INTRODUCTION

What is Computer Vision?

Endow machines with the ability to make sense of the physical world by looking at measurements of reflected light.
INTRODUCTION

INTRODUCTION

WHY IS THIS HARD?

Identify Materials

Foliage

Ceramic

Surface Properties

Wet?

Slippery?
Why is this hard?

Indirect Reflection

Observe super-position of multiple paths

Why is this hard?

Variation in appearance

Variation in appearance
WHY IS THIS HARD?

make sense of the physical world from measurements of reflected light

Seems hopeless .... except that humans, animals, birds, insects are able to do it

SO HOW DOES IT WORK?

By making assumptions about and exploiting structure in the natural world

SO HOW DOES IT WORK?

Broad Overview of (many a) Vision Algorithm

1. Understand the Image Formation Model: Scene to Image

\[ I = F(S) \]

2. Invert the Model: Gives us Multiple Physically Feasible Solutions

\[ \{S\} = F^{-1}(I) \]

3. Learn What Natural Scenes Look Like: Use to select likely scene among those that are feasible

So, we’re just guessing?

Well, sort of

ILLUSIONS

Deliberate / Artistic

Natural (rare!)
HISTORY OF CV

Attempt 1:
Let's finish this over the summer.

Slide via Lana Lazebnik

"Simple" Pattern Recognition

Slide via Lana Lazebnik

HISTORY OF CV

Face Detection in Cameras

Slide via Lana Lazebnik

HISTORY OF CV

The Smile Shutter flow

Imagine a camera smart enough to catch every smile. In Smile Shutter Mode, your Cyber-shot® camera can automatically trip the shutter at just the right instant to catch the perfect expression.

Slide via Lana Lazebnik

HISTORY OF CV


HISTORY OF CV

Lots of Exciting Research & Potential for Real World Impact!

THIS COURSE

All information @ http://www.cse.wustl.edu/~ayan/courses/cse559a/

SYLLABUS OVERVIEW

- Image formation, representation, and processing
- Low level vision
  - Photometric
  - Geometric
  - Motion
- Segmentation & Grouping
- High Level Vision (reasons with "semantic" knowledge)
  - Using deep convolutional neural networks

Slides will be posted on course website after class

PREREQUISITES

- Programming: problem sets will be in Python (but you can pick up Python+Numpy during the course).
- You should all have received an introductory e-mail. It has pointers to refresher slides on the math background. (Also linked to on the course website).
- Read through these slides
- If the material in them seems too unfamiliar, you might want to take courses on ProbStats / Linear Algebra before taking this class.

5 Problem Sets

- Math (answers to be typeset in LaTeX) and Programming (in Python)
- 15% x 5 = 75% of your grade
- To be done individually
- Roughly every two weeks (see website)
- READ collaboration and late policy
- Submitted using git.

Final Project

- To be done individually
- Open ended, mini research/implementation project
- Choose topic (suggestions on course website), submit brief proposal, and get feedback from us
- End of Term: Report = 25% of your grade

NO EXAMS
Problem Sets: Late Policy

- Any problem set turned in more than 3 late days will simply not be graded.
- Problem sets turned in 1-3 days late will be penalized.
- Initially, you will receive your graded problem set without any penalty.
- Then, we will count your total number of late days across problem sets.
- You get 3 total free late days.
- We will decide which problem sets to apply those late days for to give you maximum credit.
- Beyond that, you will lose 25% of your grade on a problem set for each day late (after deducting the free late days).
- 1 Late Day = 1 min to 24 hours late!

Project Proposals and Reports CAN NOT BE LATE

Academic Honesty

- All problem sets and projects MUST BE YOUR OWN work.
- Read through collaboration policy carefully. It is OK to discuss general concepts with your friends and classmates, or look for background knowledge online (like resources for numpy, linear algebra, etc.).
- But you can not seek or offer help for specific solutions to problem set questions.
- Do not share code or look for solutions online. PERIOD.
- Acknowledge any source of discussion or information in the problem sets (included in the homework solution template).
- ANY VIOLATIONS WILL RESULT IN FAILURE OF THE COURSE.

It does not matter how many points the violation affected. This will also be reported to the university and will quite likely attract stricter sanctions beyond the course.

- For final projects, you may use external libraries or provided source code: as long as you acknowledge them explicitly in your project report. You will be evaluated on your own contribution beyond these resources.

