Wrap-up

http://www.cs.cmu.edu/~16385/

16-385 Computer Vision
Spring 2019, Lecture 26

Course announcements

- Homework 7 is due on Sunday 5th.
  - You can use any of your remaining late days.
  - Any questions about homework 7?
  - How many of you have looked at/started/finished the homework?

- Office hours this week:
  - Yannis will have additional office hours on Thursday, 6 – 8 pm.
  - On Friday, Bruce and Yannis will switch office hours.

Class evaluation*s* – please take them!

- CMU's Faculty Course Evaluations (FCE): https://cmu.smartevals.com/
- 16-385 end-of-semester survey: Will be posted on Piazza.
- Please take both, super helpful for developing future offerings of the class.
- Thanks in advance!

Course overview

1. Image processing. ← Lectures 1 – 7
   See also 18-793: Image and Video Processing

2. Geometry-based vision. ← Lectures 7 – 12
   See also 16-822: Geometry-based Methods in Vision

3. Physics-based vision. ← Lectures 13 – 16
   See also 16-823: Physics-based Methods in Vision
   See also 15-462: Computer Graphics
   See also 15-463: Computational Photography

4. Semantic vision. ← Lectures 17 – 20
   See also 16-824: Vision Learning and Recognition
   See also 10-703: Deep Reinforcement Learning

5. Dealing with motion. ← Lectures 21 – 24
   See also 16-831: Statistical Techniques in Robotics
   See also 16-833: Robot Localization and Mapping

Image processing

Image filtering

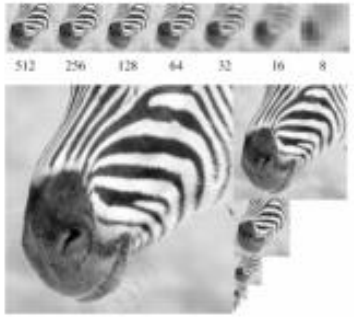


image pyramids

Fourier filtering

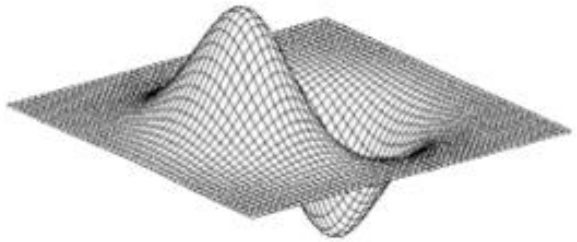
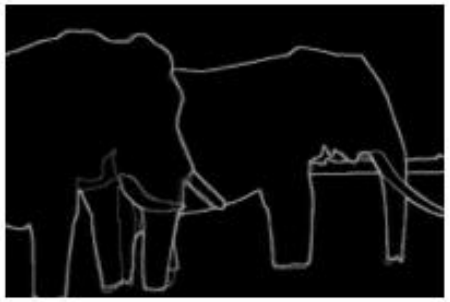
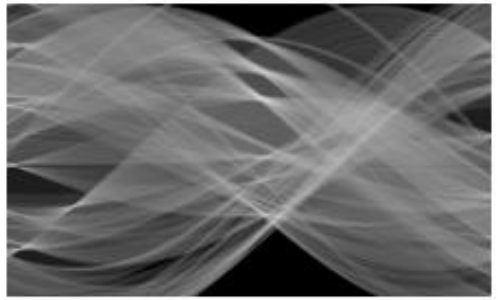


