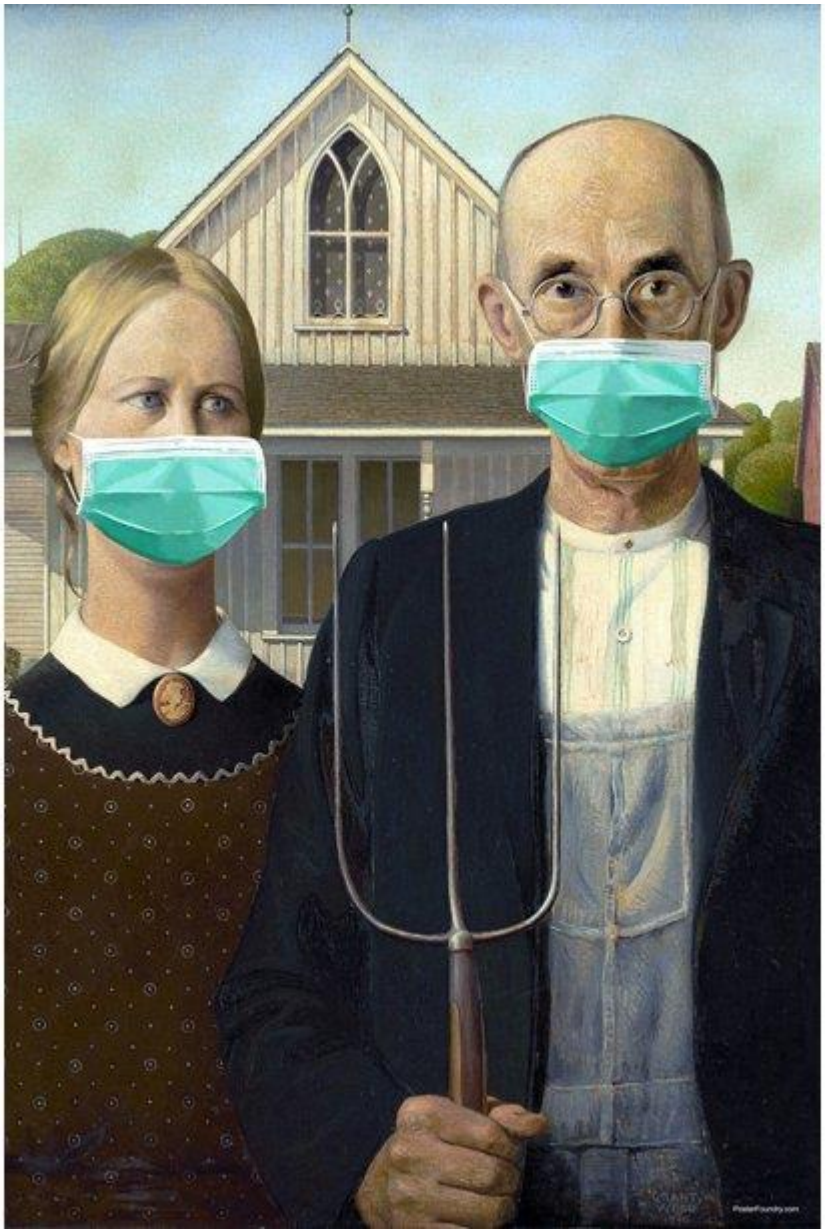
CS 180/280A: Intro to Computer Vision and Computational Photography

Instructors: Alexei Efros
Angjoo Kanazawa
GSIs: Ruilong Li
Jack Austin
Readers: Morgan Lyu
Preston McCrary
Max Vogel
UC Berkeley, Fall 2023

Covid Precautions



Today

Introductions

Why this Course?

Administrative stuff

Brief History of Visual Data

Teaching Team: professors

Angjoo Kanazawa

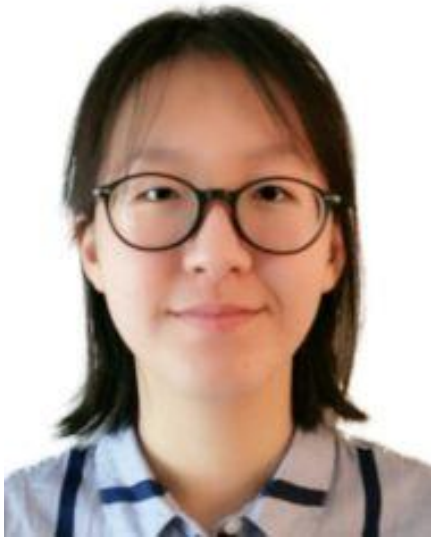
Alexei Efros

Teaching Team: GSIs

Ruilong Li

Jake Austin

Teaching Team: Readers/Tutors



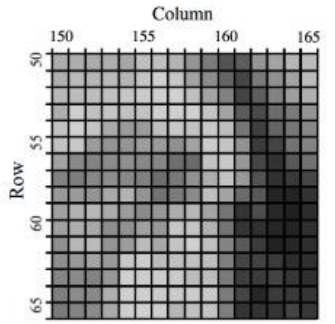
Morgan Lyu

Max Vogel

Preston McCrary

Why This Course?

Visual Computing in the old days…

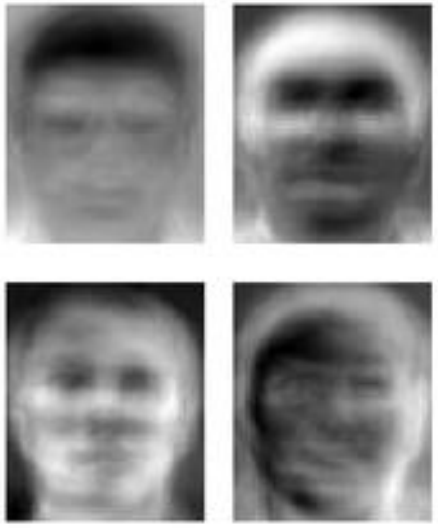


Column

Row	150	155	160	165
50	183 183 181 184 177 200 200 189 159 135 94 105 160 174 191 196			
	186 195 190 195 191 205 216 206 174 153 112 80 134 157 174 196			
	194 196 198 201 206 209 215 216 199 175 140 77 106 142 170 186			
	184 212 200 204 201 202 214 214 214 205 173 102 84 120 134 159			
55	202 215 203 179 165 165 199 207 202 208 197 129 73 112 131 146			
	203 208 166 159 160 168 166 157 174 211 204 158 69 79 127 143			
	174 149 143 151 156 148 146 123 118 203 208 162 81 58 101 125			
	143 137 147 153 150 140 121 133 157 184 203 164 94 56 66 80			
60	164 165 159 179 188 159 126 134 150 199 174 119 100 41 41 58			
	173 187 193 181 167 151 162 182 192 175 129 60 88 47 37 50			
	172 184 179 153 158 172 163 207 205 188 127 63 56 43 42 55			
	156 191 196 159 167 195 178 203 214 201 143 101 69 38 44 52			
65	154 163 175 165 207 211 197 201 201 199 138 79 76 67 51 53			
	144 150 143 162 215 212 211 209 197 198 133 71 69 77 63 53			
	140 151 150 185 215 214 210 210 211 209 135 80 45 69 66 60			
	135 143 151 179 213 216 214 191 201 205 138 61 59 61 77 63			

Image Processing
EECS 225B

Computer Graphics
CS 184



Computer Vision CS 280

Visual Computing gets interconnected

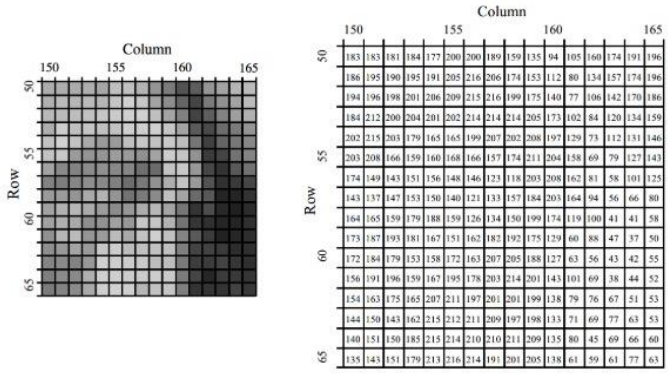
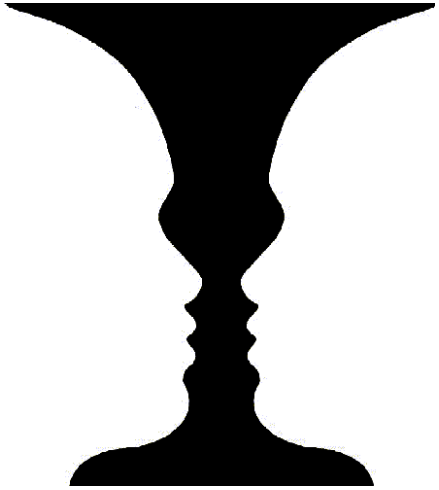


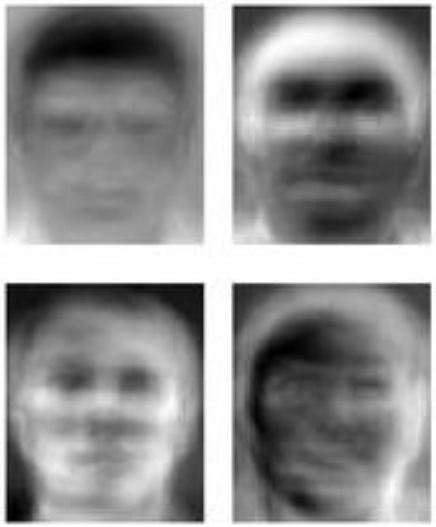
Image Processing
EECS 225B

Art History
ART 10

Computer Graphics
CS 184

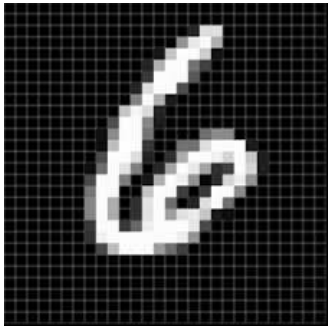


Visual Perception
PSYCH



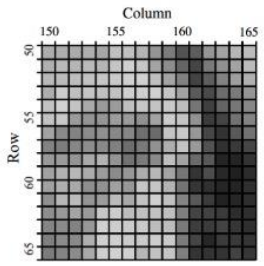
Computer Vision CS 280

Computational Photography



Machine Learning

Visual Computing gets interconnected



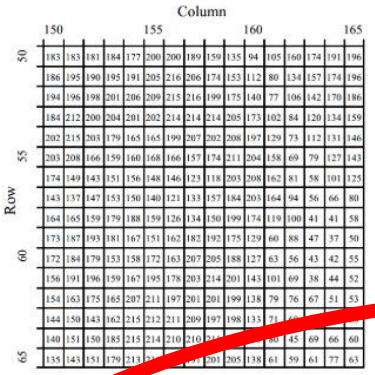
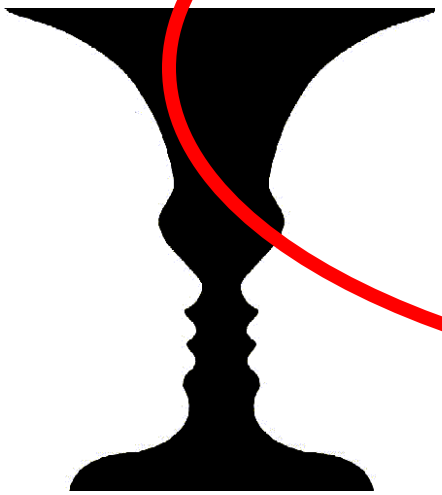


