CS 1674: Intro to Computer Vision

Matlab Tutorial

Prof. Adriana Kovashka
University of Pittsburgh

August 31, 2016

Plan for Today

- Intro
  - Course basics refresher
  - Overview of topics
  - What are images?
  - Linear algebra lightning-quick over-/re-view
- **Matlab tutorial**
- Outro
  - Overview of Homework 1 W and P (if time)

Course Info

- **Course website:** http://people.cs.pitt.edu/~kovashka/cs1674
- **Instructor**: Adriana Kovashka (kovashka@cs.pitt.edu)
  - Please use "CS1674" at the beginning of your Subject
- **Office**: Sennott Square 5129
- **Office hours**: MW, 3:30pm - 4:25pm

TAs

- **TA/Grader**: Yuhuan Jiang (yuhuan@cs.pitt.edu)
- **Additional TAs (office hours only):** Chris Thomas and Nils Murrugarra
- **TAs' office hours:** TBD
- **Your homework:** Fill out the Doodle at http://doodle.com/poll/gskprtb5uq5k85bm (ignore dates, look at days of the week, 30-min increments)

Matlab

- You can get it for free through my.pitt.edu → My Resources → Software Downloads
- Get the latest version (mostly because that's what I use)
- Make sure to check the "Image Processing Toolbox", "Computer Vision System Toolbox", and "Statistics and Machine Learning Toolbox" boxes during installation; easiest to install all

Course Components

- Written HW (11 assignments x 1% each = 11%)
- Programming HW (11 assignments x 4% each = 44%)
- Midterm exam (15%)
- Final exam (25%)
- Participation (5%)

The Rest of the Course Policies…

- Read the course website carefully!

Warnings

Warning #1

- This class is **a lot of work**
- This time I’ve opted for shorter, more manageable HW assignments, but there is more of them
- I expect you’d be spending **6-8 hours** on homework each week

- … But you get to understand algorithms and concepts in detail!

Warning #2

- Some parts will be **hard** and require that you pay close attention!
- … I will use the written HW to gauge how you're doing
- … I will also pick on students randomly to answer questions
- **Use instructor's and TAs' office hours!!!**

- … You will learn a lot!

Warning #3

- Programming assignments will be in Matlab since that's very common in computer vision, and is optimized for work with matrices
- Matlab also has great documentation
- HW1P is just Matlab practice
- Some people **won't like Matlab** (I like it!)

- … You will learn a new programming language!

Clarification from last time

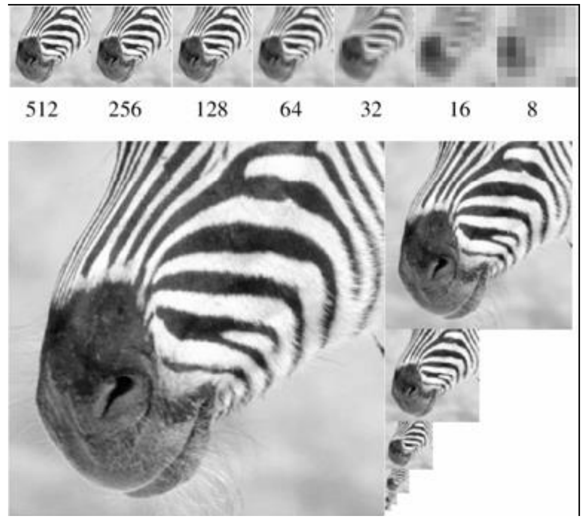
- What is the role of datasets?
  - Whatever our computer vision algorithms learn, they will learn from some (set of) datasets of images; we will also use the datasets to test our algorithms
- Why did I say it’s hard to understand deep learning (a.k.a. deep neural networks)?
  - Because most deep learning methods just take an image as input and output predictions, and they learn how to represent and examine the image *on their own*, so they appear to be “black boxes”

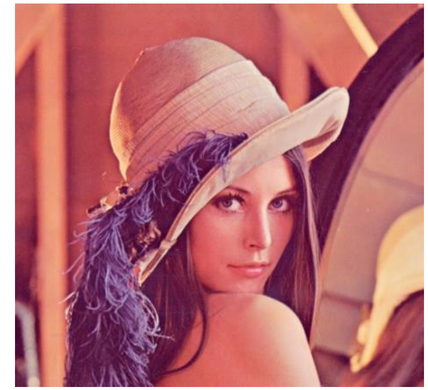
Questions?

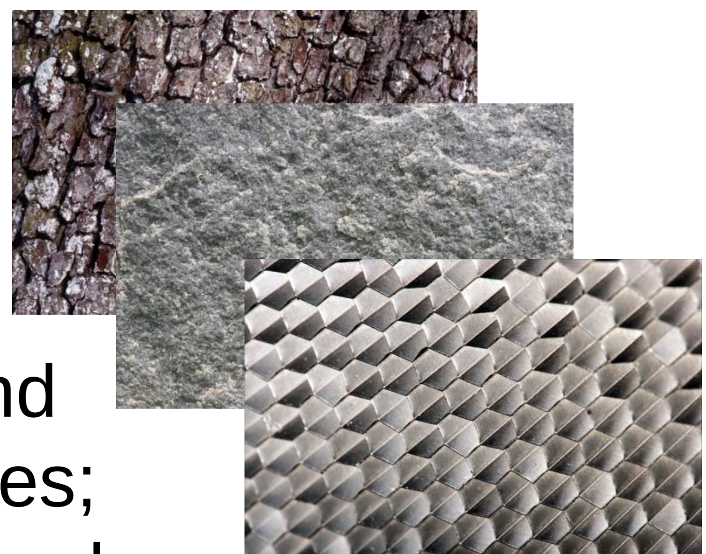
Overview of Topics

The next 13 slides will be very quick, then we'll slow down.
Ready?

Features and filters





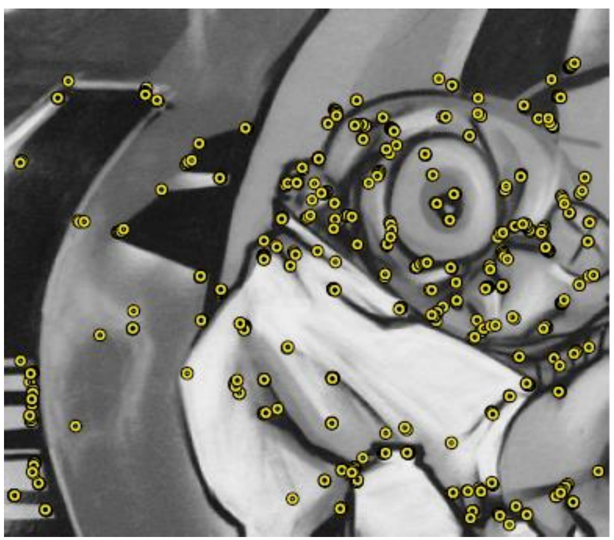


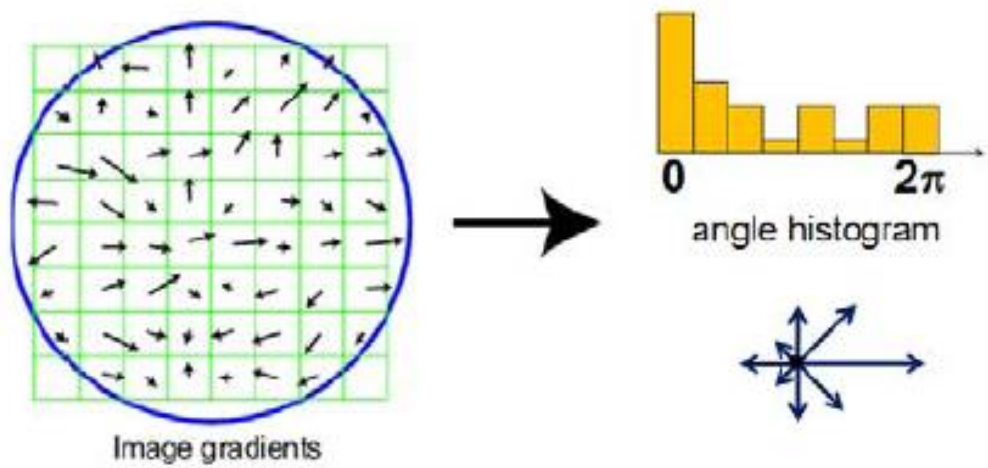
- Transforming and describing images; textures, colors, edges

