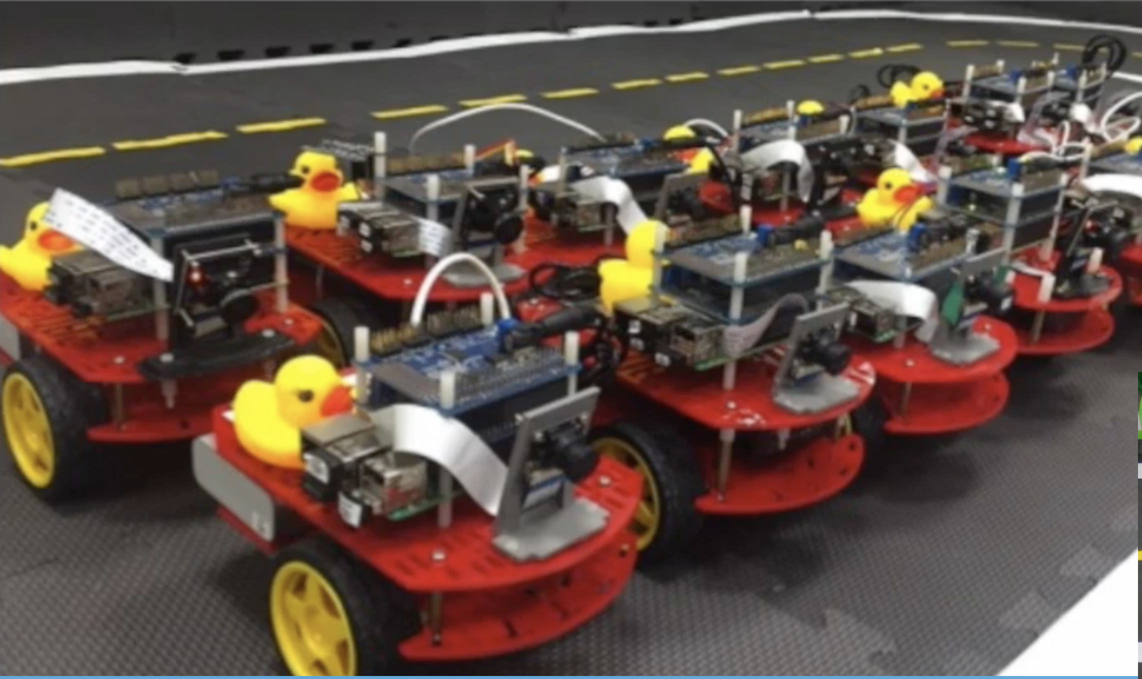


CS 3630!

Lecture 16: Computer Vision Fundamentals

Topics

1. **What is Computer Vision?**
2. **Applications of CV**
3. **Images as 2D arrays**
4. **Basic Image Processing**
5. **Image Filtering**

- Many slides borrowed from James Hays, Irfan Essa, and others.
- Intro CV course: CS 4476
    - This spring: Judy Hoffmann
    - Coming Fall: Frank Dellaert

Motivation

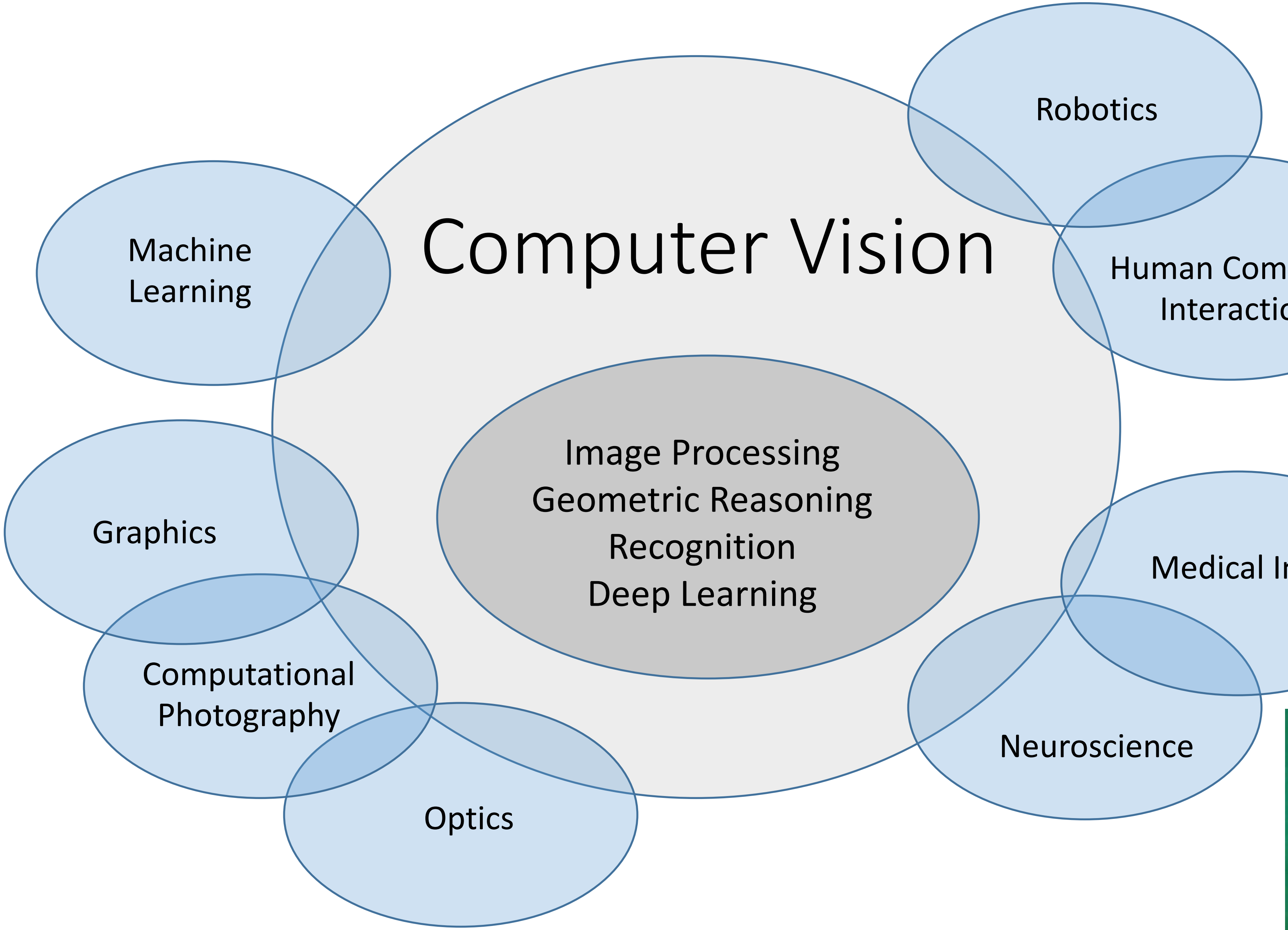
- Robots need to act in the world
- One of the cheapest and richest sensors is a camera
- Unfortunately, understanding camera images is **not** easy
- Since the sixties, researchers have tried to tackle this problem
- Since 2012, deep learning has led to incredible progress
- Perception for robotics is following closely behind

1. What is Computer Vision?

Computer Graphics: Models to Images

Comp. Photography: Images to Images

Computer Vision: Images to Models



Computer Vision

Image Processing
Geometric Reasoning
Recognition
Deep Learning

Machine Learning

Graphics

Computational Photography

Optics

Robotics

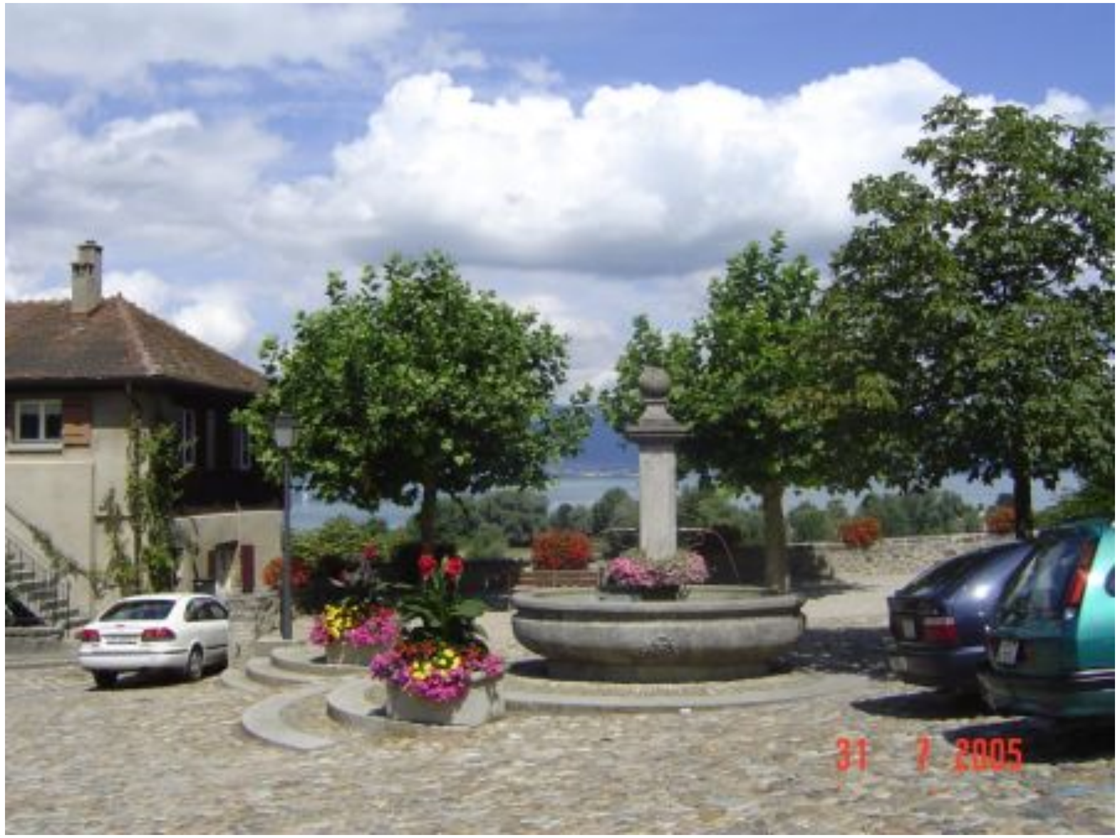
Human Computer Interaction

Medical Imaging

Neuroscience

Computer Vision

Make computers understand images and video or any visual data.



What kind of scene?

Where are the cars?

How far is the building?

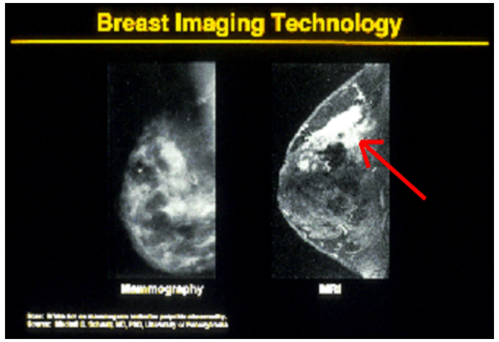
…

Vision is really hard

- Vision is an amazing feat of natural intelligence
  - Visual cortex occupies about 50% of Macaque brain
  - One third of human brain devoted to vision (more than anything else)

Why computer vision matters

Safety



Health

Security

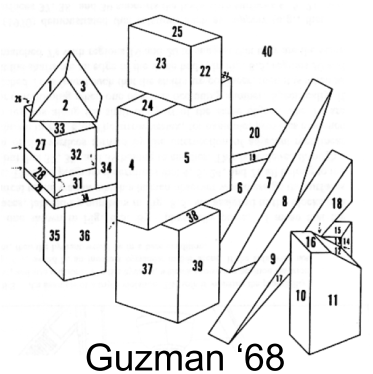
Comfort

Fun

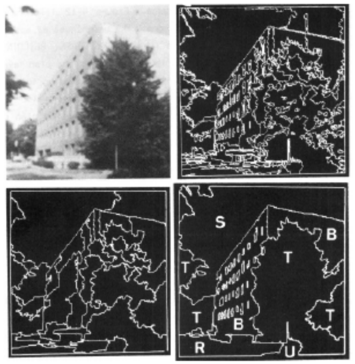
Robotics

Ridiculously brief history of computer vision

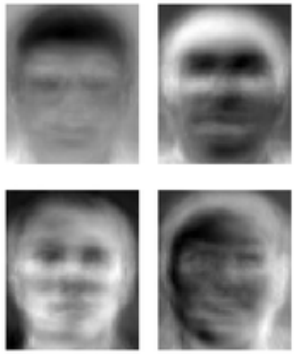
- 1966: Minsky assigns computer vision as an undergrad summer project
- 1960’s: interpretation of synthetic worlds
- 1970’s: some progress on interpreting selected images
- 1980’s: ANNs come and go; shift toward geometry and increased mathematical rigor
- 1990’s: face recognition; statistical analysis in vogue
- 2000’s: broader recognition; large annotated datasets available; video processing starts
- 2010’s: Deep learning with ConvNets
- 2020’s: Widespread autonomous vehicles?
- 2030’s: robot uprising?