Image gradients



Boundaries

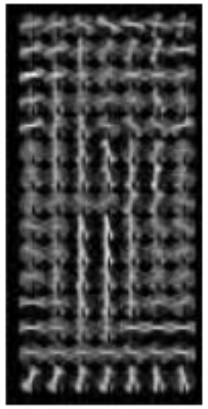


Hough Transform

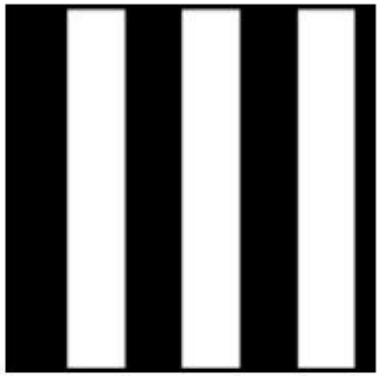
Image features

Corner detection Multi-scale detection

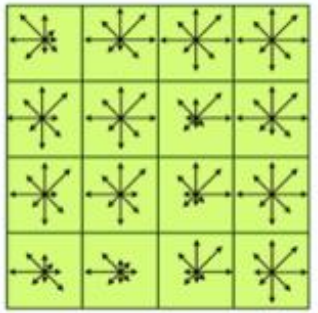
Haar-like



HOG



SURF



SIFT

2D alignment

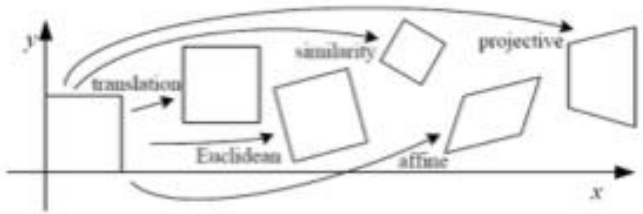


Figure 1: Basic set of 2D planar transformations

2D Transforms

DLT

RANSAC

H

Homography

Camera and multi-view geometry

$$x = \mathbf{P}\mathbf{X}$$

camera matrix

$$\mathbf{P}$$

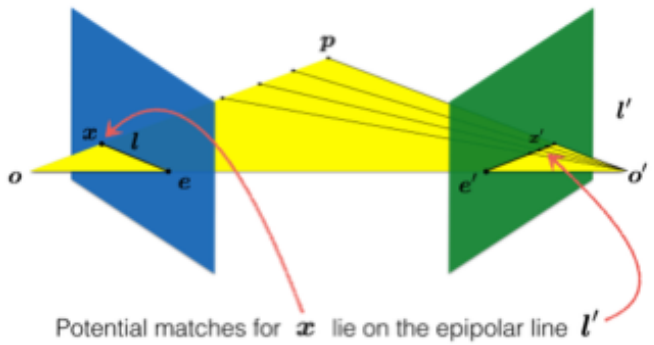
pose estimation