Image Processing
EECS 225B

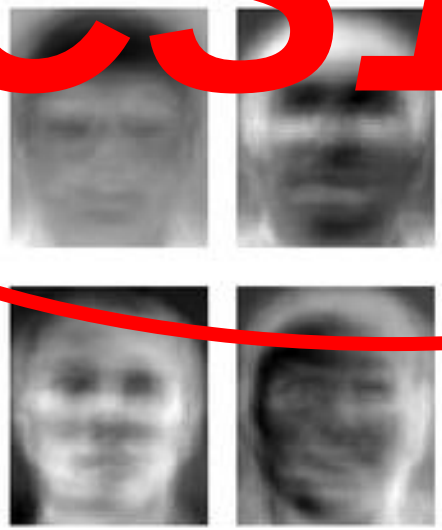
Art History
AR 10

Computer Graphics
CS 184

CS180



Visual Perception
PSYCH



Computer Vision CS 280

Computational Photography

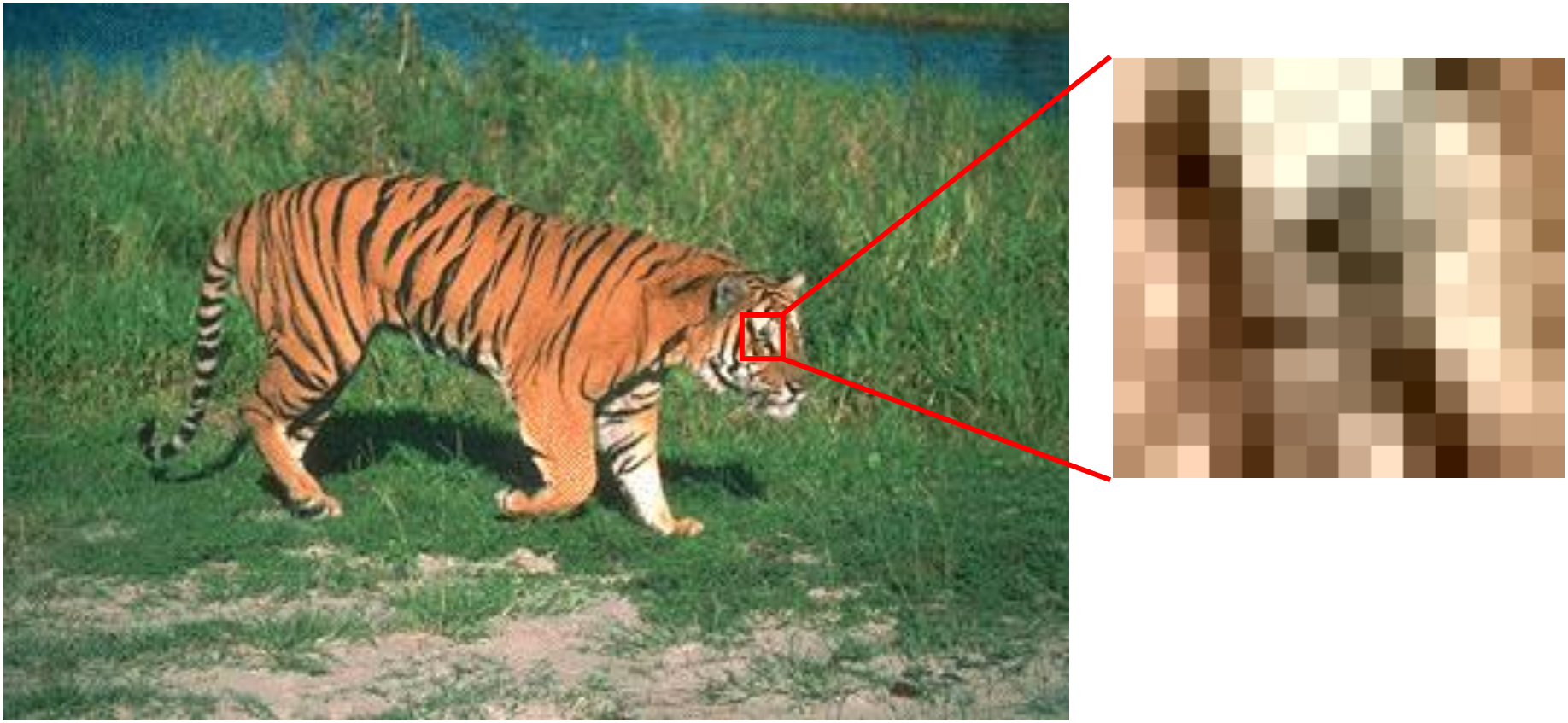
Machine Learning

CS180: Focus on Visual Data

The key objective of this class is *to become friends with every pixel!*

Course objectives

1. You will appreciate the fundamental difficulty of understanding and computing with visual data



Course objectives

2. You will get a foundation in image processing and computer vision, **from the ground up**:

- Camera basics, image formation
- Convolutions, filtering
- Image and Video Processing (filtering, anti-aliasing, pyramids)
- Image Manipulation (warping, morphing, mosaicing, matting, compositing)
- Data-driven Generative Models
- Projection, 3D, stereo
- NeRFs
- …

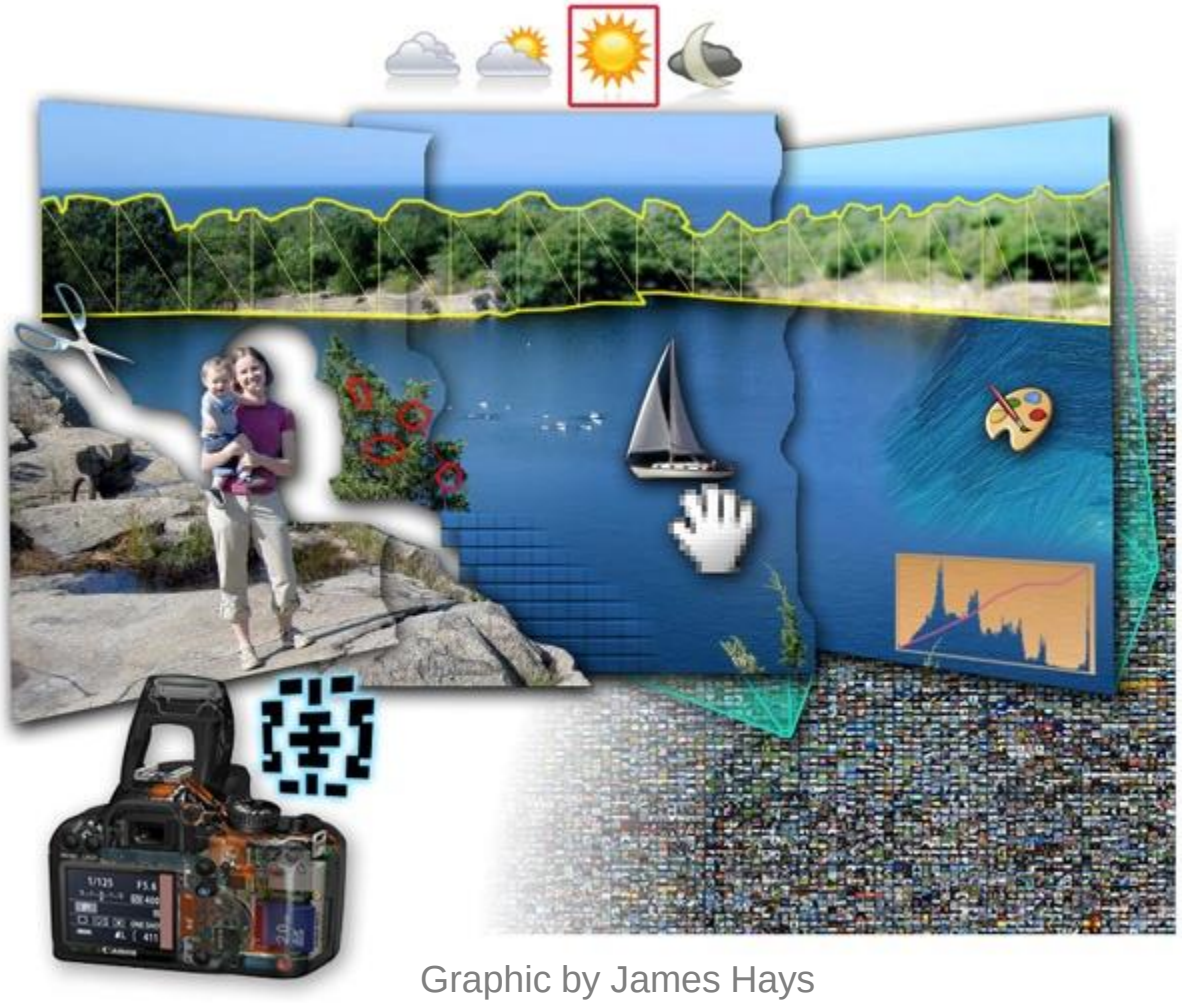
Course objectives

3. You will get a more intuitive understanding of important mathematical and computational concepts

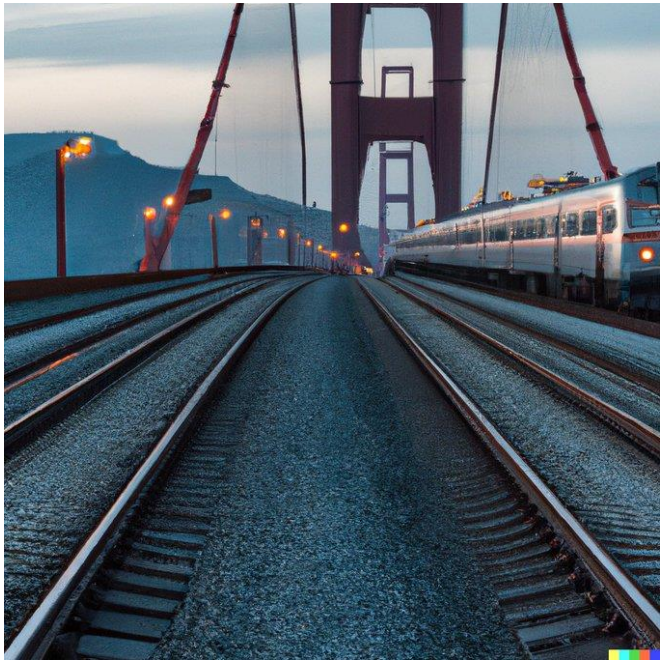
- Gradients
- Change of basis, interpolation, extrapolation, PCA
- FFT
- Dynamic programming, recursion
- Machine learning, Convolutional Neural Networks
- Large-Pixel-Models
- …