Guzman ‘68



Ohta Kanade ‘78



Turk and Pentland ‘91

2. Applications of Computer Vision

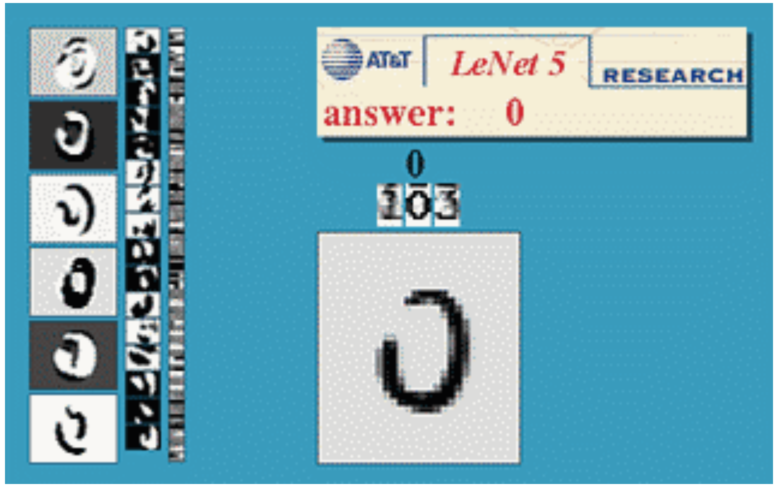
- Examples of real-world applications

Some of the following slides by Steve Seitz

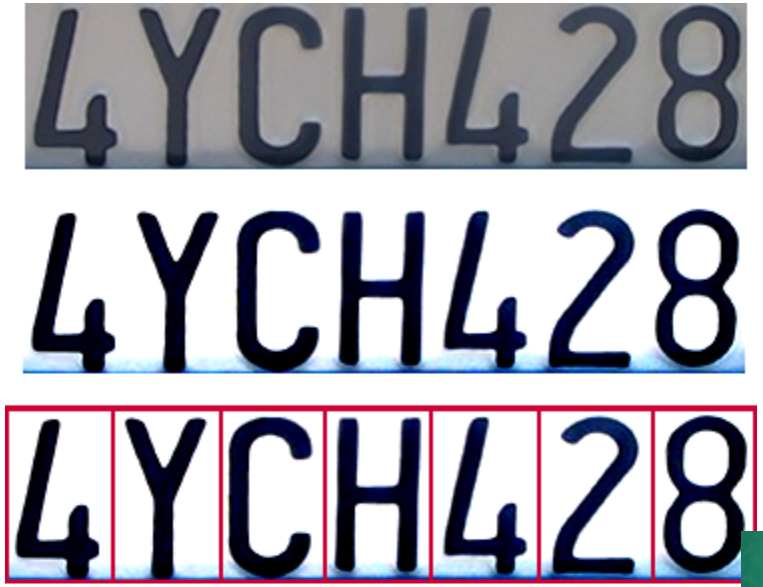
Optical character recognition (OCR)

Technology to convert scanned docs to text

- If you have a scanner, it probably came with OCR software

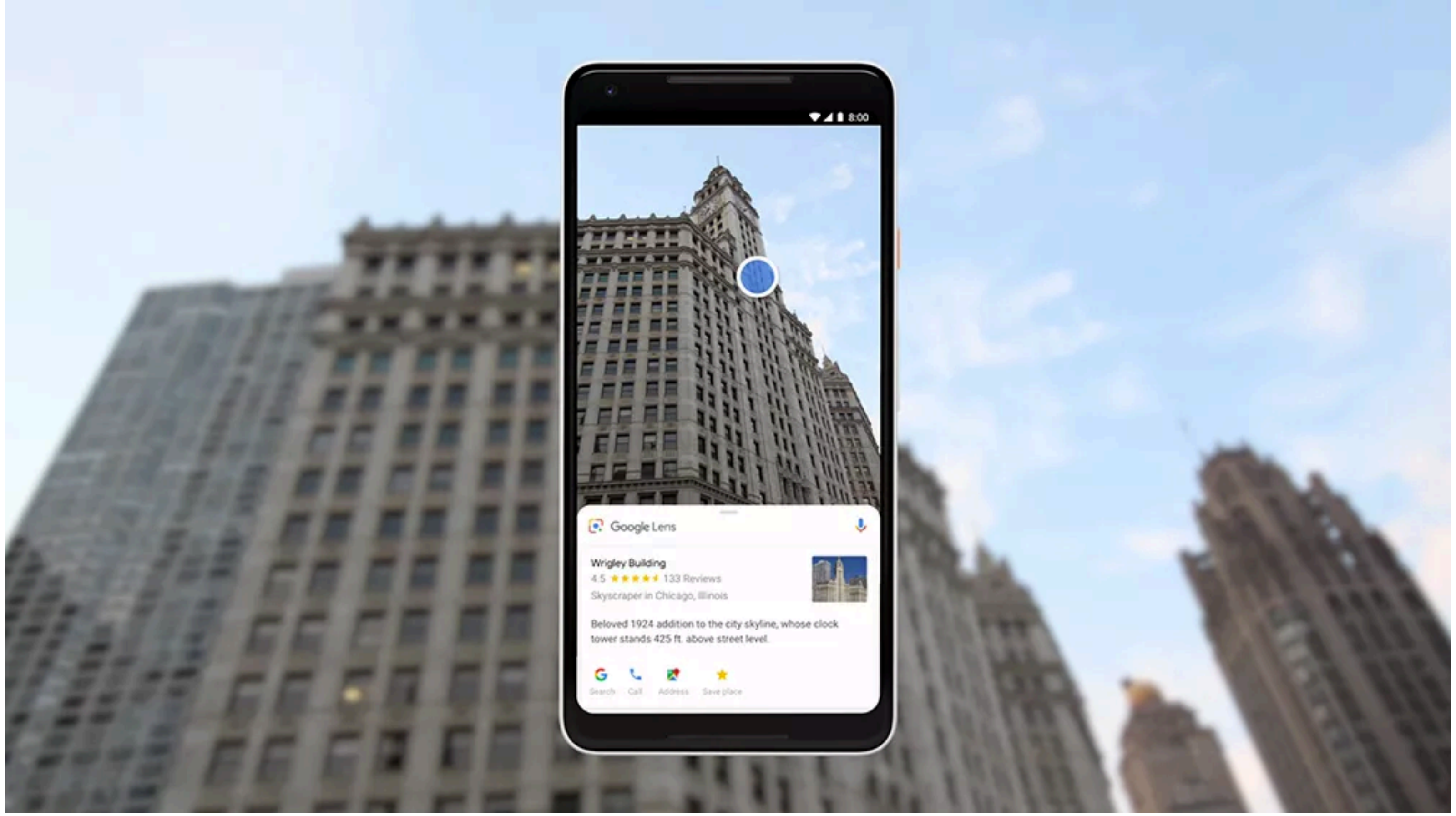


Digit recognition, AT&T labs
http://www.research.att.com/~yann/



License plate readers
http://en.wikipedia.org/wiki/Automatic_number_plate_rec

Object recognition (in mobile phones)

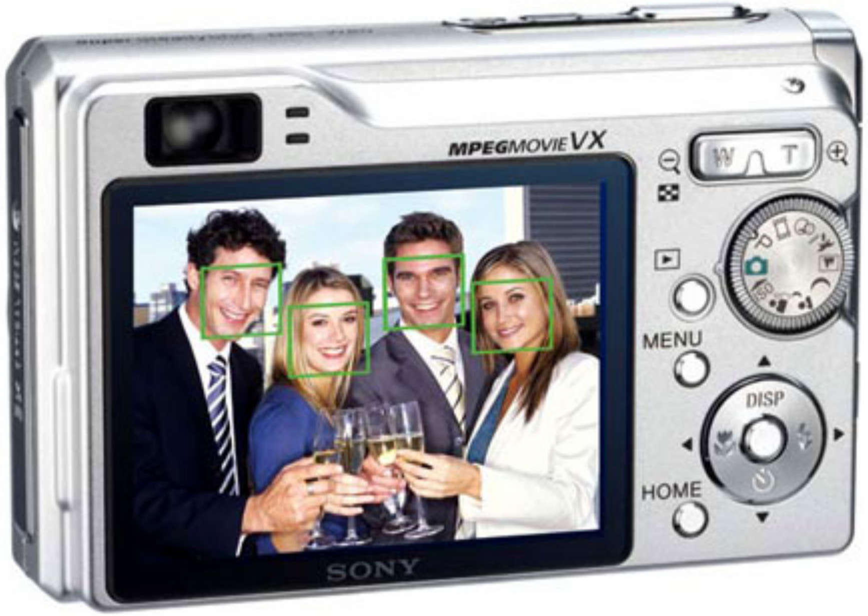


Wrigley Building
4.5 ★★★★★ 133 Reviews
Skyscraper in Chicago, Illinois

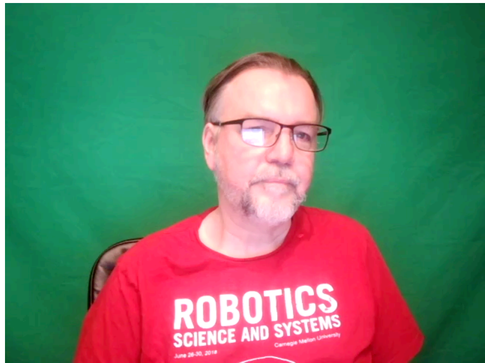
Beloved 1924 addition to the city skyline, whose clock tower stands 425 ft. above street level.

