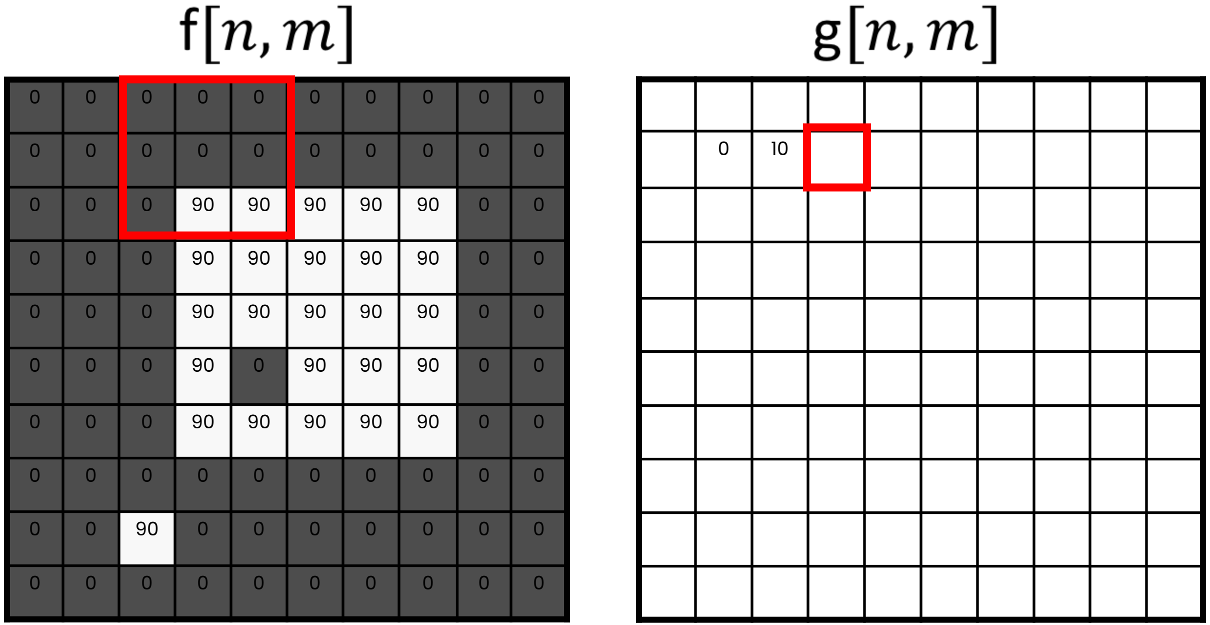
$\mathrm{f}[n,m]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$\mathrm{g}[n,m]$

	0								

Visualizing what happens with a moving average filter



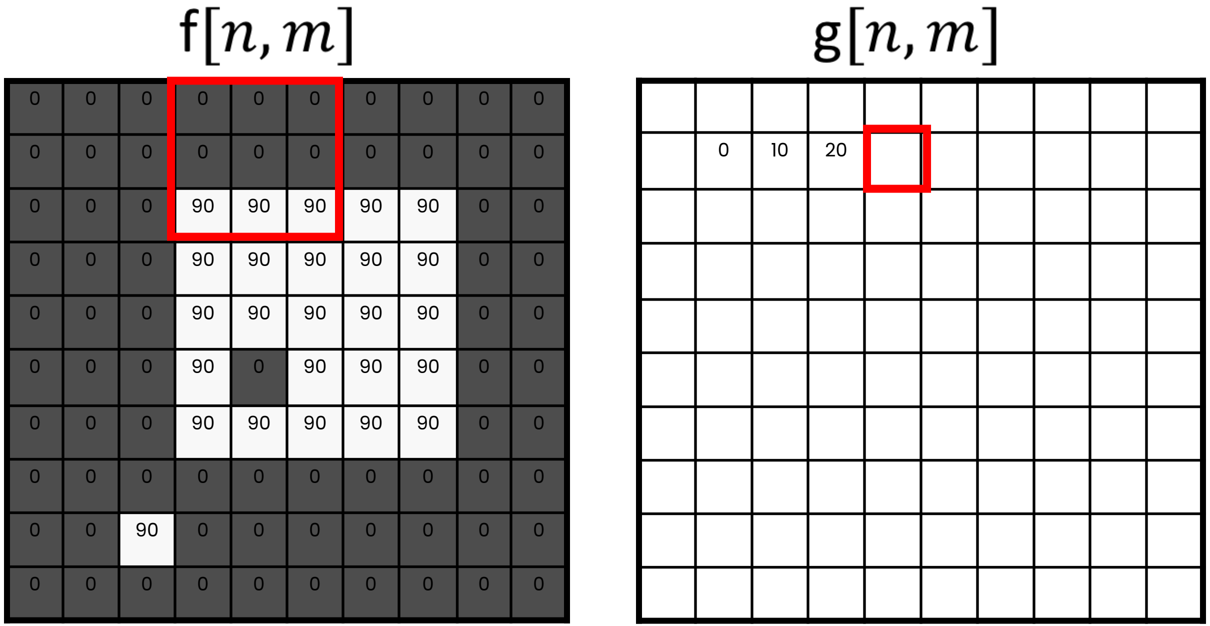
$f[n,m]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$g[n,m]$

	0	10							

Visualizing what happens with a moving average filter



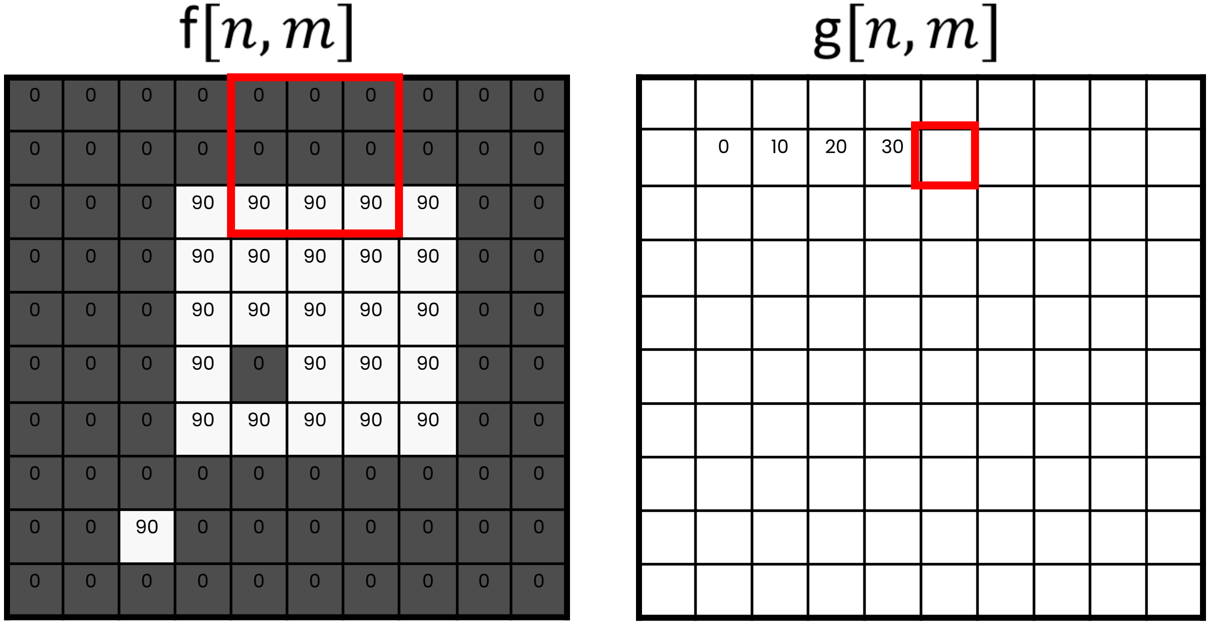
$\mathrm{f}[n,m]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$\mathrm{g}[n,m]$