Kristen Grauman

Features and filters

- Detecting distinctive + repeatable features
- Describing images with local statistics





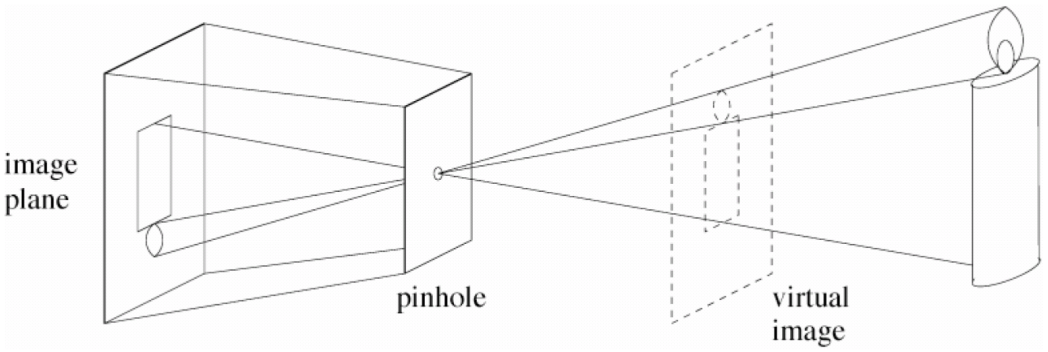
Indexing and search

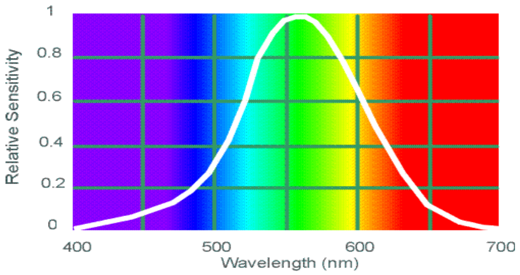
- Matching features and regions across images

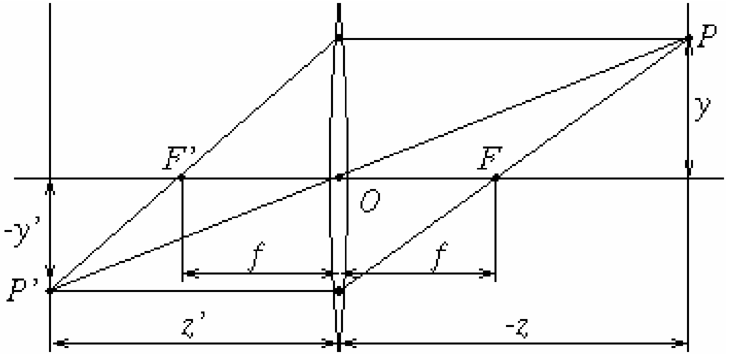
Kristen Grauman

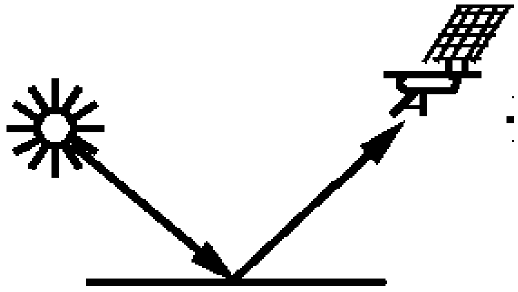
Image formation

- How does light in 3d world project to form 2d images?



