ECE276A: Sensing & Estimation in Robotics
Lecture 1: Color Vision

Lecturer:
Nikolay Atanasov: natanasov@ucsd.edu

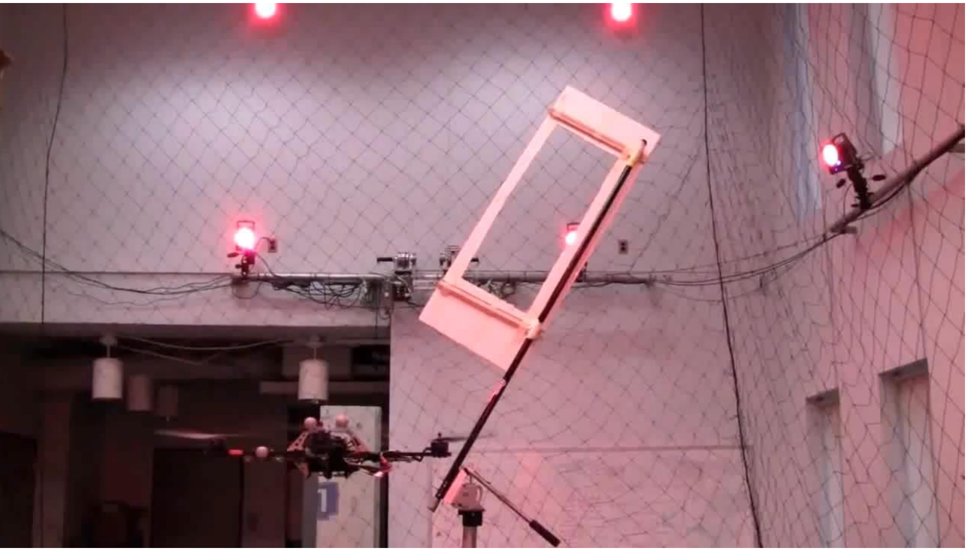
Teaching Assistants:
Siwei Guo: s9guo@eng.ucsd.edu
Anwesan Pal: a2pal@eng.ucsd.edu

UC San Diego
JACOBS SCHOOL OF ENGINEERING
Electrical and Computer Engineering

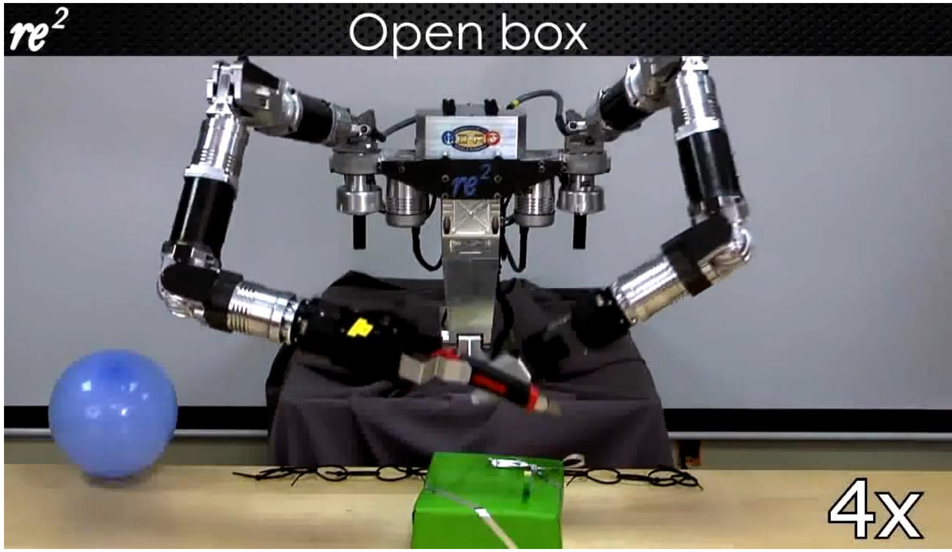
Progress in robot control

JPL-Caltech, DARPA Robotics Challenge, 2015

Boston Dynamics

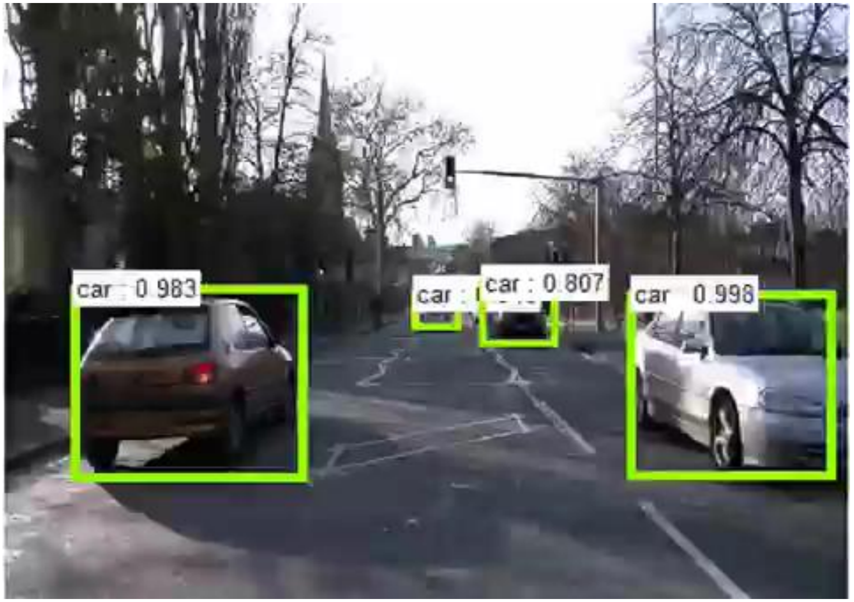


Mellinger, Michael, Kumar, IJRR’12



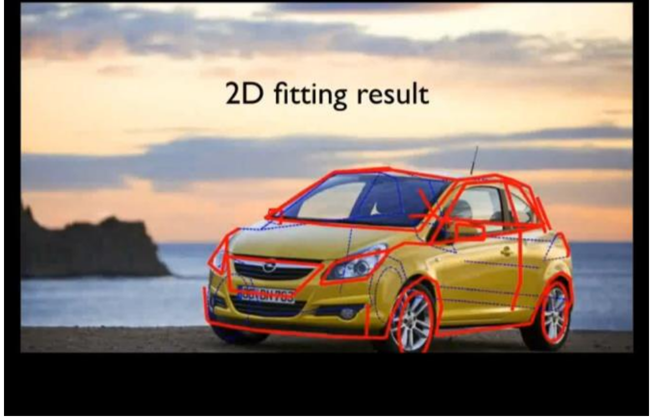
RE2 Robotics, Inc.