	0	10	20						

Visualizing what happens with a moving average filter



$f[n,m]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$g[n,m]$

	0	10	20	30					

Visualizing what happens with a moving average filter

$\mathrm{f}[n,m]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

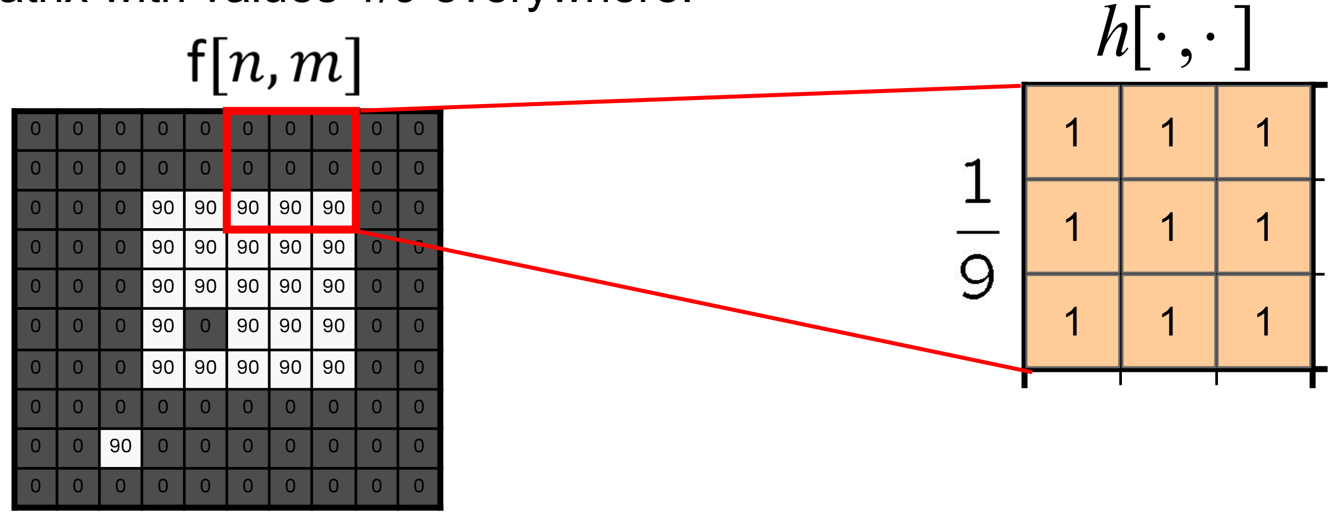
$\mathrm{g}[n,m]$

	0	10	20	30	30	30	20	10	
	0	20	40	60	60	60	40	20	
	0	30	60	90	90	90	60	30	
	0	30	50	80	80	90	60	30	
	0	30	50	80	80	90	60	30	
	0	20	30	50	50	60	40	20	
	10	20	30	30	30	30	20	10	
	10	10	10	0	0	0	0	0	

Visual interpretation of moving average

A moving average over a 3 × 3 neighborhood window

h is a 3x3 matrix with values 1/9 everywhere.



$\text{f}[n,m]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$h[\cdot,\cdot]$

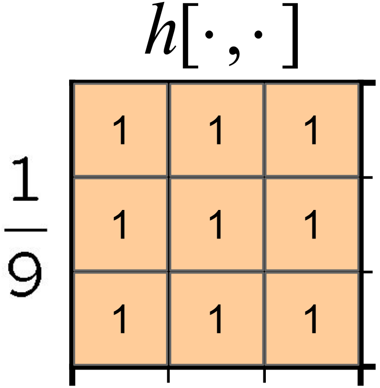
$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Visual interpretation of moving average

A moving average over a 3 × 3 neighborhood window

h is a 3x3 matrix with values 1/9 everywhere.

Q. Why are the values 1/9?



$$h[\cdot,\cdot]$$

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Filter example #1: Moving Average

In summary:

- This filter “Replaces” each pixel with an average of its neighborhood.

- Achieve smoothing effect (remove sharp features)

$$h[\cdot,\cdot]$$

$$\frac{1}{9}\begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

Mathematical interpretation of moving average

How do we represent applying this filter mathematically?

$$f[n,m] \rightarrow \boxed{\text{System } \mathcal{S}} \rightarrow g[n,m]$$

$$h[\cdot,\cdot]$$

$$\frac{1}{9}\begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline\end{array}$$

Mathematical interpretation of moving average

How do we represent applying this filter mathematically?

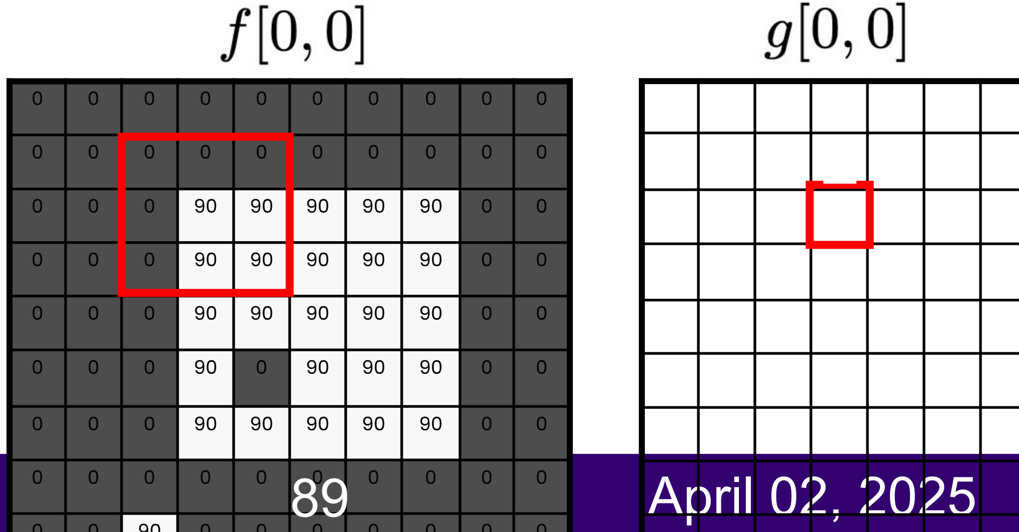
$$f[n,m] \rightarrow \boxed{\text{System } \mathcal{S}} \rightarrow g[n,m]$$

$$h[\cdot,\cdot]$$

$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

Mathematical formulation of moving average

$$f[n,m] \rightarrow \boxed{\text{System } \mathcal{S}} \rightarrow g[n,m]$$



$f[0,0]$

$g[0,0]$

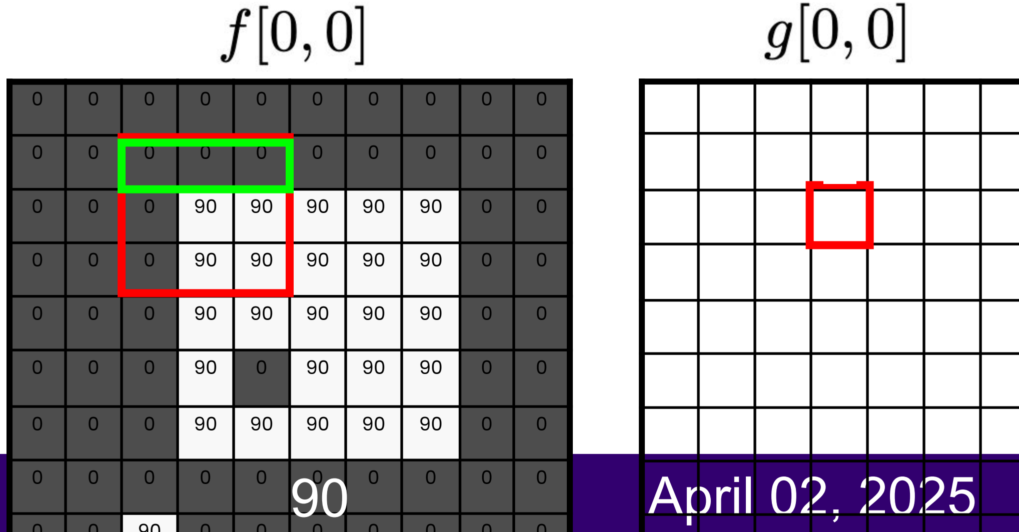
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0

Mathematical formulation of moving average

$$f[n,m] \rightarrow \boxed{\text{System } \mathcal{S}} \rightarrow g[n,m]$$

$$\boxed{g[0,0] = f[-1,-1] + f[-1,0] + f[-1,1]}$$

$$+ \; \ldots$$



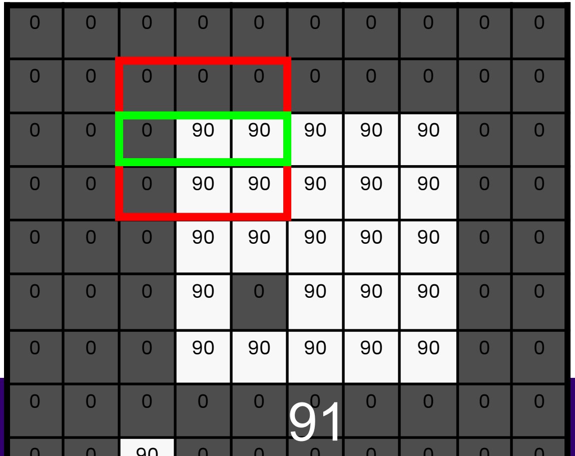
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0

Mathematical formulation of moving average

$$f[n,m] \rightarrow \boxed{\text{System } \mathcal{S}} \rightarrow g[n,m]$$

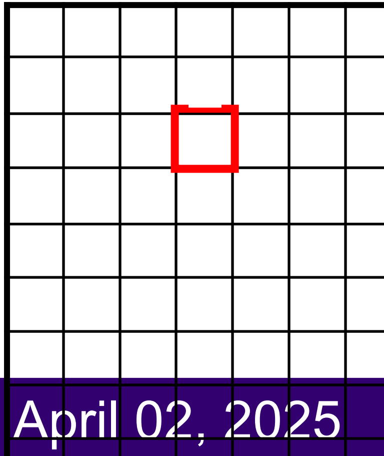
$$\begin{aligned} g[0,0] = {} & f[-1,-1] + f[-1,0] + f[-1,1] \\ & + \boxed{f[0,-1] + f[0,0] + f[0,1]} \\ & + \ \ldots \end{aligned}$$