E.g. Google Lens

Face detection

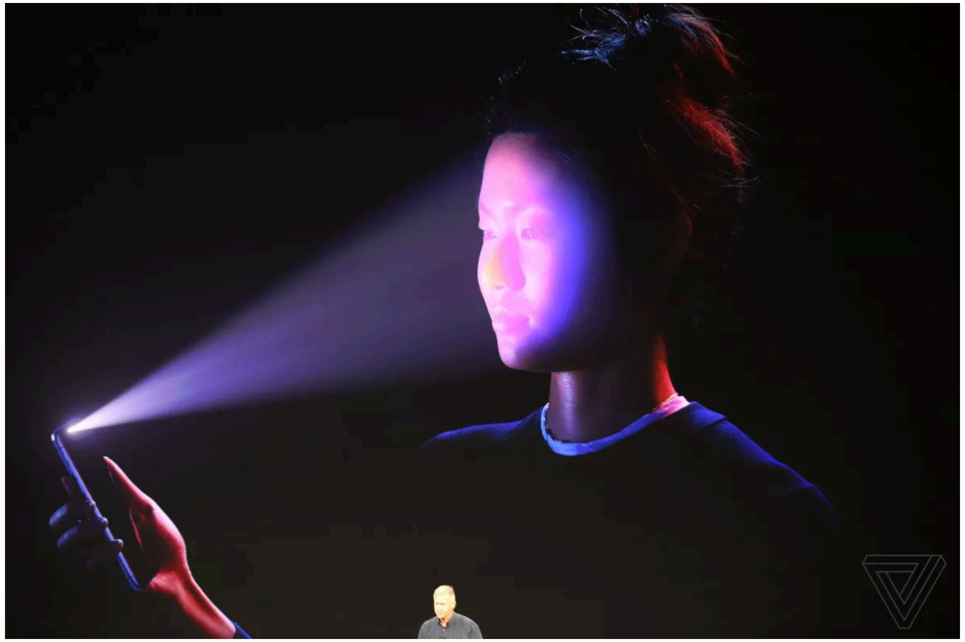


- Digital cameras (you know these as "phones") detect faces



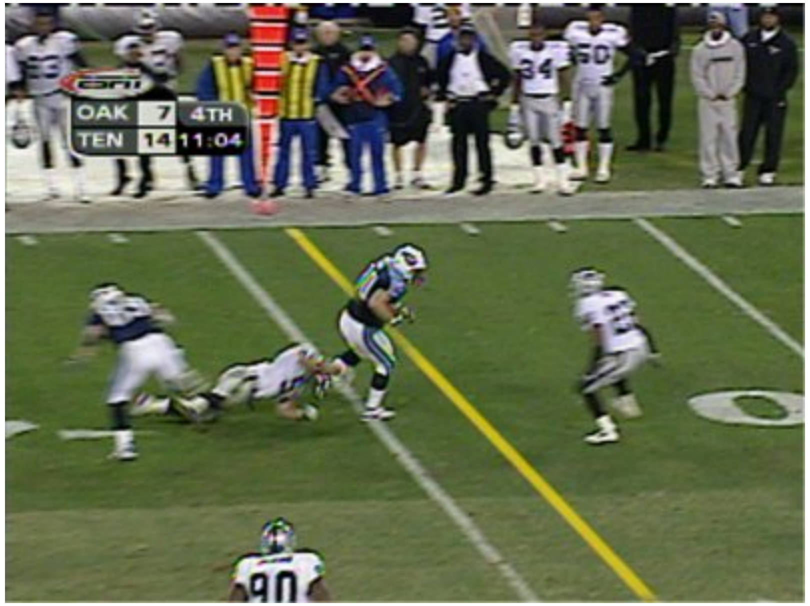
Login without a password…

Fingerprint scanners on many new laptops, other devices



Face recognition systems now widely in use on smartphones

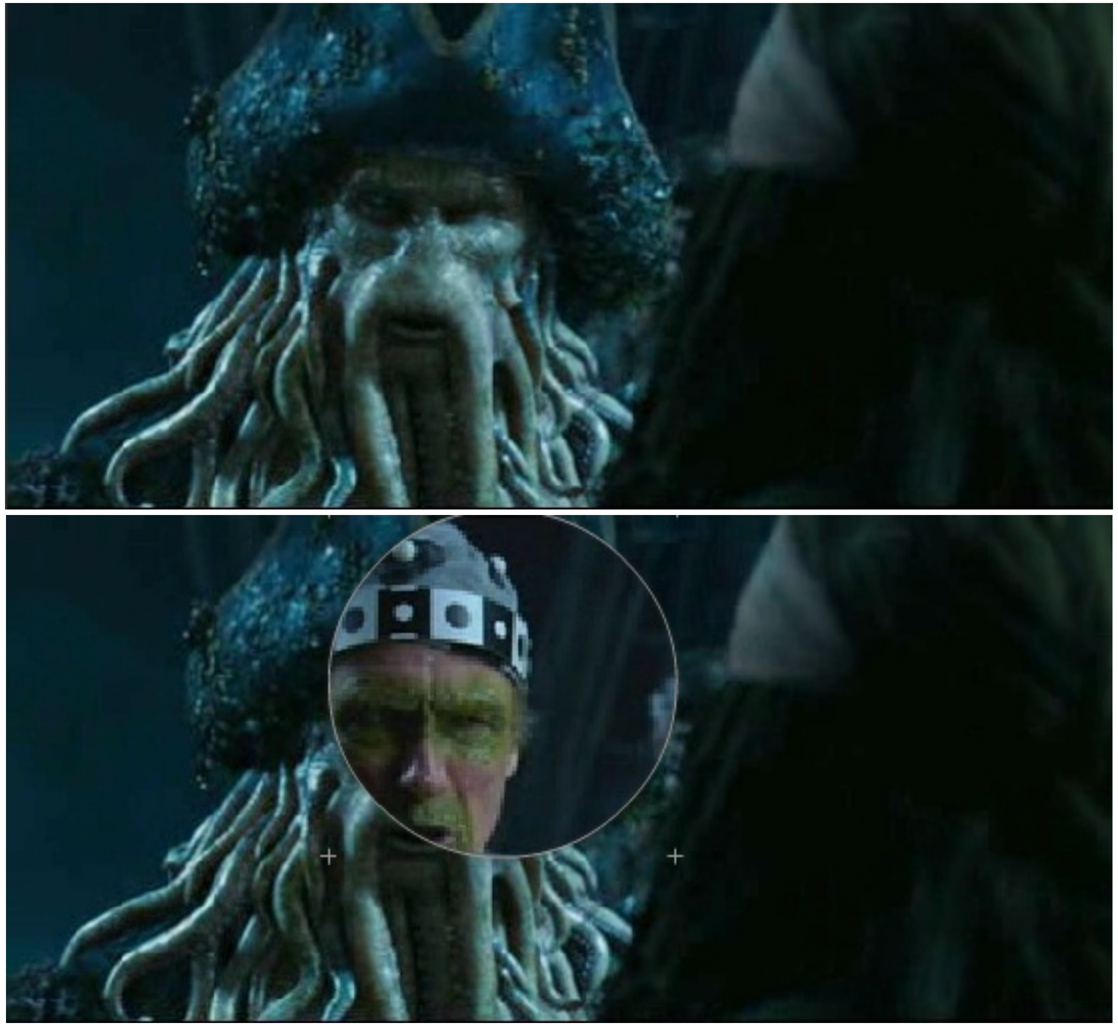
Sports



Sportvision first down line
Nice explanation on www.howstuffworks.com

http://www.sportvision.com/video.html

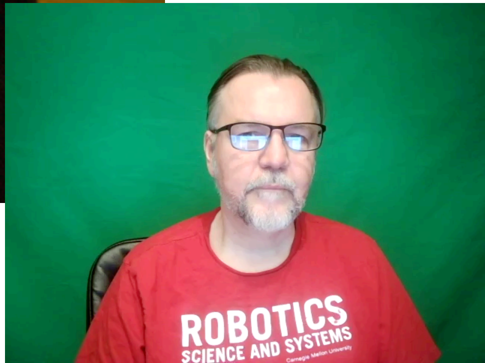
Special effects: motion capture



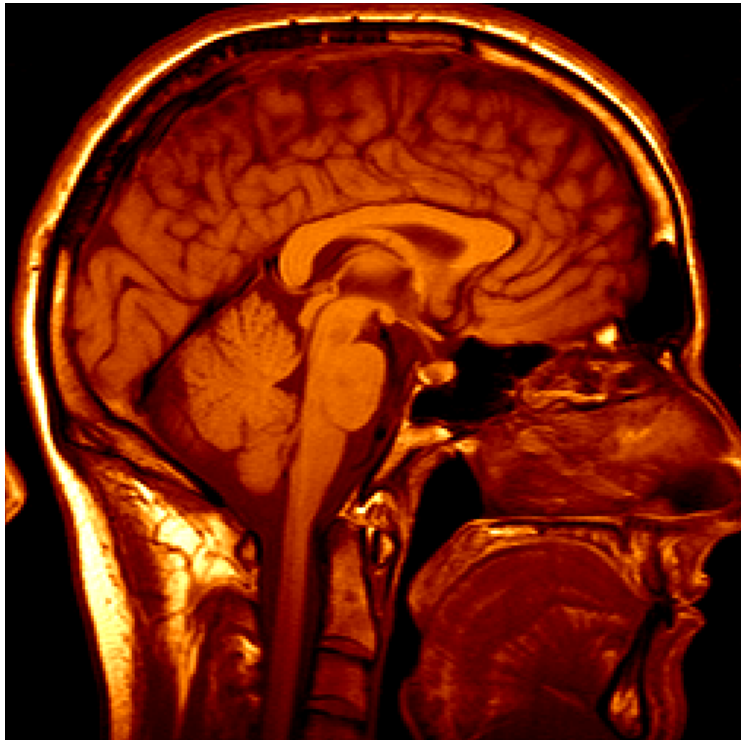
Pirates of the Carribean, Industrial Light and Magic

Augmented Reality and Virtual Reality

Magic Leap, Oculus, Hololens, etc.



Medical imaging



3D imaging
MRI, CT

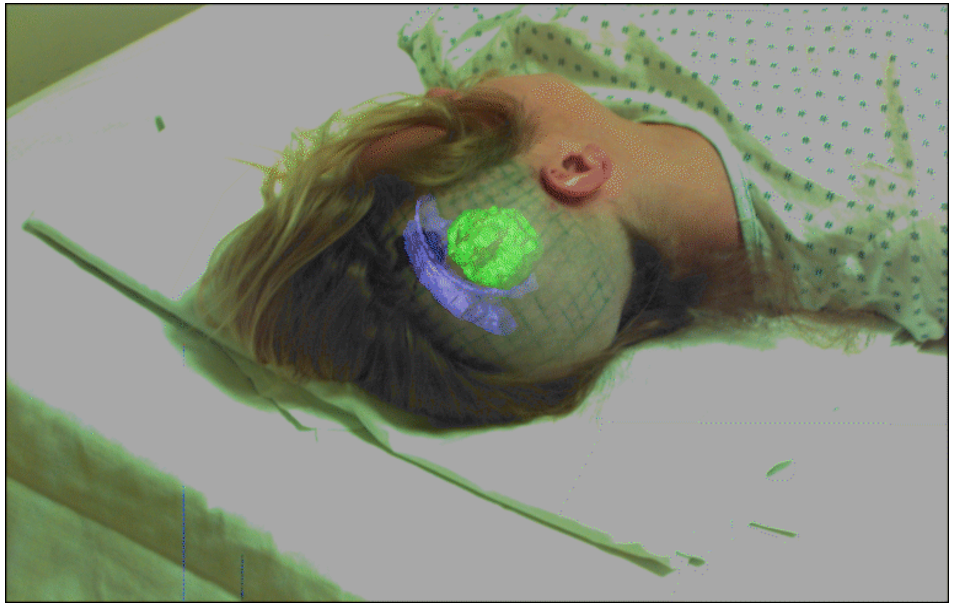
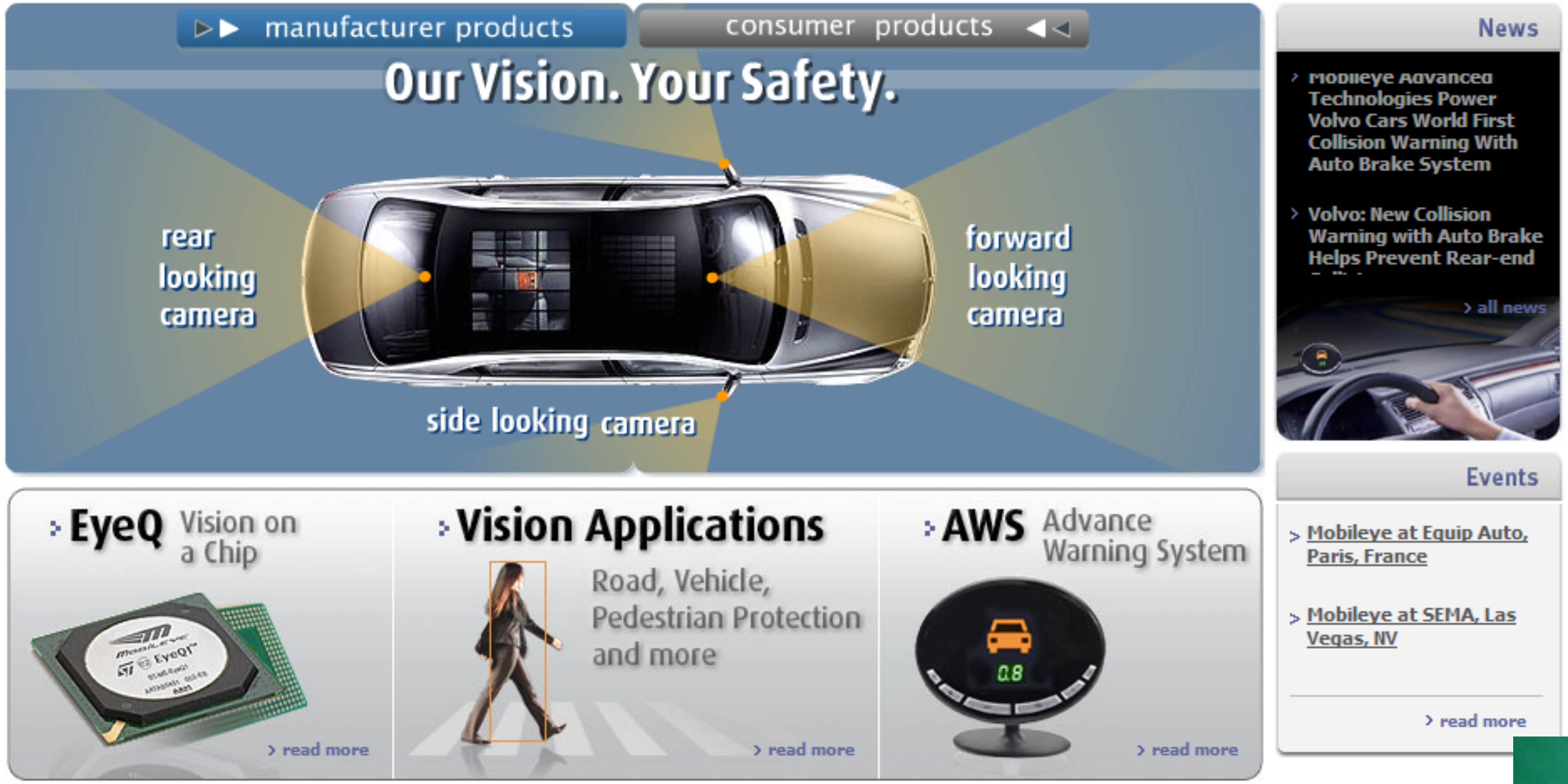


Image guided surgery
Grimson et al., MIT

Slide content courtesy of Amnon Shashua

Smart cars



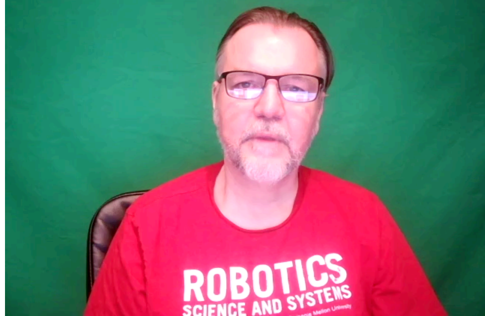
News

- Mobileye Advanced Technologies Power Volvo Cars World First Collision Warning With Auto Brake System
- Volvo: New Collision Warning with Auto Brake Helps Prevent Rear-end

Events

- Mobileye at Equip Auto, Paris, France
- Mobileye at SEMA, Las Vegas, NV

- Mobileye
  - ~~Market Capitalization: 11 Billion dollars~~
  - Bought by Intel for 15 Billion dollars