Course objectives

4. You will learn approaches for **visual synthesis**



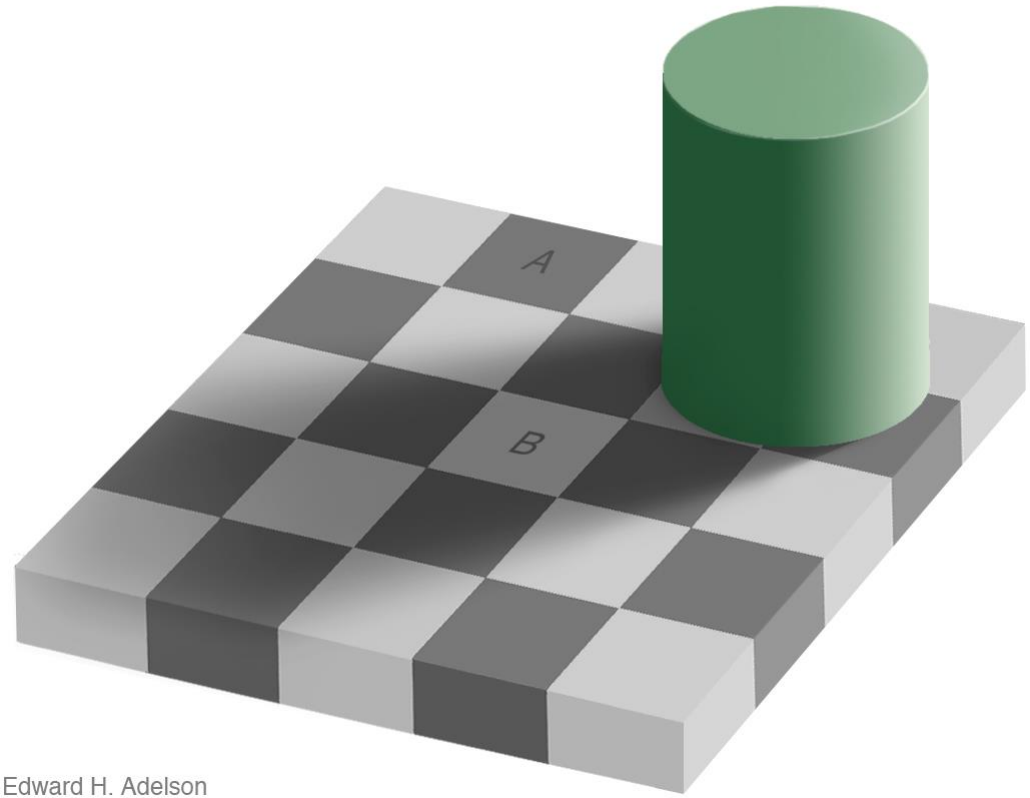
Graphic by James Hays



DALL-E + Danielle Baskin

Course objectives

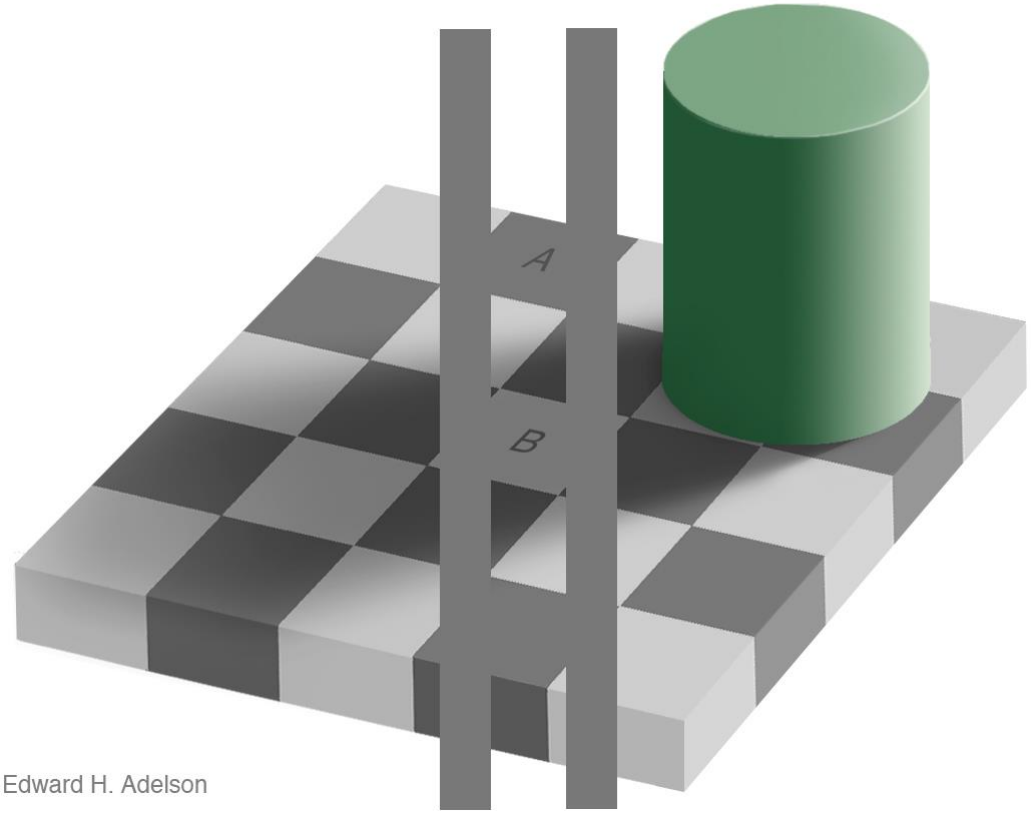
4. You'll better appreciate human visual perception



Edward H. Adelson

Course objectives

4. You'll better appreciate human visual perception



Edward H. Adelson

Different people see different things

https://en.wikipedia.org/wiki/The_dress

People see things that aren’t there

Video by Antonio Torralba (starring Rob Fergus)

But actually…

Video by Antonio Torralba (starring Rob Fergus)

Course objectives

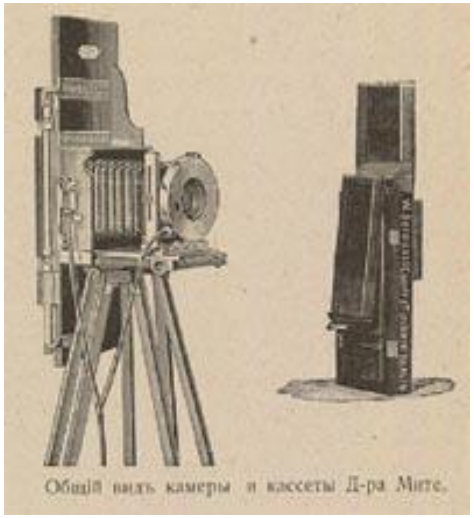
5. You will learn about the **history of ideas** in visual computing
   - Did you know Large Generative Models go back to 1940s?
   - Or that Deep Learning started with a Nobel Prize in Neuroscience in the 1960s?
   - …

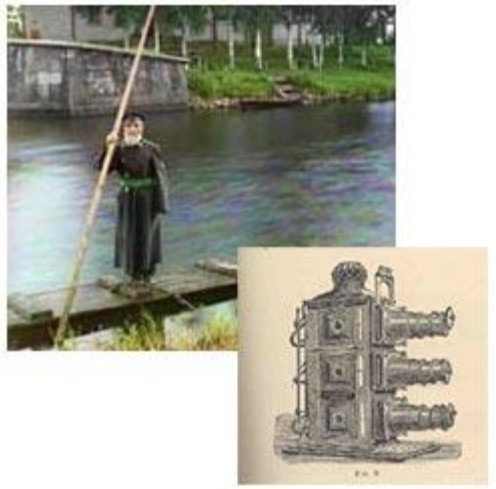
Course objectives

6. You'll have fun doing cool stuff, coding up a storm, while you **befriend the pixels**

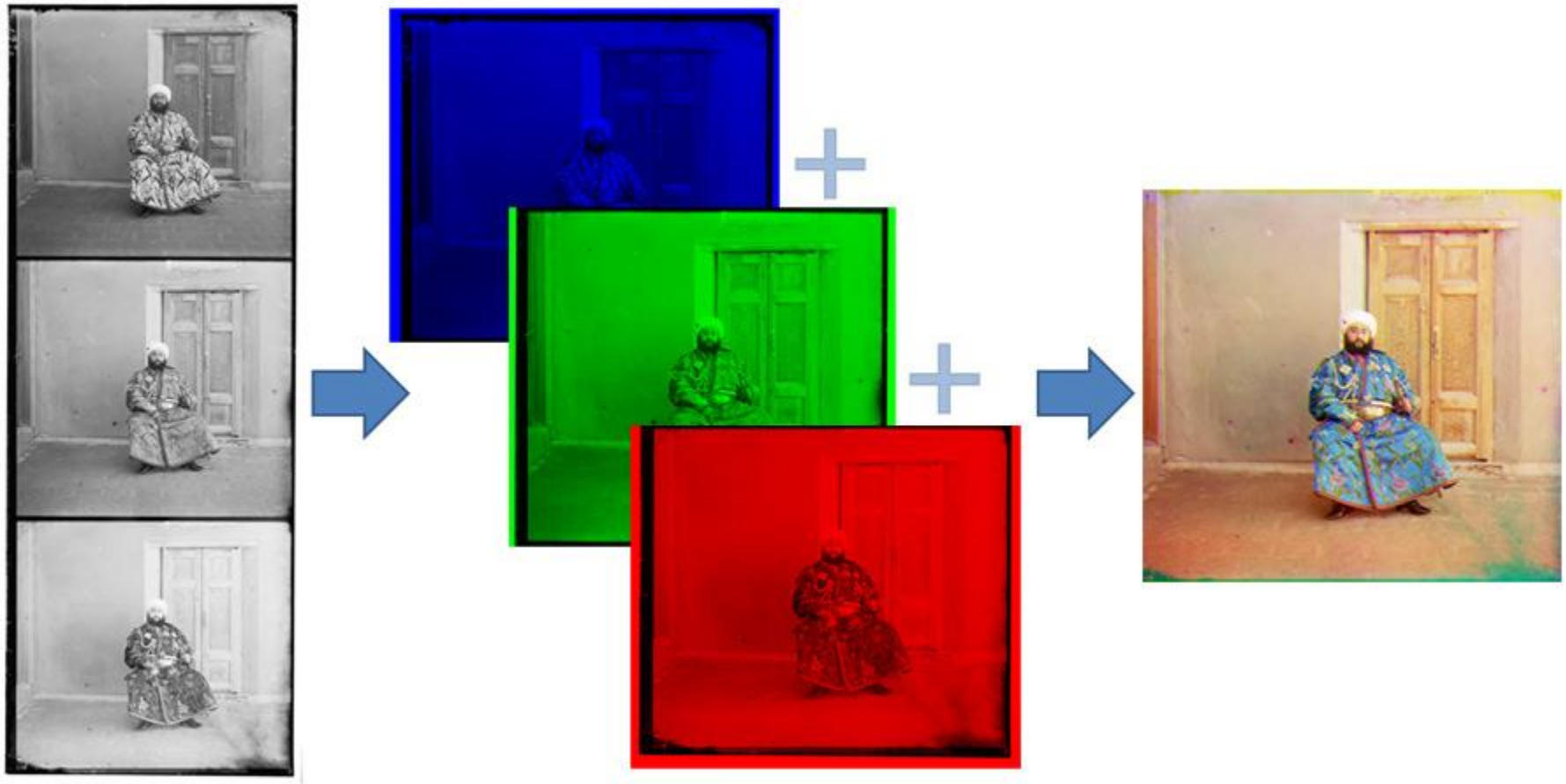
Programming Project #1

Prokudin-Gorskii's Color Photography (1907)