$f[0,0]$



0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0

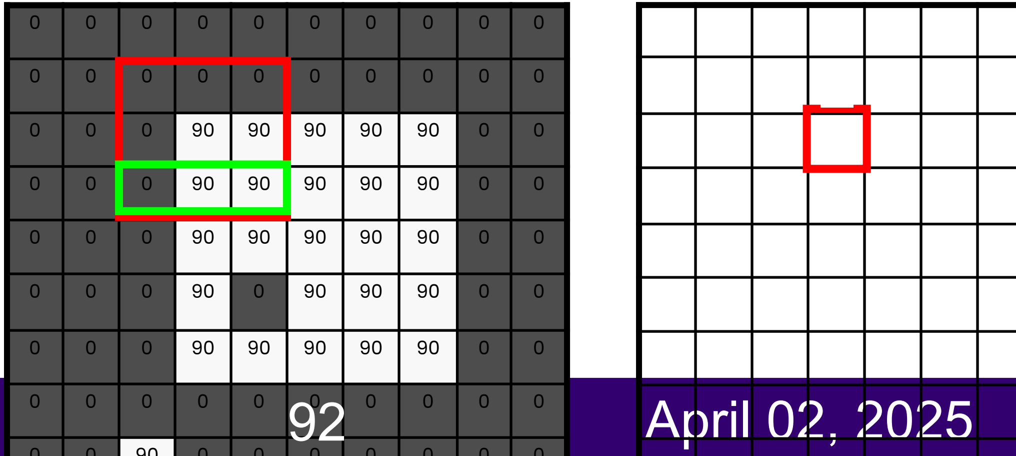
$g[0,0]$



Mathematical formulation of moving average

$$f[n,m] \rightarrow \boxed{\text{System } \mathcal{S}} \rightarrow g[n,m]$$

$$\begin{aligned} g[0,0] = {} & f[-1,-1] + f[-1,0] + f[-1,1] \\ & + f[0,-1] + f[0,0] + f[0,1] \\ & + \boxed{f[1,-1] + f[1,0] + f[1,1]} \end{aligned}$$



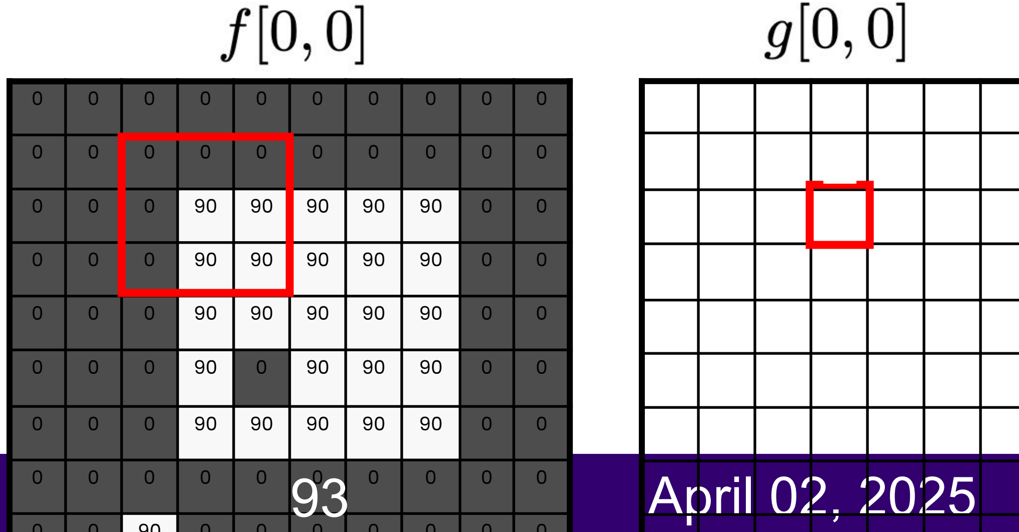
$f[0,0]$

$g[0,0]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0

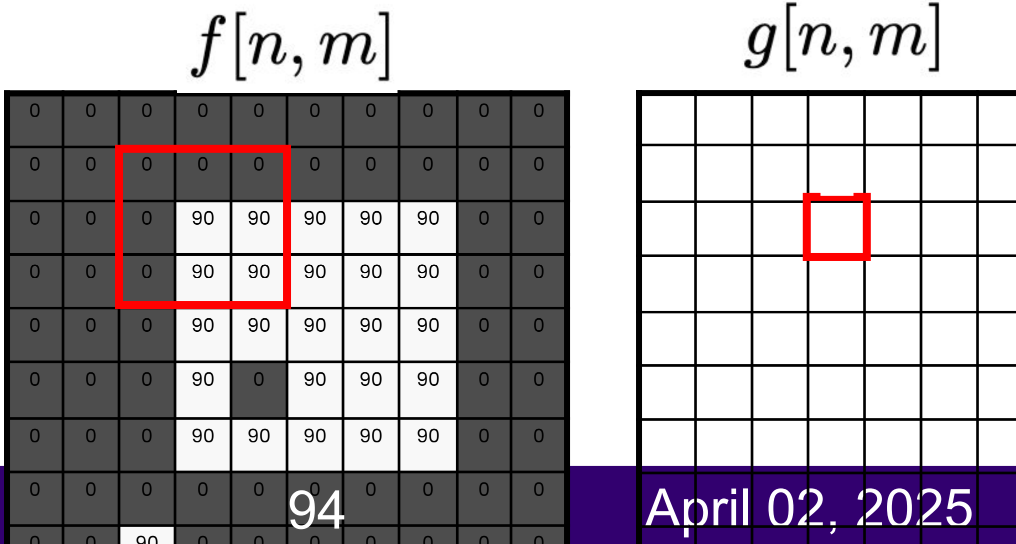
Lastly, divide by 1/9

$$g[0,0] = \frac{1}{9}[f[-1,-1] + f[-1,0] + f[-1,1] \\ + f[0,-1] + f[0,0] + f[0,1] \\ + f[1,-1] + f[1,0] + f[1,1]]$$



0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0

Now, instead of [0, 0], let’s do [n, m]



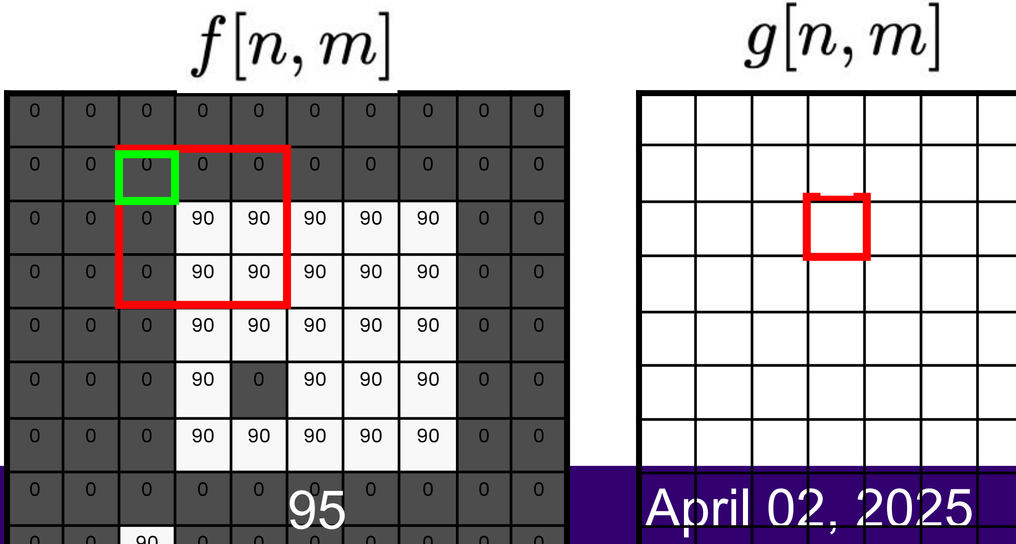
$f[n, m]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0

$g[n, m]$

Now, instead of [0, 0], let's do [n, m]

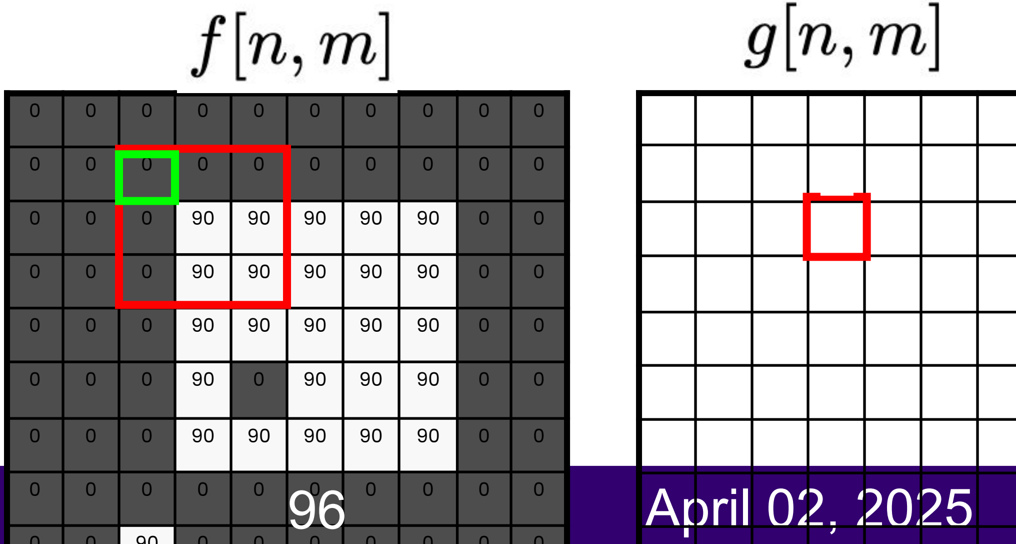
$$g[n, m] = \ \ldots$$



0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0

Now, instead of [0, 0], let’s do [n, m]