Computer Vision in space

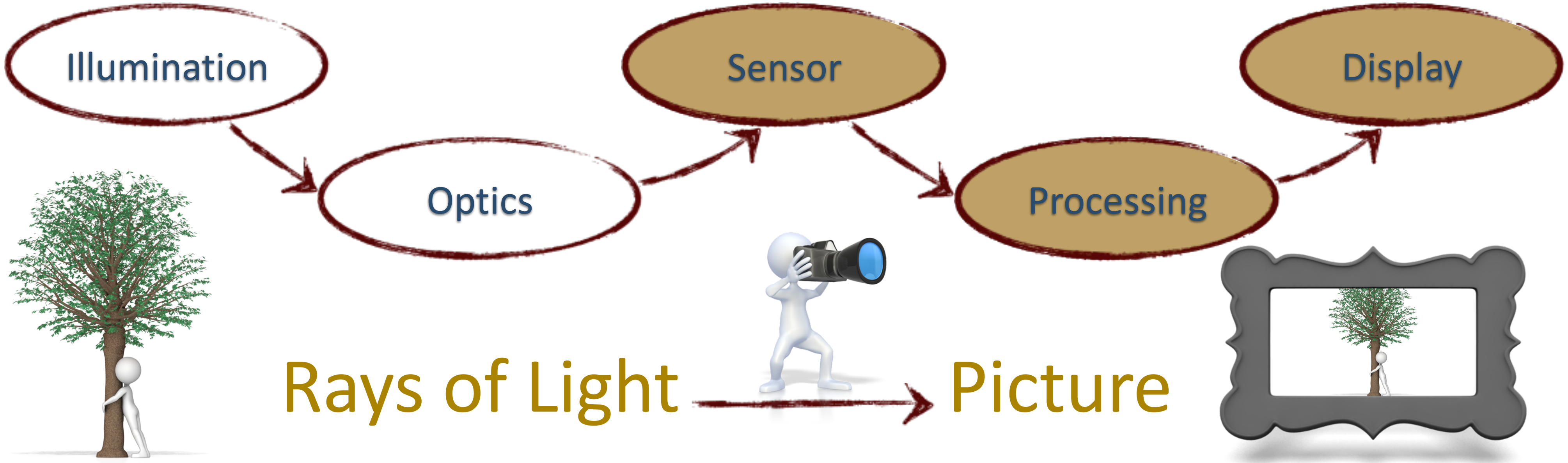
NASA'S Mars Exploration Rover Spirit captured this westward view from atop a low plateau where Spirit spent the closing months of 2007.

Vision systems (JPL) used for several tasks

- Panorama stitching
- 3D terrain modeling
- Obstacle detection, position tracking
- For more, read “Computer Vision on Mars” by Matthies et al.

3. Images as 2D Arrays

Image Acquisition Pipeline



Illumination

Optics

Sensor

Processing

Display

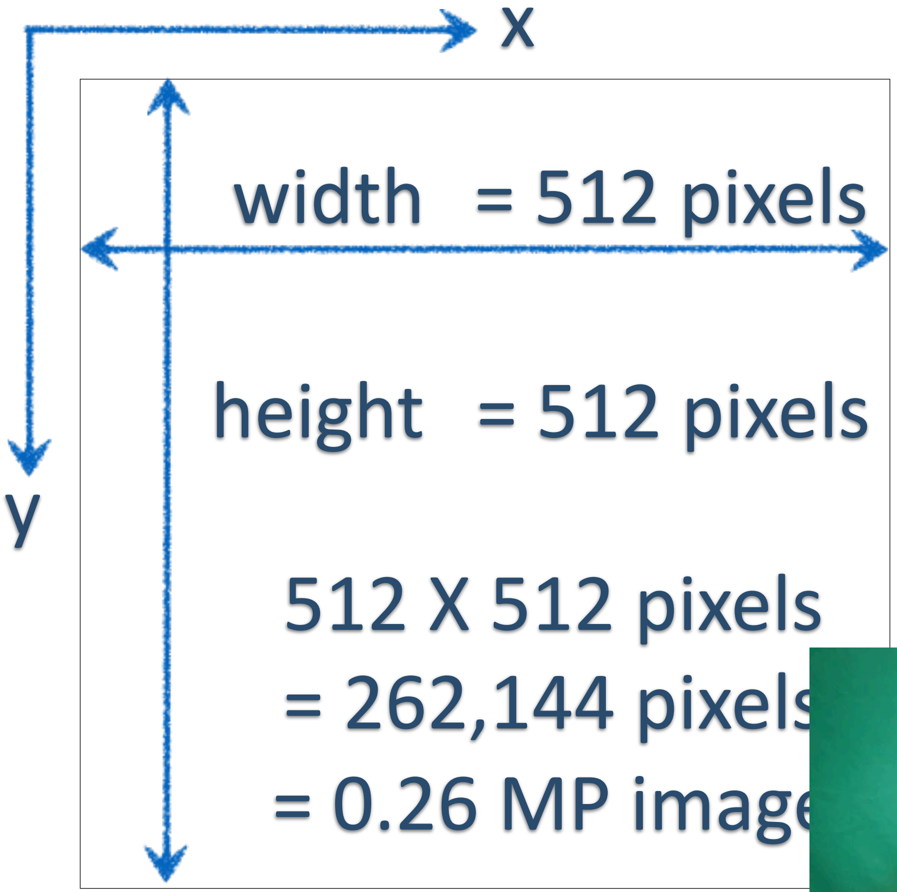
Rays of Light → Picture

- Analog (incoming light) to digital (pix

A Digital Image (W X H)

Georgia Tech's Mascot Buzz

Georgia Tech's Mascot Buzz, in Black and White

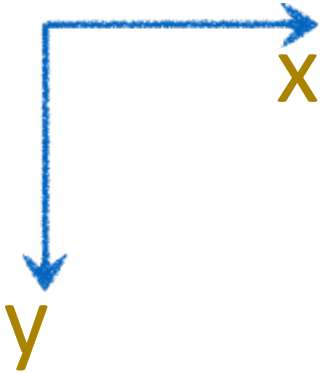


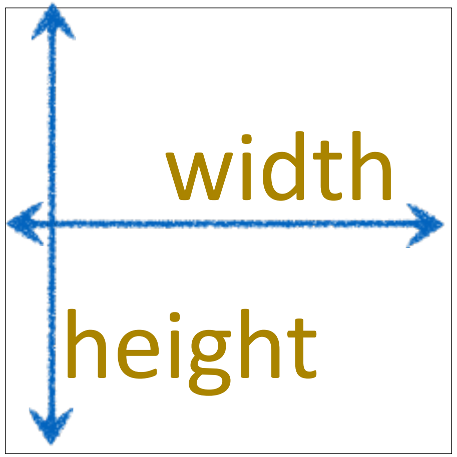
width = 512 pixels

height = 512 pixels

512 X 512 pixels
= 262,144 pixels
= 0.26 MP image

A Digital Image!

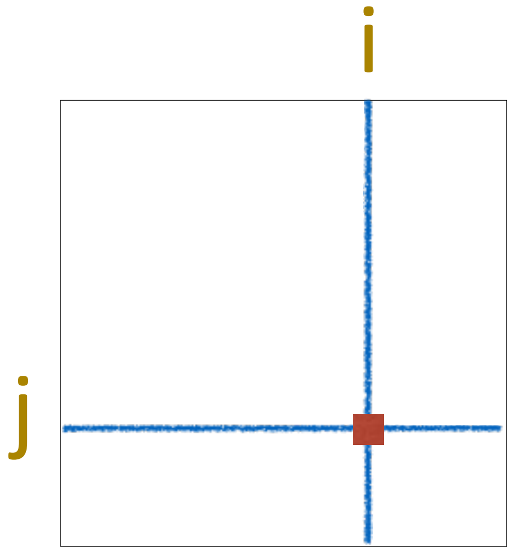




- Numeric representation in 2-D (x and y)
- Referred to as *I(x,y)* in continuous function form, *I(i,j)* in discrete
- Image Resolution: expressed in terms of Width and Height of the image

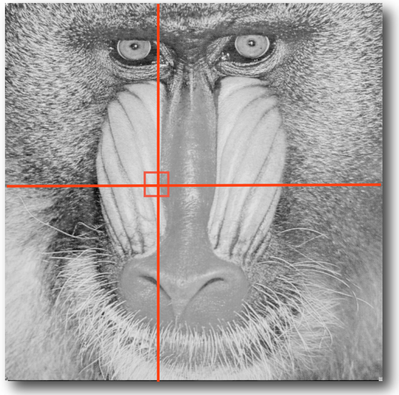
Pixel

A “picture element” that contains the light intensity at some location *(i,j)* in the image



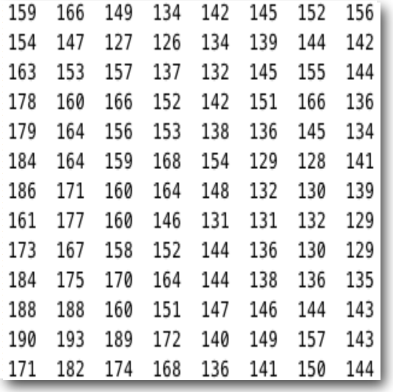
$I(i,j)$ = Some Numeric Value

Characteristics of a Digital Image



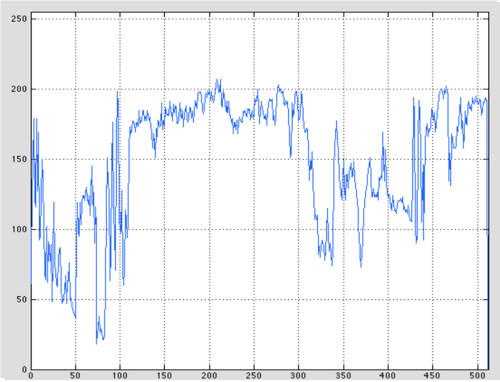
Original Image

Zoomed In



159	166	149	134	142	145	152	156
154	147	127	126	134	139	144	142
163	153	157	137	132	145	155	144
178	160	166	152	142	151	166	136
179	164	156	153	138	136	145	134
184	164	159	168	154	129	128	141
186	171	160	164	148	132	130	139
161	177	160	146	131	131	132	129
173	167	158	152	144	136	130	129
184	175	170	164	144	138	136	135
188	188	160	151	147	146	144	143
190	193	189	172	140	149	157	143
171	182	174	168	136	141	150	144

Values



Plots of Values at a Slice

- A two-dimensional array of pixels and respective intensities
- Image can be represented as a Matrix
- Intensity Values range from 0 = Black to 255 = White

Common data types

Data types used to store pixel values:

- `unsigned char`
- `uint8`
- `unsigned char 8bit`
- 2^n (2^1, 2^2, 2^4, 2^8, etc.)

Digital Image Formats

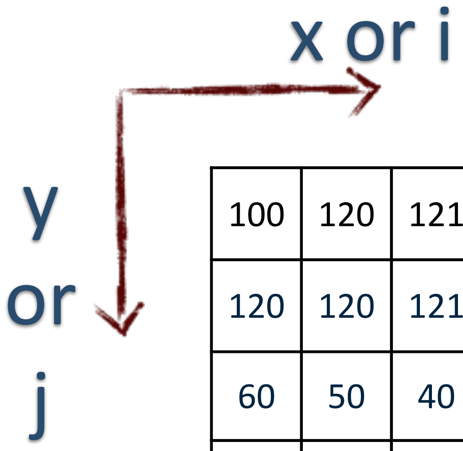
Images can also be 16, 24, 32 bits-per-pixel:

- 24 bits per pixel usually means 8 bits per color
- At the two highest levels, the pixels themselves can carry up to 16,777,216 different colors

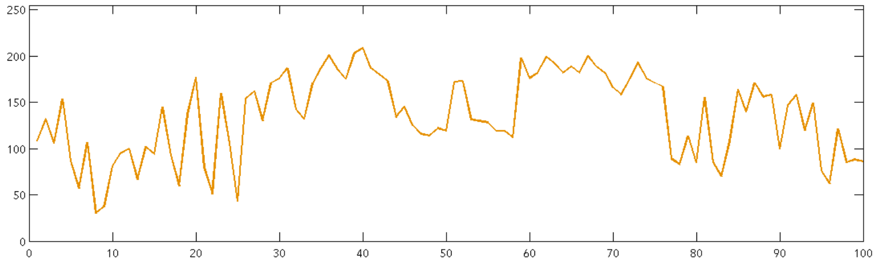
Common raster image formats:

- GIF, JPG, PPM, TIF, BMP, etc.