$$\mathbf{X}$$

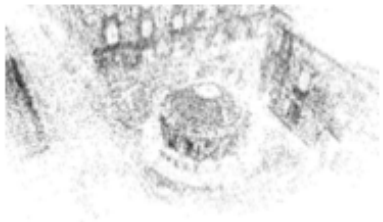
triangulation

$$\mathbf{F}$$

fundamental matrix

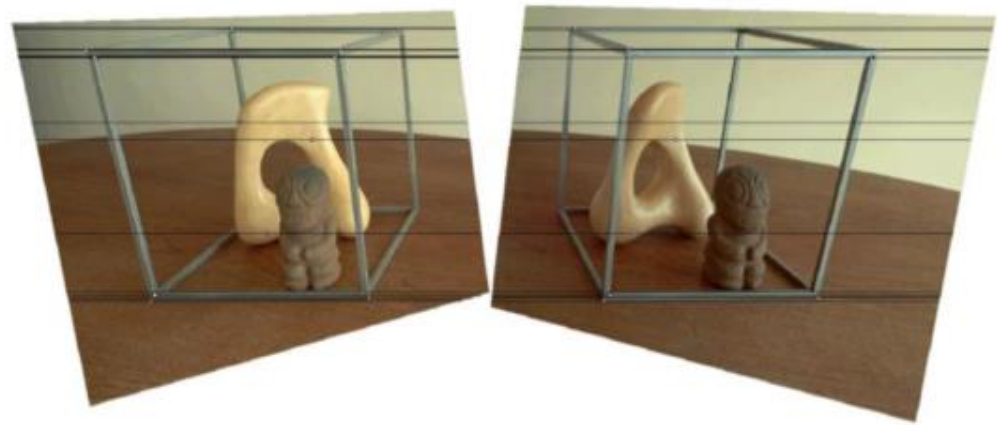


epipolar geometry

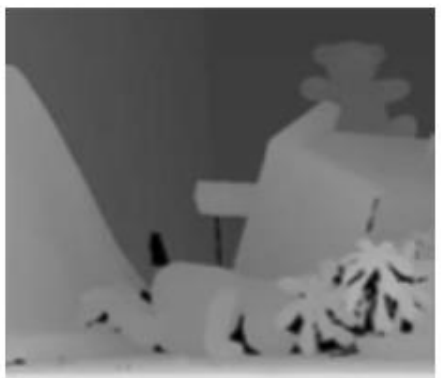


Reconstruction

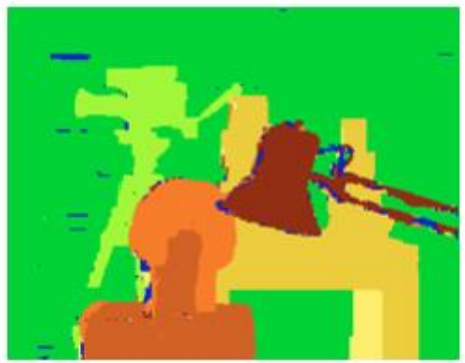
Stereo



Stereo Rectification

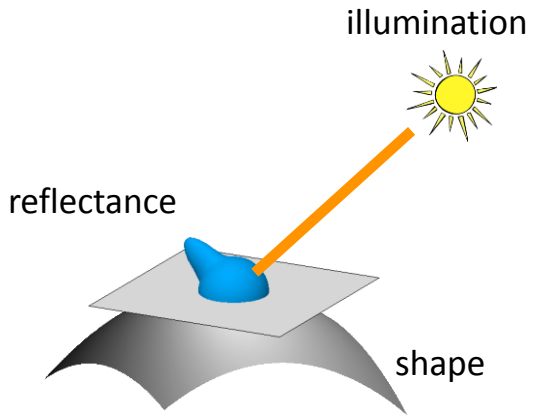


Block matching



Energy minimization

Image formation and physics



Radiometry and image formation

Photometric stereo

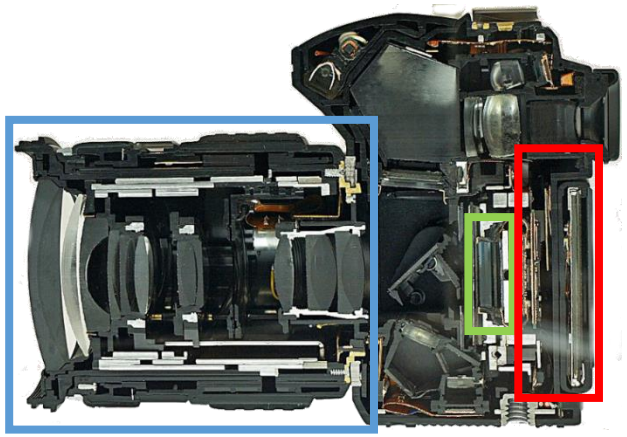
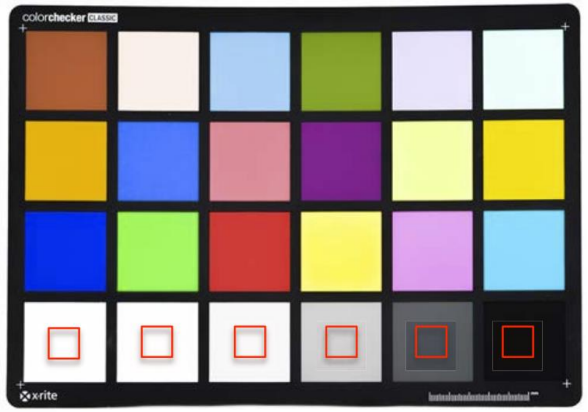