Progress in robot perception

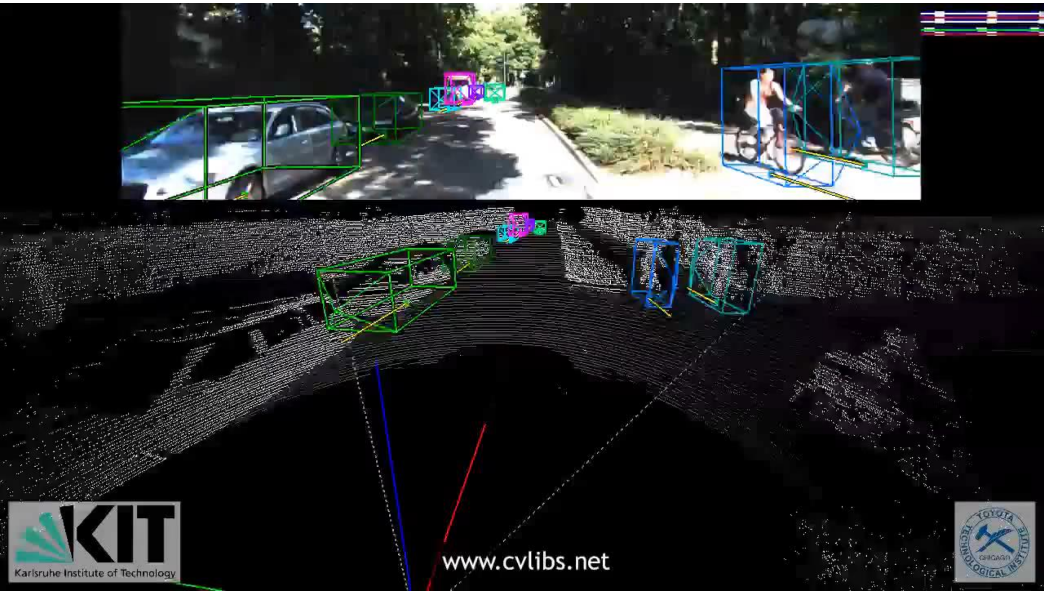


Ren, He, Girshick, Sun, NIPS'15

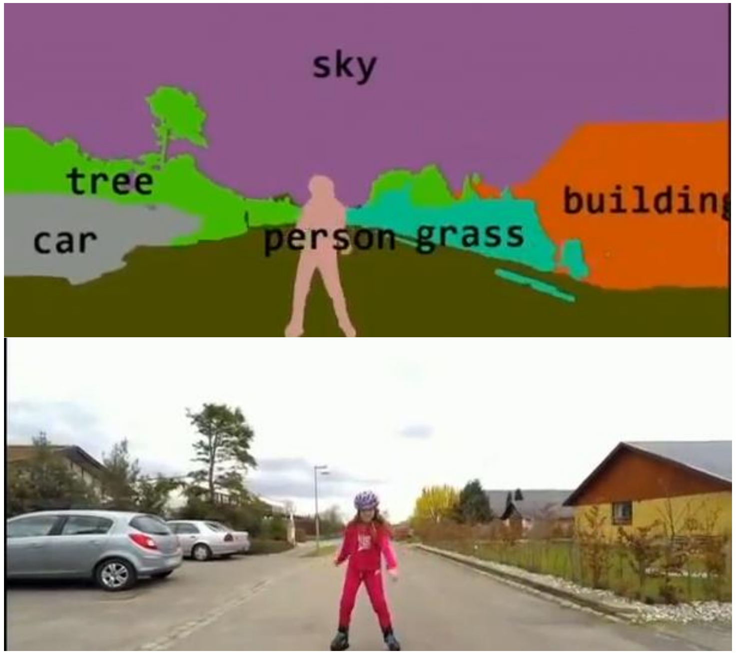
Zhu, Zhou, Daniilidis, ICCV'15



Newcombe, Fox, Seitz, CVPR'15



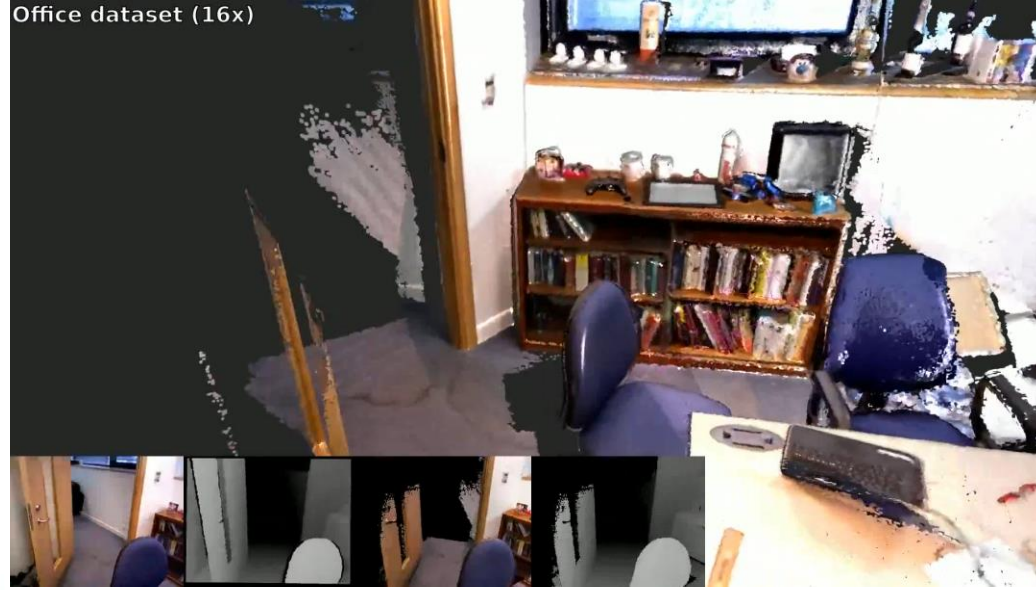
Geiger, Lenz, Urtasun, CVPR'12



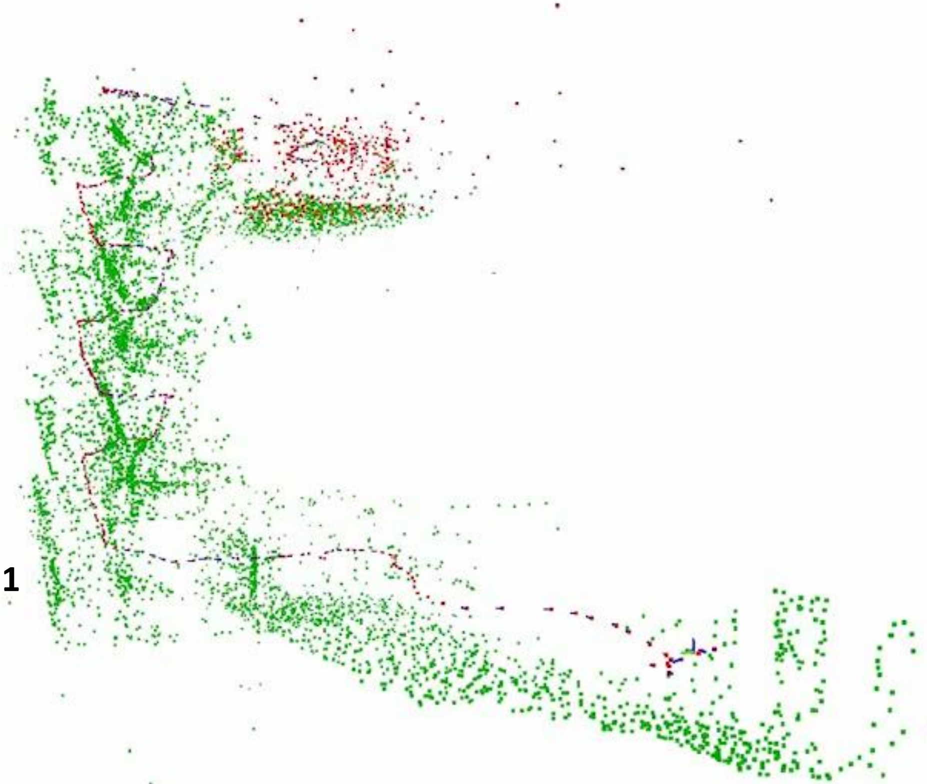
Microsoft Ignite, 2015

Localization & Mapping

Goal: determine the robot pose over time and build a map of the environment

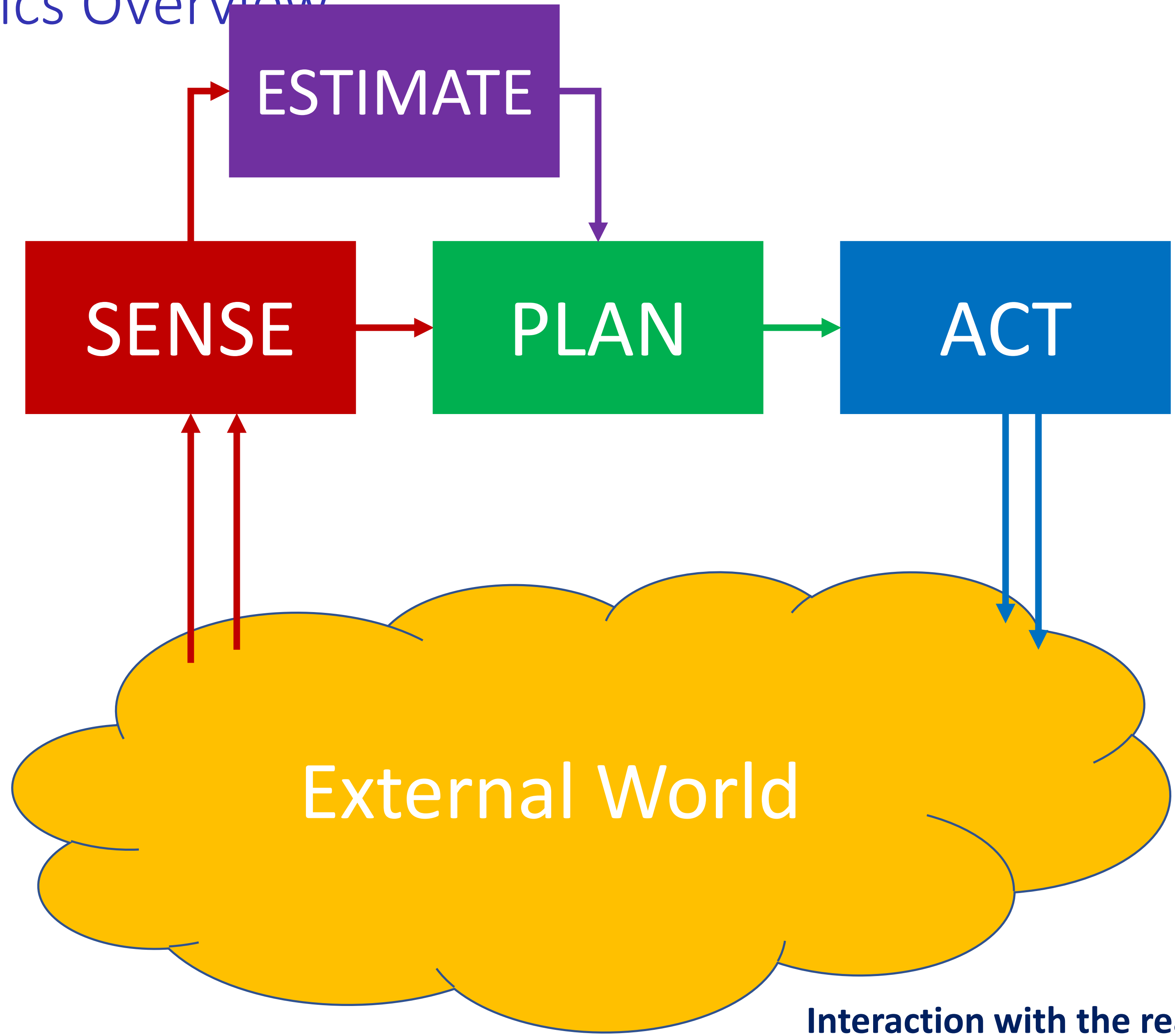


Whelan, Leutenegger, Salas-Moreno, Glocker, Davison, RSS'15



[1] Forster, Carlone, Dellaert, Scaramuzza, RSS'15
[2] Kummerle, Grisetti, Strasdat, Konolige, Burgard, ICRA'11
[3] Kaess, Ranganathan, Dellaert, T-RO'08
[4] Mourikis, Roumeliotis, ICRA'07
[5] Google Project Tango

Robotics Overview



Interaction with the real world introduces uncertainty!

Robotics Overview

- **Common Sensors**:
  - Images from cameras
  - Sounds from microphones
  - Distances from IR, sonar, laser range finders
  - Tactile bump switches
  - Magnetic sensors
  - Acceleration and angular velocity from inertial measurement units

- **Common Actuators**:
  - Joint angles for legged robots and articulated robot arms
  - Pan-tilt heads
  - Steering, throttle for wheeled robots
  - Thrust for quadrotors

Robotics Overview

- The field of robotics is an amalgam of several research areas:
  - **Computer vision & signal processing:** algorithms to deal with real world signals in real time (e.g., filter sound signals, convolve images with edge detectors, recognize objects)
  - **Machine learning**: algorithms to improve performance based on previous results and data (supervised, unsupervised, and reinforcement learning)
  - **Control theory**: algorithms to estimate robot and world states and plan and execute robot actions
  - **Optimization**: algorithms to choose the best robot behavior according to a suitable criterion from a set of available alternatives
- The key to robotics is the ability to deal with uncertainty (**Probability theory is important too!**)
  - Sensor noise & actuator slippage
  - Environment changes (outdoor sun, moving to different rooms, people)
  - Real-time operation

Main themes

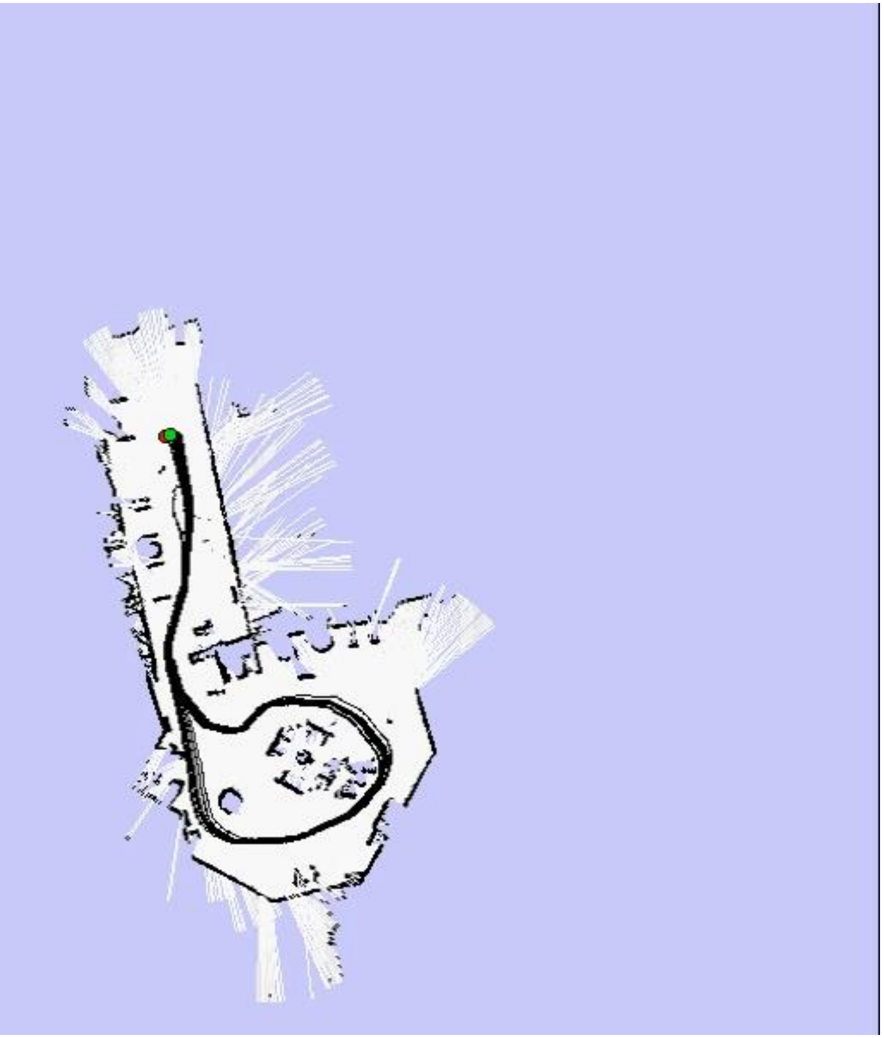
- **Noise**: how to model uncertainty using probability distributions
- **Perception:** how to recognize objects and geometry in the environment
- **Estimation**: how to estimate robot and environment state variables given uncertain measurements
- **Planning/Sequential decision making**: how to choose the most appropriate action at each time
- **Control/Dynamics**: how to control forces that act on the robot and the resulting acceleration; how to take world changes in time into account
- **Learning**: how to incorporate prior experience to improve robot performance