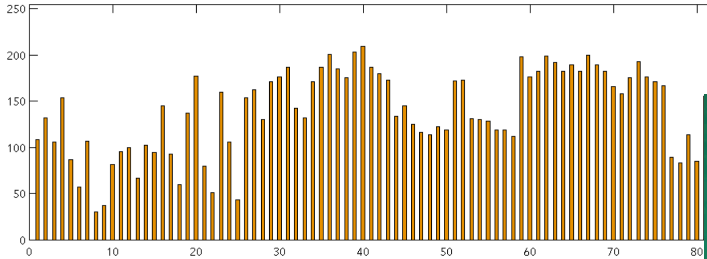
Digital Image is a Function



100	120	121	122	30	40
120	120	121	122	70	40
60	50	40	41	7	8
100	120	121	122	1	0
200	120	200	122	12	14
200	220	225	250	30	40



Continuous Signal



Discrete Signa

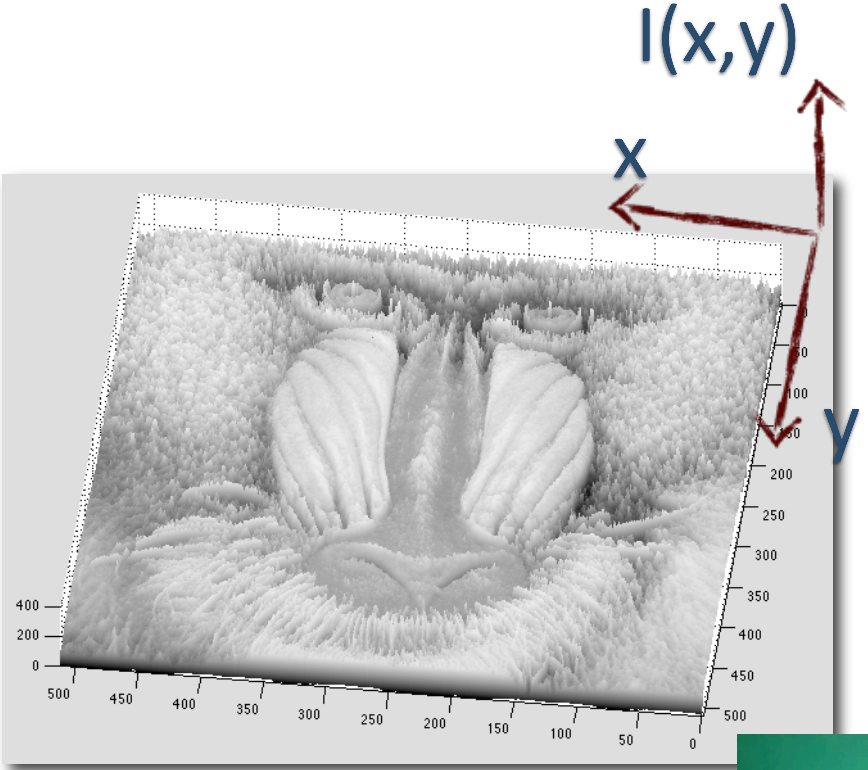
Slide adapted from Steve Seitz and Aaron Bobick

Digital Image is a Function

x or i

y or j

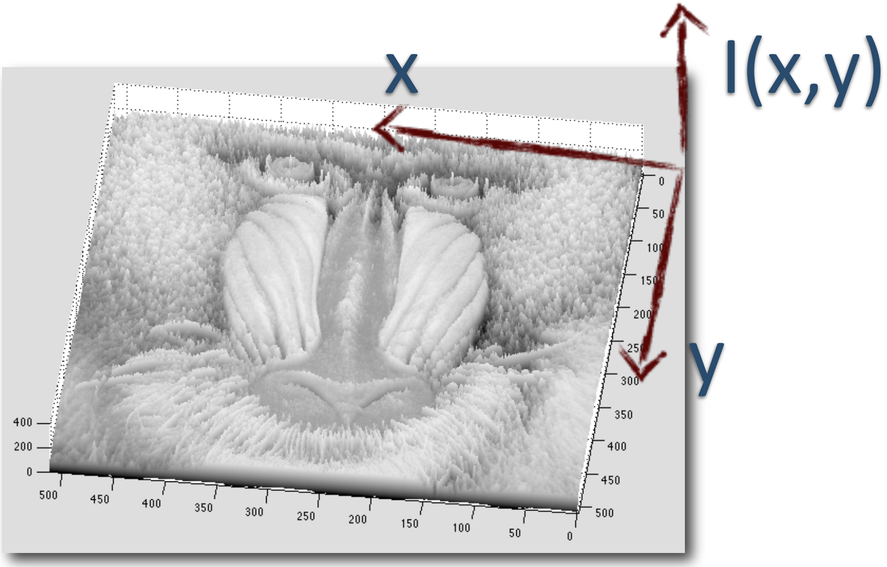
100	120	121	122	30	40
120	120	121	122	70	40
60	50	40	41	7	8
100	120	121	122	1	0
200	120	200	122	12	14
200	220	225	250	30	40



Slide adapted from Steve Seitz and Aaron Bobick

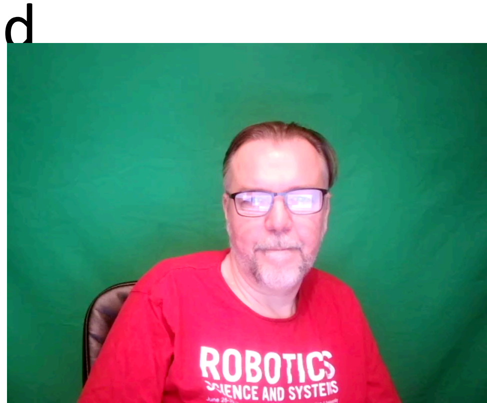
Digital Image is a Function

100	120	121	122	30	40
120	120	121	122	70	40
60	50	40	41	7	8
100	120	121	122	1	0
200	120	200	122	12	14
200	220	225	250	30	40

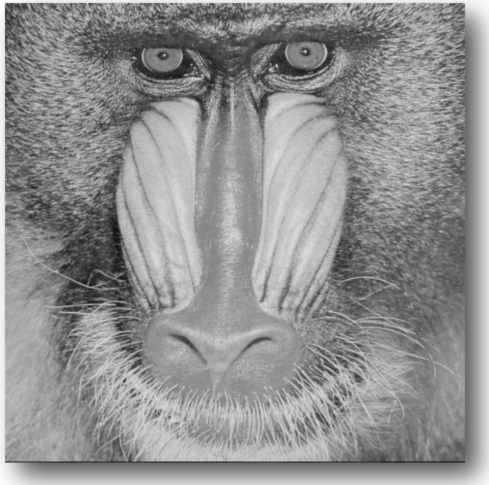


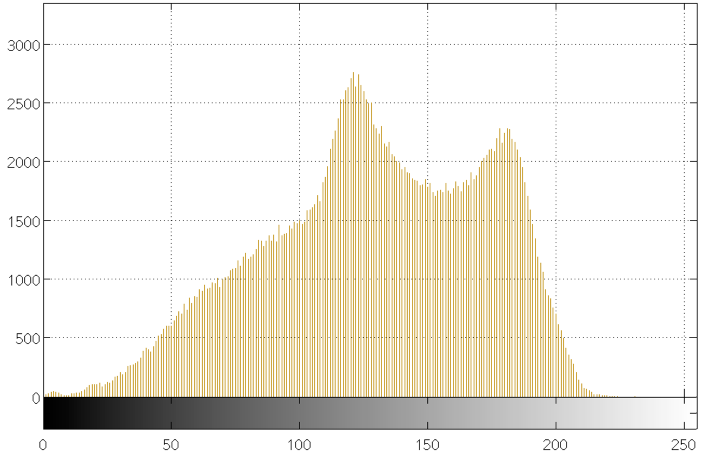
- Typically, the functional operation requires discrete values
  - Sample the two-dimensional (2D) space on a regular grid
  - Quantize each sample (rounded to “nearest integer”)
- Matrix of integer values (Range: 0-255)

Slide adapted from Steve Seitz and Aaron Bobick



Digital Image Statistics





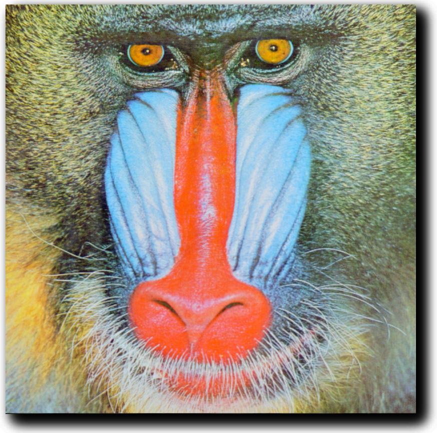
Pixel Counts

Image Histogram

Intensity Bins

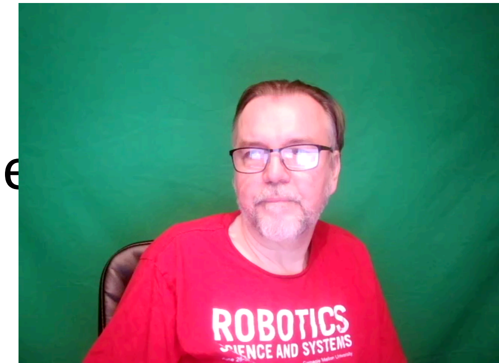
- Image statistics - average, median, mode
    - Scope - entire image or smaller windows/regions
- Histogram - distribution of pixel intensities in the image
    - Can be separate for each channel, or region-based too

Color Digital Image: An Example



Color Red Channel Green Channel Blue Channel

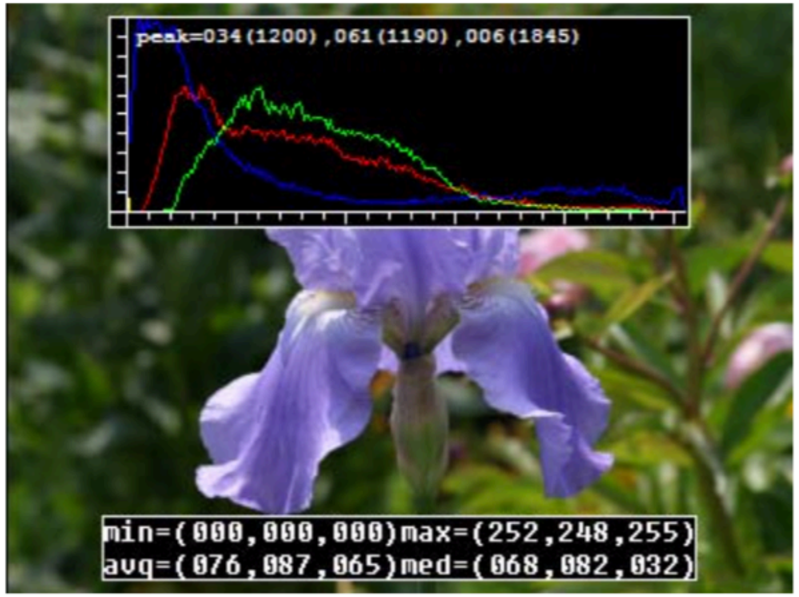
- Color image = 3 color channels (images, with their own intensities) blended together
- Makes 3D data structure of size: Width X Height X Channels
- Each pixel has therefore 3 intensities: Red (R), Green (G), Blue



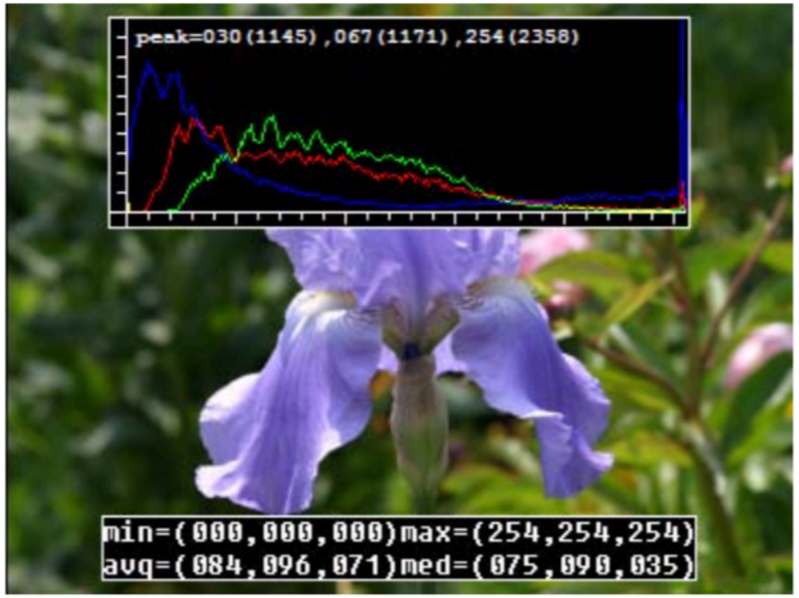
4. Basic Image Processing

- Contrast
- Brightness
- Gamma
- Histogram equalization
- Arithmetic
- Compositing

Contrast



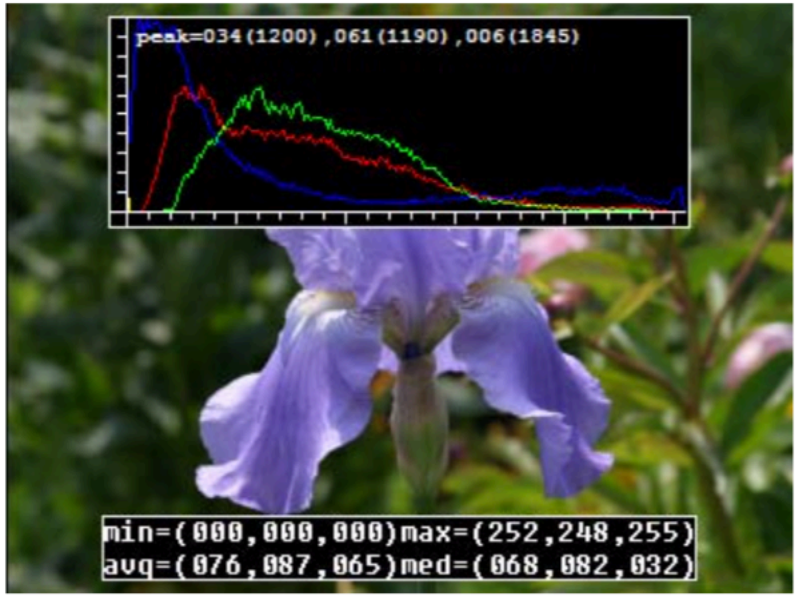
```
peak=034(1200),061(1190),006(1845)
min=(000,000,000)max=(252,248,255)
avg=(076,087,065)med=(068,082,032)
```