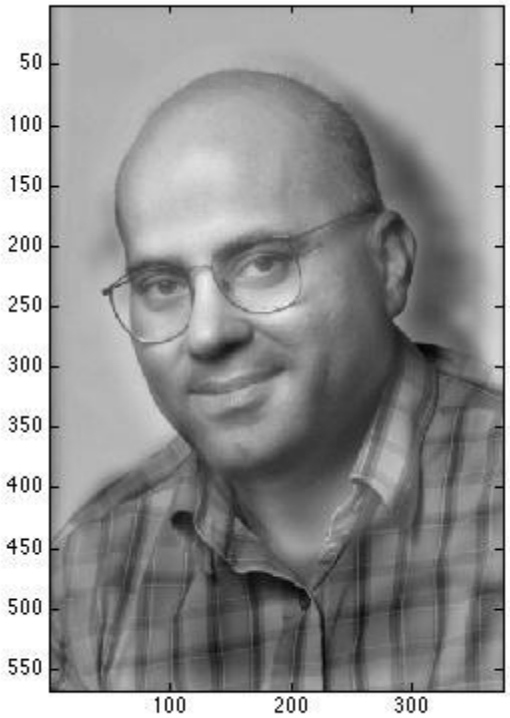


Programming Project #1



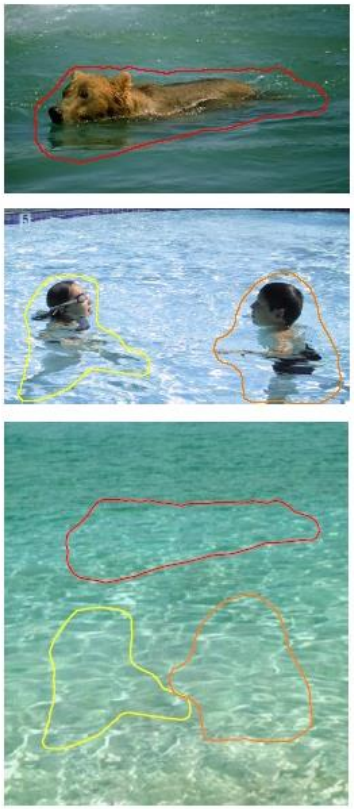
Project 2: Fun with frequencies

Project 2: Fun with frequencies



Prof. Christos Papadimalik

Project 2: Fun with Frequencies

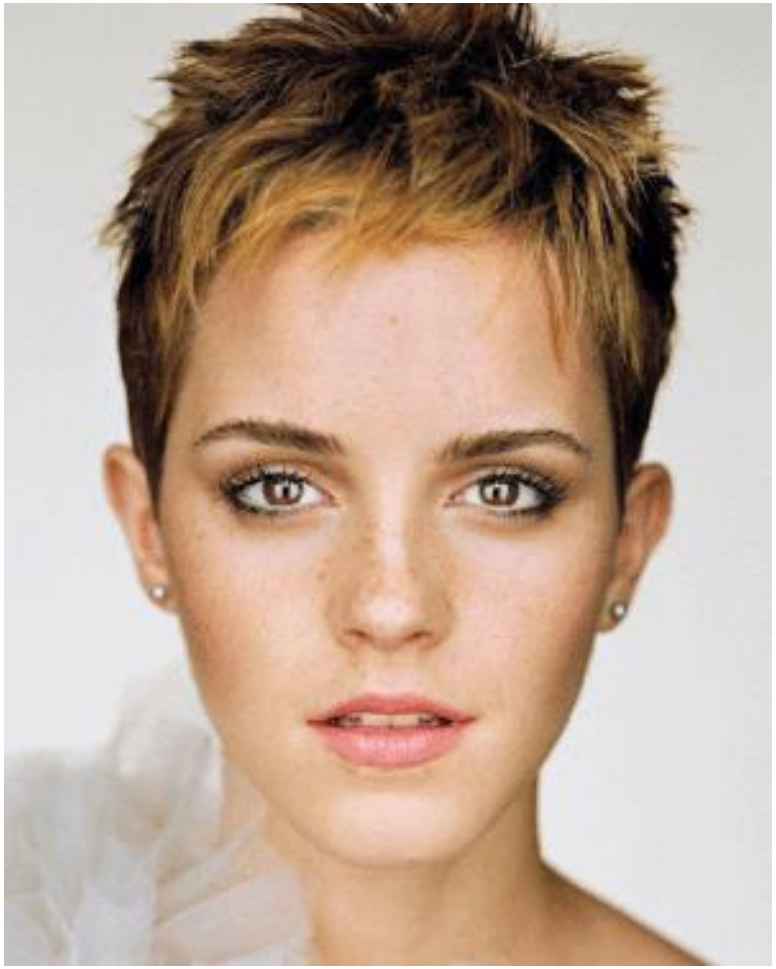


sources/destinations

cloning

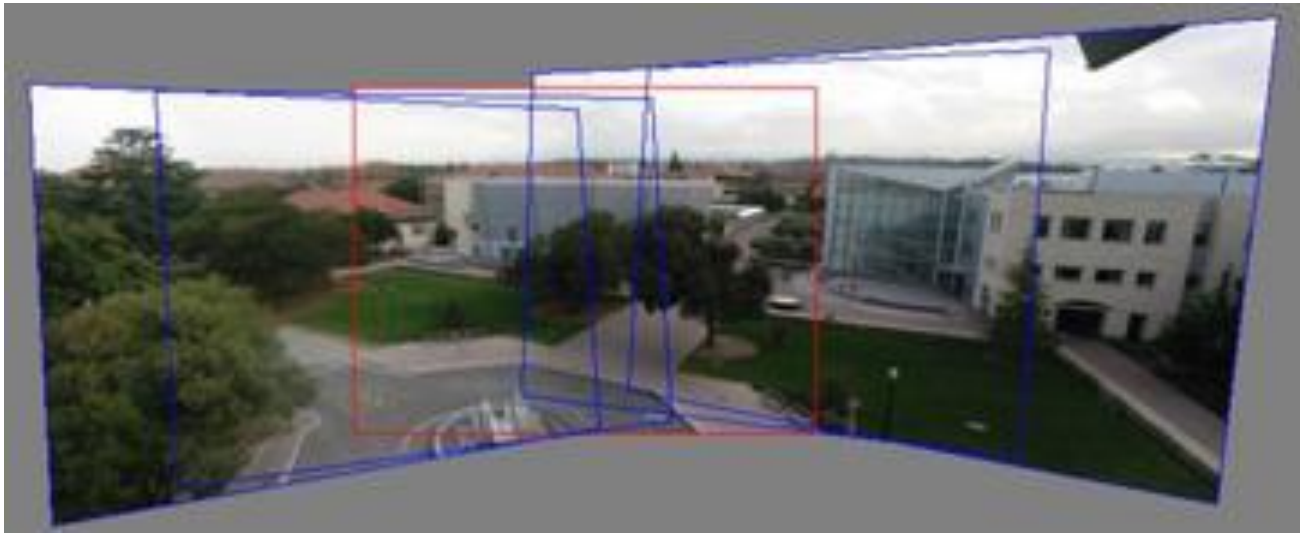
seamless cloning

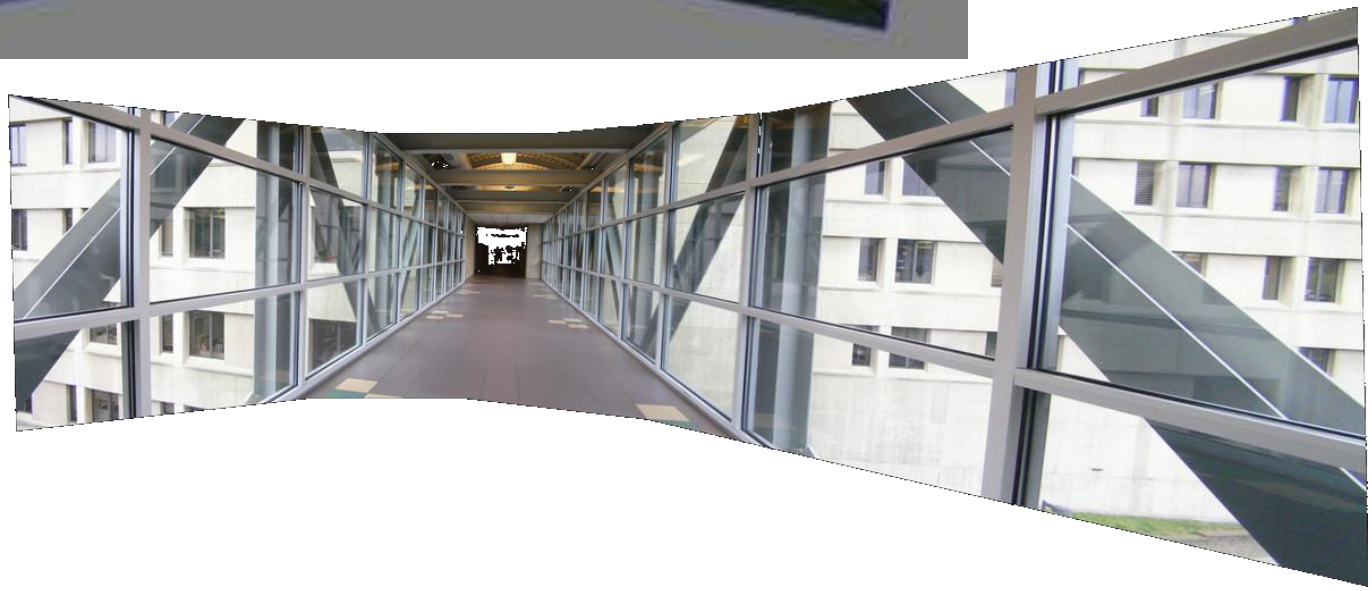
Project 3: Face modeling and morphing



Project 4: Panorama Stitching

Photo Mosaics





Project 5: TBD

(depends on if we could get GPU donation)

Final Project

Something cool!!!

- We will have some pre-canned projects
- Will also have some suggestions, cool datasets, etc
- Or you can do whatever you want!