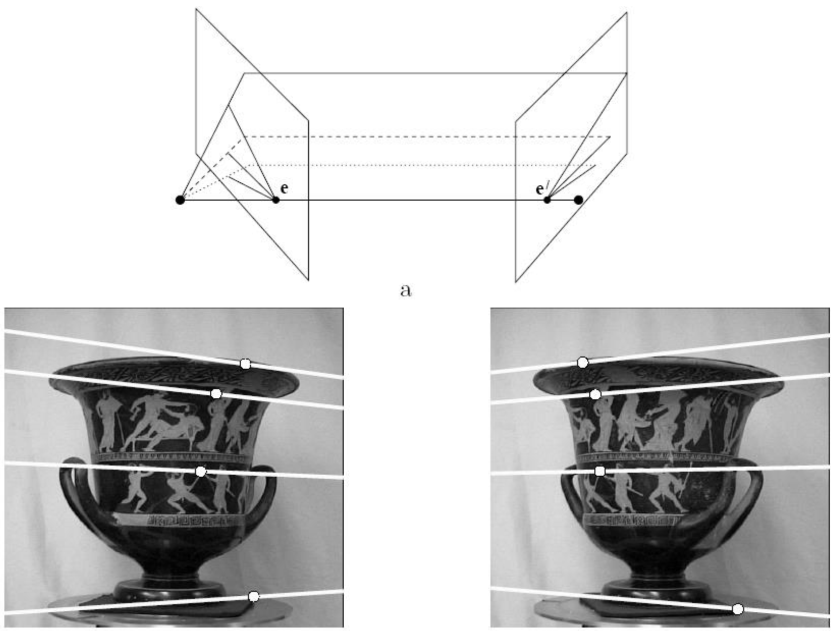




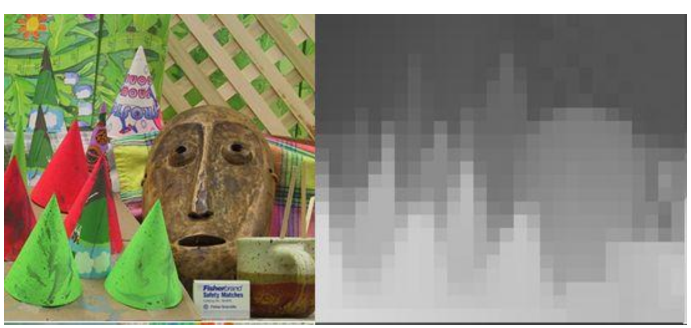


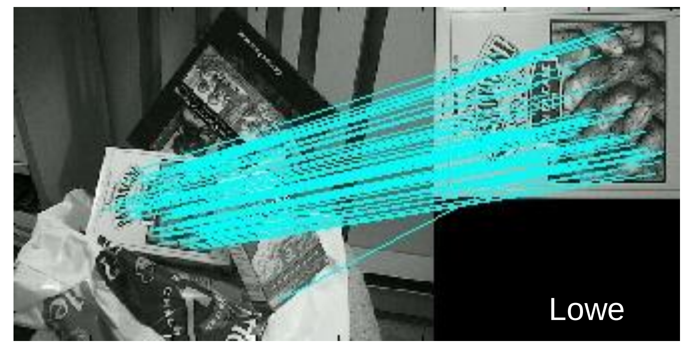
Multiple views

- Multi-view geometry, matching, invariant features, stereo vision



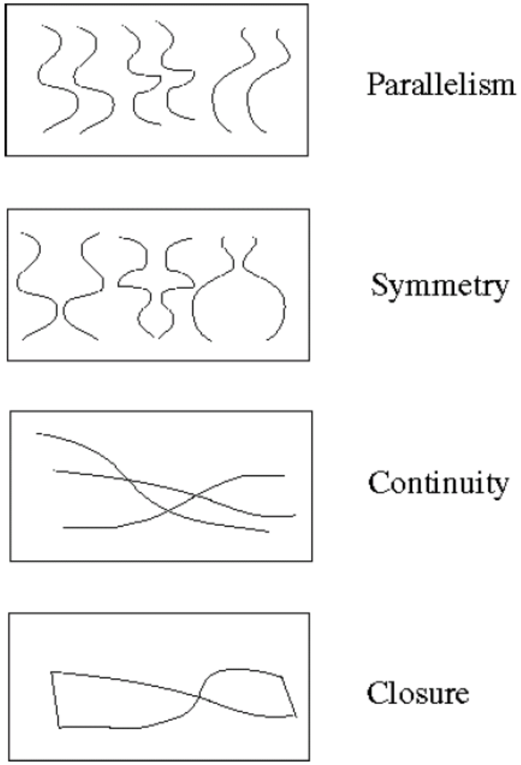
Hartley and Zisserman





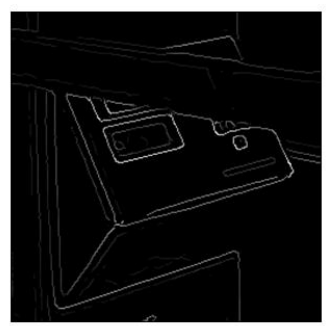
Kristen Grauman

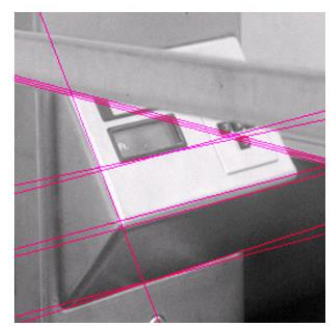
Grouping and fitting

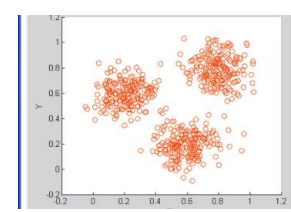


[fig from Shi et al]

- Clustering, segmentation, fitting; what parts belong together?

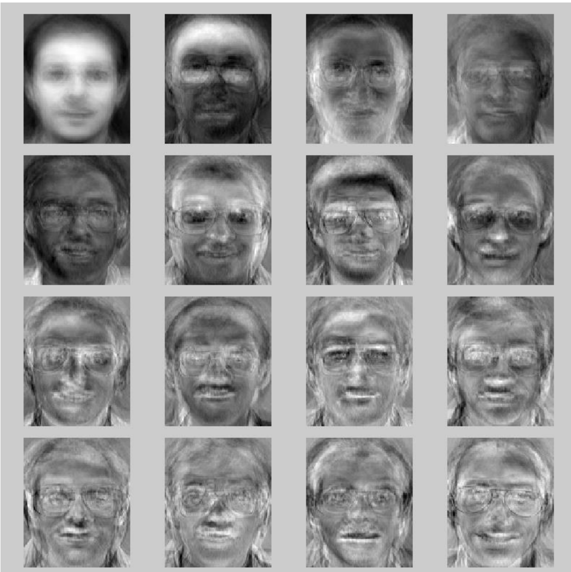


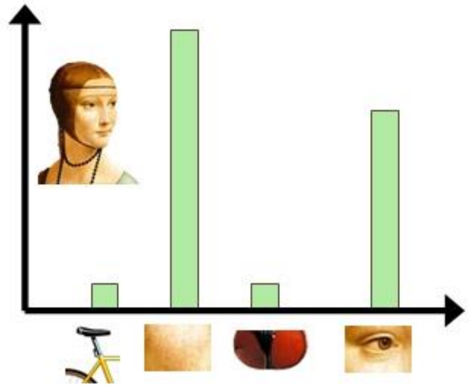


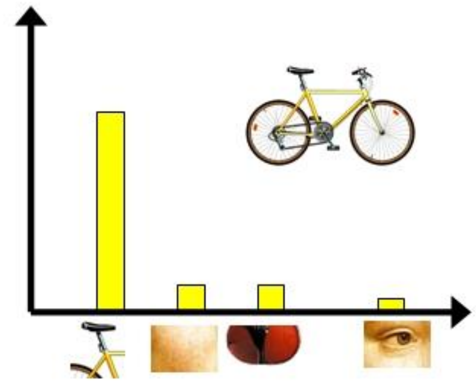


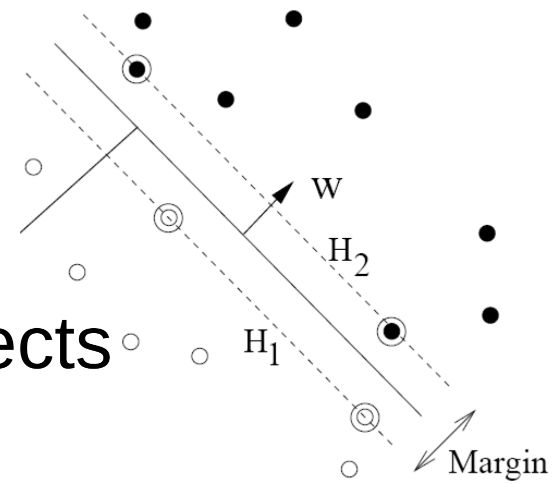
Kristen Grauman

Visual recognition







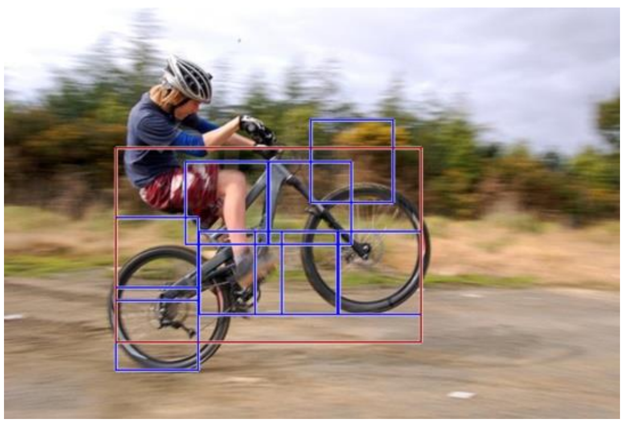


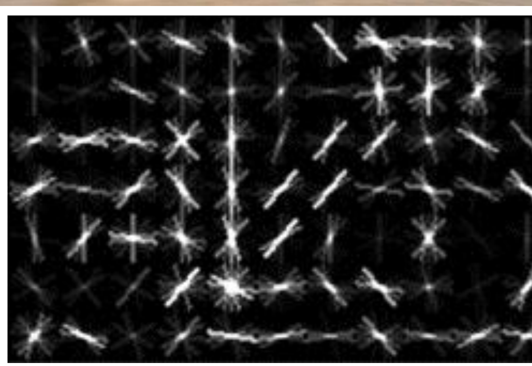
- Recognizing objects and categories, learning techniques