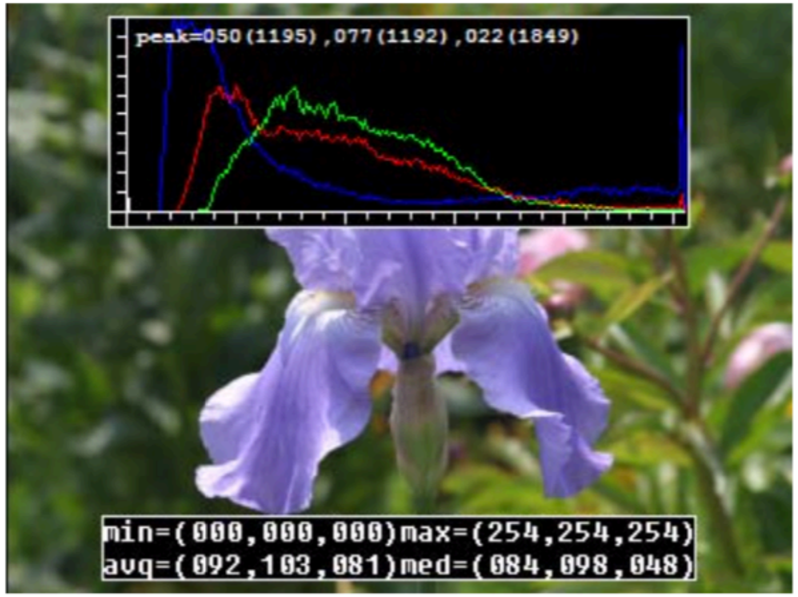


```
peak=030(1145),067(1171),254(2358)
min=(000,000,000)max=(254,254,254)
avg=(084,096,071)med=(075,090,035)
```

- g(x) = a f(x), a=1.1

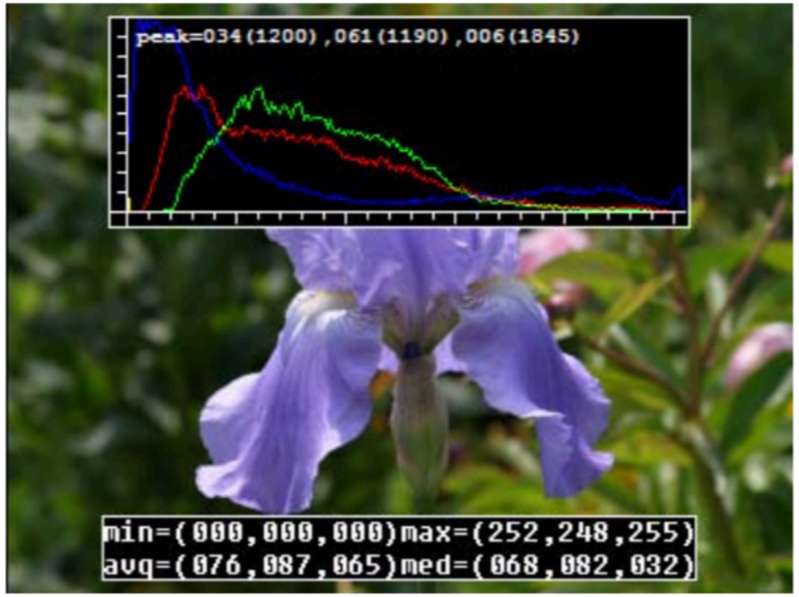
Brightness

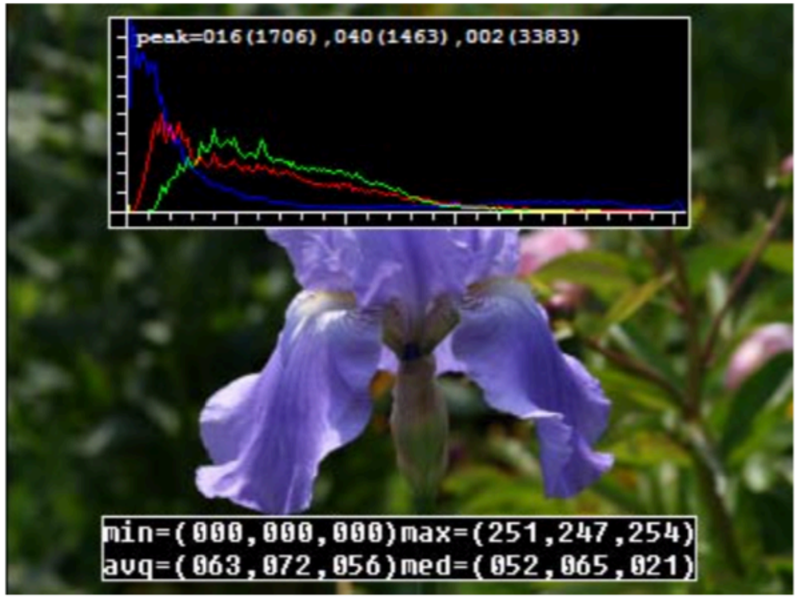




- g(x) = f(x) + b, b=16

Gamma correction

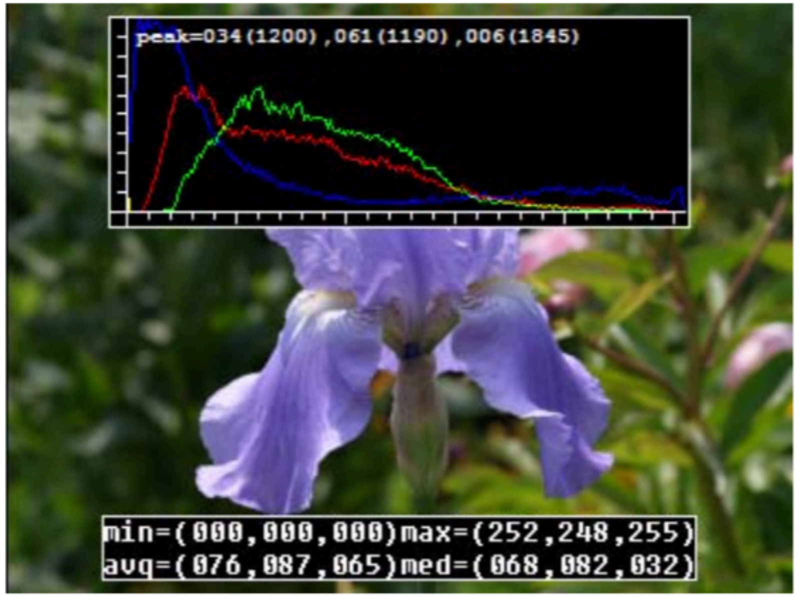


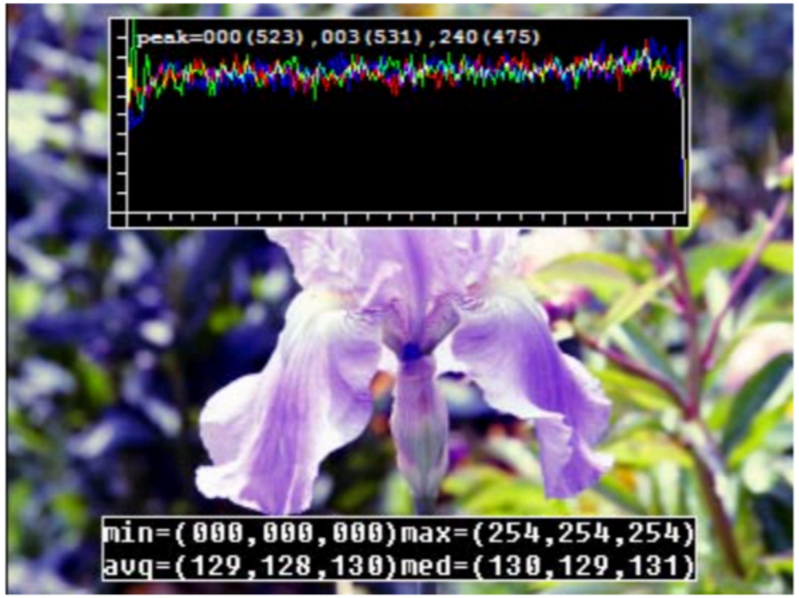


$$g(\boldsymbol{x}) = [f(\boldsymbol{x})]^{1/\gamma}$$

- gamma = 1.2

Histogram Equalization





- Non-linear transform to make histogram flat
- Still a per-pixel operation $g(x) = h(f(x))$

Point-Process: Pixel/Point Arithmetic

120	122	140	142	143
121	120	141	144	147
122	121	144	146	11
125	121	144	145	10
126	121	145	147	13

+

120	122	140	142	143
121	80	40	144	10
122	81	40	0	151
125	80	40	0	152
126	70	40	0	153

=

240	244	280	284	286
121	200	181	288	157
122	202	184	146	162
125	201	184	145	164
126	191	185	147	166

120	122	140	142	143
121	120	141	144	147
122	121	144	146	11
125	121	144	145	10
126	121	145	147	13

-

120	122	140	142	143
121	80	40	144	10
122	81	40	0	151
125	80	40	0	152
126	70	40	0	153

=

0	0	0	0	0
0	40	101	0	137
0	40	104	146	-140
0	40	104	145	-142
0	191	185	147	-140

Pixel/Point Arithmetic: An Example

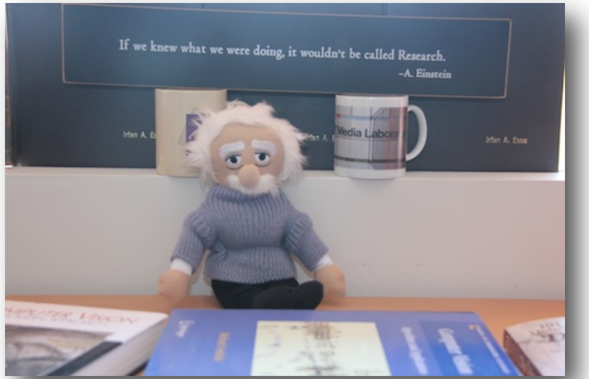

If we knew what we were doing, it wouldn't be called Research.
-A. Einstein

Image 1

-

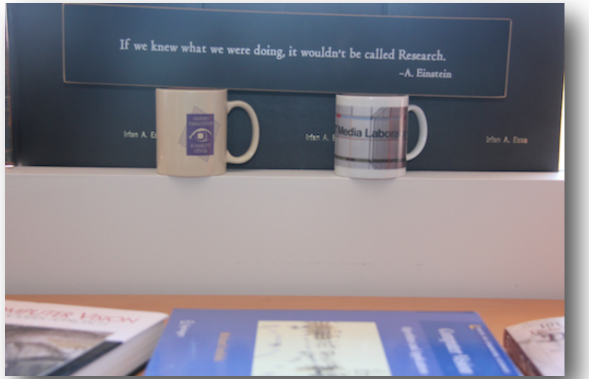

If we knew what we were doing, it wouldn't be called Research.
-A. Einstein

Image 2

=

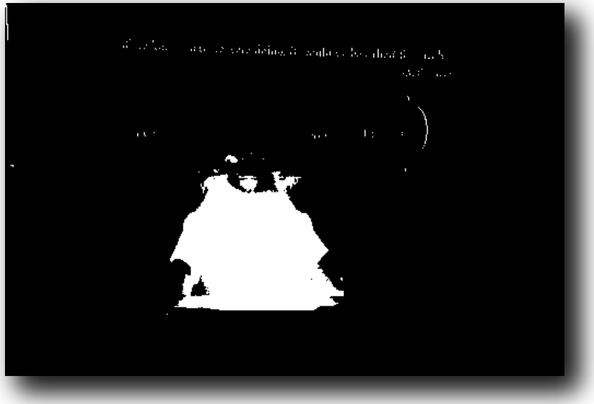


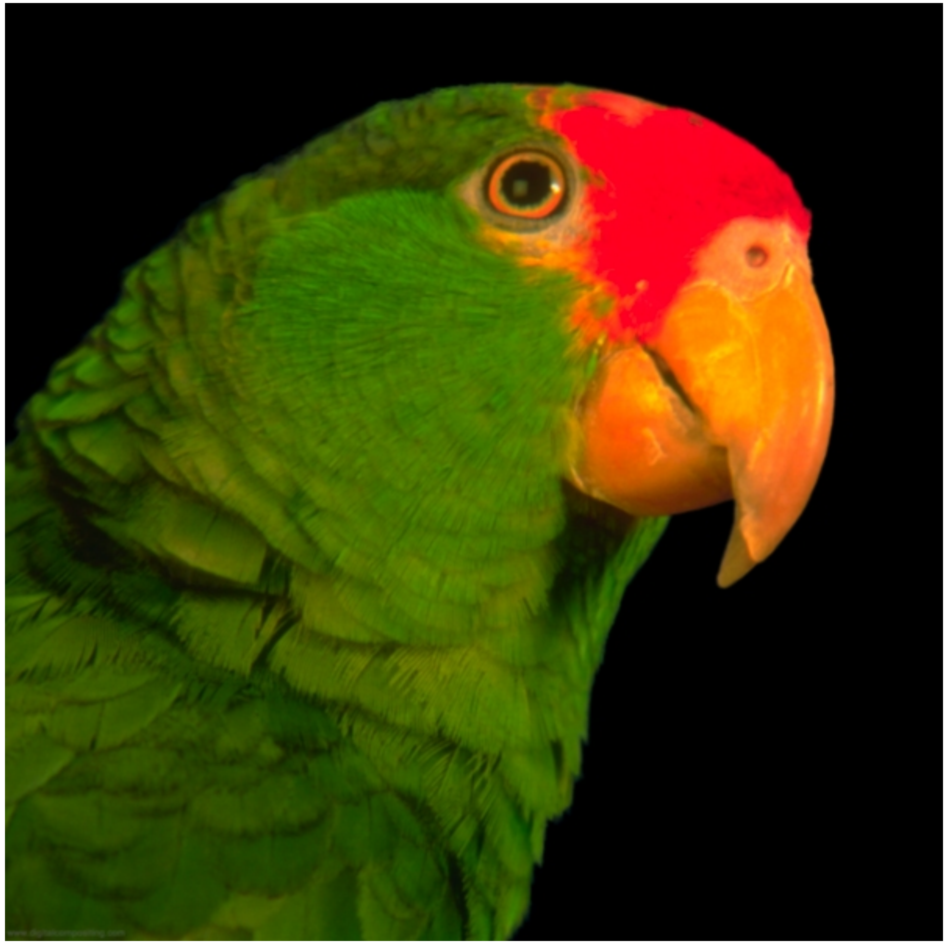
Image 1 - Image 2

Binary(Image 1 - Image 2)