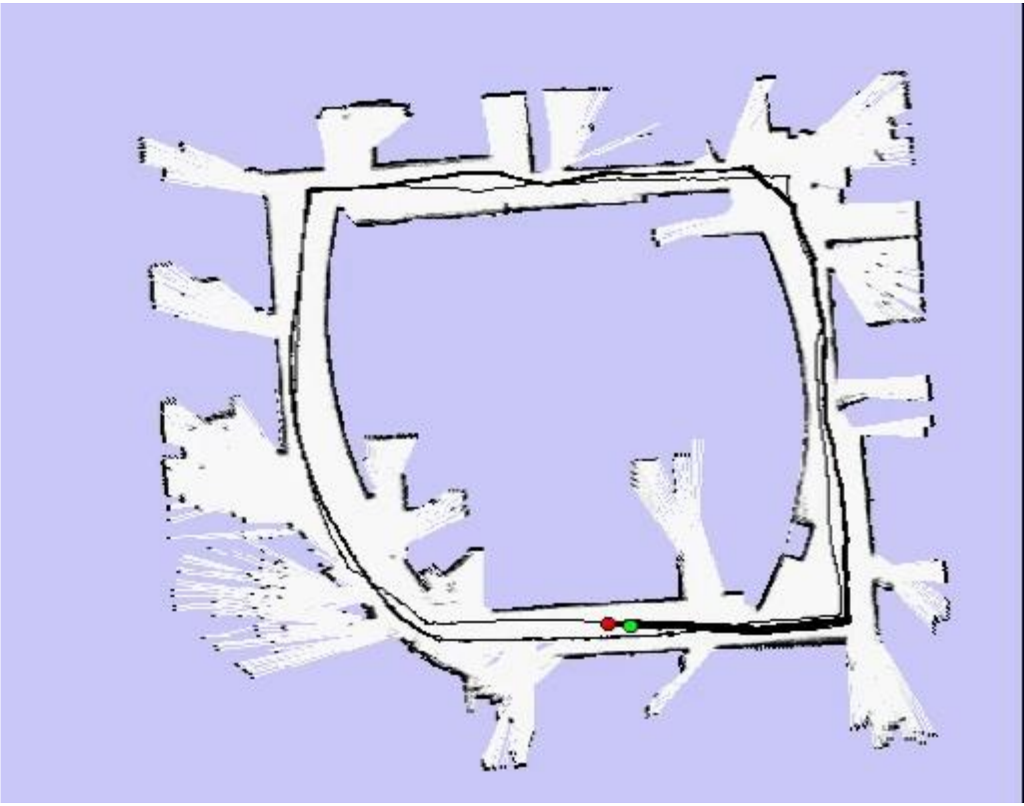
A few robotic success stories…

and connections with material covered in this course

Mapping

[Haehnel and Burgard]





FastSLAM: particle filter + occupancy grid mapping

Driverless Cars

- **Ernst Dickmanns / Mercedes Benz (1995):**
  - 1758 km: Paris highway and Munich → Odense
  - Longest autonomous stretch: 158 km
  - Lane changes up to 140 km/h

Kalman filtering, LQR, mapping, terrain & object recognition

D


Versuchsfahrzeug für
autonome Mobilität
und RechnerSehen
VaMoRs

ion

Driverless Cars

- **Ernst Dickmanns / Mercedes Benz (1995):**
  - 1758 km: Paris highway and Munich → Odense
  - Longest autonomous stretch: 158 km
  - Lane changes up to 140 km/h
- **DARPA Grand Challenge**: first long-distance driverless car competition
  - 2004: CMU vehicle drove 7.36 out of 150 miles
  - 2005: 5 teams finished, Stanford team won

Kalman filtering, LQR, mapping, terrain & object recognition

Driverless Cars

Kalman filtering, LQR, mapping, terrain & object recognition

Driverless Cars

- **Ernst Dickmanns / Mercedes Benz (1995):**
  - 1758 km: Paris highway and Munich → Odense
  - Longest autonomous stretch: 158 km
  - Lane changes up to 140 km/h
- **DARPA Grand Challenge**: first long-distance driverless car competition
  - 2004: CMU vehicle drove 7.36 out of 150 miles
  - 2005: 5 teams finished, Stanford team won nova-race
- **DARPA Urban Challenge (2007)**
  - Urban environment: other vehicles present
  - 6 teams finished (CMU won)

Kalman filtering, LQR, mapping, terrain & object recognition

Driverless Cars

- **Ernst Dickmanns / Mercedes Benz (1995):**
  - 1758 km: Paris highway and Munich → Odense
  - Longest autonomous stretch: 158 km
  - Lane changes up to 140 km/h
- **DARPA Grand Challenge**: first long-distance driverless car competition
  - 2004: CMU vehicle drove 7.36 out of 150 miles
  - 2005: 5 teams finished, Stanford team won nova-race
- **DARPA Urban Challenge (2007)**
  - Urban environment: other vehicles present
  - 6 teams finished (CMU won)
- **Google/Waymo Self-Driving Car**
  - 2010: Mountain View ➔ Santa Monica; >200,000 miles; Lombard, Golden Gate, Tahoe, Pacific Coast Highway
  - by Oct 2016: 2M miles with only minor accidents

Kalman filtering, LQR, mapping, terrain & object recognition

Driverless Cars

Kalman filtering, LQR, mapping, terrain & object recognition

Autonomous Helicopter Flight

[Abbeel, Coates & Ng]

Kalman filtering, model-predictive control, LQR, system ID, trajectory learning

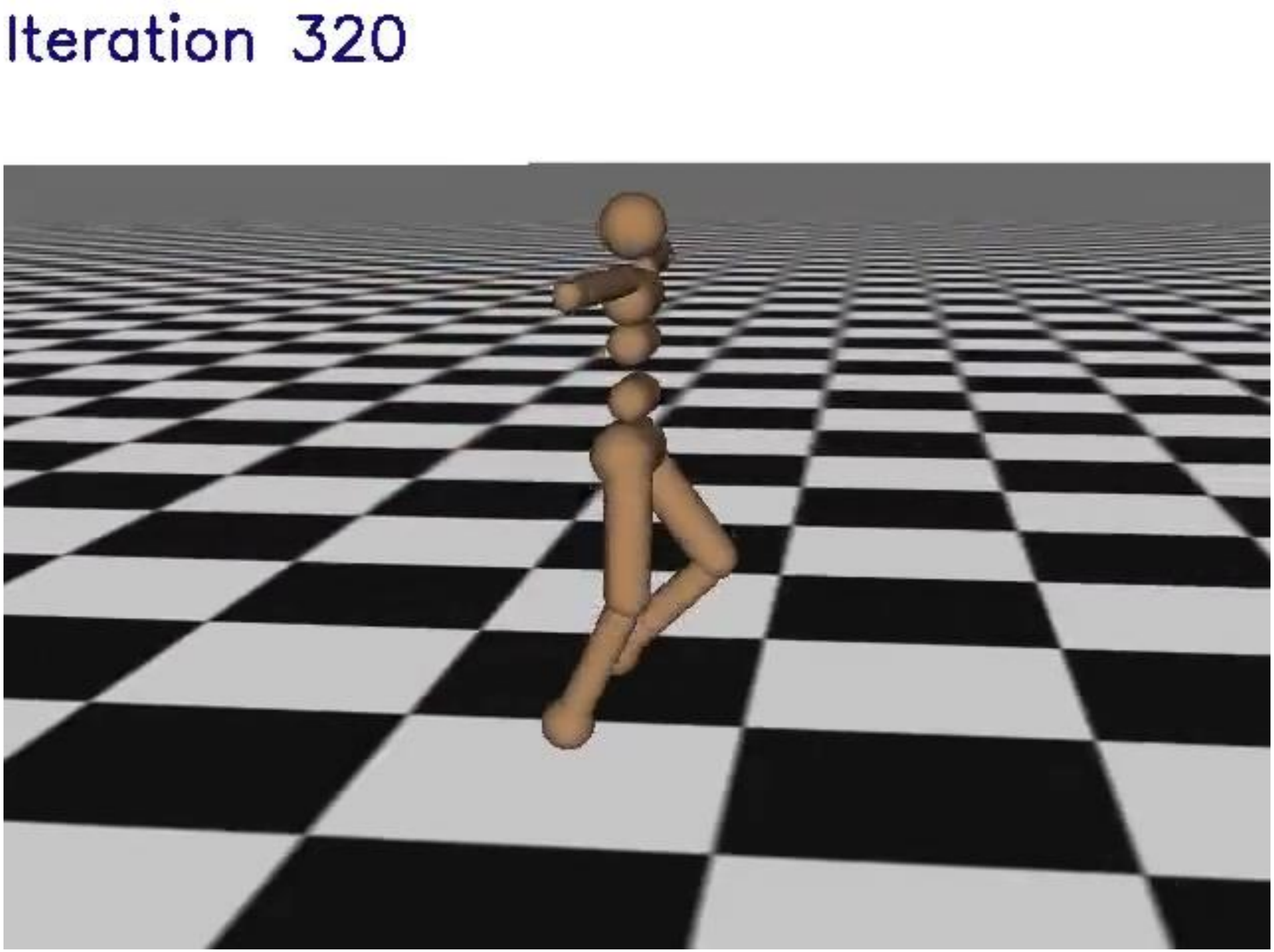
Four-legged Locomotion

[Kolter, Abbeel & Ng]

value iteration, receding horizon control, motion planning, inverse reinforcement learning

Learning Locomotion

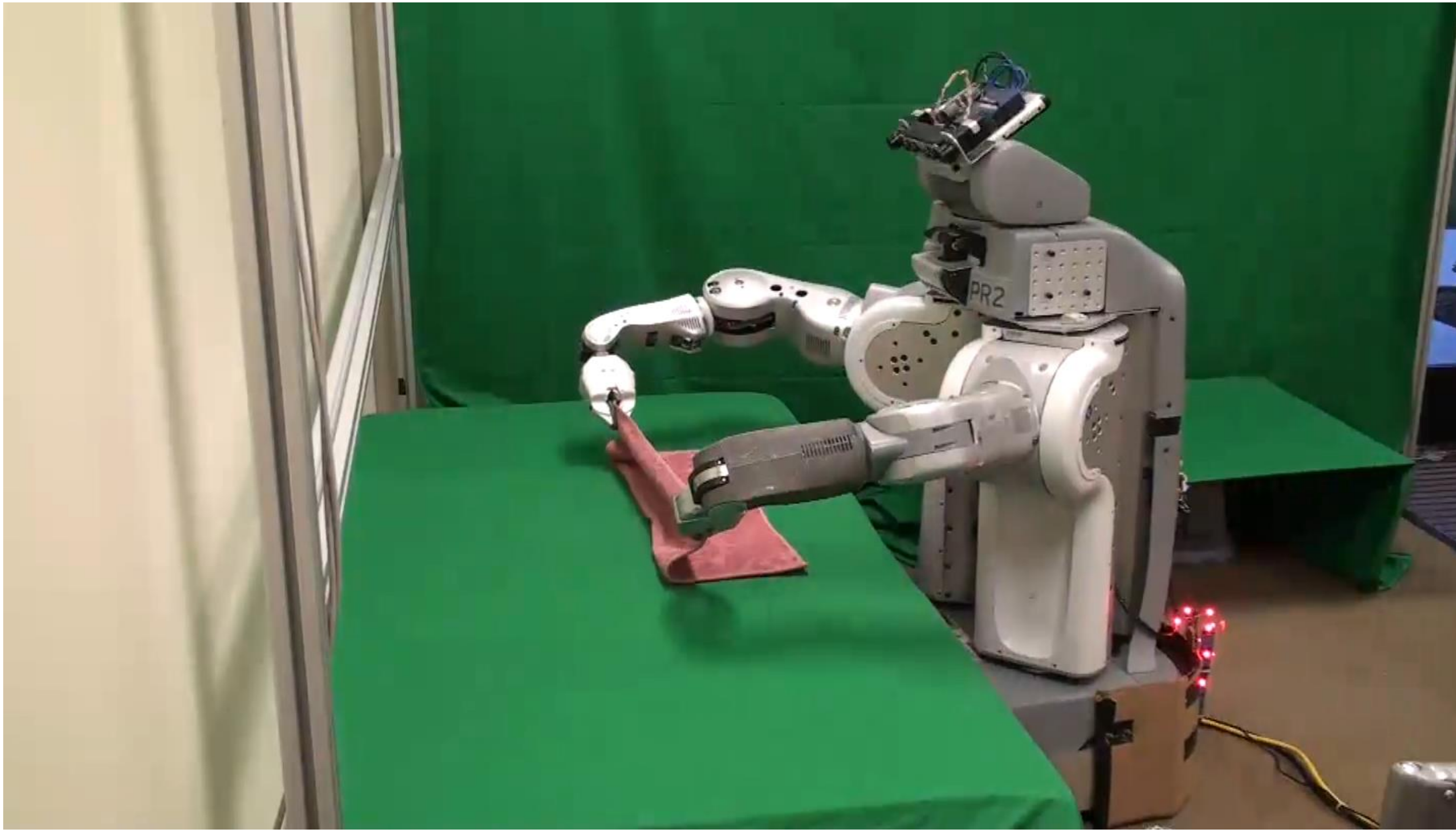
[Schulman, Abbeel, et al.]