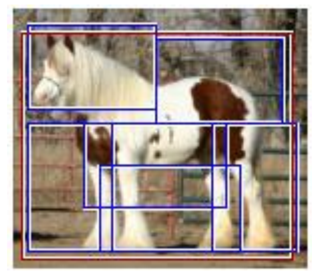
Kristen Grauman

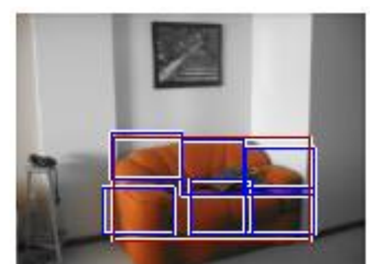
Object detection

- Detecting novel instances of objects
- Classifying regions as one of several categories



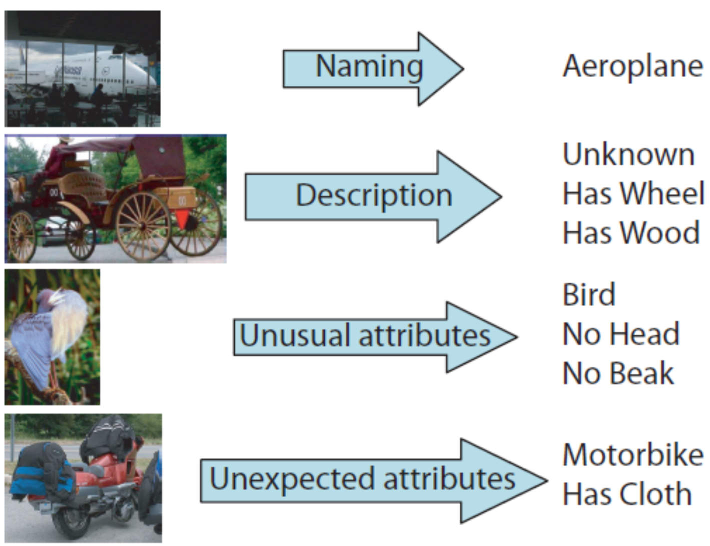






Attribute-based description

- Describing the high-level properties of objects
- Allows recognition of unseen objects



otter

black: yes
white: no
brown: yes
stripes: no
water: yes
eats fish: yes

polar bear

black: no
white: yes
brown: no
stripes: no
water: yes
eats fish: yes

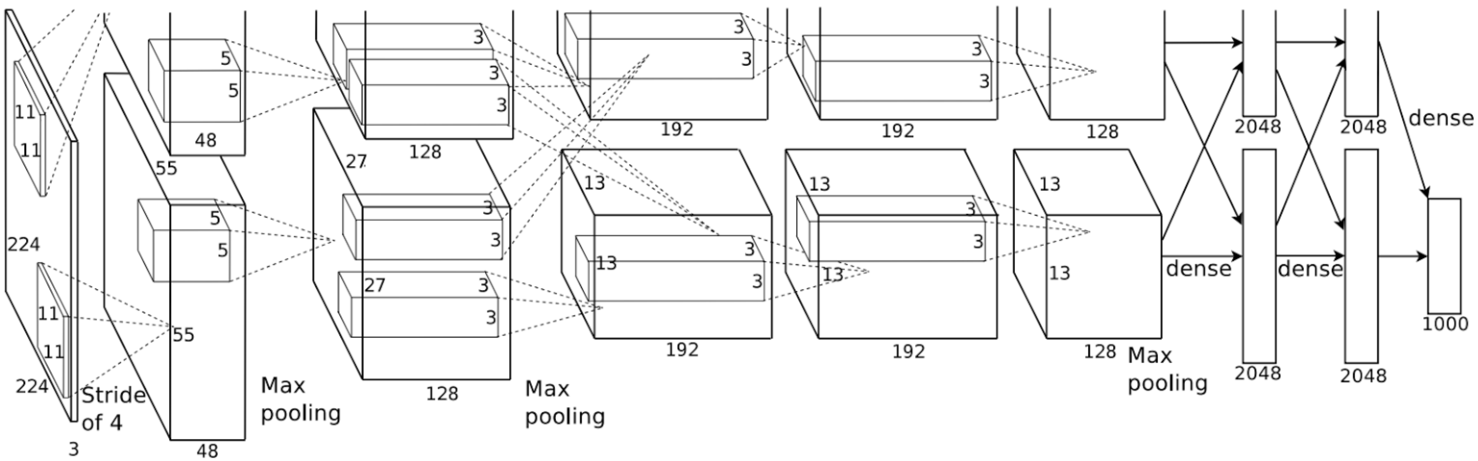
zebra

black: yes
white: yes
brown: no
stripes: yes
water: no
eats fish: no

Convolutional neural networks

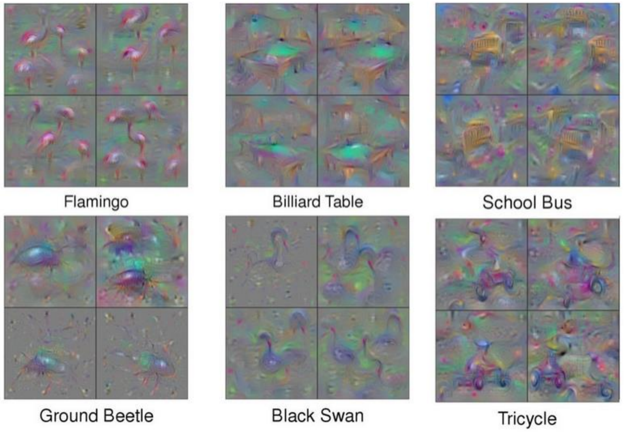
- State-of-the-art on many recognition tasks