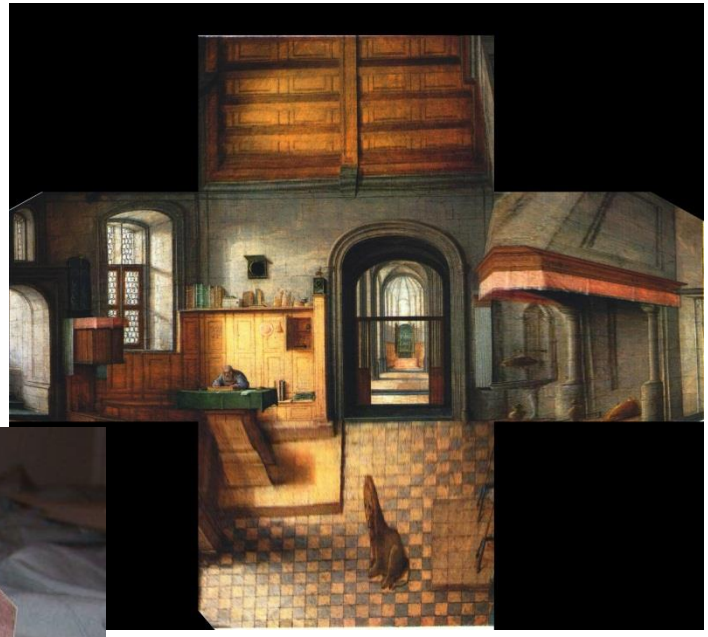
(can be done in groups of 2 or 3)

Example Pre-canned Project

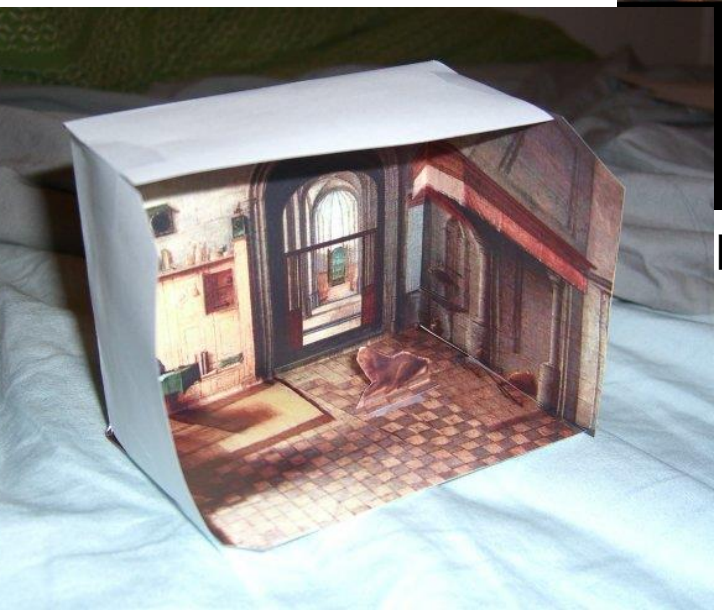
Tour Into the Picture

Paper Pop-up

Step 1: define planes



p 2: rectify each plane



Step 3: compute 3D box coords

Sample final project in my class

Everybody Dance Now

https://www.youtube.com/watch?v=PCBTZh41Ris&feature=youtu.be

For each project:

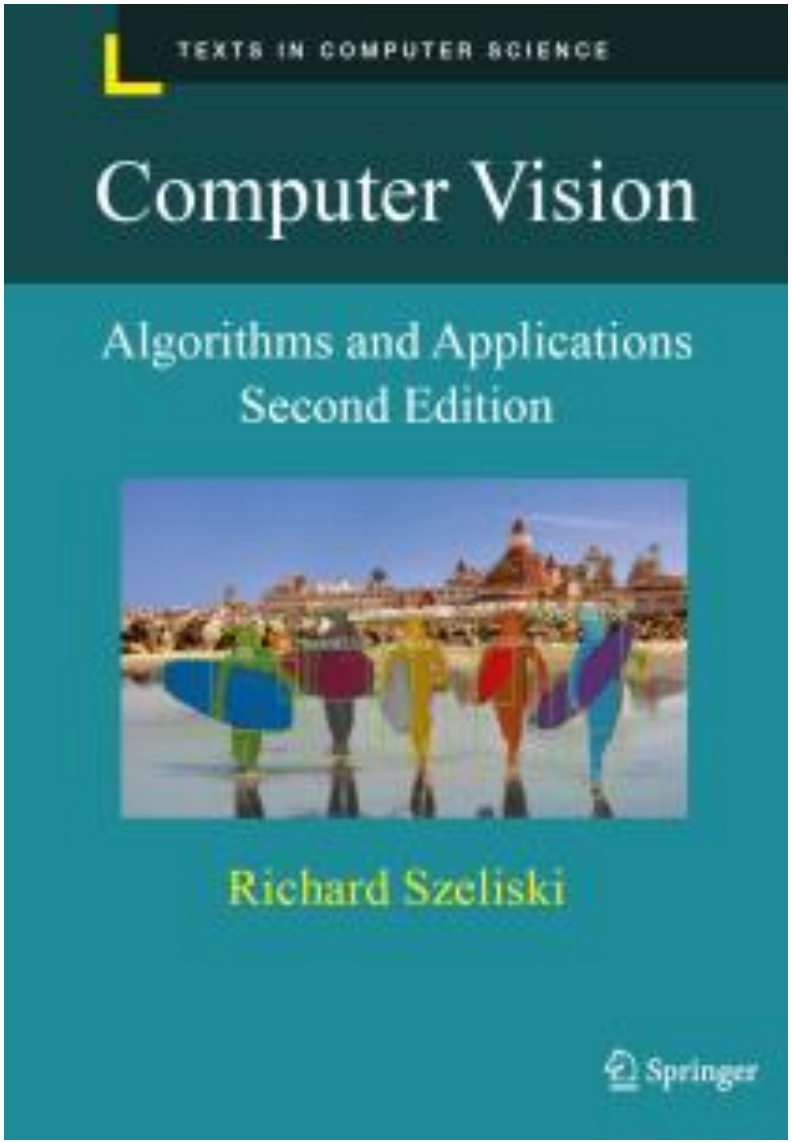
Derive the **math**, implement stuff **from scratch**, and apply it to your **own** photos

Every person does their own project (except final projects)

Reporting via web page (plus submit code)

Afterwards, vote for class favorite(s)!

Textbook



http://szeliski.org/Book/

Class Organization / Administrivia

General

Prerequisites

- Linear algebra!!! (EE16A, Math 54, or Math 110)
- Good programming skills (at least CS61B)
- Deep Learning experience strongly recommended!

Emphasis on programming projects!

- Building something from scratch

Graduate Version:

- Final project required (not pre-canned), including conference-style report paper

Administrative Stuff

Grading

- Programming Project (60%)
- Exam + possible popup quizzes (20%)
- Final Project (20%)
- Class Participation: priceless

Late Policy

- Five (5) **emergency** late days for semester. The expectation is you will never use them.
- Max 10% of full credit afterwards

Extra Points

- Most projects will have optional “bells & whistles”
- These extra points could be used to pad scores on other projects (but not exams!)

Rule # 1:
No lecture recordings

This is **an in-person class**. You are to come to the lecture and ask questions! Attendance is required.

Only available by request for truly exceptional circumstances, e.g. severe illness

Rule # 2: Deadline is a deadline

In real life there's no slip days

This is a FUN but INTENSE class, projects come **one after another**

Slip days are for **emergencies.** If nothing dramatic happened to you during the semester, you should have all your slip days left.

Projects are time consuming. Start early!!!

Rule # 3:
TA’s don’t debug code

TA’s don’t debug code for you.

Part of the skill is to learn how to ask questions to debug the issue without presenting the code

Visualize the results and send those to figure out what is wrong

Use the pixels – become friends with visual debugging

Getting help outside of class

Course Web Page

- http://inst.eecs.berkeley.edu/~cs180

Discussion board:

- Ed

Discussion Section:

- Ruilong: Tuesdays, Cory 247 1pm - 2pm (Capacity 63)
- Jake: TBD

Office hours

- For instructors: after lecture
- For others, see webpage

Academic Integrity

- Can discuss projects, but **never share code**
- Don’t search for code or copy from a friend
- If you’re not sure if it’s allowed, ask
- Cite any sources and inspirations

Our GPT policy

- GPT is a wonderful tool
  - And so is calculator, Wolfram Alpha, Wikipedia, Stack Overflow, etc.
- but before you use a calculator, it's important to learn how to do long division by hand.
- In this course, we want you to do things from scratch.
  - So, no Stack Overflow, no searching for code, no fancy libraries, and no GPT
- You can use GPT (sparingly) to debug your code (if nothing else works), but please acknowledge and submit transcript
- Can use whatever for "bells and whistles"

Waitlists

- To keep this course live, we are limited by room size (~300 people)
- However, we expect 50-70 people to drop after the first two projects ☺
  - So, if you are on waitlist, etc, you have good chance to get into class
  - But need to start doing projects!

Warning:
historically high GPA of this course

- Survivor bias
- High class GPA != easy course
- This is a FUN but INTENSE class
- You write the code from scratch, that’s the point.
- Rubrics are fuzzy, goals are ill-defined, that’s the point.

Why you should NOT take this class

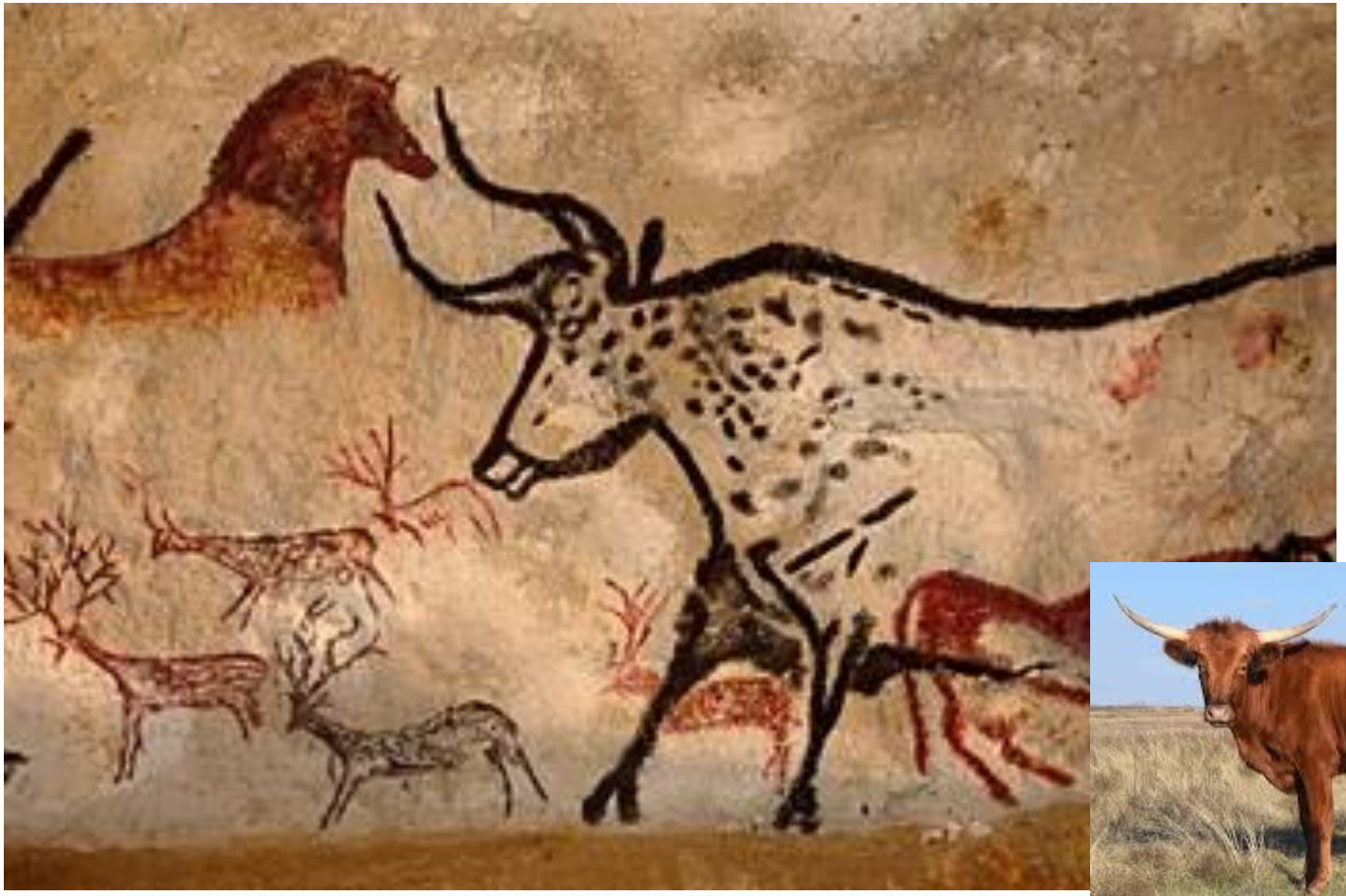
- Project-based class
  - No canned problem sets
  - Not theory-heavy (but will read a few research papers)
  - No clean rubrics
  - Open-ended by design
  - Will not copy advanced topics, but will try to make sure everyone understands the basics super-well
- Need time to think, not just hack
  - **Creativity** is a class requirement
  - We already expect you to know Deep Learning!
- Lots of work…There are easier classes if
  - you just need some units
  - you care more about the grade than about learning stuff
- Not worth it if you don’t enjoy it