policy gradients, value function approximation

Manipulation

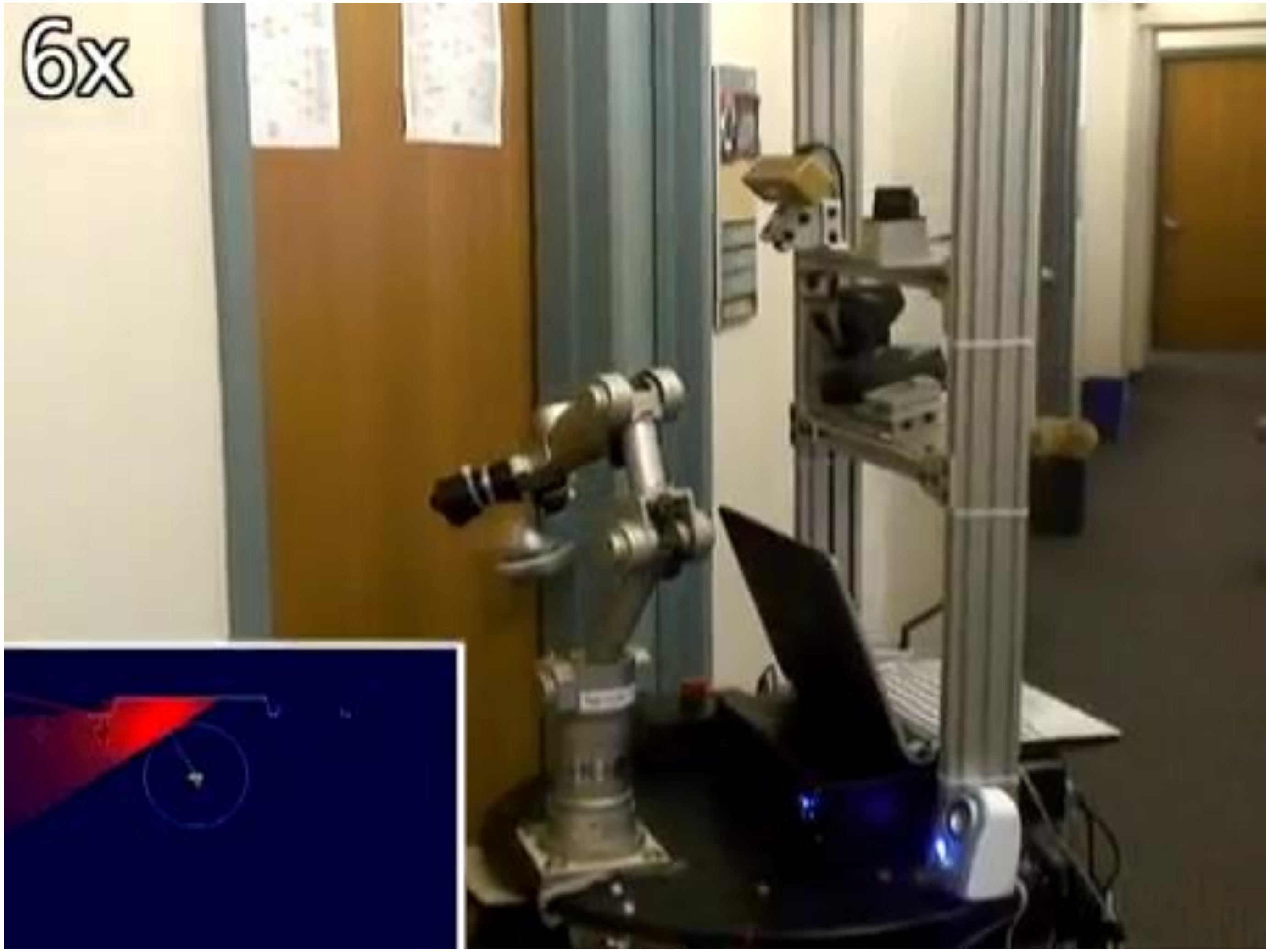
[Maitin-Shepard et al., 2010]



localization, motion planning for navigation and grasping, grasp point selection, visual recognition

The household robot

[Quigley, Gould, Saxena, Ng, et al.]



SLAM, localization, motion planning for navigation and grasping, grasp point selection, visual category recognition (speech recognition and synthesis)

ECE 276A: Sensing & Estimation in Robotics

- The class will cover theoretical topics in:
  - **Sensing**: rigid body motion, projective geometry, features, optical flow, object recognition
  - **Estimation**: regression, maximum likelihood estimation, classification, probabilistic models, filtering, mapping, hidden Markov models
- References (**not required!**):
  - An Invitation to 3-D Vision: Ma, Kosecka, Soatto & Sastry
  - Probabilistic Robotics: Thrun, Burgard & Fox
  - Bayesian Filtering and Smoothing: Sarkka
  - Nonlinear Gaussian Filtering: Theory, Algorithms, and Applications: Huber
- Course website: https://natanaso.github.io/ece276a
- Includes links to: (**SIGN UP!**)
  - Discussion: **Piazza**
  - Homework + Report Submission: **GradeScope**
  - Project Submission: **TritonEd**
  - Grades: **GradeScope**
  - **TA session:** once per week on Thursday or Friday - **TBD**

ECE 276A: Sensing & Estimation in Robotics

- Four assignments (roughly 25% each, detailed rubric online)
  - Project 1: Color Segmentation
  - Project 2: Orientation Tracking
  - Project 3: SLAM
  - Project 4: Gesture Recognition

- Each assignment includes:
  - theoretical homework
  - programming assignment in **python**
  - project report

- Letter grades will be assigned based on the class performance, i.e., there will be a “curve”.

- A test set will be released for each project a few days before the deadline. Your report should include results on **both** the test set and the training set.

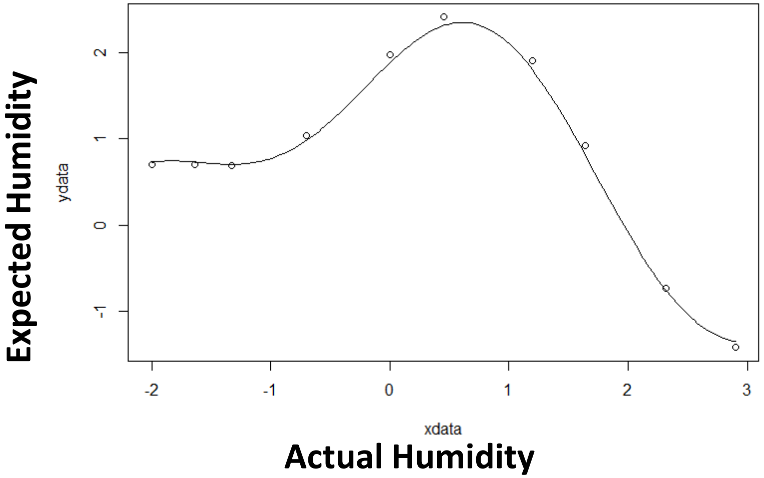
Report Structure

1. Introduction
   It is important to monitor the humidity of plants and choose optimal watering times. In this paper, we present an approach to select the best watering time in the week from given historical humidity data.

2. Problem Formulation
   Let $f: \mathbb{R} \to \mathbb{R}$ *be the average historical weakly humidity.*
   Problem*: Find a watering time* $t^* \in \mathbb{R}$ *such that* $t^* = \underset{t}{\operatorname{argmin}}\, f(t)$

3. Technical Approach
   The minimum of a function appears at one of its critical points $\{s \in \mathbb{R} \mid f'(s) = 0\}$. We find all the roots of f' and select the smallest one as the optimal watering time.

4. Results and Discussion
   The method performs well as shown in Fig. 1. The performance could be improved if real-time humidity measurements are used to update f.

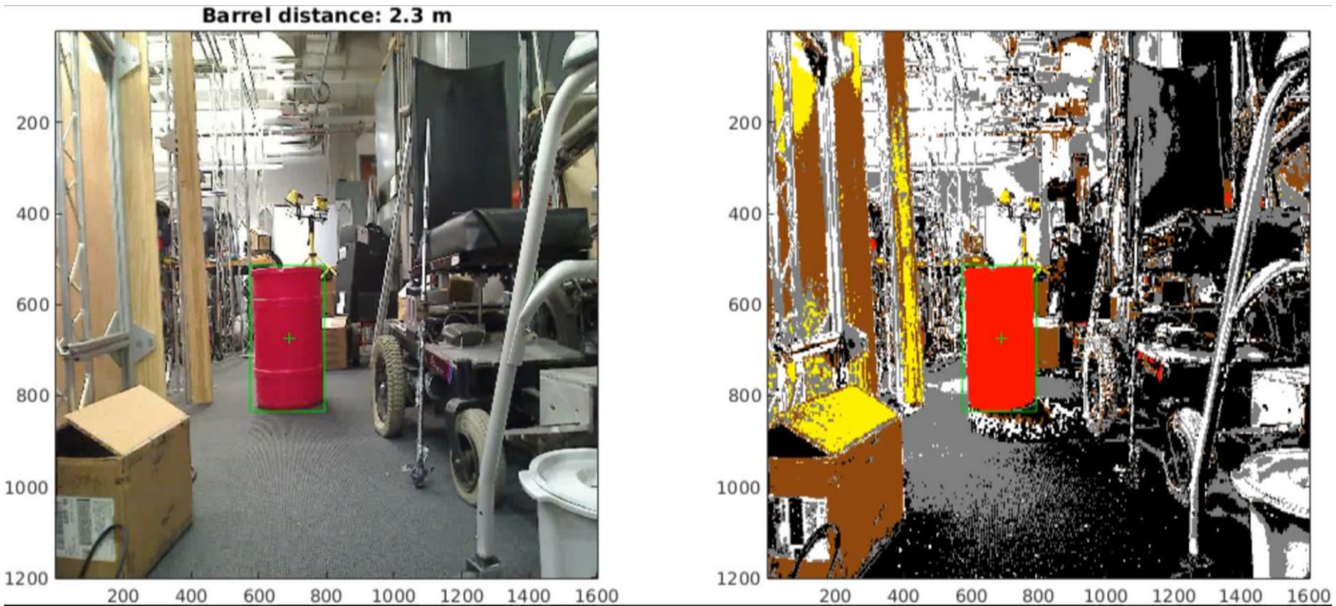


Syllabus Snapshot

Date	Lecture	Materials	Assignment out/due
Oct 02	Introduction, Color Vision, Matrix Calculus	System-discretization, Matrix-functions	
Oct 04	Probability Theory		P1
Oct 09	Supervised Learning		
Oct 11	Expectation Maximization	Tomasi-EM	
Oct 16	Rigid Body Motion	RigidBodyMotion	
Oct 18	Bayes Filter, Kalman Filter		P2
Oct 23	EKF, UKF	Kraft-UKF	P1
Oct 25	Projective Geometry, Panorama		
Oct 30	Sensor Models I		
Nov 01	Sensor Models II		
Nov 06	Gaussian Mixture and Particle Filter, Monte Carlo Sampling		
Nov 08	Markov Localization		P2, P3
Nov 13	Occupancy Grids, SLAM	Thrun-SLAM	
Nov 15	Kalman Smoother, Factor Graphs		
Nov 20	Robust Estimation: Hough, RANSAC, IRLS, Kabsch, ICP		
Nov 22	Visual Features, Optical Flow		
Nov 27	TBD		
Nov 29	Hidden Markov Models, Forward-Backward Procedure	Rabiner-HMM	P3, P4
Dec 04	Viterbi Decoding, Baum-Welch Algorithm		
Dec 06	TBD		
Dec 13			P4