Image



Prediction

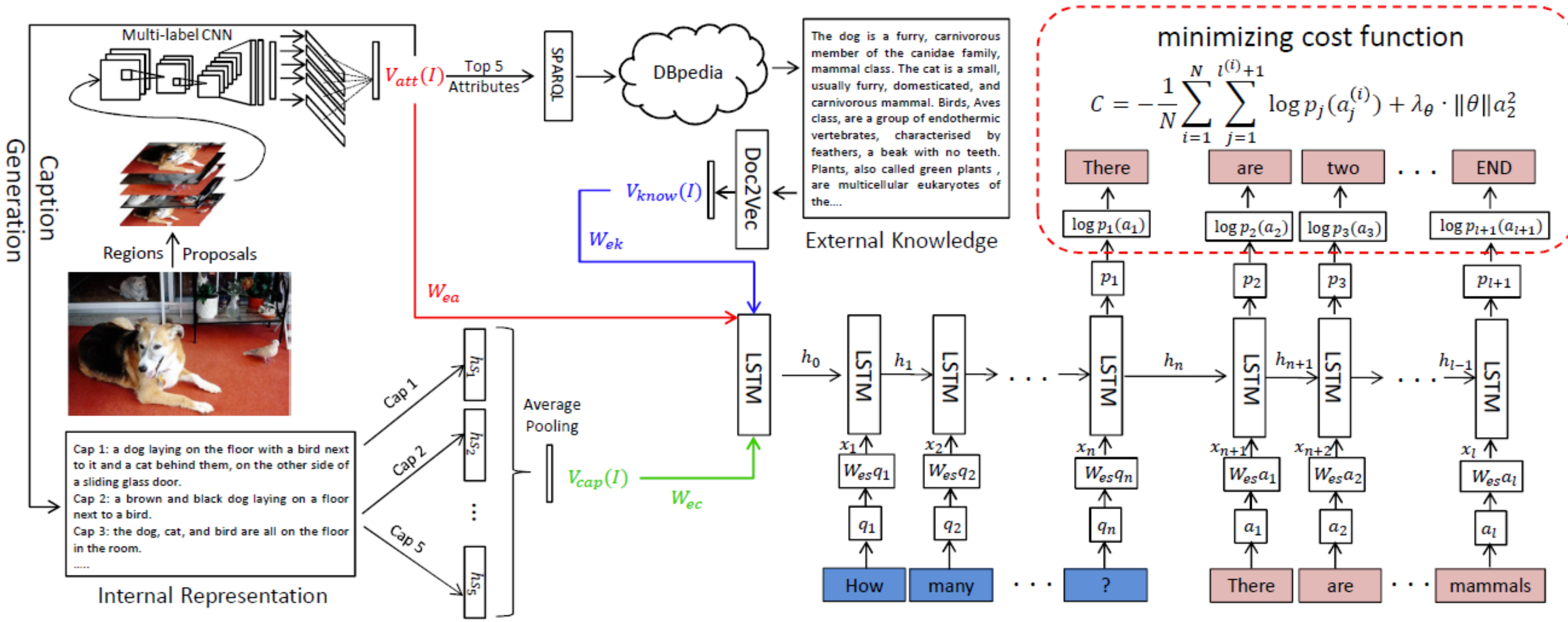
Krizhevsky et al.



Yosinski et al., ICML DL workshop 2015

Recurrent neural networks

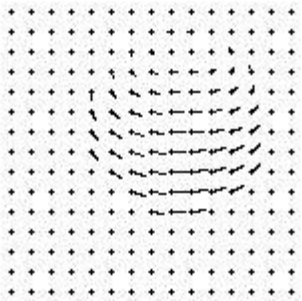
- Sequence processing, e.g. question answering

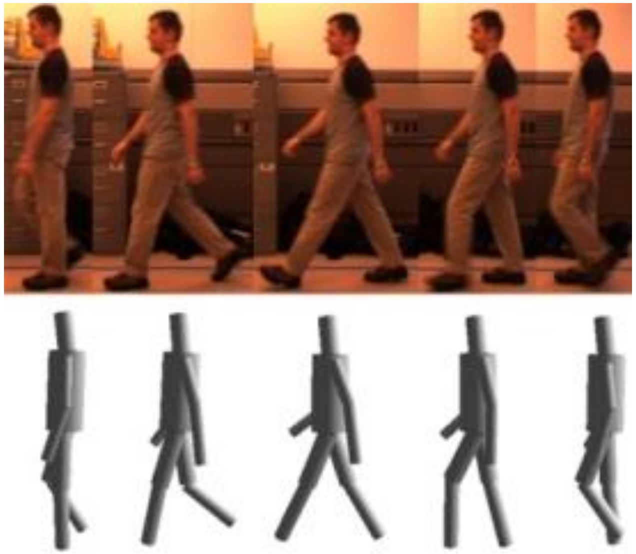


Wu et al., CVPR 2016

Motion and tracking

- Tracking objects, video analysis



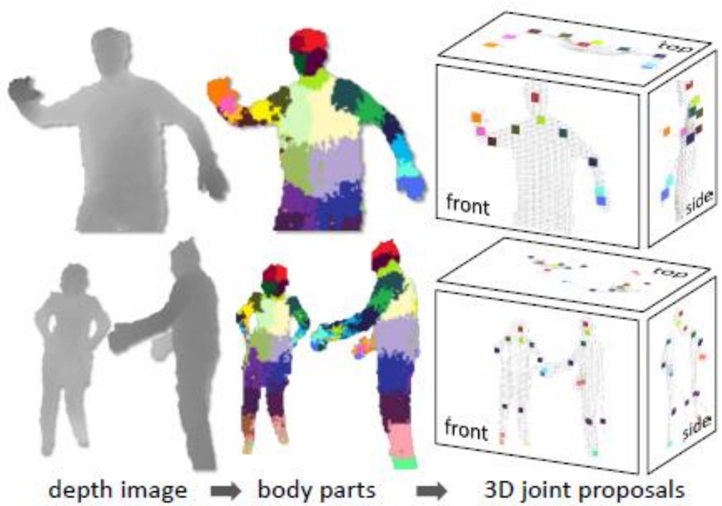


Tomas Izo

Kristen Grauman

Pose and actions

- Automatically annotating human pose (joints)
- Recognizing actions in first-person video