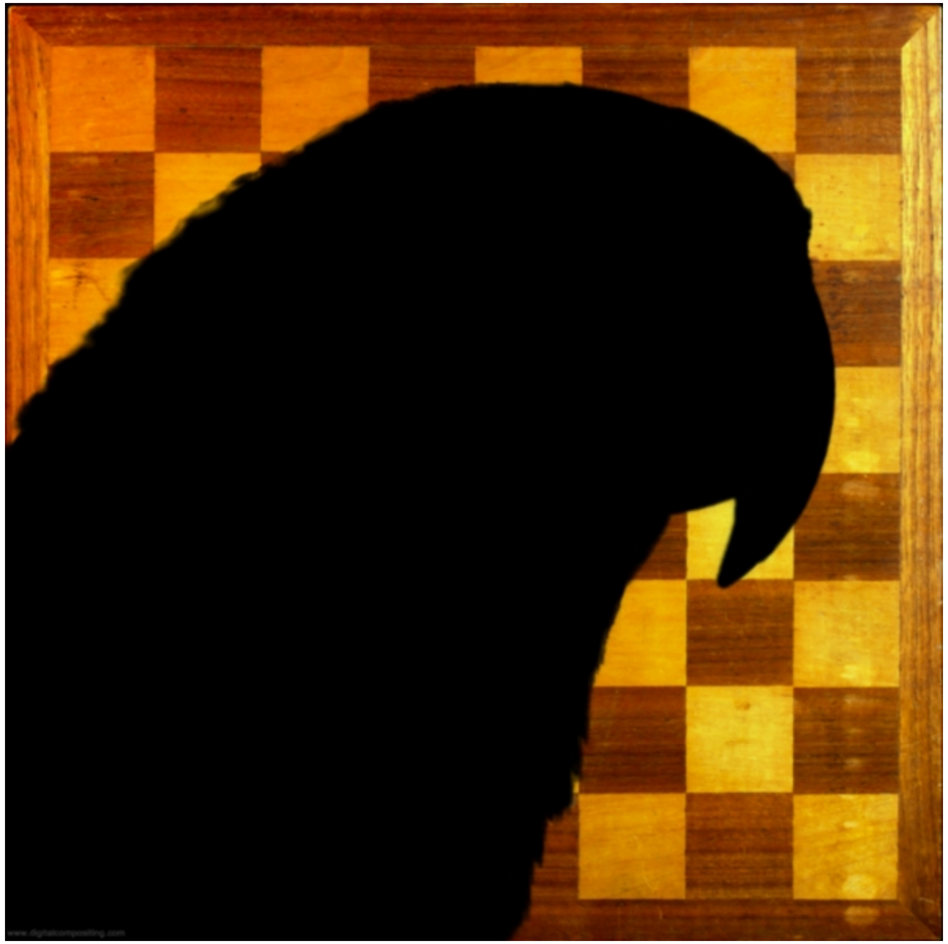
Matte: an alpha image

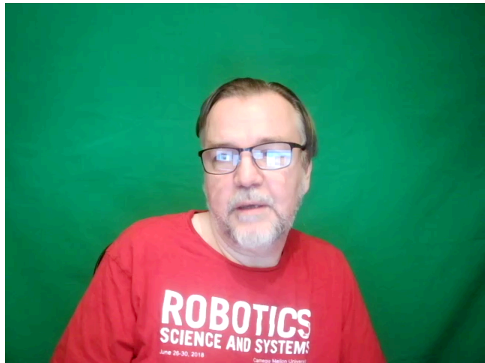
www.digitalcompositing.com

aF

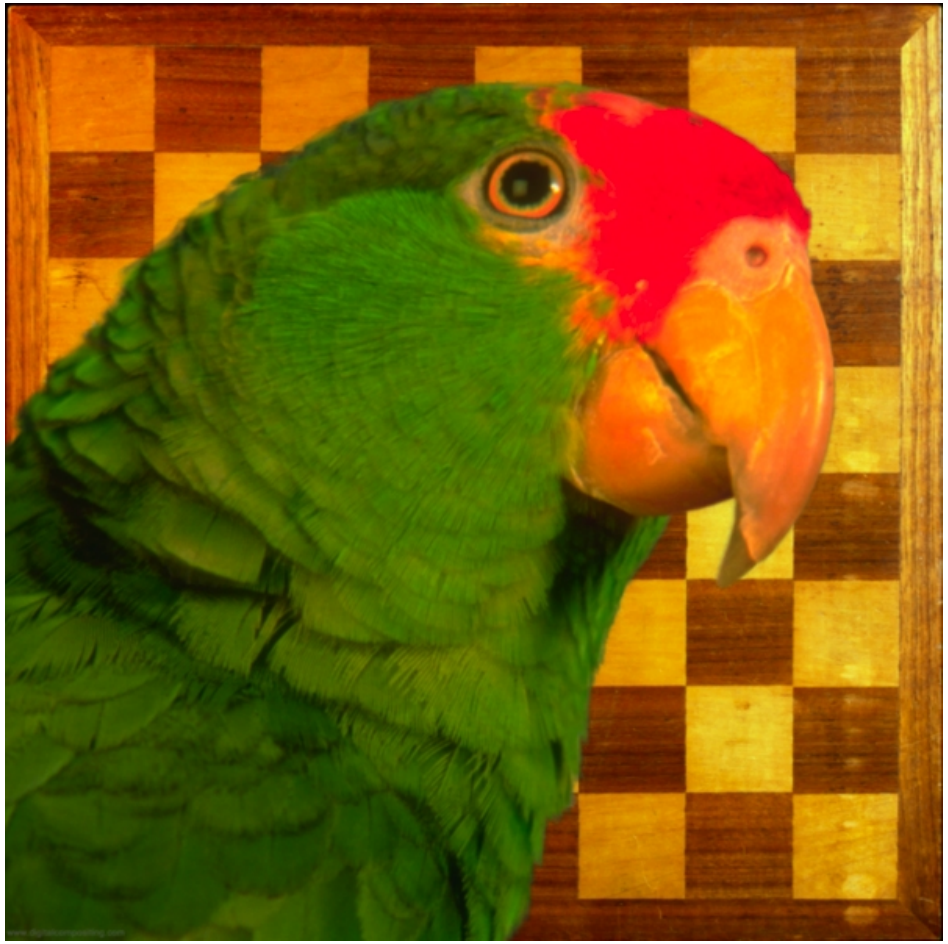


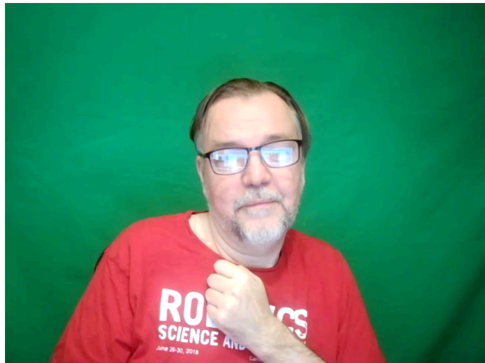
(1-a)B





KeyMix: aF + (1-a)B





5. Image Filtering

Image filtering: compute function of local neighborhood at each position

- Very important!
  - Enhance images
    - Denoise, resize, increase contrast, etc.
  - Extract information from images
    - Texture, edges, distinctive points, etc.
  - Detect patterns
    - Template matching
  - Deep Convolutional Networks

Example: box filter

$$g[\cdot\, ,\cdot\,]$$

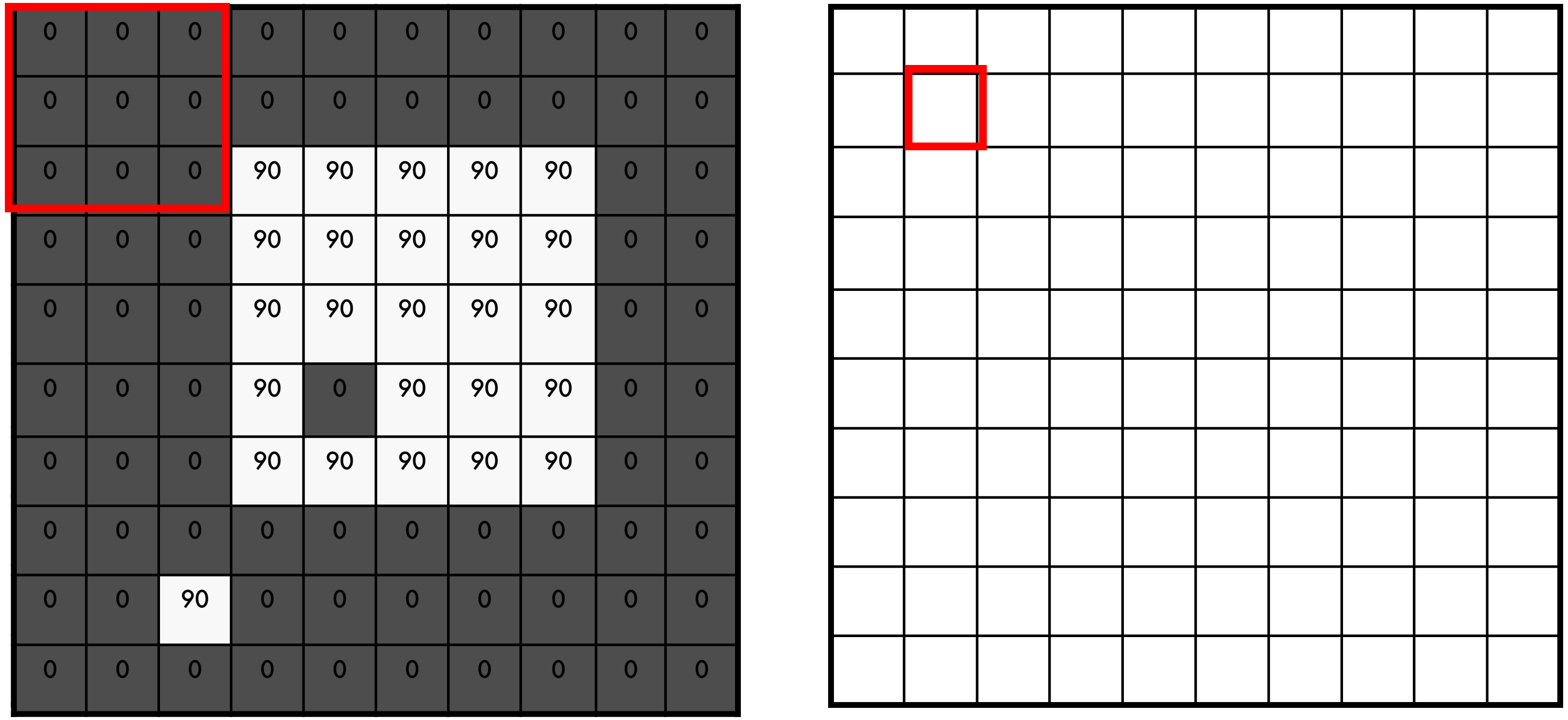
$$\frac{1}{9}\begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