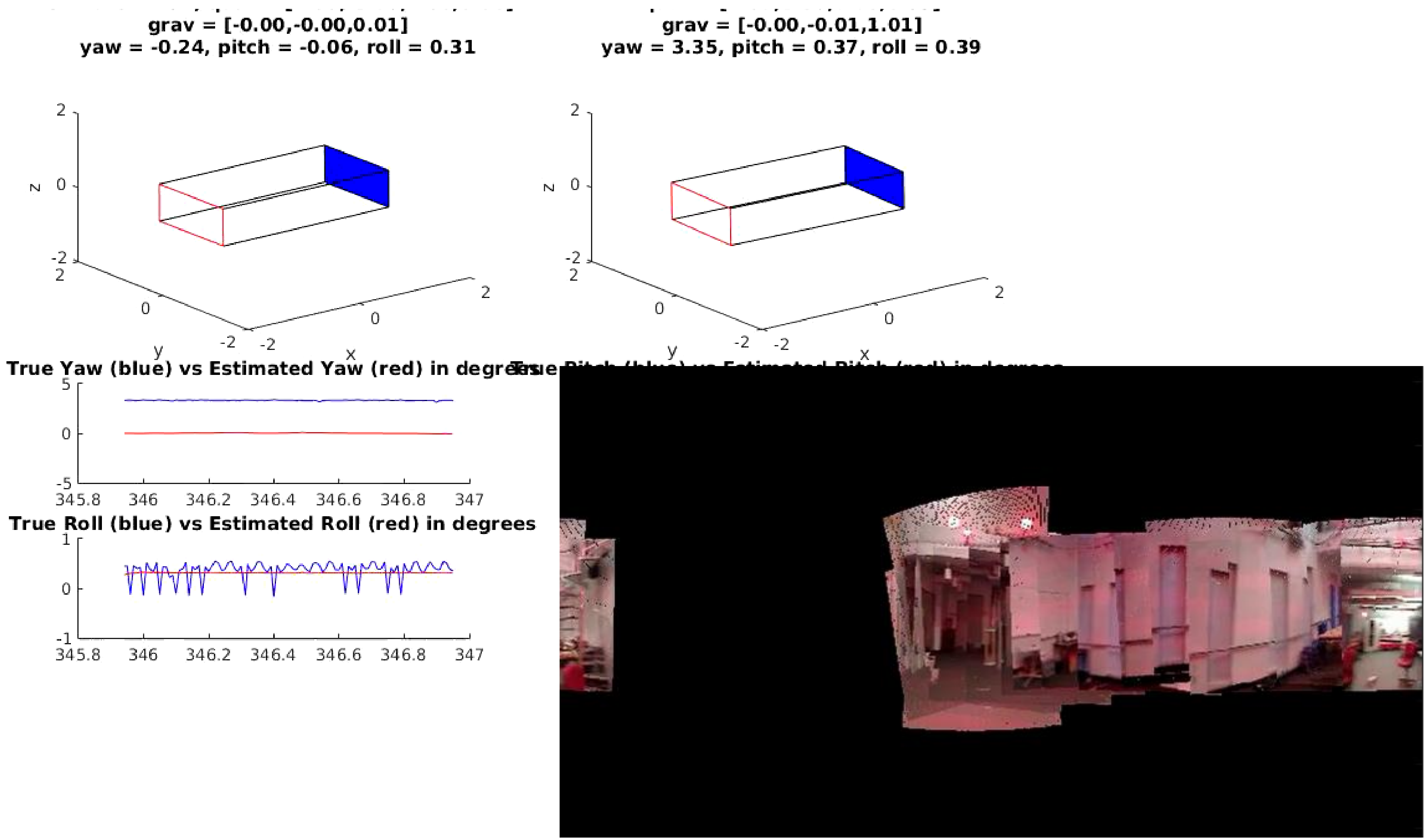
Color Segmentation

- train a Gaussian mixture color model to detect a red barrel in images



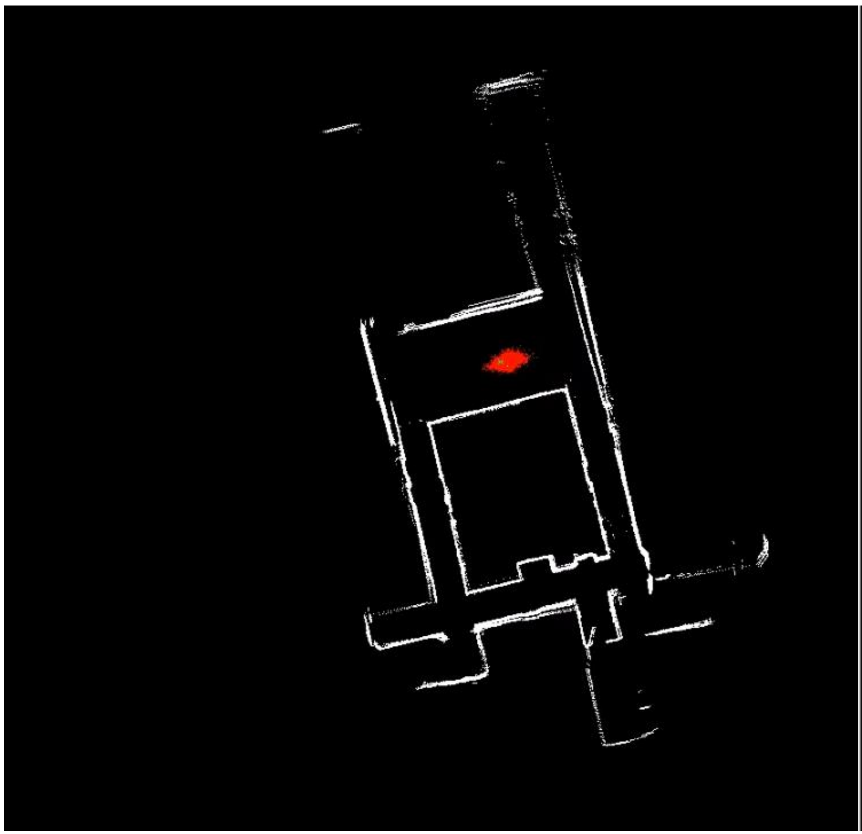
Orientation Tracking

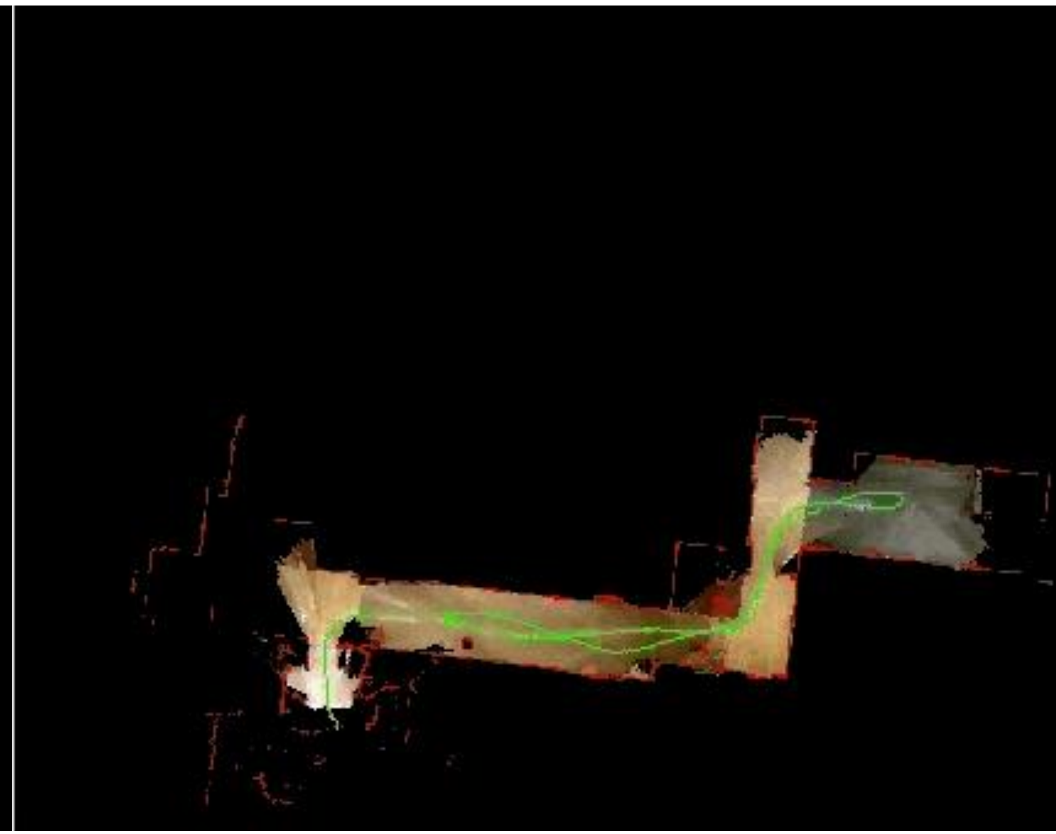
- use a Kalman filter to track the 3-D orientation of a rotating body using IMU measurements and construct a panorama using RGB images



SLAM

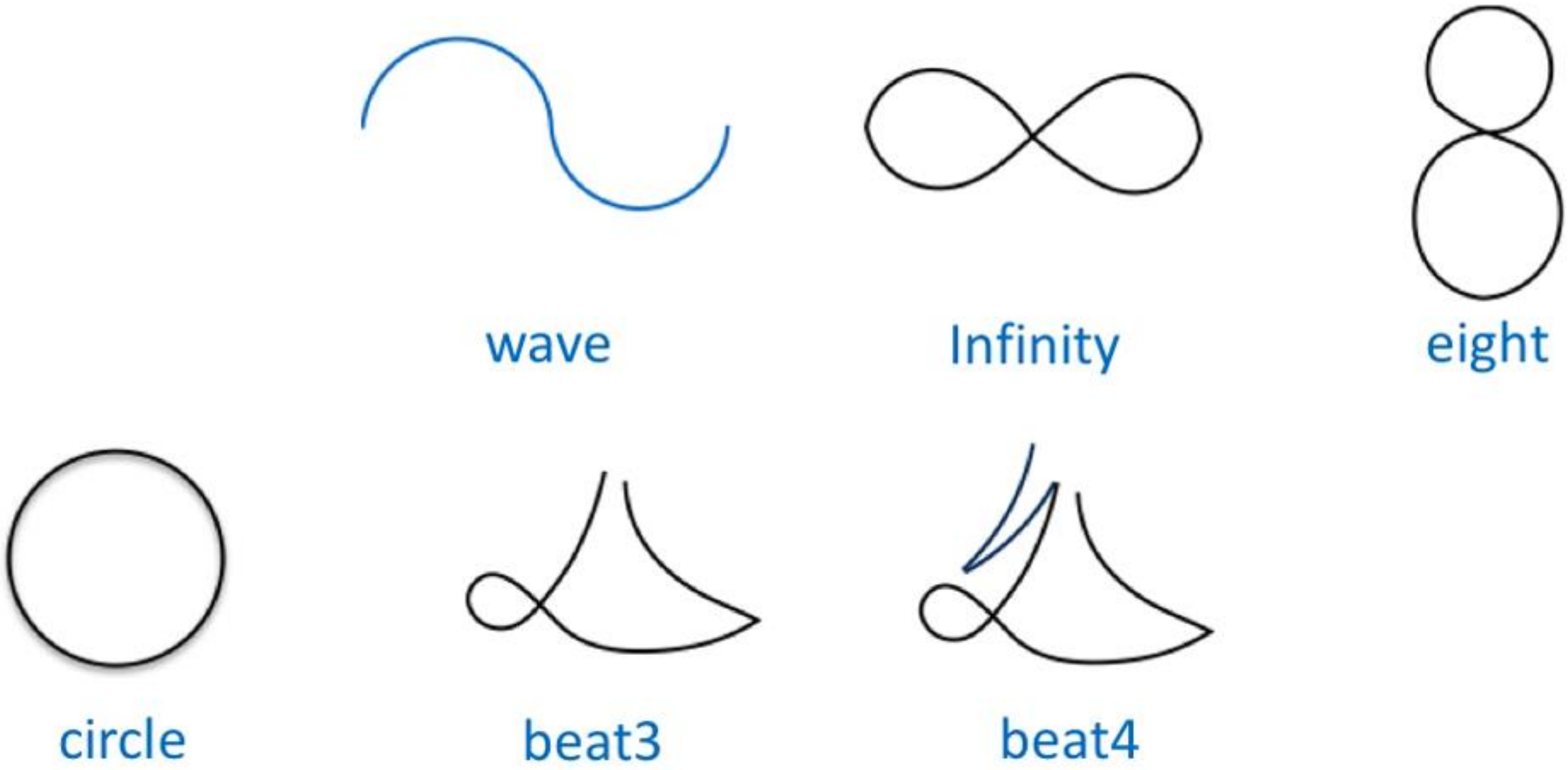
- implement robot localization & mapping using odometry, IMU, laser, RGBD measurements from a humanoid robot