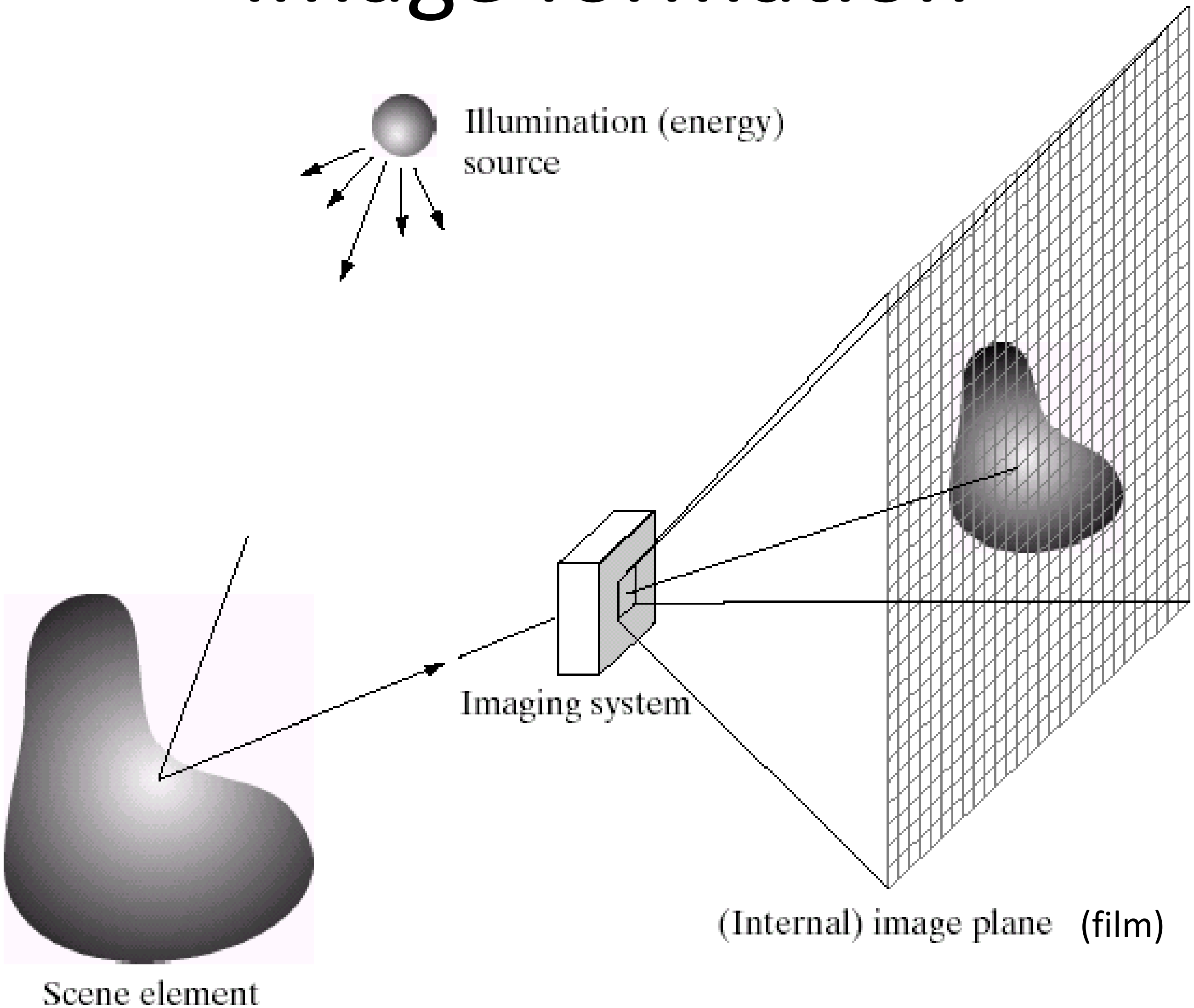


What are images?

What are images? (in Matlab)

- Matlab treats images as matrices of numbers
- To proceed, let’s talk very briefly about how images are formed

Image formation



Slide credit: Derek Hoiem

Digital camera

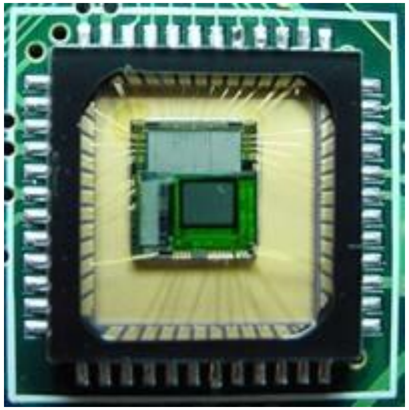
A digital camera replaces film with a sensor array

- Each cell in the array is light-sensitive diode that converts photons to electrons

http://electronics.howstuffworks.com/cameras-photography/digital/digital-camera.htm

Slide by Steve Seitz

Digital images



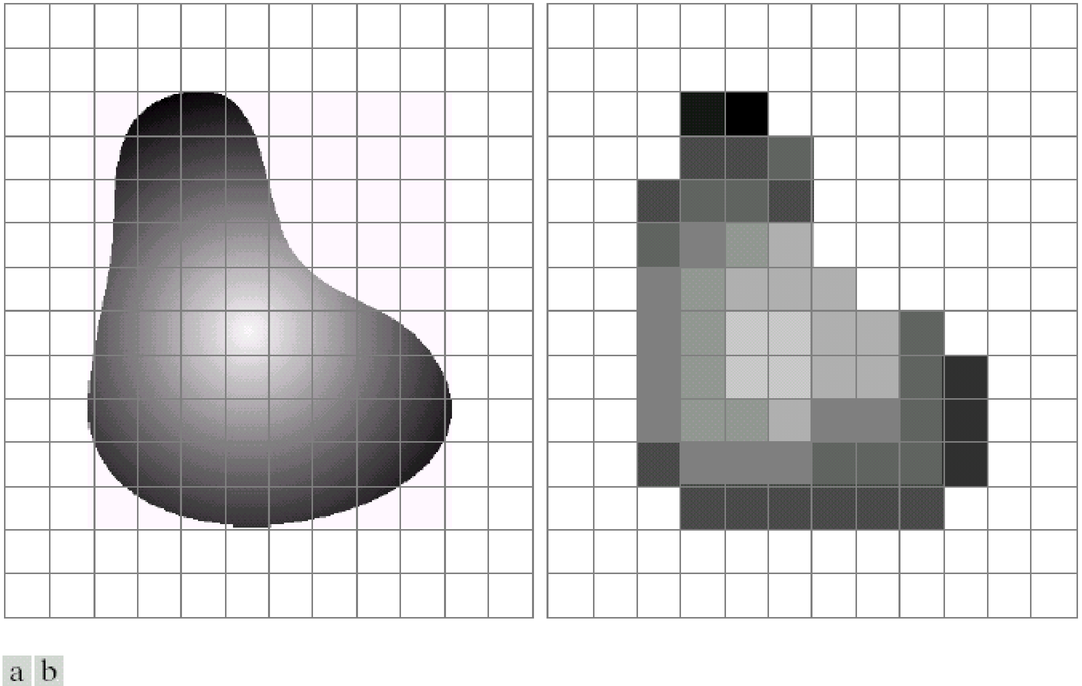


FIGURE 2.17 (a) Continuos image projected onto a sensor array. (b) Result of image sampling and quantization.

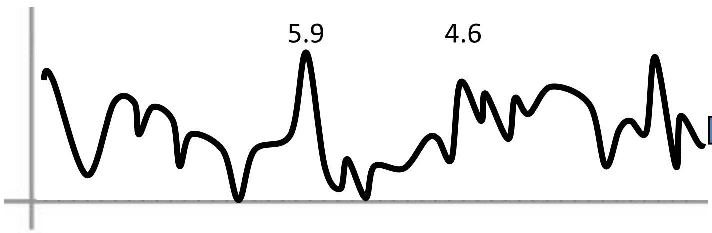
- **Sample** the 2D space on a regular grid
- **Quantize** each sample (round to nearest integer)

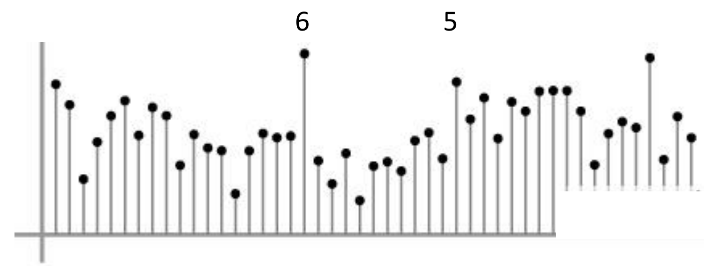
Slide credit: Derek Hoiem, Steve Seitz

Digital images

- **Sample** the 2D space on a regular grid
- **Quantize** each sample (round to nearest integer)
- What does quantizing signal look like?

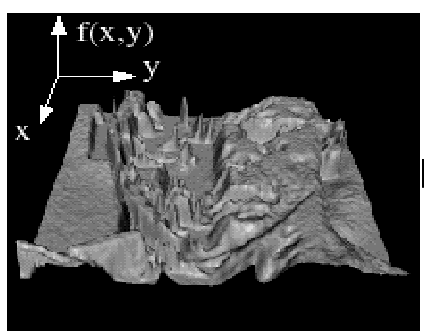
1D





- Image thus represented as a matrix of integer values.