Now… reasons TO take this class

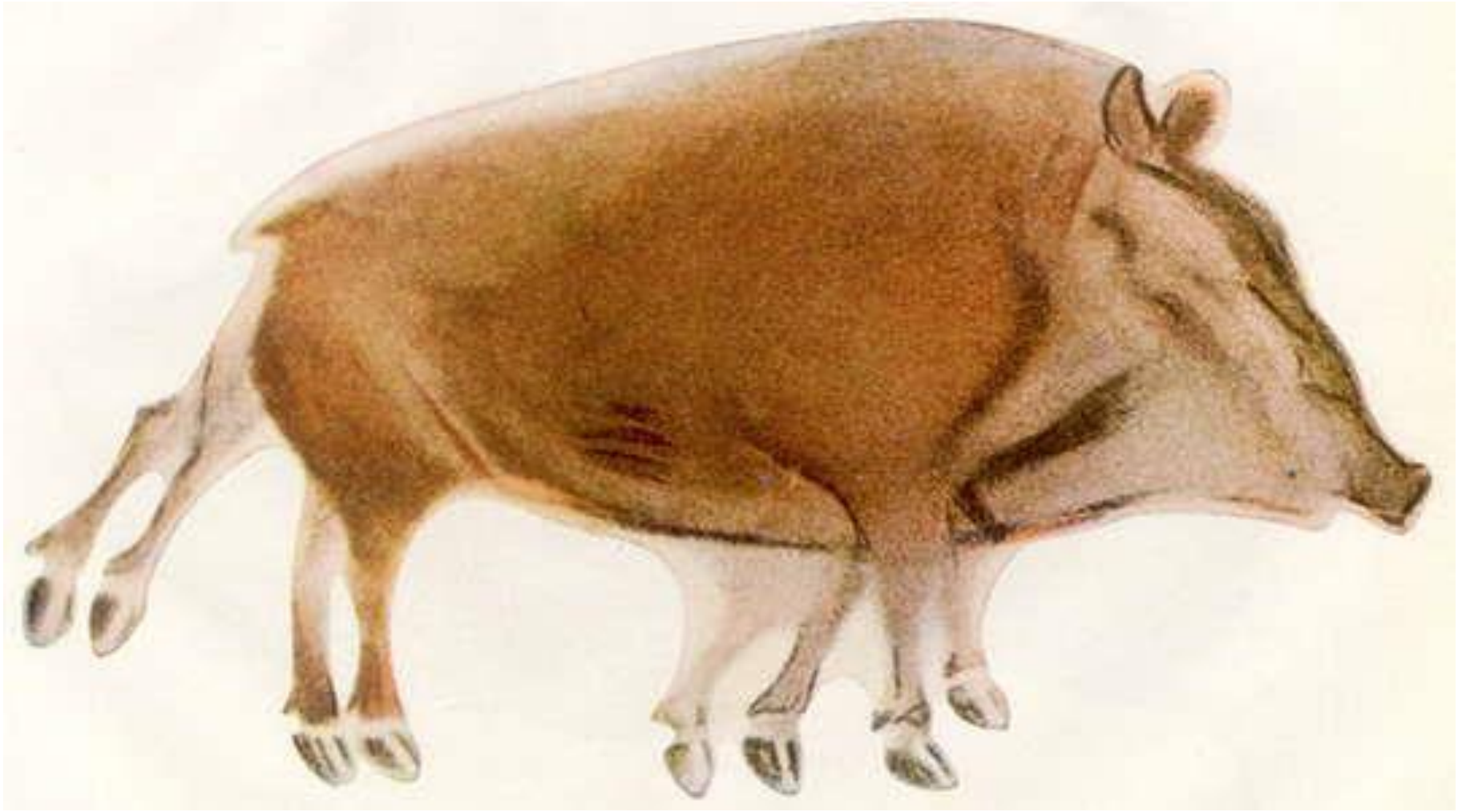
- It's your reward after 3 grueling years ☺
- You get to work with pictures, unleash your creative potential
- Interested in grad school? ☺

A Brief History of the Visual Data

Depicting Our World: The Beginning

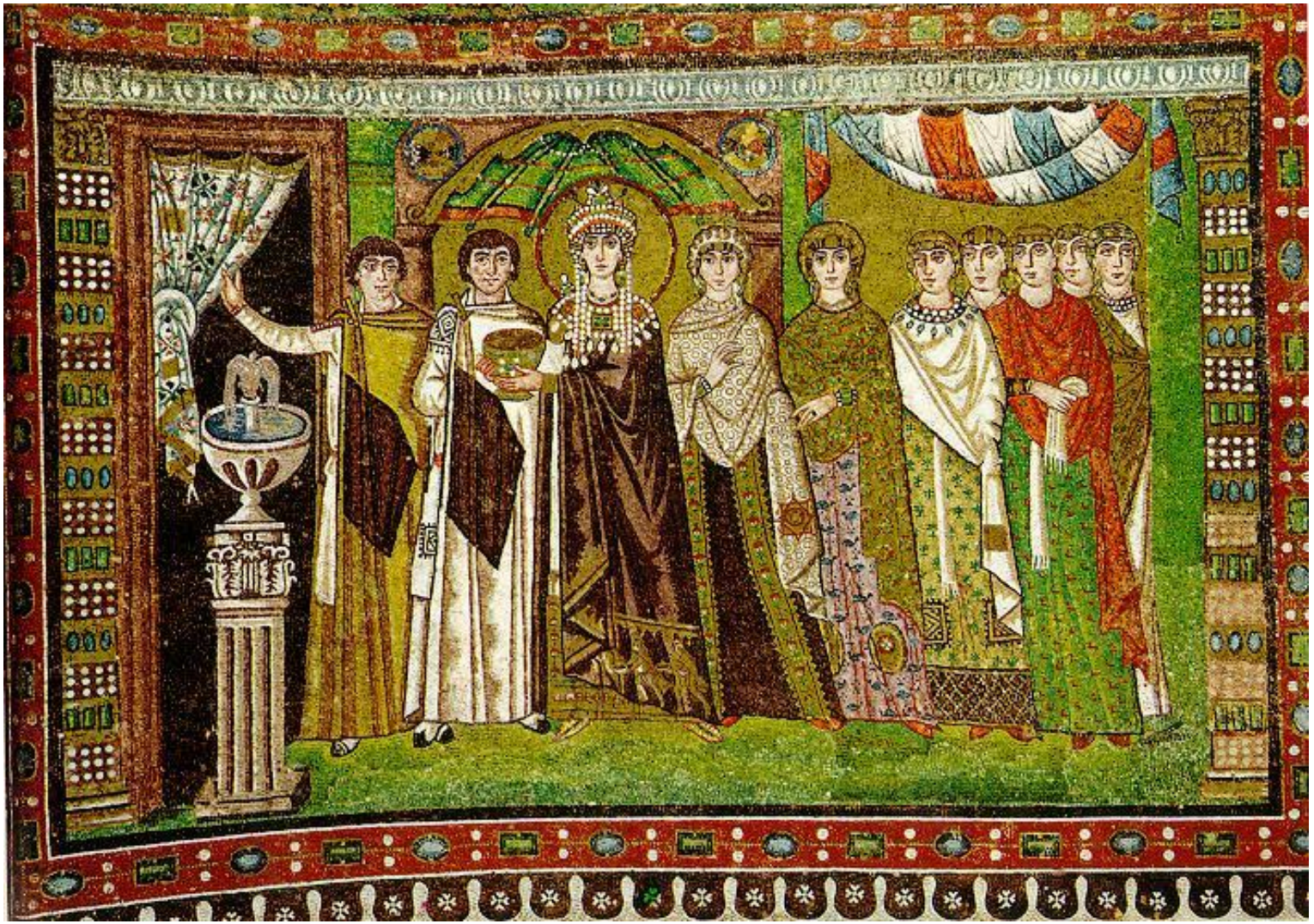


Prehistoric Painting, Lascaux Cave, France
~ 13,000 -- 15,000 B.C.



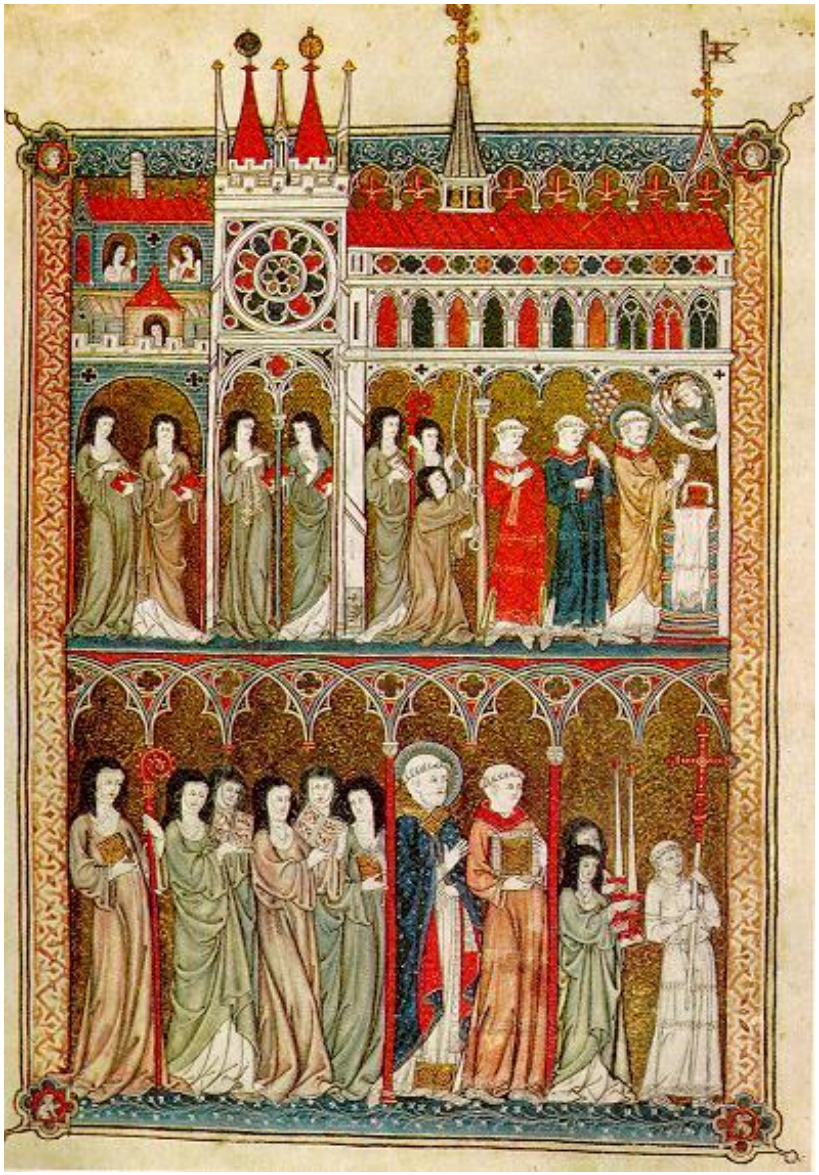
Prehistoric Cave Painting, Altamira
~ 20,000 – 15,000 B.C.