Gesture Recognition

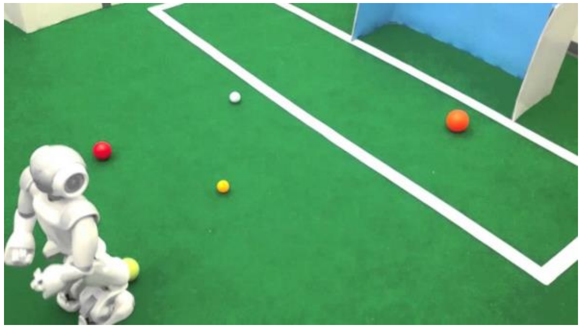
- implement a Hidden Markov Model to predict different hand gestures from raw IMU data



Vision

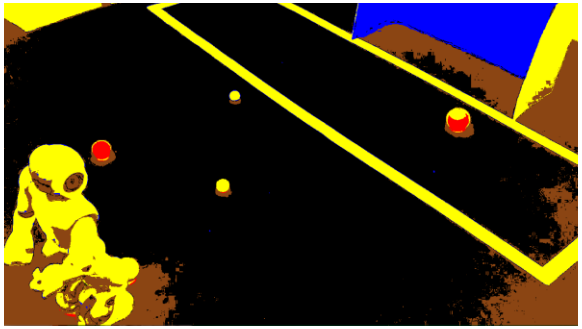
- The process of extracting information from an image
- Goal: identify objects and their relative locations

Robot Soccer Example: real-time robot vision system



RGB color image at 30 fps from camera

↓ Color segmentation

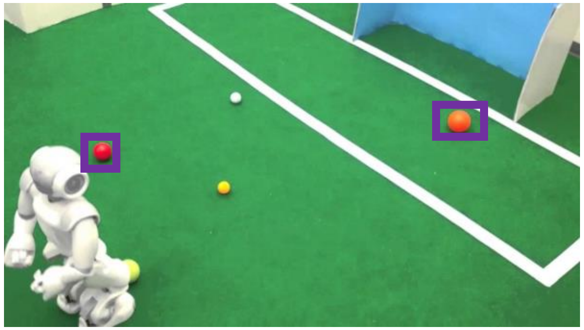


Each pixel is labelled by symbolic colors

↓ Run length encoding

More efficient computational data structure

↓ Union-find algorithm for merging of run-lengths

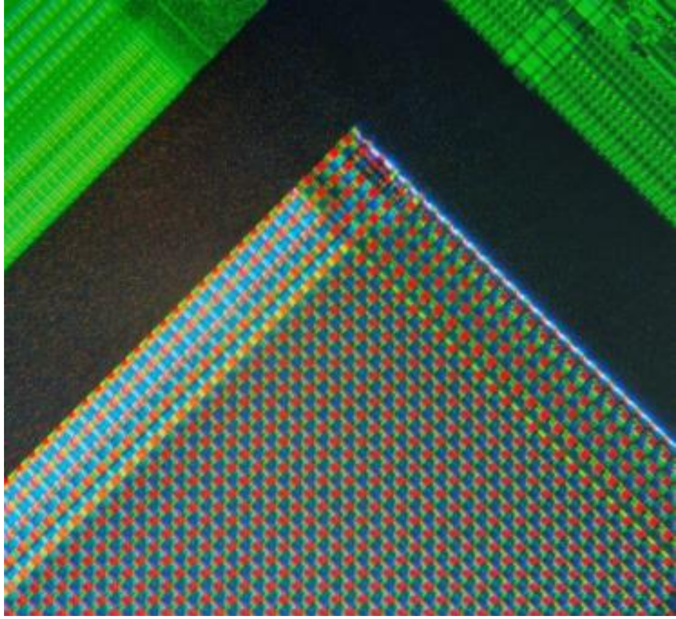


Connected components or superpixels/regions/blobs

↓ Use centroid, bounding box, major/minor axis, etc. to determine ball vs square etc.

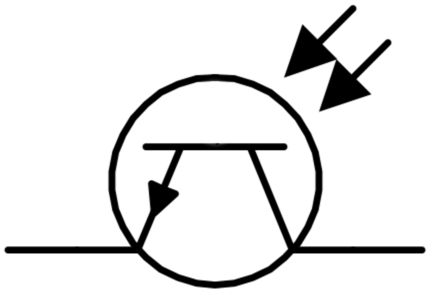
Classify objects based upon shape statistics

Color Imaging



- **Image sensor**: converts the variable attenuation of light/electromagnetic radiation into small bursts of current

- Analog imaging technology uses charge-coupled devices (CCD) or complementary metal-oxide semiconductors (CMOS)



- CCD/CMOS photosensor array:
  - A phototransistor converts light into current
  - Each transistor charges a capacitor to measure:
    #photons/sampling time
  - R,G,B filters are used to modify the absorption profiles of photons

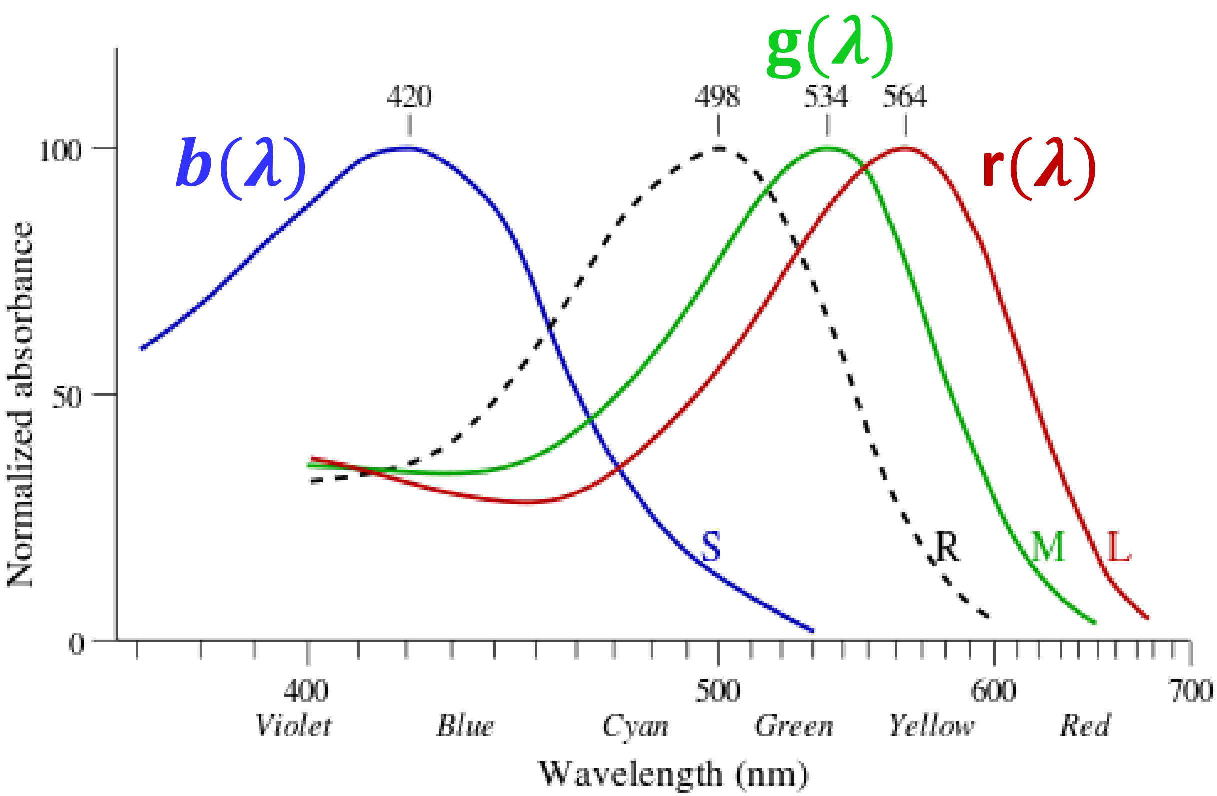
- The R,G,B transistor values are combined using an A/D converter to get pixel values:

  R = 127, G = 200, B = 103 (**24-bit color**)

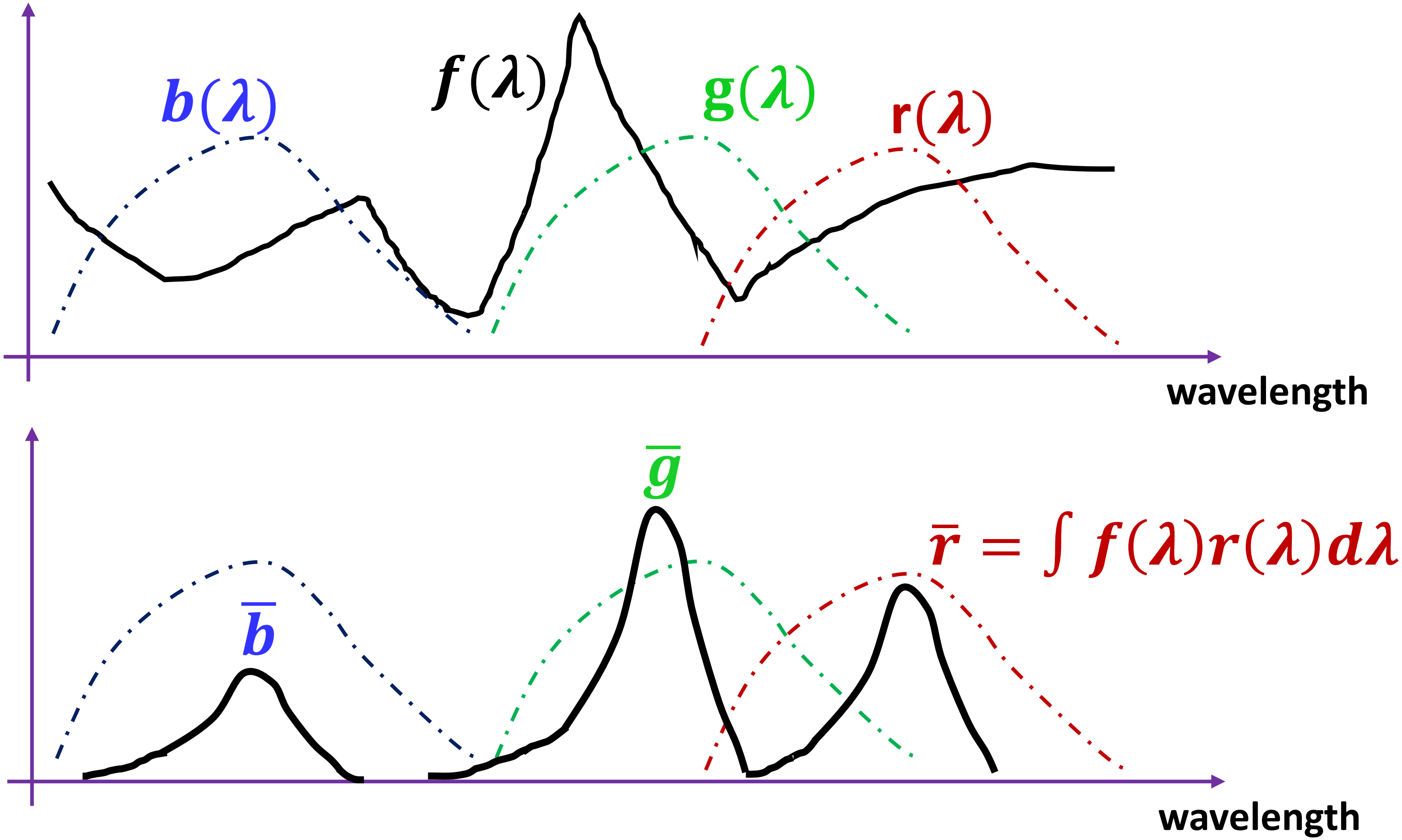
  8 bits (0-255)

Why RGB, why 3?