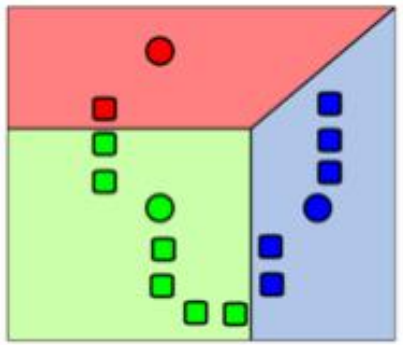
Image processing pipeline



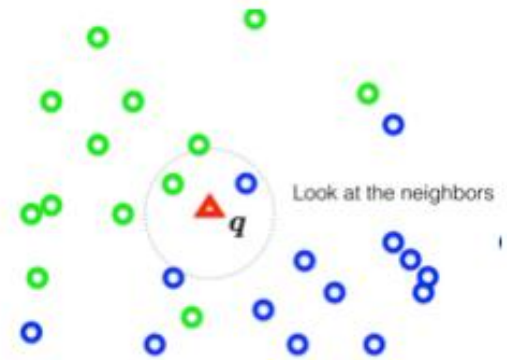
Radiometric and color calibration

Object recognition

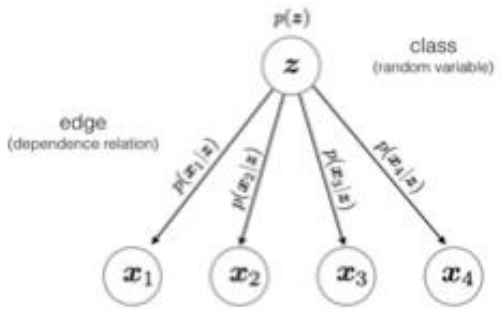
Bag-of-words



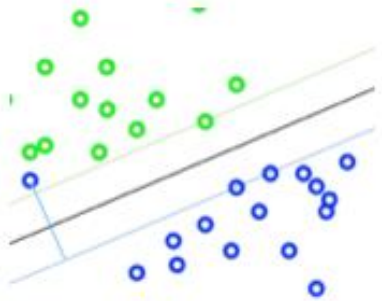
K-means



Nearest Neighbor

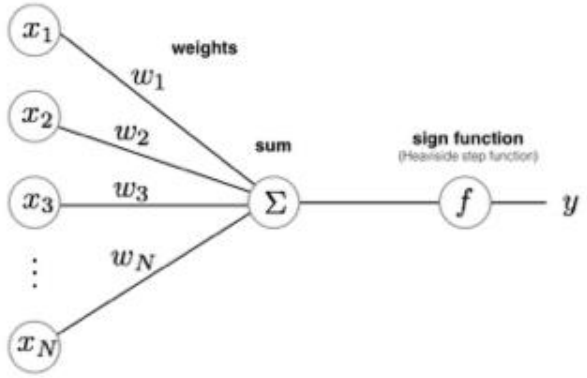


Naive Bayes

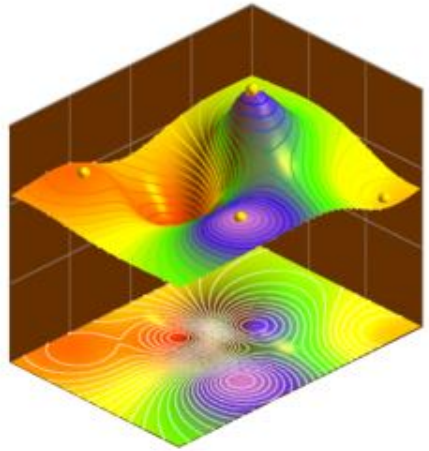


SVM

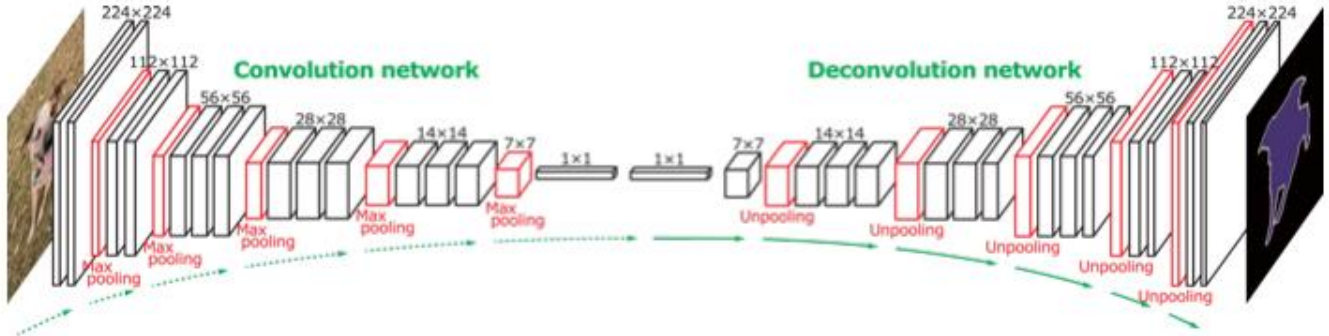
Neural networks



Perceptron



Gradient Decent



Convolutional Neural Networks

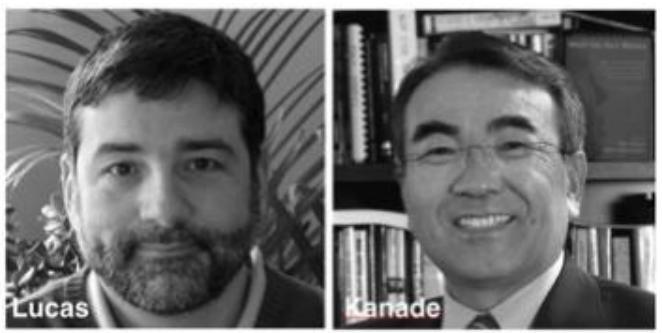
Optical flow and alignment

$$\begin{bmatrix} I_x(\boldsymbol{p}_1) & I_y(\boldsymbol{p}_1) \\ I_x(\boldsymbol{p}_2) & I_y(\boldsymbol{p}_2) \\ \vdots & \vdots \\ I_x(\boldsymbol{p}_{25}) & I_y(\boldsymbol{p}_{25}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(\boldsymbol{p}_1) \\ I_t(\boldsymbol{p}_2) \\ \vdots \\ I_t(\boldsymbol{p}_{25}) \end{bmatrix}$$

Constant Flow

$$\min_{\boldsymbol{u},\boldsymbol{v}} \sum_{ij} \left\{ E_d(i,j) + \lambda E_s(i,j) \right\}$$

Horn-Schunck