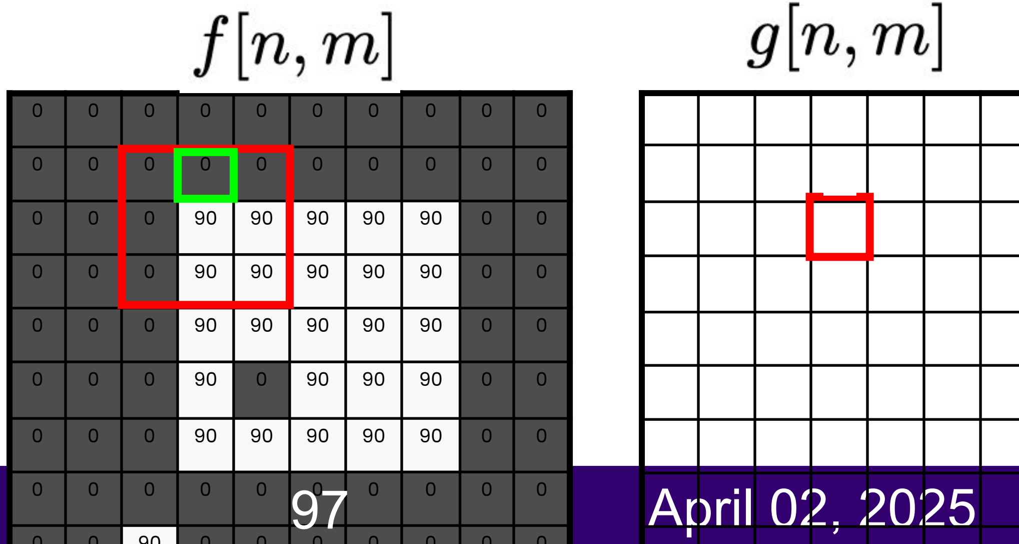
$$g[n,m] = \boxed{f[n-1,m-1]} + \ \ldots$$



0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0

Now, instead of [0, 0], let’s do [n, m]

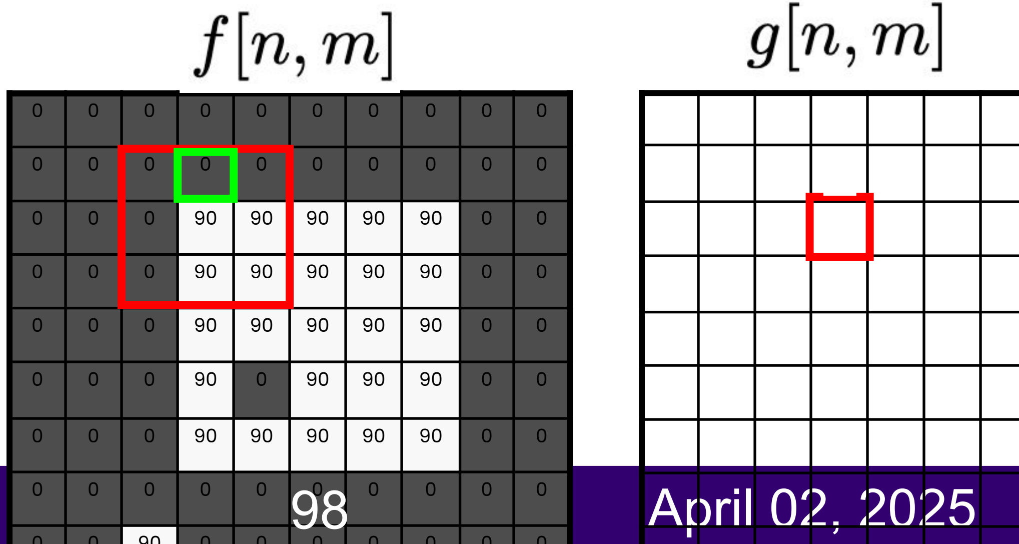
$$g[n,m] = f[n-1, m-1] + \ \ldots$$



0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0

Now, instead of [0, 0], let's do [n, m]

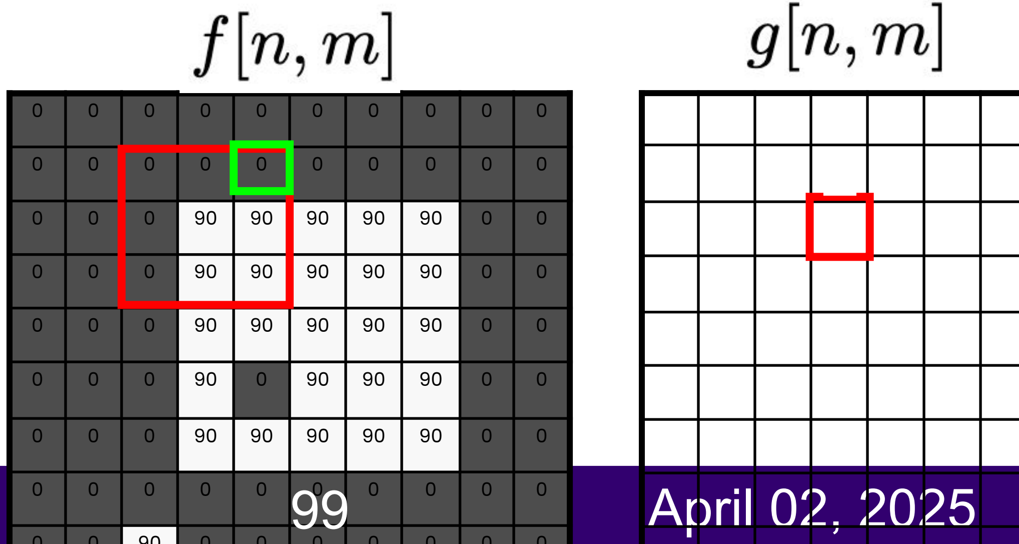
$$g[n,m] = f[n-1,m-1] + \boxed{f[n-1,m]} + \;\ldots$$



0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0

Now, instead of [0, 0], let’s do [n, m]

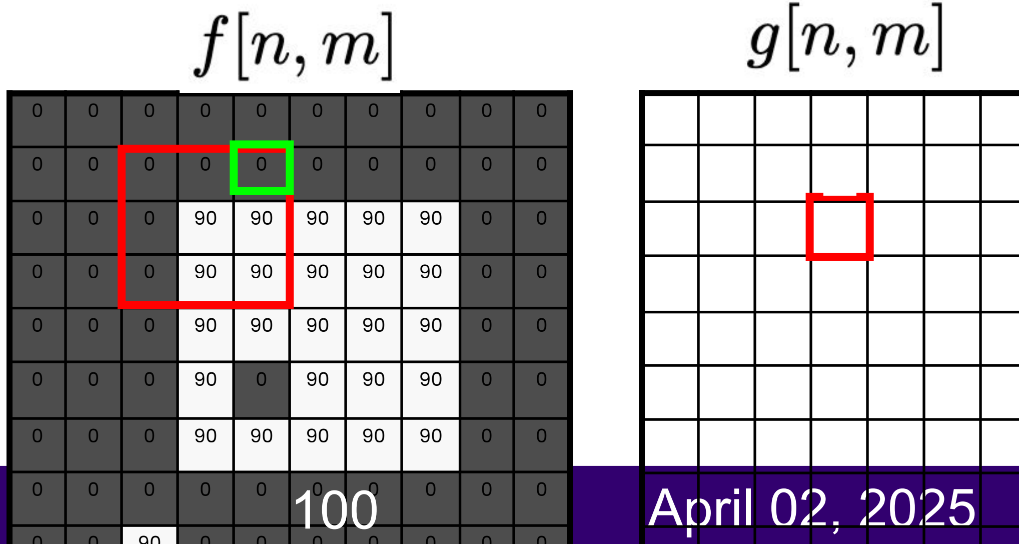
$$g[n,m] = f[n-1, m-1] + f[n-1, m] + \ \dots$$



0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0

Now, instead of [0, 0], let’s do [n, m]