Slide credit: David Lowe (

Image filtering

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

$f[.,.]$ $\qquad$ $h[.,.]$



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

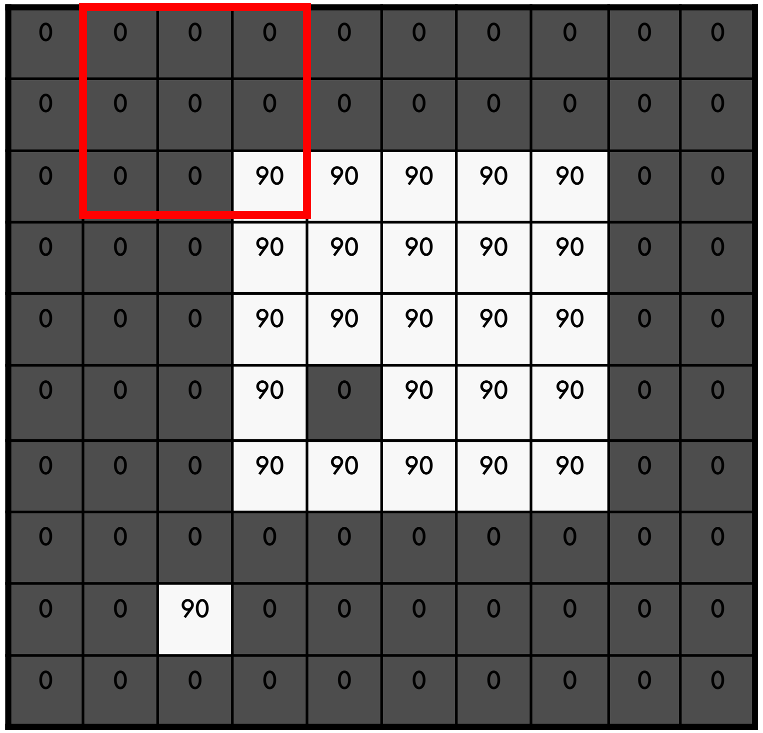
$$h[m,n]=\sum_{k,l} g[k,l]\, f[m+k,n+l]$$

Credit:

Image filtering

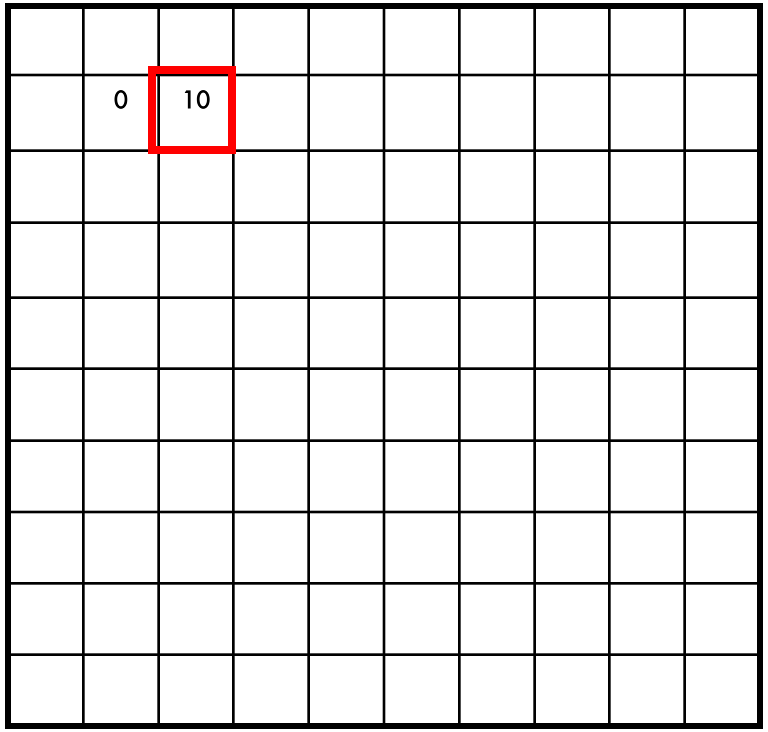
$$g[\cdot\,,\cdot\,]\ \frac{1}{9}\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}$$

$f[.,.]$



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

$h[.,.]$



	0	10							

$$h[m,n]=\sum_{k,l}g[k,l]\,f[m+k,n+l]$$

Credit:

Image filtering

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

$f[.,.]$

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

$h[.,.]$

	0	10	20						

$$h[m,n]=\sum_{k,l} g[k,l]\, f[m+k,n+l]$$

Credit:

Image filtering

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

$f[.,.]$

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

$h[.,.]$

	0	10	20	30					

$$h[m,n]=\sum_{k,l} g[k,l]\, f[m+k,n+l]$$

Credit:

Image filtering

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

$f[.,.]$

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

$h[.,.]$

	0	10	20	30	30				

$$h[m,n]=\sum_{k,l} g[k,l]\, f[m+k,n+l]$$

Credit:

Image filtering

$g[\cdot,\cdot]\ \frac{1}{9}$

1	1	1
1	1	1
1	1	1

$f[.,.]$

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

$h[.,.]$

	0	10	20	30	30				
				?					

$$h[m,n]=\sum_{k,l} g[k,l]\, f[m+k,n+l]$$

Credit:

Image filtering

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

$f[.,.]$

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

$h[.,.]$

	0	10	20	30	30				
						?			
				50					

$$h[m,n]=\sum_{k,l} g[k,l]\, f[m+k,n+l]$$

Credit:

Image filtering

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

$$f[.,.]$$

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

$$h[.,.]$$

	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	

$$h[m,n]=\sum_{k,l} g[k,l]\, f[m+k,n+l]$$

Credit:

Box Filter

What does it do?

- Replaces each pixel with an average of its neighborhood

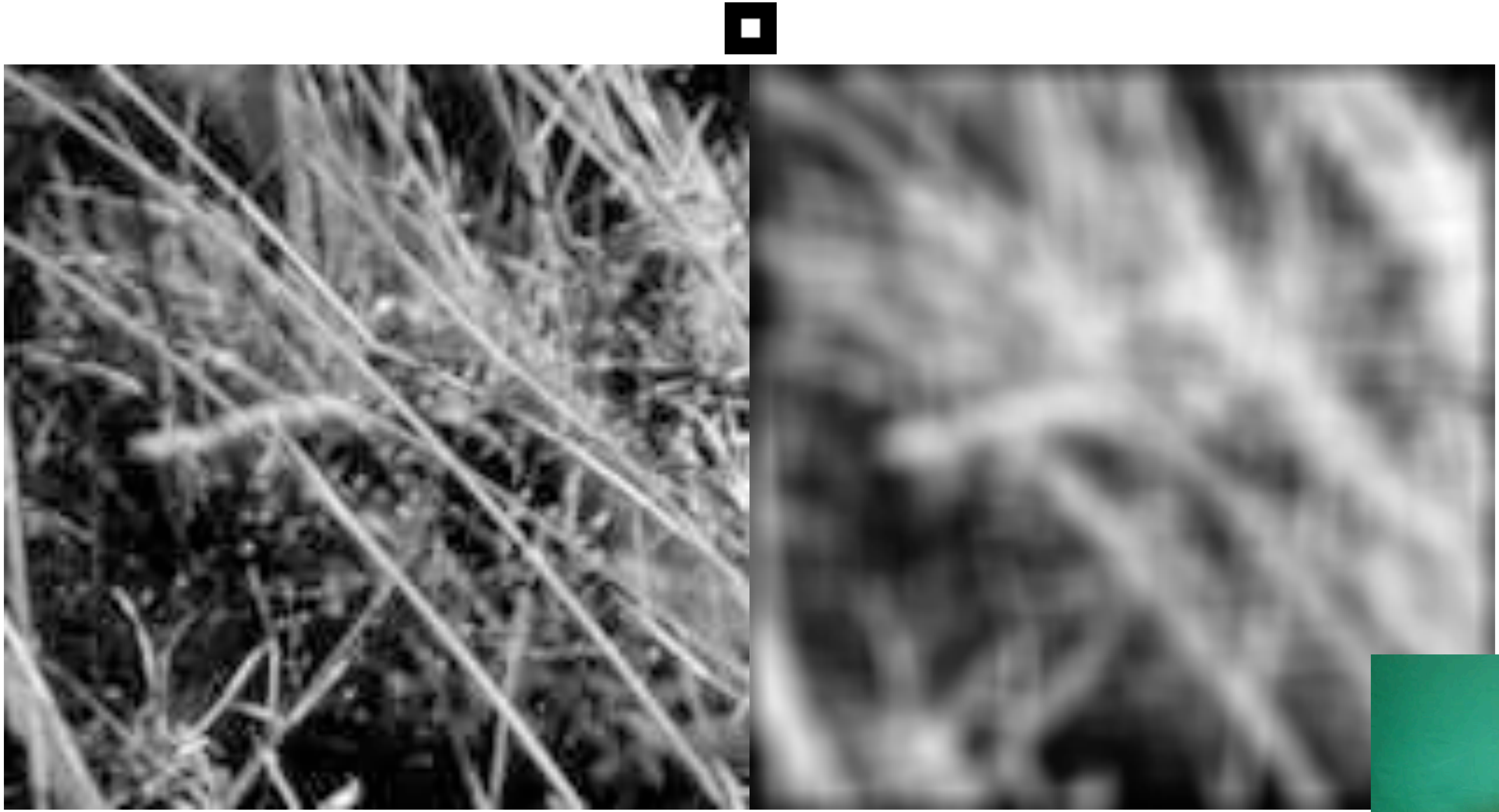
- Achieve smoothing effect (remove sharp features)

$$g[\cdot\,,\cdot\,]$$

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

Slide credit: David Lowe (

Smoothing with box filter

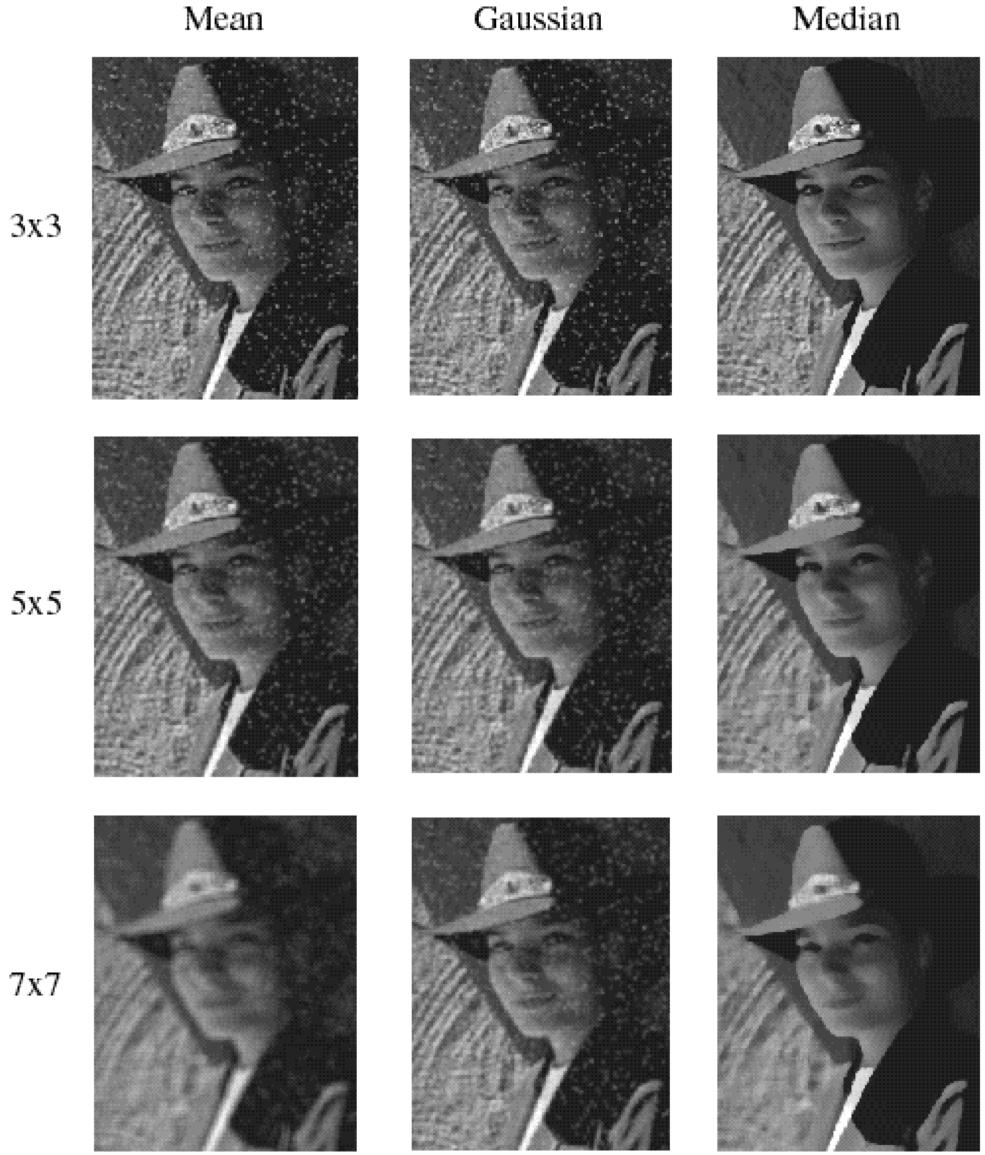


Median filters

- A **Median Filter** operates over a window by selecting the median intensity in the window.
- What advantage does a median filter have over a mean filter?
- Is a median filter a kind of convolution?

Slide by Steve Seitz

Comparison: salt and pepper noise



Slide by Steve Seitz

Summary

1. Computer Vision defined
2. Applications of CV are plentiful!
3. Images are 2D arrays of pixel values
4. Basic image processing: contrast, intensity, histogram eq., arithmetic
5. Image filtering: convolution (linear) and non-linear (median)