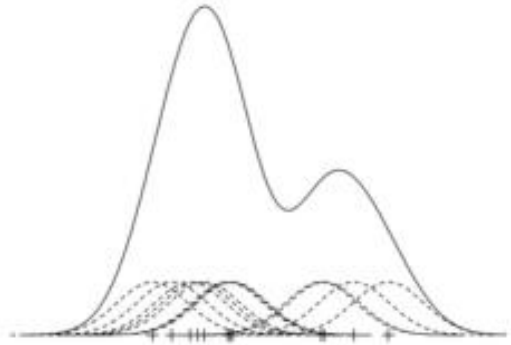


Lucas-Kanade
(Forward additive)

Baker-Matthews
(Inverse Compositional)

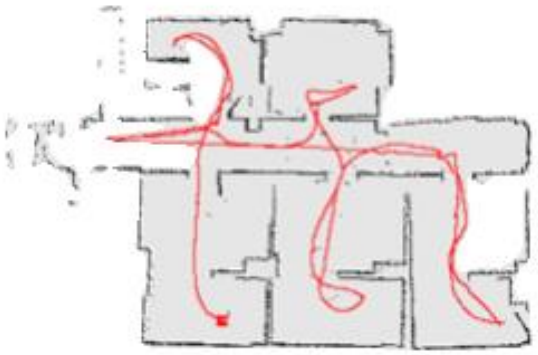
Tracking in videos

KLT



Mean shift

Kalman Filtering



SLAM

Things you should know how to do

1. Detect lines (circles, shapes) in an image.
2. Perform automatic image warping and basic AR.
3. Reconstruct 3D scene structure from two images.
4. Do photometric stereo and render simple images.
5. Recognize objects using a bag-of-words model.
6. Recognize objects using deep CNNs.
7. Track objects in video.

Questions?

Do you plan on taking any other vision courses?

Which part of the class did you like the most?

Which part of the class did you like the least?

Any topics you wanted to learn more about?

Any topics you wanted to learn less about?

Would the class work better if we did learning first?

Which was your favorite homework?

Which was your least favorite homework?

How does homework difficulty compare to other classes?

Would it be better if homeworks were in Python?