Todo: Testing Environment & Submission Using git

- Read Problem Set section
- Install Anaconda
- Setup a LaTeX environment, or familiarize yourself with one of the online LaTeX editors.
- Set up git. Read tutorials on public key authentication if you are not familiar.
- Generate a public-private key pair if you don’t have one. Set up your environment to use the private key to authenticate with git.
- Submit your public key using the link e-mailed to you.
- Then as soon as possible, try to complete Problem Set 0. This is a trivial problem set to test your setup (LaTeX, Anaconda, Git).
- Ungraded, but mandatory!
This course

All information @ http://www.cse.wustl.edu/~ayan/courses/cse559a/

TODO: Testing Environment & Submission Using git

About 10-20 minutes after you upload your public key, you should be able to clone the pset0 repository as:

git clone cse559@euclid.seas.wustl.edu:wustl.key/pset0

- **IMPORTANT**: Replace wustl.key with your actual WUSTL key username.
- If it asks you for a password, your public key authentication setup isn’t working.
- If the clone is successful, you will see a newly created pset0 directory. Look at the pset0.pdf file inside the directory.
- Complete the homework (write and run the code, complete the report to create solution.pdf).
- Then add your code and solution.pdf files, commit & push to submit, and pull to verify submission.

Work on your problem set.

- You need to push. To determine whether your problem set is on time or late, we will look at when it was pushed, not committed!
- Verify that it was submitted properly

This will show you a message with a timestamp and all the files included in your submission.
- Hold on to this repository. Once we have graded your submission, you can do a git pull to retrieve comments and feedback.

- Sign up for Piazza!
  - Ask ALL course-related questions on piazza
    - Make private posts if you wish.
  - Answer others’ questions (subject to collaboration policy)

- Office Hours: TBD

Questions?
THE PINHOLE CAMERA

Record Intensity at each sensor Location

Plane with Sensor Elements

Image = Array of Numbers

Form Image on Sensor Plane

Physical Object

Sensor Plane

Object

1D for simplicity

THE PINHOLE CAMERA

Light

Sensor Plane

Object

Why can't we get an image to form by just holding the sensor in front of the object?

THE PINHOLE CAMERA

Sensor Plane

Object

Because the object isn't just reflecting in straight lines!
A point on sensor plane receives light from many distinct points on the object. No image since intensities "washed out".

Every point on the sensor plane corresponds to a unique ray.
THE PINHOLE CAMERA

By Similar Triangles: \( s_y = -\frac{f}{z} \)
THE PINHOLE CAMERA

We will go in-depth into camera projection in a few weeks....

By Similar Triangles: \[ s_y = -f \frac{y}{z}, \quad s_x = -f \frac{x}{z} \]
\[
(x, y, z) \Rightarrow \left( -f \frac{x}{z}, -f \frac{y}{z} \right)
\]

stay parallel!

Patterns on a "Fronto-parallel Pane" scaled by the same z

\[
(x, y, z) \Rightarrow \left( -f \frac{x}{z}, -f \frac{y}{z} \right)
\]

THE PINHOLE CAMERA

We will go in-depth into camera projection in a few weeks....

Different z will get scaled differently:
"perspective distortion"

\[
(x, y, z) \Rightarrow \left( -f \frac{x}{z}, -f \frac{y}{z} \right)
\]

THE PINHOLE CAMERA

We will go in-depth into camera projection in a few weeks....

Different z will get scaled differently:
"perspective distortion"

\[
(x, y, z) \Rightarrow \left( -f \frac{x}{z}, -f \frac{y}{z} \right)
\]
THE PINHOLE CAMERA

Not new to digital cameras!

Basic principle known to Mozi (470-390 BCE), Aristotle (384-322 BCE)

Source: A. Efros

Not new to digital cameras!

Basic principle known to Mozi (470-390 BCE), Aristotle (384-322 BCE)

Drawing aid for artists: described by Leonardo da Vinci (1452-1519)

Image via Subhramsv Maji

Source: A. Efros

THE PINHOLE CAMERA

Eclipsomania 2017!

SAFE WAY TO VIEW ECLIPSE

Eclipsomania 2017!
Accidental Pinhole and Pinspeck Cameras: Revealing the scene outside the picture
Antonio Torralba, William T. Freeman CVPR 2012
http://people.csail.mit.edu/torralba/research/accidentalcameras/

Turning windows and shadows into pinhole and anti-pinhole cameras.