2D



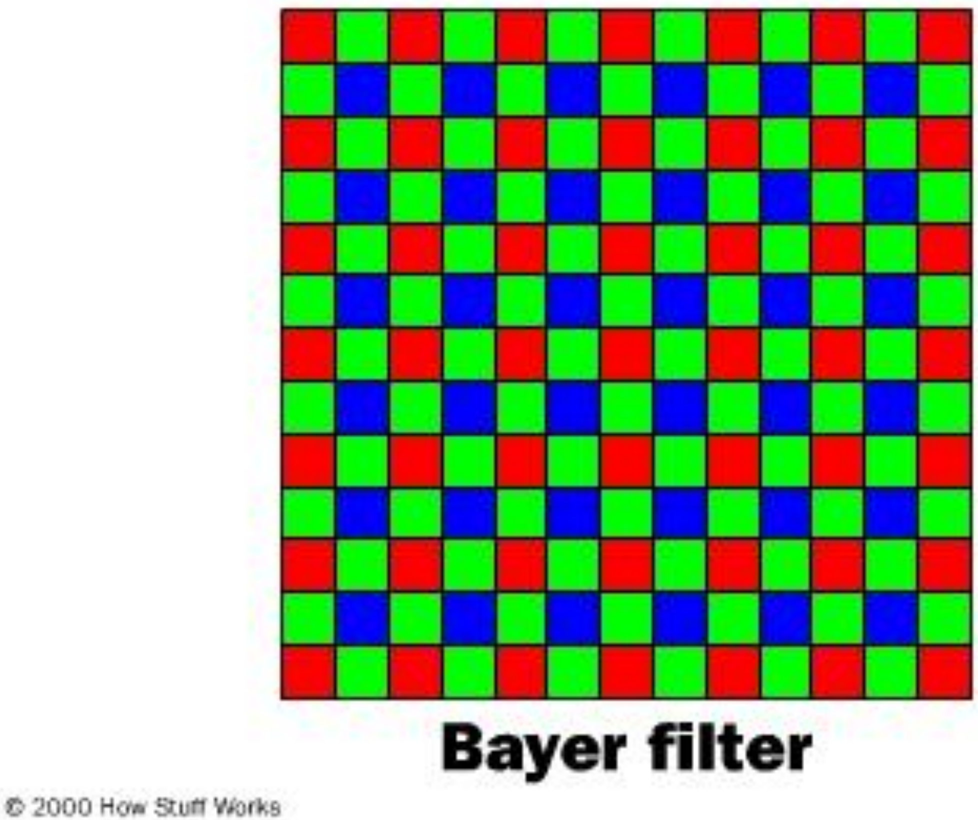
j

i

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

Adapted from S. Seitz

Digital color images



Slide credit: Kristen Grauman

Digital color images

Color images,
RGB color space:

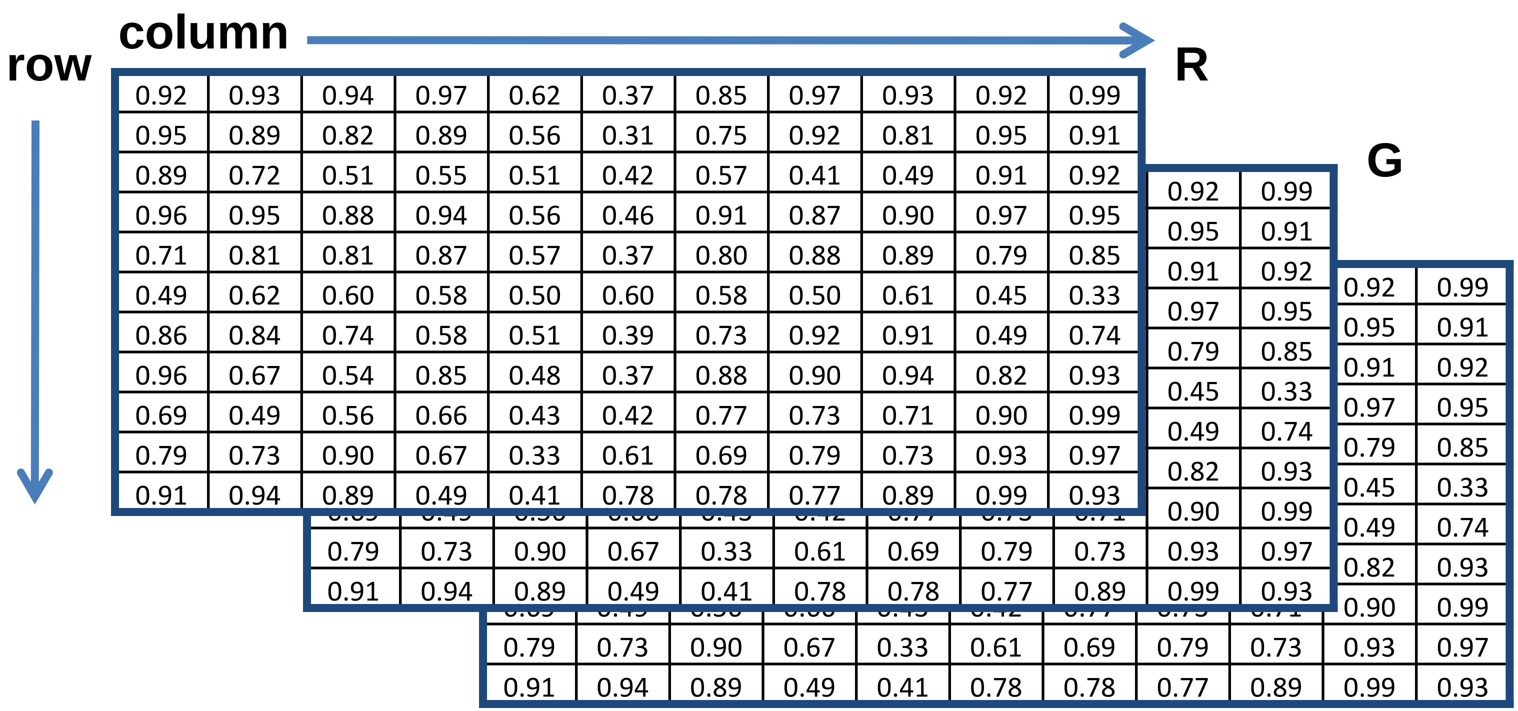
Split image into
three channels

R G B

Adapted from Kristen Grauman

Images in Matlab

- Color images represented as a matrix with multiple channels (=1 if grayscale)
- Suppose we have a NxM RGB image called "im"
  - im(1,1,1) = top-left pixel value in R-channel
  - im(y, x, b) = y pixels **down**, x pixels **to right** in the bth channel
  - im(N, M, 3) = bottom-right pixel in B-channel
- imread(filename) returns a uint8 image (values 0 to 255)
  - Convert to double format (values 0 to 1) with double



R

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

G

									0.92	0.99
									0.95	0.91
									0.91	0.92
									0.97	0.95
									0.79	0.85
									0.45	0.33
									0.49	0.74
									0.82	0.93
									0.90	0.99
0.79	0.73	0.90	0.67	0.33	0.61	0.69	0.79	0.73	0.93	0.97
0.91	0.94	0.89	0.49	0.41	0.78	0.78	0.77	0.89	0.99	0.93

B

									0.92	0.99
									0.95	0.91
									0.91	0.92
									0.97	0.95
									0.79	0.85
									0.45	0.33
									0.49	0.74
									0.82	0.93
									0.90	0.99
0.79	0.73	0.90	0.67	0.33	0.61	0.69	0.79	0.73	0.93	0.97
0.91	0.94	0.89	0.49	0.41	0.78	0.78	0.77	0.89	0.99	0.93

Adapted from Derek Hoiem

Review and Tutorial

- Linear algebra
  - Very brief, all you need to know for most of course
  - Exception: Last three lectures before midterm (some more review then)
  - Raise your hand if you've had a linear algebra course
- Matlab: http://www.cs.pitt.edu/~kovashka/cs1674/tutorial.m

Vectors and Matrices

- Vectors and matrices are just collections of ordered numbers that represent something: movements in space, scaling factors, word counts, movie ratings, pixel brightnesses, etc.
- We'll define some common uses and standard operations on them.