Depicting Our World: Middle Ages



The Empress Theodora with her court.
Ravenna, St. Vitale 6th c.

Depicting Our World: Middle Ages



Nuns in Procession. French ms. ca. 1300.

Beginnings of the Renaissance

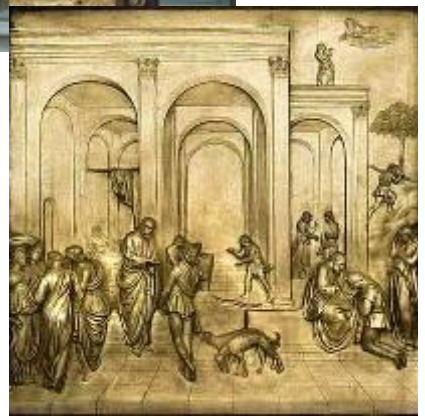
Giotto, *The Mourning of Christ*, c.1305

Depicting Our World: Renaissance

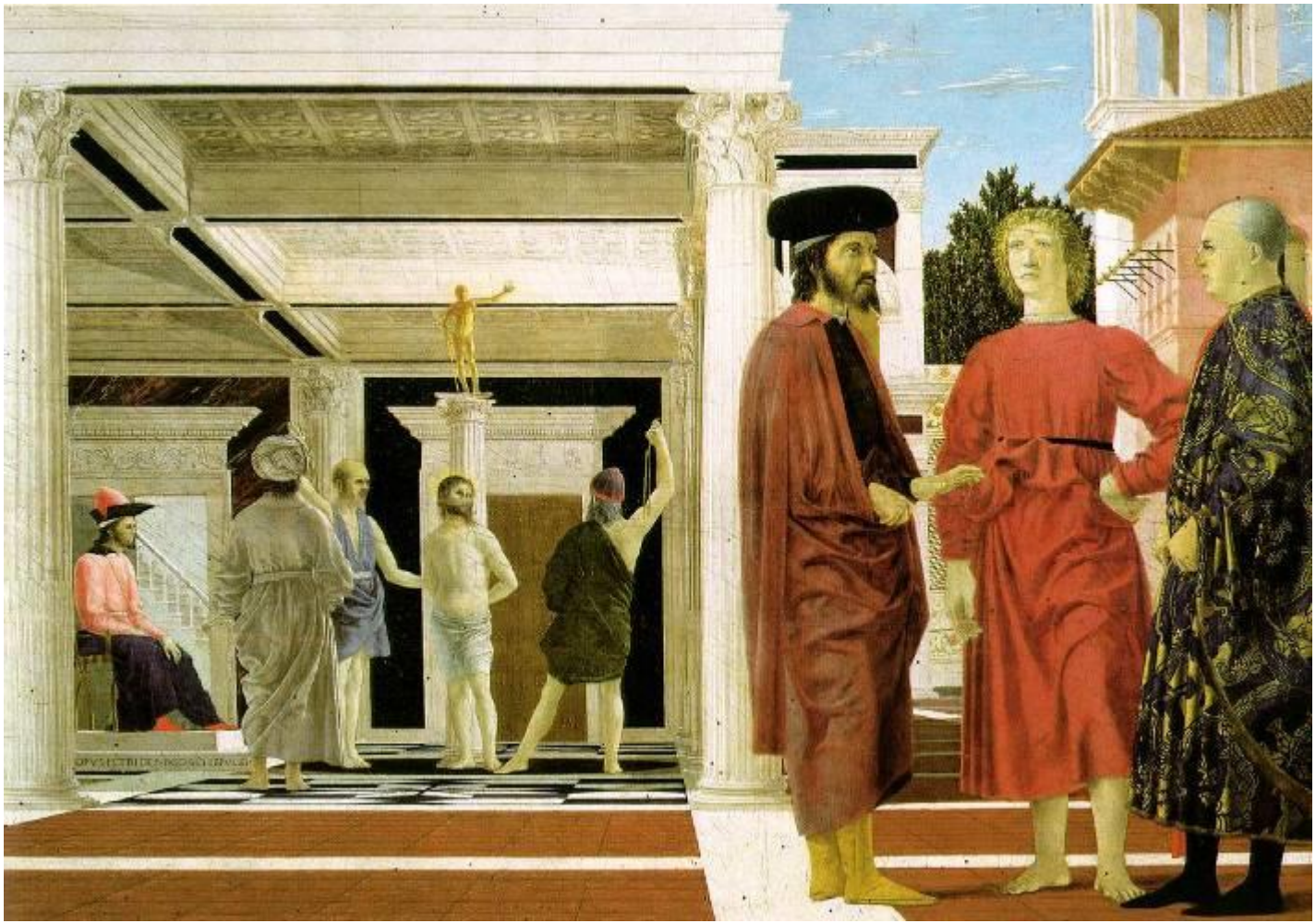
North Doors (1424)

Lorenzo Ghiberti (1378-1455)

East Doors (1452)

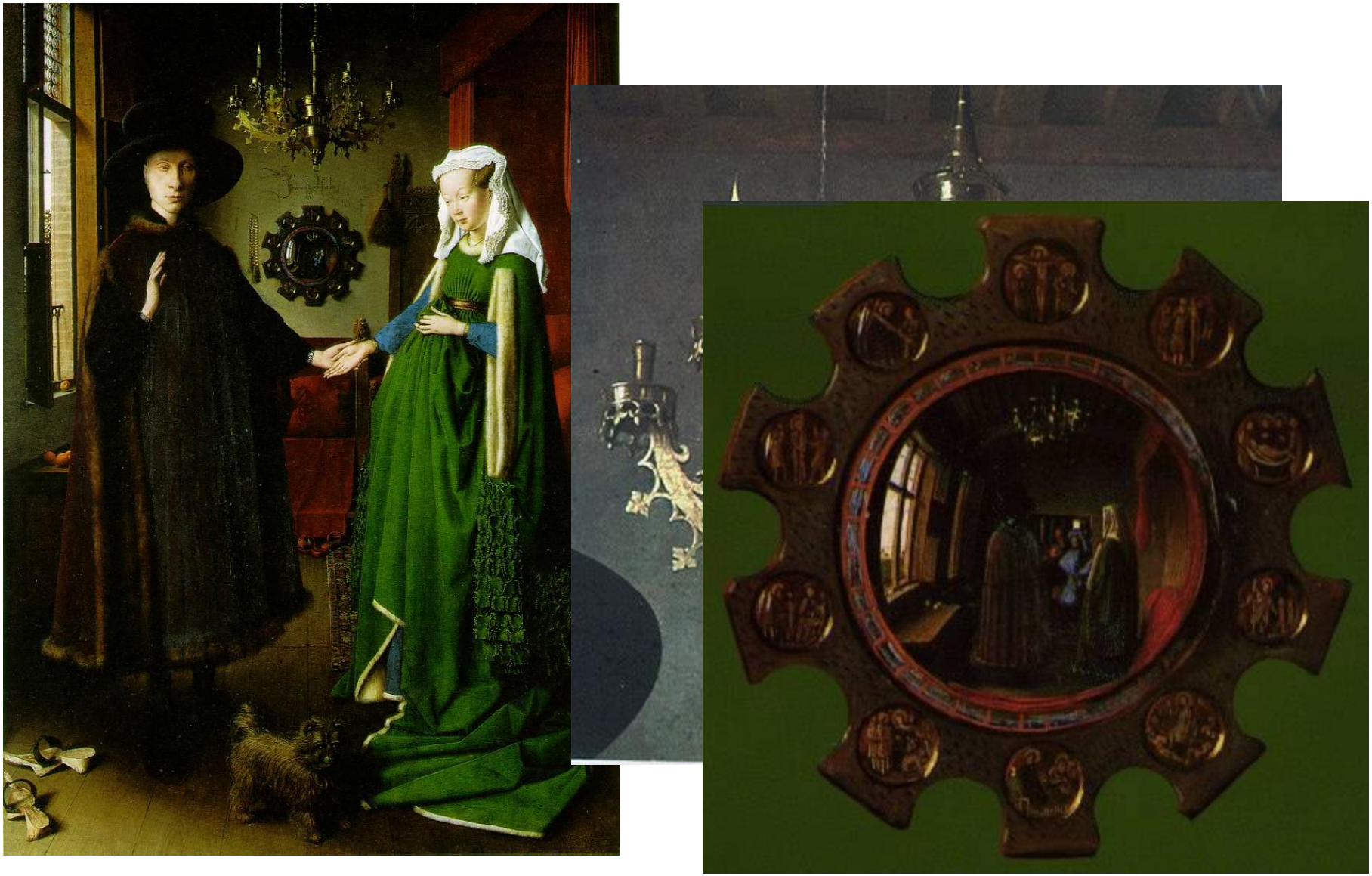


Depicting Our World: Renaissance



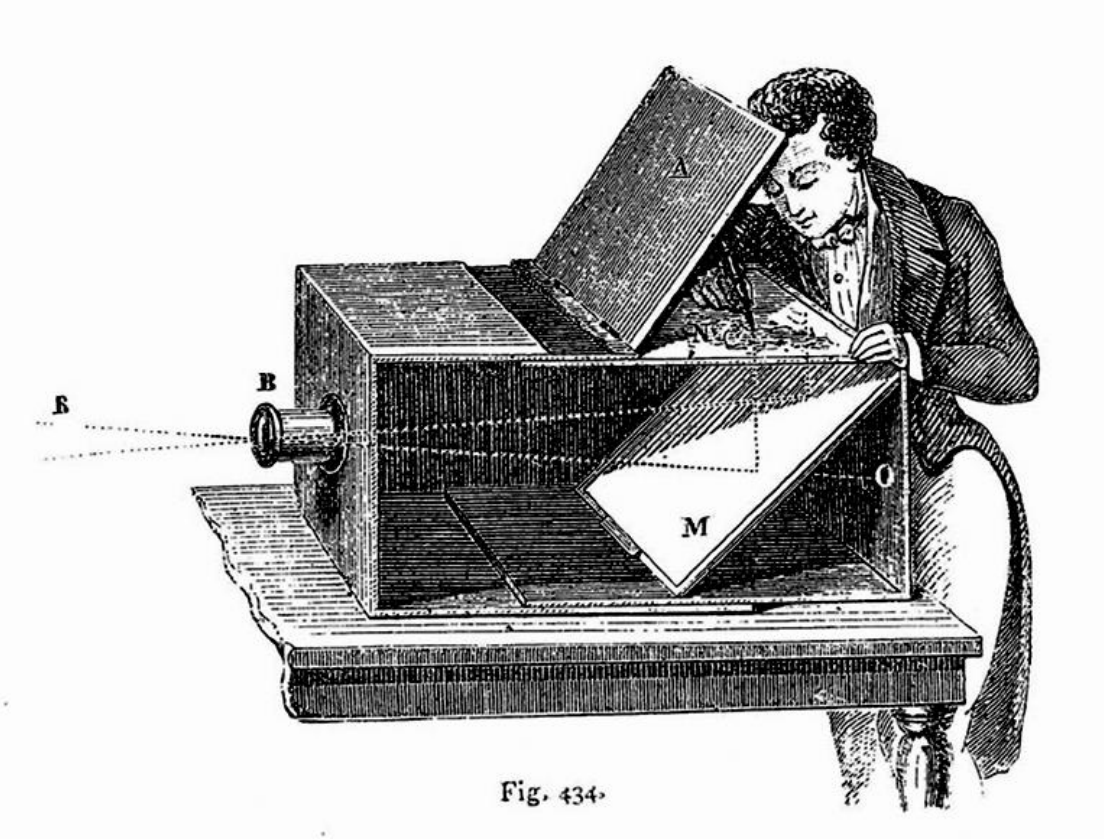
Piero della Francesca,
The Flagellation (c.1469)