- Retina: 2 types of photoreceptors: **rod** & **cone cells** (S,M,L)
- Rod cells are relatively insensitive to wavelength but highly sensitive to intensity and thus are mostly saturated in their response during normal daylight conditions



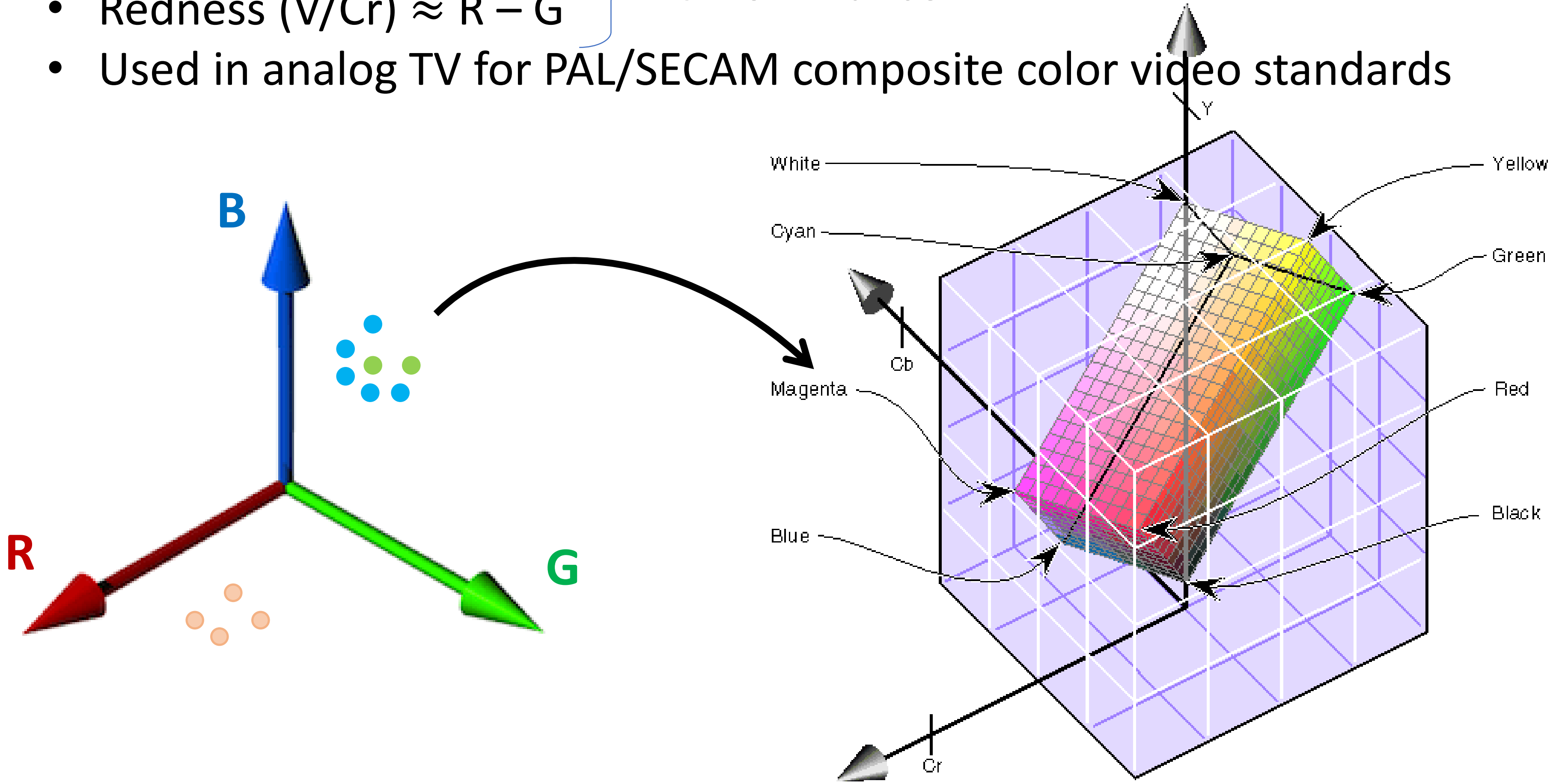
- Given an arbitrary light spectral distribution $f(\lambda)$, the cone cells act as filters that provide a **convolution-like signal** to the brain:



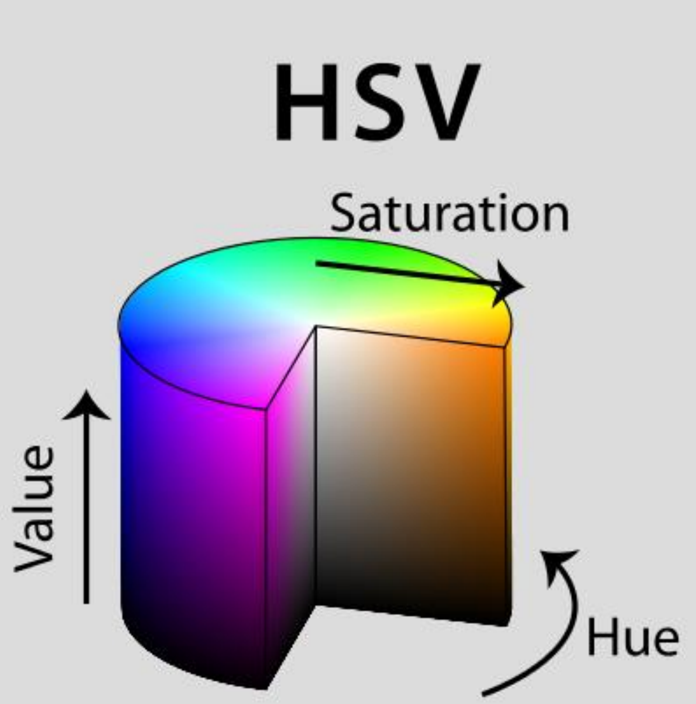
- Color blind people are deficient in 1 or more of these cones
- Other animals (e.g., fish) have more than 3 cones

Luma-Chroma Color Space

- **YUV (YCbCr)**: a linear transformation of RGB
  - Luminance/Brightness (Y) ~ (R+G+B)/3 } Gray-scale image
  - Blueness (U/Cb) ≈ B - G
  - Redness (V/Cr) ≈ R – G } Chrominance
  - Used in analog TV for PAL/SECAM composite color video standards



Other Color Spaces



- **HSV**: cylindrical coordinates of RGB points
  - **Hue (H)**: angular dimension (red ~ 0º, green ~ 120º, blue ~ 240º)
  - **Saturation (S)**: pure red has saturation 1, while tints have saturation < 1.
  - **Value/Brightness (V)**: achromatic/gray colors ranging from black (V = 0, bottom) to white (V=1, top)

- **LAB**: nonlinear transformation of RGB; device independent
  - Lightness (L): from black (L=0) to white (L=100)
  - Position between green and red/magenta (A)
  - Position between blue and yellow (B)

Image Formation

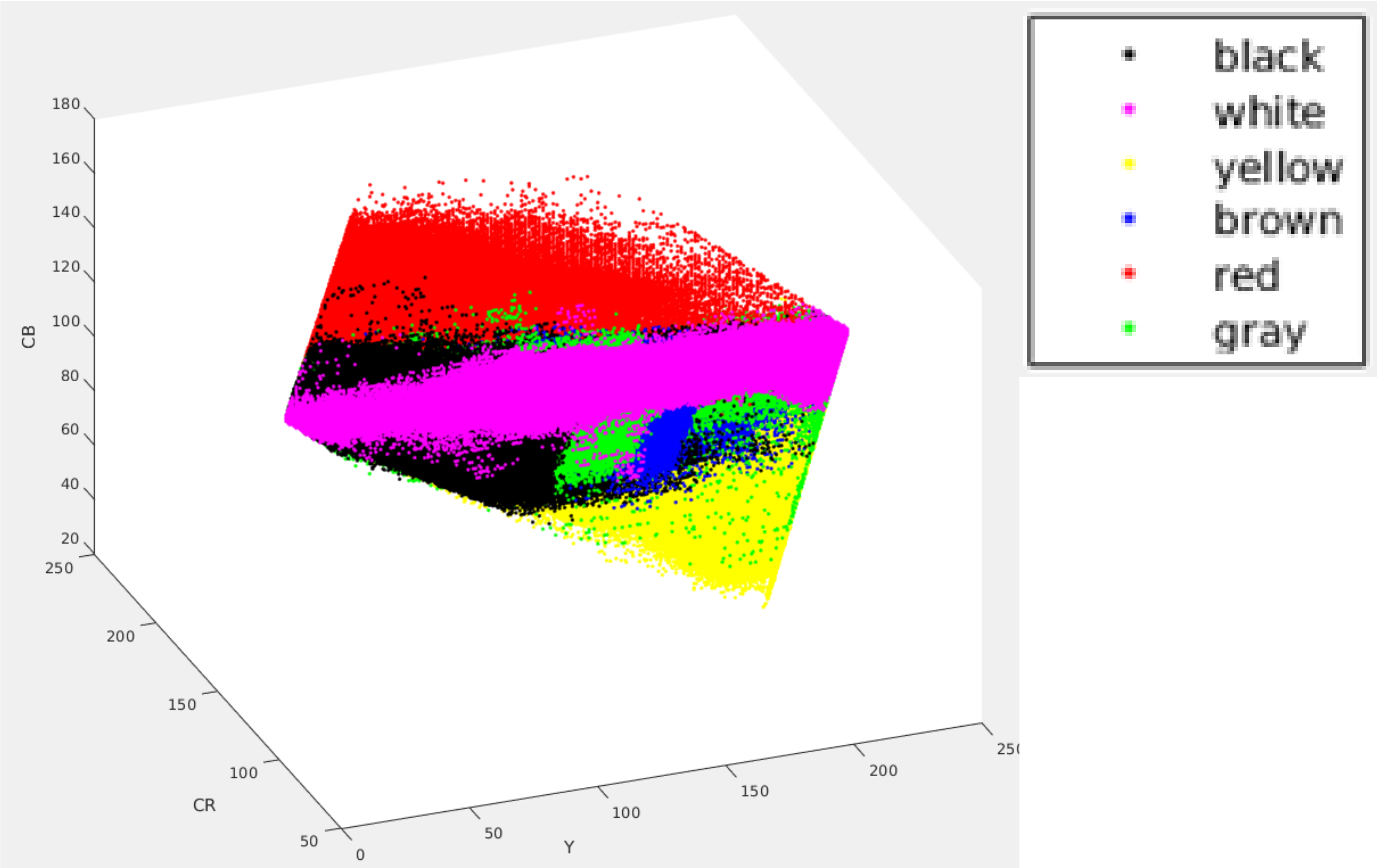
- Pixel values depend on:
  - Scene **geometry**
  - Scene **photometry** (illumination and reflective properties)
  - Scene **dynamics** (moving objects)

- Using camera images to infer a representation of the world is challenging because the shape, material properties, and motion of the observed scene are in general unknown

- **Color segmentation**: aims to segment the color space into a set of discrete volumes
  - Each pixel is a **3-D vector**: $x = (Y, Cb, Cr)$
  - Discrete color labels: $w \in \{1, \ldots, N\}$

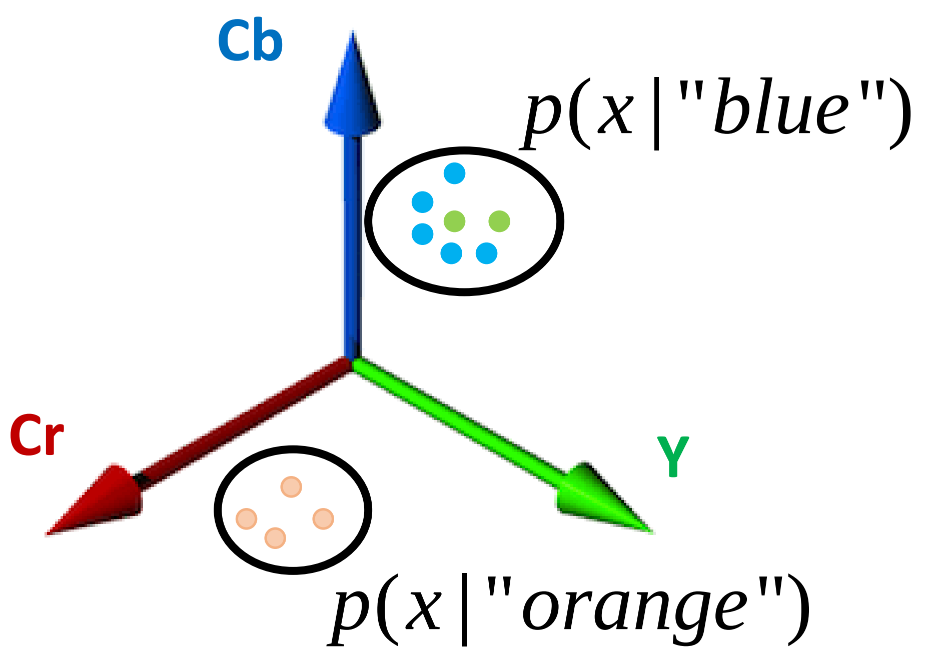
YCbCr Image Space

- Each pixel is a **3-D vector**: $x = (Y, Cb, Cr)$
- Discrete color labels: $w \in \{1, \ldots, N\}$



Bayes decision theory

- Pixel values are noisy
- Learn a **probabilistic model** $\mathrm{p}(w \mid x)$ of the color classes w given color-space **training data** $\mathrm{D} = \{(x_i, w_i)\}$
- Define a color map that transforms a color-space input to a discrete color label: $x \rightarrow \arg\max_w p(w \mid x)$



Vectors

- A **vector** $x \in \mathbb{R}^d$ with d dimensions is a collection of scalars $x_i \in \mathbb{R}$ for $i = 1, \dots, d$ organized in a column:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix} \qquad x^T = [x_1 \quad x_2 \quad \cdots \quad x_d]$$

- **Linearly independent vectors**: a set of vectors $\{x_i \in \mathbb{R}^d, i = 1, \dots, n\}$ such that no nontrivial linear combination of them gives the zero vector:

$$\sum_{i=1}^{n} a_i x_i = 0 \quad \Longrightarrow \quad a_i = 0, \forall i$$

- A set of d linearly independent vectors $x_i \in \mathbb{R}^d$ forms a **basis** for the vector space of all $d \times 1$ vectors

- The set if all linear combinations of a specified set of vectors is a vector space called the **span** of the set of vectors.