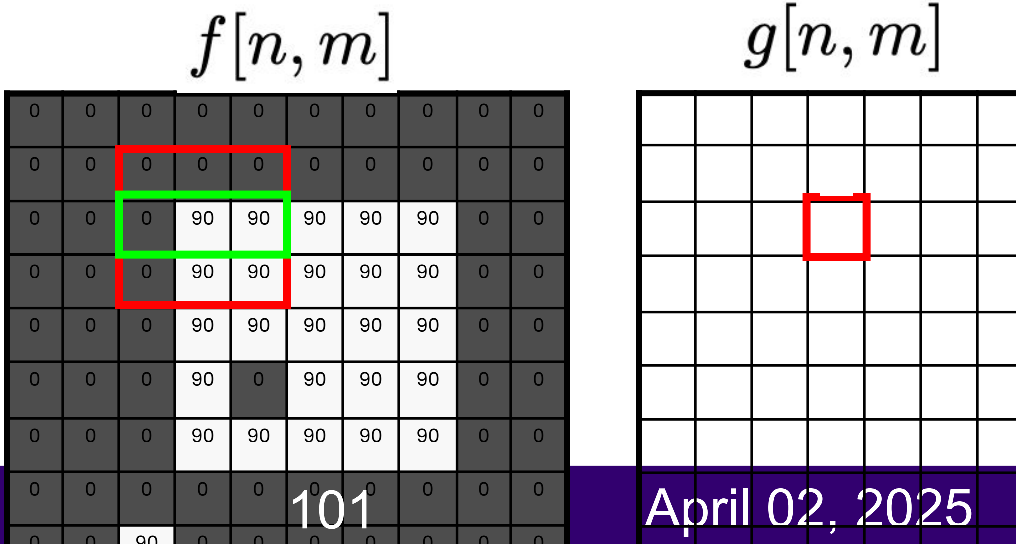
$$g[n,m] = f[n-1,m-1] + f[n-1,m] + \boxed{f[n-1,m+1]}$$



0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0

Now, instead of [0, 0], let’s do [n, m]

$$g[n,m] = f[n-1,m-1] + f[n-1,m] + f[n-1,m+1]$$
$$\boxed{+ f[n,m-1] + f[n,m] + f[n,m+1]}$$



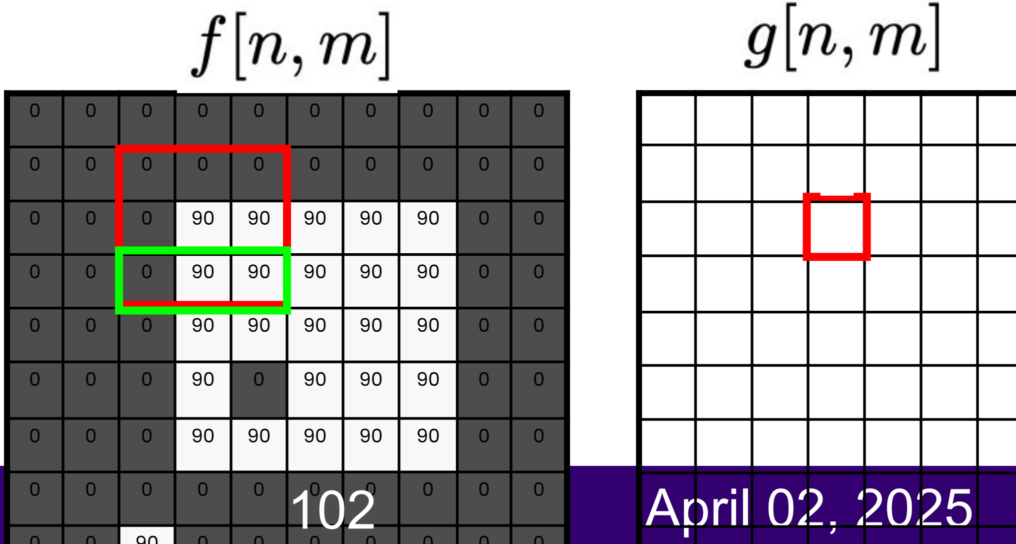
$f[n,m]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0

$g[n,m]$

Now, instead of [0, 0], let’s do [n, m]

$$\begin{aligned} g[n,m] = {} & f[n-1,m-1] + f[n-1,m] + f[n-1,m+1] \\ & + f[n,m-1] + f[n,m] + f[n,m+1] \\ & \boxed{+ f[n+1,m-1] + f[n+1,m] + f[n+1,m+1]} \end{aligned}$$



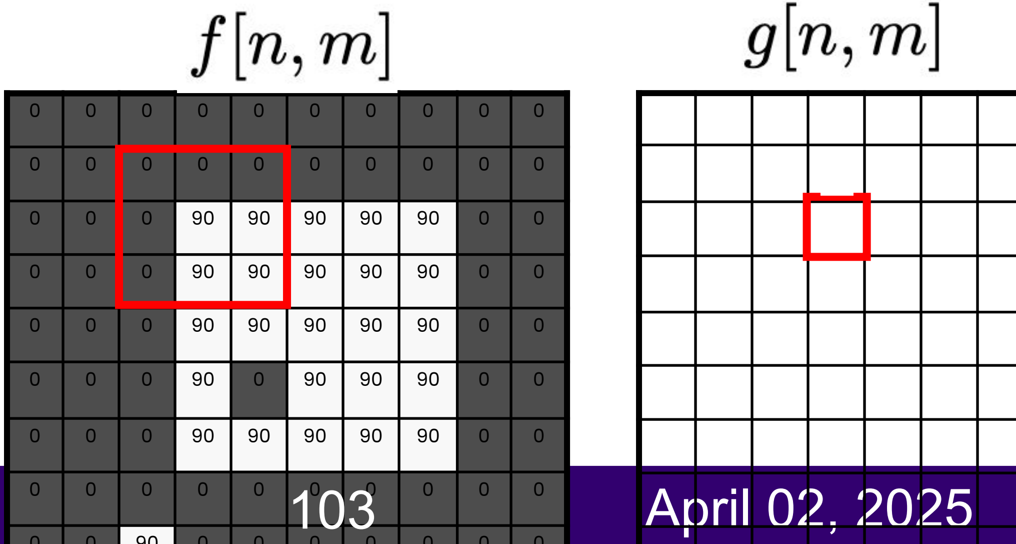
$f[n,m]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0

$g[n,m]$

Lastly, divide by 1/9

$$
\begin{aligned}
g[n,m] = \frac{1}{9}[f[n-1,m-1] &+ f[n-1,m] + f[n-1,m+1] \\
&+ f[n,m-1] + f[n,m] + f[n,m+1] \\
&+ f[n+1,m-1] + f[n+1,m] + f[n+1,m+1]]
\end{aligned}
$$



$f[n,m]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0

$g[n,m]$

Mathematical formulation of moving average

We can re-write the equation using summations

$$g[n,m] = \frac{1}{9} \sum_{k=??}^{??} \sum_{l=??}^{??} f[k,l]$$

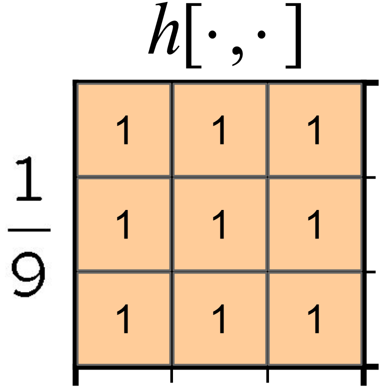
$$h[\cdot,\cdot] \qquad \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Q. What values will **k** take?

Mathematical formulation of moving average

How do we represent applying this filter mathematically?

$$g[n,m] = \frac{1}{9} \sum_{k=n-1}^{n+1} \sum_{l=??}^{??} f[k,l]$$



$$h[\cdot,\cdot] \quad \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

k goes from n-1 to n+1

Mathematical formulation of moving average

How do we represent applying this filter mathematically?

$$g[n,m] = \frac{1}{9} \sum_{k=n-1}^{n+1} \sum_{l=??}^{??} f[k,l]$$

$$h[\cdot,\cdot]$$

$$\frac{1}{9}\begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

Q. What values will **l** take?

Mathematical formulation of moving average

How do we represent applying this filter mathematically?

$$g[n,m] = \frac{1}{9} \sum_{k=n-1}^{n+1} \sum_{l=m-1}^{m+1} f[k,l]$$

$$h[\cdot,\cdot]$$

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

l goes from m-1 to m+1

Math formula for the moving average filter

A moving average over a 3 × 3 neighborhood window

We can write this operation mathematically:

$$g[n,m] = \frac{1}{9} \sum_{k=n-1}^{n+1} \sum_{l=m-1}^{m+1} f[k,l]$$

$$h[\cdot,\cdot]$$

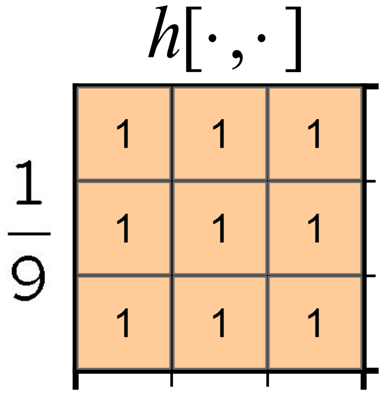
$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Rewriting this formula