Vector

- A column vector $\mathbf{v} \in \mathbb{R}^{n \times 1}$ where

$$\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$$

- A row vector $\mathbf{v}^T \in \mathbb{R}^{1 \times n}$ where

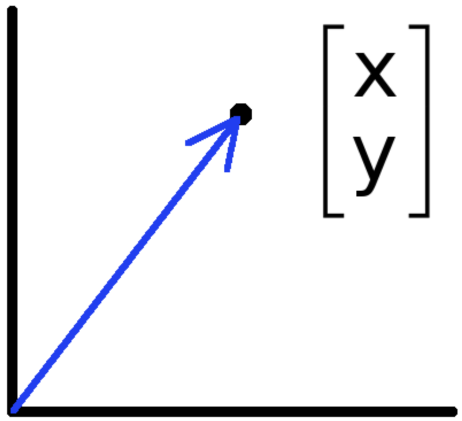
$$\mathbf{v}^T = \begin{bmatrix} v_1 & v_2 & \dots & v_n \end{bmatrix}$$

T denotes the transpose operation

Vector

- You'll want to keep track of the orientation of your vectors when programming in MATLAB.
- You can transpose a vector *V* in MATLAB by writing **V'**.

Vectors have two main uses



- Vectors can represent an offset in 2D or 3D space
- Points are just vectors from the origin

- Data can also be treated as a vector
- Such vectors don't have a geometric interpretation, but calculations like "distance" still have value

Matrix

- A matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ is an array of numbers with size $m \downarrow$ by $n \rightarrow$, i.e. *m* rows and *n* columns.

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ \vdots & & & & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \dots & a_{mn} \end{bmatrix}$$

- If $m = n$, we say that $\mathbf{A}$ is square.

Matrix Operations

- Addition

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} + \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} a+1 & b+2 \\ c+3 & d+4 \end{bmatrix}$$

  – Can only add matrices with matching dimensions, or a scalar to a matrix.

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} + 7 = \begin{bmatrix} a+7 & b+7 \\ c+7 & d+7 \end{bmatrix}$$

- Scaling

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \times 3 = \begin{bmatrix} 3a & 3b \\ 3c & 3d \end{bmatrix}$$

Matrix Operations

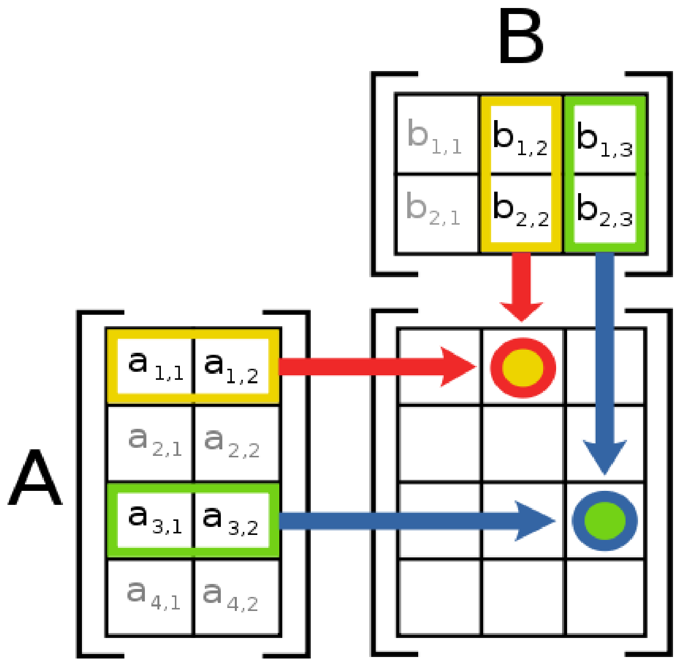
- Inner product (*dot* · product) of vectors
  - Multiply corresponding entries of two vectors and add up the result
  - We won't worry about the geometric interpretation for now

$$\mathbf{x}^T\mathbf{y} = \begin{bmatrix} x_1 & \dots & x_n \end{bmatrix} \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} = \sum_{i=1}^{n} x_i y_i \qquad \text{(scalar)}$$

Matrix Operations

- Multiplication

- The product AB is:



- Each entry in the result is (that row of A) dot product with (that column of B)

Matrix Operations

- Multiplication example:

A x B

$$\begin{bmatrix} 1 & 3 \\ 5 & 7 \end{bmatrix}$$

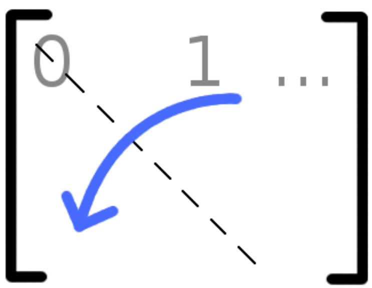
$$\begin{bmatrix} 0 & 2 \\ 4 & 6 \end{bmatrix} \quad \begin{bmatrix} \; & 14 \\ \; & \; \end{bmatrix}$$

$$0 \cdot 3 + 2 \cdot 7 = 14$$

  - Each entry of the matrix product is made by taking the dot product of the corresponding row in the left matrix, with the corresponding column in the right one.

Matrix Operations

- Transpose – flip matrix, so row 1 becomes column 1



$$\begin{bmatrix} 0 & 1 \\ 2 & 3 \\ 4 & 5 \end{bmatrix}^T = \begin{bmatrix} 0 & 2 & 4 \\ 1 & 3 & 5 \end{bmatrix}$$

- A useful identity:

$$(ABC)^T = C^T B^T A^T$$

Special Matrices

- Identity matrix **I**
  - Square matrix, 1's along diagonal, 0's elsewhere
  - **I** · [another matrix] = [that matrix]

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Diagonal matrix
  - Square matrix with numbers along diagonal, 0's elsewhere
  - A diagonal · [another matrix] scales the rows of that matrix

$$\begin{bmatrix} 3 & 0 & 0 \\ 0 & 7 & 0 \\ 0 & 0 & 2.5 \end{bmatrix}$$

1-Minute Break

Matlab Tutorial

http://www.cs.pitt.edu/~kovashka/cs1674/tutorial.m

We'll cover parts 1-4, do parts 5-6 at home.

Homework 1W and 1P

http://people.cs.pitt.edu/~kovashka/cs1674/hw1w.html

http://people.cs.pitt.edu/~kovashka/cs1674/hw1p.html

Next Time / Homework

- Finish the Matlab tutorial on your own + post on Piazza if questions

- No class Monday (Labor Day), but HW1W due
- Wednesday: Image filtering, HW1P due
- Reading for Wednesday: Szeliski Sec. 3.2

- Reminder: Fill out Doodle!