Norm

- A **norm** on a vector space V over a subfield F is a function $\|\cdot\|: V \to \mathbb{R}$ such that for all $a \in F$ and all $x, y \in V$
  - $\|ax\| = |a|\|x\|$ (absolute homogeneity)
  - $\|x + y\| \leq \|x\| + \|y\|$ (triangle inequality)
  - $\|x\| \geq 0$ (non-negativity)
  - $\|x\| = 0$ iff $x = 0$ (definiteness)

- The **Euclidean norm** of a vector $x \in \mathbb{R}^d$ is $\|x\|_2 \coloneqq \sqrt{x^T x}$ and satisfies:
  - $\max_{1 \leq i \leq d} |x_i| \leq \|x\|_2 \leq \sqrt{d} \max_{1 \leq i \leq d} |x_i|$
  - $|x^T y| \leq \|x\|_2 \|y\|_2$ (Cauchy-Schwarz Inequality)

Matrices

- A **matrix** $A \in \mathbb{R}^{m \times n}$ is a rectangular array of scalars $x_{ij} \in \mathbb{R}$ for $i = 1, \dots, m$ and $j = 1, \dots, n$
- The entries of the **transpose** $A^T \in \mathbb{R}^{n \times m}$ of a matrix $A \in \mathbb{R}^{m \times n}$ are: $A_{ij}^T = A_{ji}$. The transpose satisfies: $(AB)^T = B^T A^T$
- The **trace** of square matrix $A \in \mathbb{R}^{n \times n}$ is the sum of its diagonal entries:

$$tr(A) := \sum_{i=1}^{n} A_{ii} \qquad tr(ABC) = tr(BCA) = tr(CAB)$$

- The **determinant** of square matrix $A \in \mathbb{R}^{n \times n}$ is:

$$det(A) := \sum_{j=1}^{n} A_{ij} C_{ij} \qquad det(AB) = \det(A)\det(B) = \det(BA)$$

where C_{ij} is the **cofactor** of the entry A_{ij} and is equal to $(-1)^{i+j}$ times the determinant of the $(n-1) \times (n-1)$ submatrix that results when the i^{th}-row and j^{th}-col of A are deleted. This recursive definition uses the fact that the determinant of a scalar is the scalar itself

Matrices

- The **adjugate** $adj(A)$ is the transpose of the matrix of cofactors of A
- The **inverse** A^{-1} of A exists iff $det(A) \neq 0$ and satisfies

$$A^{-1} = \frac{adj(A)}{\det(A)} \qquad (AB)^{-1} = B^{-1}A^{-1}$$

- If $A \in \mathbb{R}^{n\times n}$ and $p \in \mathbb{R}^n$ is a nonzero vector such that for $\lambda > 0$, $Ap = \lambda p$, then p is an **eigenvector** corresponding to the **eigenvalue** λ.
- A real matrix can have complex eigenvalues and eigenvectors, which appear in conjugate pairs. The n eigenvalues of $A \in \mathbb{R}^{n\times n}$ are precisely the n roots of the **characteristic polynomial** of A, given by $p(s) \coloneqq det(sI - A)$.
- The roots of a polynomial are continuous functions of its coefficients and hence the eigenvalues of a matrix are continuous functions of its entries.

$$tr(A) \coloneqq \sum_{i=1}^{n} \lambda_i \qquad det(A) \coloneqq \prod_{i=1}^{n} \lambda_i$$

Quadratic Forms

- The product $x^T Qx$ for $Q \in \mathbb{R}^{n \times n}$ and $x \in \mathbb{R}^n$ is called a **quadratic form** and without loss of generality Q can be assumed **symmetric,** $Q = Q^T$ because for all $x \in \mathbb{R}^n$:
$$\frac{1}{2} x^T (Q + Q^T) x = x^T Q x$$
- A symmetric matrix $Q \in \mathbb{R}^{n \times n}$ is **positive semidefinite** if $x^T Qx \geq 0$ for all $x \in \mathbb{R}^n$
- A symmetric matrix $Q \in \mathbb{R}^{n \times n}$ is **positive definite** if it is positive semidefinite and if $x^T Qx = 0$ implies $x = 0$
- All eigenvalues of a symmetric matrix are **real**. Hence, all eigenvalues of a positive semidefinite matrix are nonnegative and all eigenvalues of a positive definite matrix are positive.
- The **Schur complement** of block D of matrix $M = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$ is $S_D = A - BD^{-1}C$
- A symmetric matrix $M = \begin{bmatrix} A & B \\ B^T & D \end{bmatrix}$ is positive semi-definite **if and only if** both A and S_A are positive semi-definite (or both D and S_D are positive semi-definite).

Matrix Inversion Lemma

- The Woodbury matrix identity:

$$(A+BDC)^{-1} = A^{-1} - A^{-1}B\left(D^{-1}+CA^{-1}B\right)^{-1}CA^{-1}$$

- Block Matrix inversion:

Schur complement

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} I & 0 \\ D^{-1}C & I \end{bmatrix}^{-1} \begin{bmatrix} A-BD^{-1}C & 0 \\ 0 & D \end{bmatrix}^{-1} \begin{bmatrix} I & BD^{-1} \\ 0 & I \end{bmatrix}^{-1}$$

$$= \begin{bmatrix} I & 0 \\ -D^{-1}C & I \end{bmatrix} \begin{bmatrix} \left(A-BD^{-1}C\right)^{-1} & 0 \\ 0 & D^{-1} \end{bmatrix} \begin{bmatrix} I & -BD^{-1} \\ 0 & I \end{bmatrix}$$

$$= \begin{bmatrix} (A-BD^{-1}C)^{-1} & -(A-BD^{-1}C)^{-1}BD^{-1} \\ -D^{-1}C(A-BD^{-1}C)^{-1} & D^{-1}+D^{-1}C(A-BD^{-1}C)^{-1}BD^{-1} \end{bmatrix}$$

Matrix Inversion Lemma

- Square Completion:

$$\frac{1}{2}x^T Ax + b^T x + c = \frac{1}{2}\left(x+A^{-1}b\right)^T A\left(x+A^{-1}b\right) + c - \frac{1}{2}b^T A^{-1}b$$

Matrix Exponential

- The **matrix exponential** of $A \in \mathbb{R}^{n \times n}$ is defined by the power series

$$e^A = \sum_{k=0}^{\infty} \frac{1}{k!} A^k$$

- Properties:
  - $e^A e^{-A} = I, \qquad det(e^A) = e^{tr(A)}$
  - If $AB = BA$, then $e^A e^B = e^B e^A = e^{(A+B)}$
  - If B^{-1} exists, then $e^{BAB^{-1}} = Be^A B^{-1}$
  - $e^{A^T} = [e^A]^T$, which implies that:
    - If A is symmetric, then e^A is symmetric
    - If A is skew-symmetric, then e^A is orthogonal

- The matrix exponential appears in
  - solutions of linear equations of **ordinary differential equations**
  - the relationship between **rotation matrices** and **skew-symmetric matrices**

Linear ODE Solutions

- **Continuous-time linear time-varying system**:

$$\dot{x}(t) = A(t)x(t) + B(t)u(t), \;\; x(t_0) = x_0, \;\; t \geq t_0$$

- Solution: $$x(t) = \Phi(t, t_0)x_0 + \int_{t_0}^{t} \Phi(t, s)B(s)u(s)ds$$

- The **transition matrix** $\Phi(t, t_0)$ is given by the **Peano-Baker series**:

$$\Phi(t,\tau) = I + \int_{\tau}^{t} A(s_1)ds_1 + \int_{\tau}^{t}\int_{\tau}^{s_1} A(s_1)A(s_2)ds_2 ds_1 + \int_{\tau}^{t}\int_{\tau}^{s_1}\int_{\tau}^{s_2} A(s_1)A(s_2)A(s_3)ds_3 ds_2 ds_1 + \ldots$$

- If A is **time-invariant**, then $\Phi(t, \tau) = e^{A(t-\tau)}$

- **Discrete-time**: $$x_{t+1} = A_t x_t + B_t u_t, \;\; x_{t_0} = x_0, \;\; t \geq t_0$$

$$x_t = \Phi\left(t, t_0\right) x_0 + \sum_{j=t_0}^{t-1} \Phi\left(t, j+1\right) B_j u_j, \;\; t \geq t_0 + 1$$

$$\Phi\left(t, j\right) = \begin{cases} A_{t-1}A_{t-2}\cdots A_j, & t \geq j+1 \\ I, & t = j \end{cases}$$

Derivatives (numerator layout)

$$\frac{\partial y}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial y}{\partial x_1} & \cdots & \frac{\partial y}{\partial x_n} \end{bmatrix} = \left[\nabla_{\mathbf{x}} y\right]^T \qquad \text{(gradient transpose)}$$

$$\frac{\partial \mathbf{y}}{\partial x} = \begin{bmatrix} \frac{\partial y_1}{\partial x} \\ \vdots \\ \frac{\partial y_m}{\partial x} \end{bmatrix} \qquad \frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \cdots & \frac{\partial y_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_m}{\partial x_1} & \cdots & \frac{\partial y_m}{\partial x_n} \end{bmatrix} = D_{\mathbf{x}}\mathbf{y} \qquad \text{(Jacobian)}$$

$$\frac{\partial Y}{\partial x} = \begin{bmatrix} \frac{\partial Y_{11}}{\partial x} & \cdots & \frac{\partial Y_{1n}}{\partial x} \\ \vdots & \ddots & \vdots \\ \frac{\partial Y_{m1}}{\partial x} & \cdots & \frac{\partial Y_{mn}}{\partial x} \end{bmatrix} \qquad \frac{\partial y}{\partial X} = \begin{bmatrix} \frac{\partial y}{\partial x_{11}} & \cdots & \frac{\partial y}{\partial x_{p1}} \\ \vdots & \ddots & \vdots \\ \frac{\partial y}{\partial x_{1q}} & \cdots & \frac{\partial y}{\partial x_{pq}} \end{bmatrix}$$

Matrix Calculus (numerator layout)

1. $$\frac{d}{dX_{ij}} X = e_i e_j^T$$

2. $$\frac{d}{dx} Ax = A$$

3. $$\frac{d}{dx} x^T A x = x^T \left(A + A^T \right)$$

4. $$\frac{d}{dx} M^{-1}(x) = -M^{-1}(x) \frac{dM(x)}{dx} M^{-1}(x)$$

5. $$\frac{d}{dX} tr\left(AX^{-1}B \right) = -\left(X^{-1}BAX^{-1} \right)^T$$

6. $$\frac{d}{dX} \log \det X = X^{-T}$$