We are almost done. Let’s rewrite this formula a little bit

Let $k' = n - k$

$$g[n,m] = \frac{1}{9} \sum_{k=n-1}^{n+1} \sum_{l=m-1}^{m+1} f[k,l]$$



$h[\cdot,\cdot]$

$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

Rewriting this formula

We are almost done. Let's rewrite this formula a little bit

Let $k' = n - k$

therefore, $k = n - k'$

$$g[n,m] = \frac{1}{9} \sum_{k=n-1}^{n+1} \sum_{l=m-1}^{m+1} f[k,l]$$

Now we can replace k in the equation above

$$h[\cdot,\cdot]$$

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Rewriting this formula

We are almost done. Let’s rewrite this formula a little bit

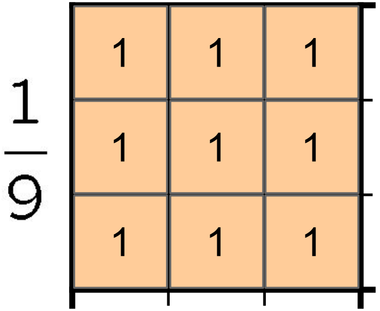
Let $k' = n - k$

therefore, $k = n - k'$

$$g[n,m] = \frac{1}{9} \sum_{k=n-1}^{n+1} \sum_{l=m-1}^{m+1} f[k,l]$$

$$g[n,m] = \frac{1}{9} \sum_{n-k'=n-1}^{n-k'=n+1} \sum_{l=m-1}^{m+1} f[n-k',l]$$

$h[\cdot,\cdot]$

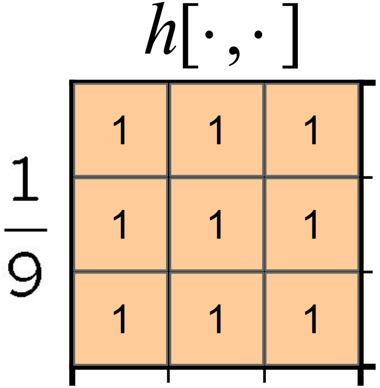


$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

Rewriting this formula

So now we have this:

$$g[n,m] = \frac{1}{9} \sum_{n-k'=n-1}^{n-k'=n+1} \sum_{l=m-1}^{m+1} f[n-k', l]$$



$$h[\cdot,\cdot]$$

$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

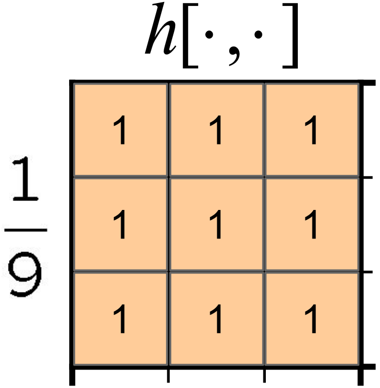
Rewriting this formula

So now we have this:

$$g[n,m] = \frac{1}{9} \sum_{n-k'=n-1}^{n-k'=n+1} \sum_{l=m-1}^{m+1} f[n-k',l]$$

We can simplify the equations in red:

$$g[n,m] = \frac{1}{9} \sum_{k'=1}^{k'=-1} \sum_{l=m-1}^{m+1} f[n-k',l]$$



$$h[\cdot,\cdot] \quad \frac{1}{9}\begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

Rewriting this formula

So now we have this:

$$g[n,m] = \frac{1}{9} \sum_{\color{red}{k'=1}}^{\color{red}{k'=-1}} \sum_{l=m-1}^{m+1} f[n-k',l]$$

Remember that summations are just for-loops!!

$$h[\cdot,\cdot]$$

$$\frac{1}{9}\begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

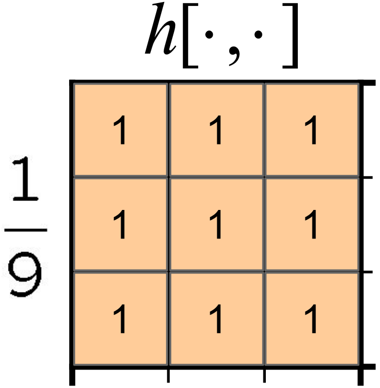
Rewriting this formula

So now we have this:

$$g[n,m] = \frac{1}{9} \sum_{k'=1}^{k'=-1} \sum_{l=m-1}^{m+1} f[n-k',l]$$

Remember that summations are just for-loops!!

$$g[n,m] = \frac{1}{9} \sum_{k'=-1}^{1} \sum_{l=m-1}^{m+1} f[n-k',l]$$



$$h[\cdot,\cdot]$$

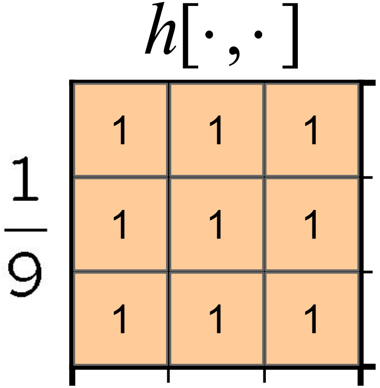
$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Rewriting this formula

One last change: since there are no more k and only k',
let's just write k' as k

$$g[n,m] = \frac{1}{9} \sum_{k'=-1}^{1} \sum_{l=m-1}^{m+1} f[n-k',l]$$

$$g[n,m] = \frac{1}{9} \sum_{k=-1}^{1} \sum_{l=m-1}^{m+1} f[n-k,l]$$



$$h[\cdot,\cdot] \quad \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Mathematical interpretation of moving average

Let’s repeat for l, just like we did for k

$$g[n,m] = \frac{1}{9}\sum_{k=n-1}^{n+1}\sum_{l=m-1}^{m+1} f[k,l]$$

$$= \frac{1}{9}\sum_{k=-1}^{1}\sum_{l=-1}^{1} f[n-k,m-l]$$

$$h[\cdot,\cdot]$$

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Mathematical interpretation of moving average

Let’s repeat for l, just like we did for k

$$g[n,m] = \frac{1}{9} \sum_{k=n-1}^{n+1} \sum_{l=m-1}^{m+1} f[k,l]$$

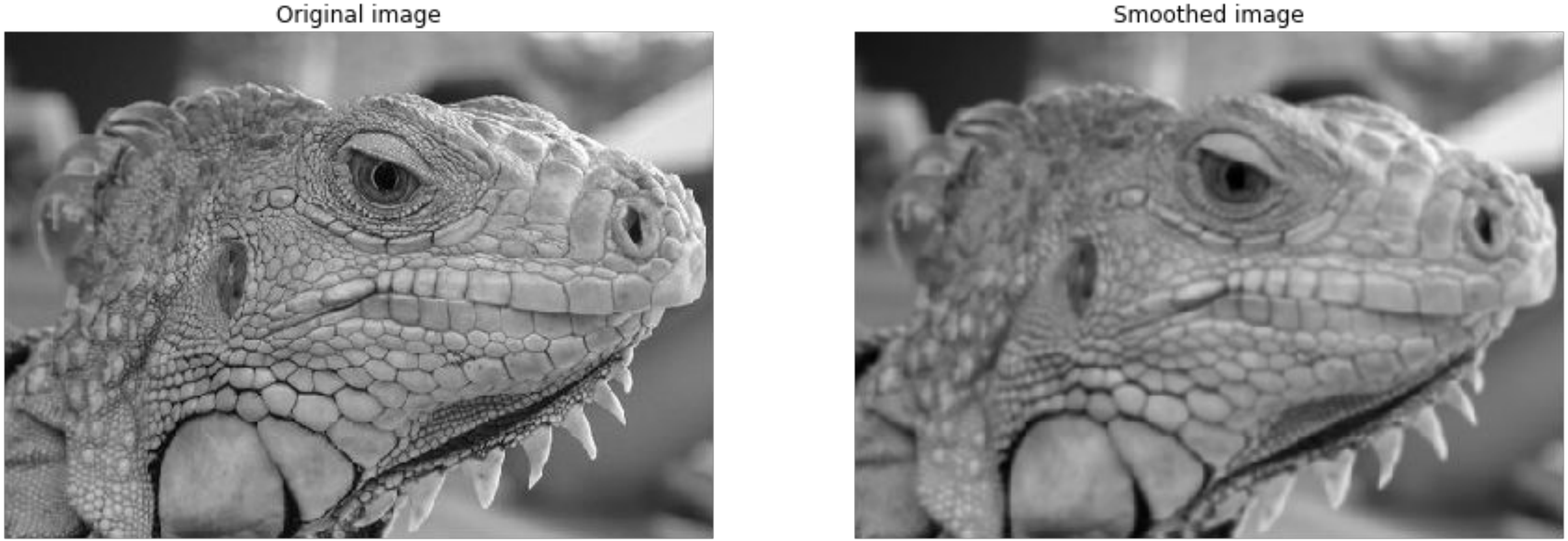
$$= \frac{1}{9} \sum_{k=-1}^{1} \sum_{l=-1}^{1} f[n-k, m-l]$$

$$h[\cdot,\cdot]$$

$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Q: how to use a larger 5x5 filter?

Filter example #1: Moving Average

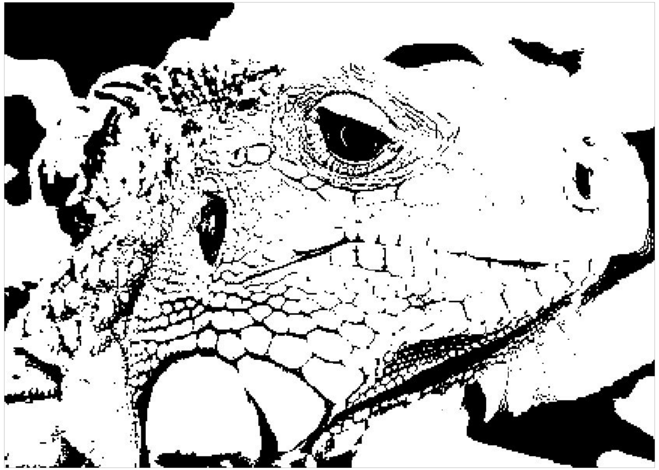


Original image

Smoothed image

Filter example #2: Image Segmentation

Q. How would you use pixel values to design a filter to segment an image so that you only keep around the edges?

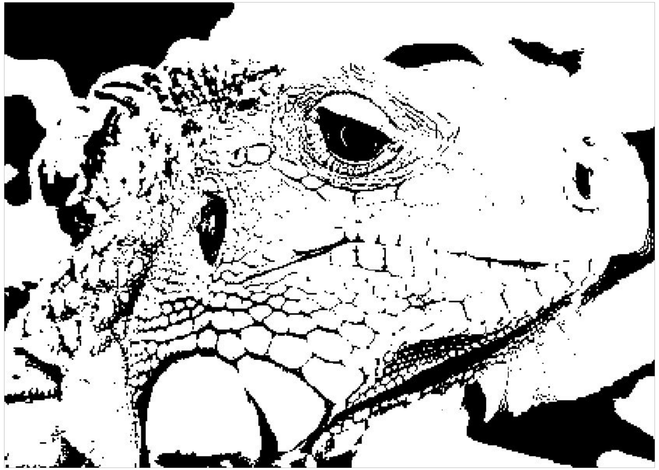


Filter example #2: Image Segmentation

- Use a simple pixel threshold:

$$g[n,m] = \begin{cases} 255, & f[n,m] > 100 \\ 0, & \text{otherwise.} \end{cases}$$

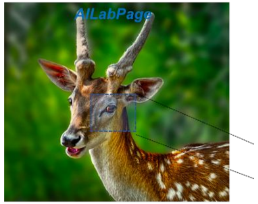
Exercise: Is this linear or non-linear operation?

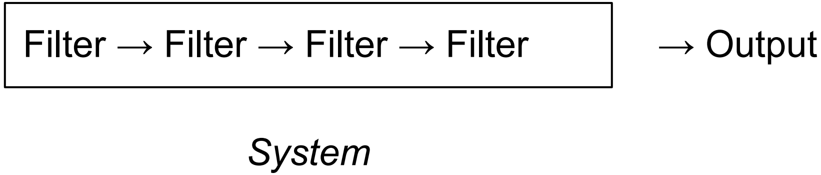


Summary so far

- Beyond examples we have seen today, there are A HUGE number of possible filters we can design.
- Discrete systems, with filters, convert input discrete signals and convert them into something more meaningful.
- What are ways we can category the space of possible systems?

From a signal processing view





Filter → Filter → Filter → Filter

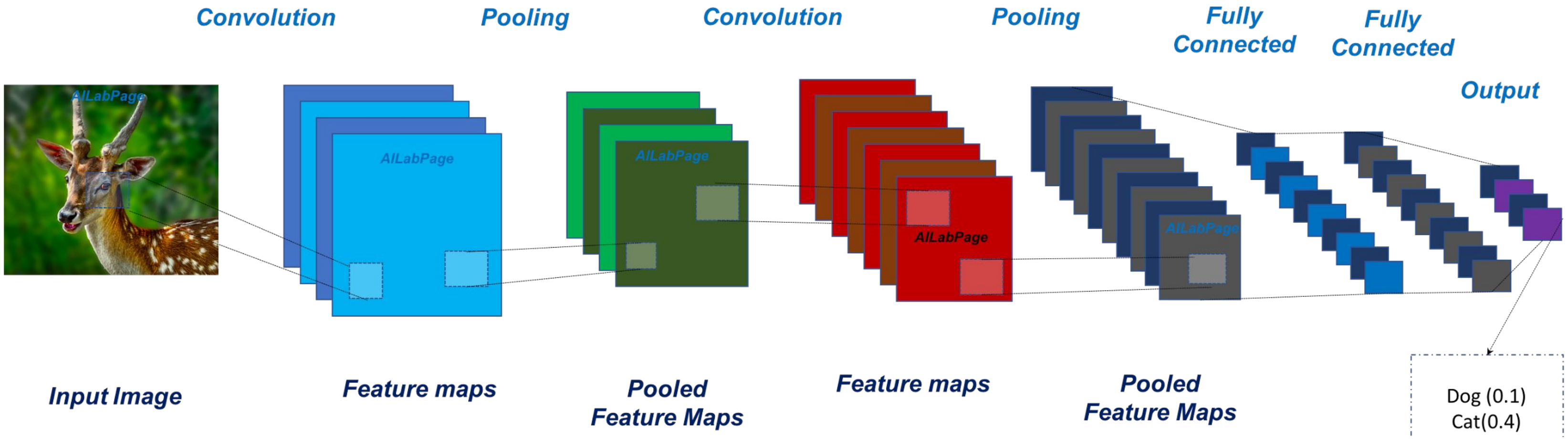
→ Output

System

Input Image

In ML langauge

- Neural networks and specifically **convolutional** neural networks are a sequence of filters (except they are a non-linear system) that contains multiple individual linear sub-systems.
- filter as layer & system as model



Today’s agenda

- Color spaces
- Image sampling and quantization
- Image histograms
- Images as functions
- Filters
- Properties of systems

Properties of systems

- Amplitude properties:
  - Additivity

$$\mathcal{S}[f_i[n,m] + f_j[n,m]] = \mathcal{S}[f_i[n,m]] + \mathcal{S}[f_j[n,m]]$$

Example question:

Q. Is the moving average filter additive?

$$\mathcal{S}[f_i[n,m] + f_j[n,m]] = \mathcal{S}[f_i[n,m]] + \mathcal{S}[f_j[n,m]]$$

$$h[\cdot,\cdot]$$

$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

How would you prove it?

$$g[n,m] = \frac{1}{9}\sum_{k=-1}^{1}\sum_{l=-1}^{1} f[n-k, m-l]$$

Example question:

$$\mathcal{S}[f_i[n,m] + f_j[n,m]] = \mathcal{S}[f_i[n,m]] + \mathcal{S}[f_j[n,m]]$$

Let $f'[n,m] = f_i[n,m] + f_j[n,m]$

$$h[\cdot,\cdot]$$

$$\frac{1}{9}\begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

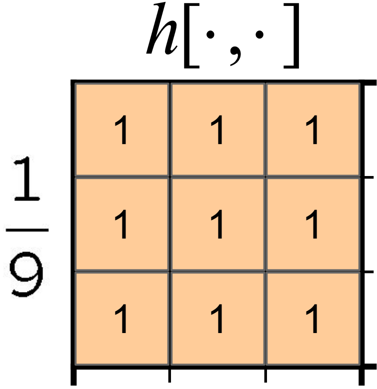
$$g[n,m] = \frac{1}{9}\sum_{k=-1}^{1}\sum_{l=-1}^{1} f[n-k, m-l]$$

Example question:

$$\mathcal{S}[f_i[n,m] + f_j[n,m]] = \mathcal{S}[f_i[n,m]] + \mathcal{S}[f_j[n,m]]$$

Let $f'[n,m] = f_i[n,m] + f_j[n,m]$

$$\mathcal{S}[f_i[n,m] + f_j[n,m]] = \mathcal{S}[f'[n,m]]$$



$h[\cdot,\cdot]$

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

$$g[n,m] = \frac{1}{9}\sum_{k=-1}^{1}\sum_{l=-1}^{1} f[n-k, m-l]$$

Example question:

$$\mathcal{S}[f_i[n,m] + f_j[n,m]] = \mathcal{S}[f_i[n,m]] + \mathcal{S}[f_j[n,m]]$$

Let $f'[n,m] = f_i[n,m] + f_j[n,m]$

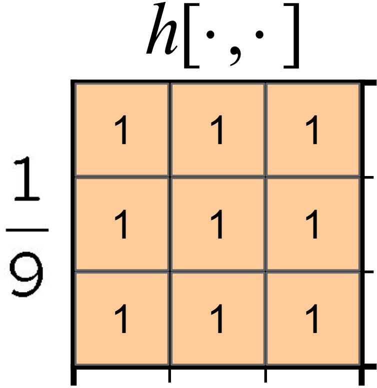
$$\begin{aligned}\mathcal{S}[f_i[n,m] + f_j[n,m]] &= \mathcal{S}[f'[n,m]] \\ &= \frac{1}{9}\sum_{k=-1}^{1}\sum_{l=-1}^{1} f'[n-k, m-l]\end{aligned}$$

$$h[\cdot,\cdot]$$

$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

$$g[n,m] = \frac{1}{9}\sum_{k=-1}^{1}\sum_{l=-1}^{1} f[n-k, m-l]$$

Example question:



$$h[\cdot,\cdot] \quad \frac{1}{9}\begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline\end{array}$$

$$\mathcal{S}[f_i[n,m] + f_j[n,m]] = \mathcal{S}[f_i[n,m]] + \mathcal{S}[f_j[n,m]]$$

Let $f'[n,m] = f_i[n,m] + f_j[n,m]$

$$g[n,m] = \frac{1}{9}\sum_{k=-1}^{1}\sum_{l=-1}^{1} f[n-k, m-l]$$

$$\begin{aligned}\mathcal{S}[f_i[n,m] + f_j[n,m]] &= \mathcal{S}[f'[n,m]] \\ &= \frac{1}{9}\sum_{k=-1}^{1}\sum_{l=-1}^{1} f'[n-k, m-l] \\ &= \frac{1}{9}\sum_{k=-1}^{1}\sum_{l=-1}^{1} [f_i[n-k, m-l] + f_j[n-k, m-l]]\end{aligned}$$

Example question:

$$h[\cdot,\cdot] \quad \frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

$$g[n,m] = \frac{1}{9}\sum_{k=-1}^{1}\sum_{l=-1}^{1} f[n-k,m-l]$$

$$\mathcal{S}[f_i[n,m] + f_j[n,m]] = \mathcal{S}[f_i[n,m]] + \mathcal{S}[f_j[n,m]]$$

Let $f'[n,m] = f_i[n,m] + f_j[n,m]$

$$\begin{aligned}
\mathcal{S}[f_i[n,m] + f_j[n,m]] &= \mathcal{S}[f'[n,m]] \\
&= \frac{1}{9}\sum_{k=-1}^{1}\sum_{l=-1}^{1} f'[n-k,m-l] \\
&= \frac{1}{9}\sum_{k=-1}^{1}\sum_{l=-1}^{1} [f_i[n-k,m-l] + f_j[n-k,m-l]] \\
&= \frac{1}{9}\sum_{k=-1}^{1}\sum_{l=-1}^{1} f_i[n-k,m-l] + \frac{1}{9}\sum_{k=-1}^{1}\sum_{l=-1}^{1} f_j[n-k,m-l]]
\end{aligned}$$

Example question:

$$h[\cdot,\cdot]$$

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\mathcal{S}[f_i[n,m] + f_j[n,m]] = \mathcal{S}[f_i[n,m]] + \mathcal{S}[f_j[n,m]]$$

Let $f'[n,m] = f_i[n,m] + f_j[n,m]$

$$g[n,m] = \frac{1}{9}\sum_{k=-1}^{1}\sum_{l=-1}^{1} f[n-k, m-l]$$

$$\begin{aligned}
\mathcal{S}[f_i[n,m] + f_j[n,m]] &= \mathcal{S}[f'[n,m]] \\
&= \frac{1}{9}\sum_{k=-1}^{1}\sum_{l=-1}^{1} f'[n-k, m-l] \\
&= \frac{1}{9}\sum_{k=-1}^{1}\sum_{l=-1}^{1} [f_i[n-k, m-l] + f_j[n-k, m-l]] \\
&= \frac{1}{9}\sum_{k=-1}^{1}\sum_{l=-1}^{1} f_i[n-k, m-l] + \frac{1}{9}\sum_{k=-1}^{1}\sum_{l=-1}^{1} f_j[n-k, m-l]] \\
&= \mathcal{S}[f_i[n,m]] + \mathcal{S}[f_j[n,m]]
\end{aligned}$$

Properties of systems

- Amplitude properties:
  - Additivity

$$\mathcal{S}[f_i[n,m] + f_j[n,m]] = \mathcal{S}[f_i[n,m]] + \mathcal{S}[f_j[n,m]]$$

Properties of systems

- Amplitude properties:
    - Additivity

$$\mathcal{S}[f_i[n,m] + f_j[n,m]] = \mathcal{S}[f_i[n,m]] + \mathcal{S}[f_j[n,m]]$$

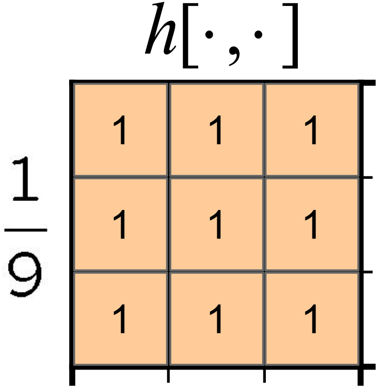
    - Homogeneity

$$\mathcal{S}[\alpha f[n,m]] = \alpha \mathcal{S}[f[n,m]]$$

Another question:

Q. Is the moving average filter homogeneous?

$$\mathcal{S}[\alpha f[n,m]] = \alpha \mathcal{S}[f[n,m]]$$



$h[\cdot,\cdot]$

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

Practice proving it at home using:

$$g[n,m] = \frac{1}{9}\sum_{k=-1}^{1}\sum_{l=-1}^{1} f[n-k, m-l]$$

What we covered today

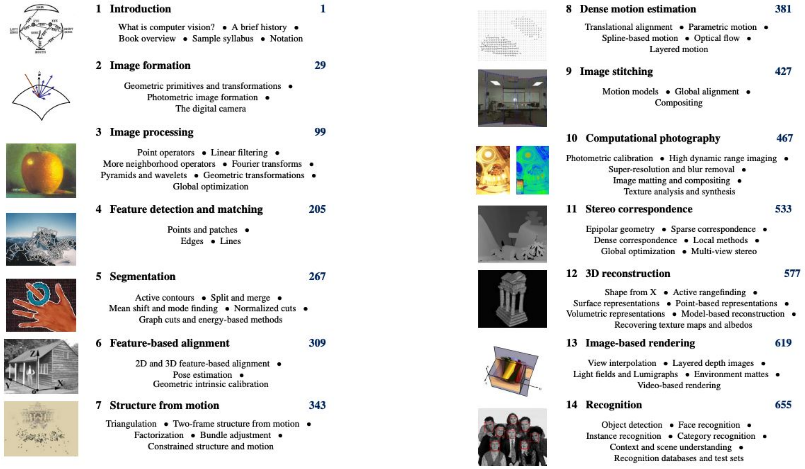
- Color spaces
- Image sampling and quantization
- Image histograms
- Images as functions
- Filters
- Properties of systems

Classic v.s. Deep Learning ?!

Should I take CSE455? Or should I go ahead to take the deep learning class?

History

- Recent neural networks is invented before 2000’s
- But here is the popular agenda in 2000’s



Deep Learning

- is a powerful **tool**
- immediately useful (for your problem, and maybe for job hunting)
- it’s not the problem (at least not CV problem)

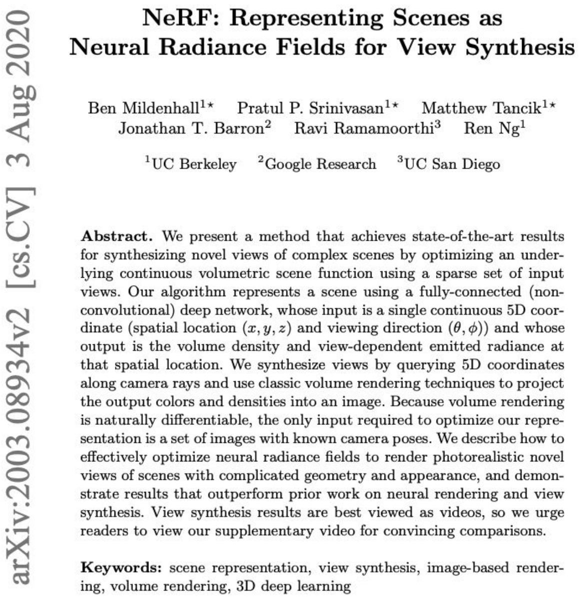
Why basics?

- Learning the problem (CSE455)
- and knowing the tool (any other deep learning classes)

Recall the groundbreaking NeRF paper

“NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis”, ECCV 2020

- Image as a function + neural network
- One of the most important paper in the recent CV development
- >10,000 citations in 5 years



arXiv:2003.08934v2 [cs.CV] 3 Aug 2020

NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis

Ben Mildenhall[1⋆] Pratul P. Srinivasan[1⋆] Matthew Tancik[1⋆]
Jonathan T. Barron[2] Ravi Ramamoorthi[3] Ren Ng[1]

[1]UC Berkeley [2]Google Research [3]UC San Diego

Abstract. We present a method that achieves state-of-the-art results for synthesizing novel views of complex scenes by optimizing an underlying continuous volumetric scene function using a sparse set of input views. Our algorithm represents a scene using a fully-connected (non-convolutional) deep network, whose input is a single continuous 5D coordinate (spatial location (x, y, z) and viewing direction (θ, ϕ)) and whose output is the volume density and view-dependent emitted radiance at that spatial location. We synthesize views by querying 5D coordinates along camera rays and use classic volume rendering techniques to project the output colors and densities into an image. Because volume rendering is naturally differentiable, the only input required to optimize our representation is a set of images with known camera poses. We describe how to effectively optimize neural radiance fields to render photorealistic novel views of scenes with complicated geometry and appearance, and demonstrate results that outperform prior work on neural rendering and view synthesis. View synthesis results are best viewed as videos, so we urge readers to view our supplementary video for convincing comparisons.

Keywords: scene representation, view synthesis, image-based rendering, volume rendering, 3D deep learning

Why basics?

- Learning the problem (CSE455)
  - and knowing the tool (any other deep learning classes)
- Our goal in CSE455:
  - learning foundation, ideas, and basics
  - build your taste and intuition to computer vision

- Classic approaches are still useful nowadays
  - especially in scale (cheap!!)

Next time:

Linear systems and convolutions