Depicting Our World: Toward Perfection



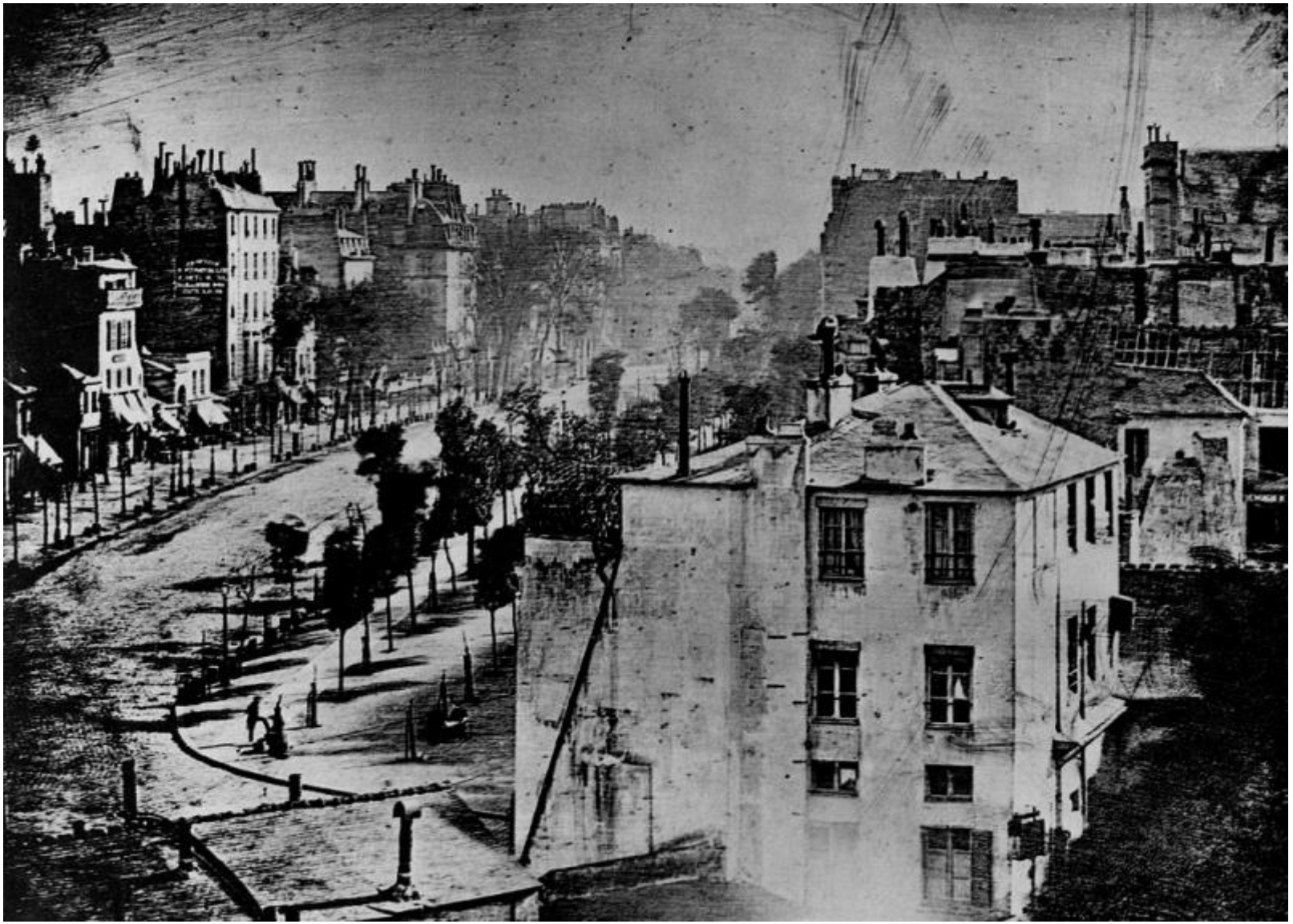
Jan van Eyck, *The Arnolfini Marriage (c.1434)*

Depicting Our World: Toward Perfection



Lens Based Camera Obscura, 1568

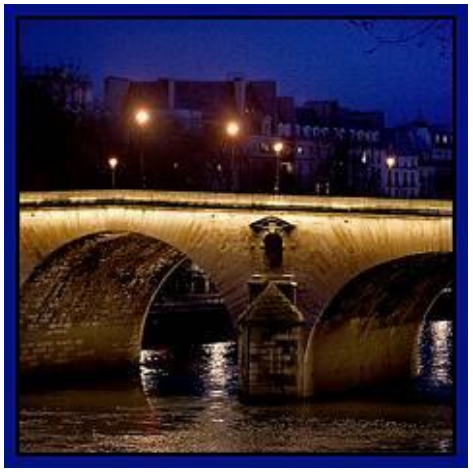
Depicting Our World: Perfection!



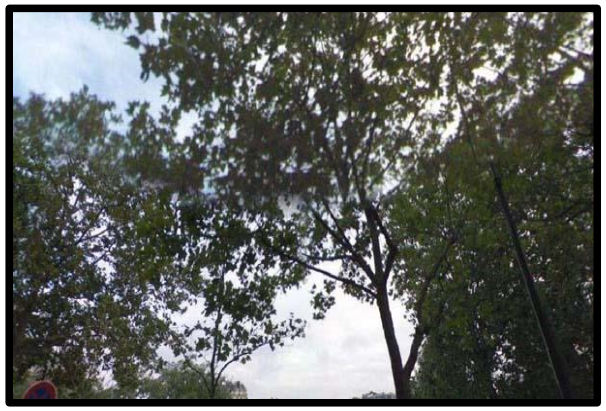
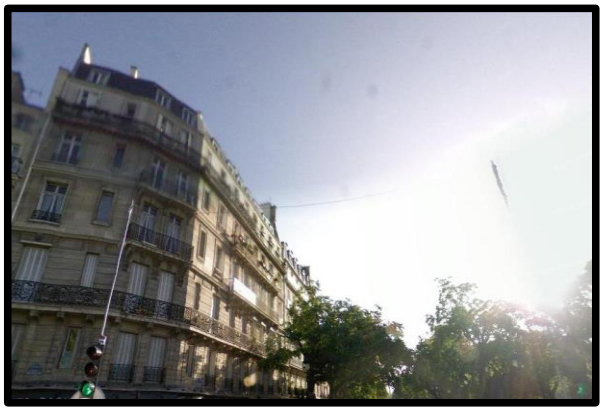
Boulevard du Temple, Louis Daguerre, 1838

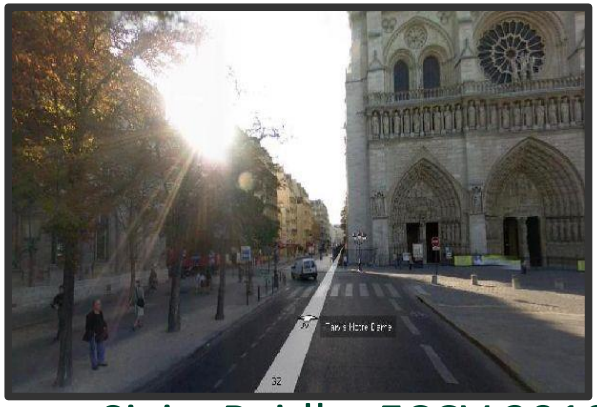
Depicting Our World: Realism?

Paris, according to Flickr



Paris, according to Google StreetView

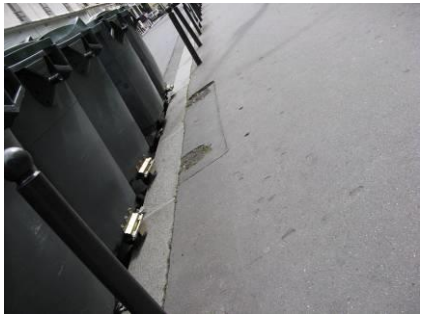



Knopp, Sivic, Pajdla, ECCV 2010

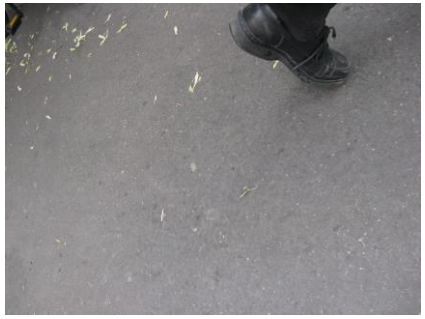
Paris, according to me









After realism…

Monet,
La rue Montorgueil

Depicting Our World: Ongoing Quest

Pablo Picasso

David Hockney

Better than realism?

David Hockney, Place Furstenberg (1985)

Which one is right?

Multiple viewpoints

David Hockney,
Place Furstenberg,
1985

Single viewpoint

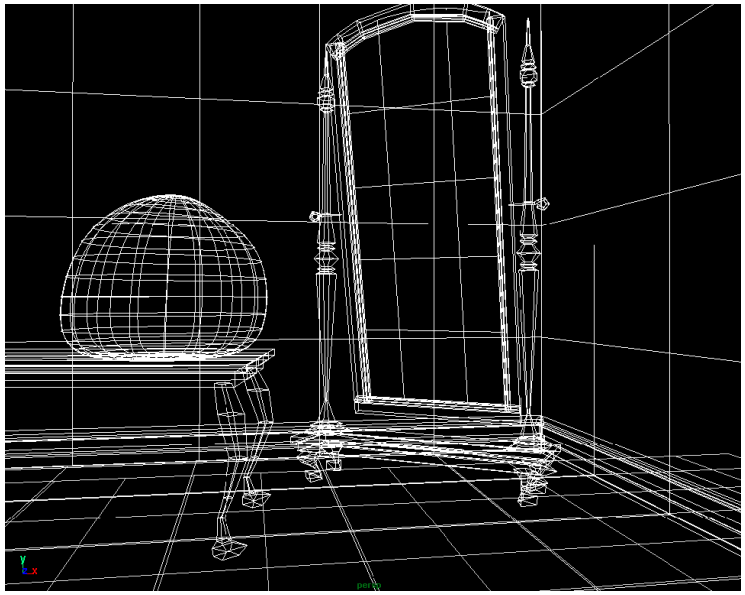
Alyosha Efros
Place Furstenberg,
2009

Depicting Our World: Ongoing Quest

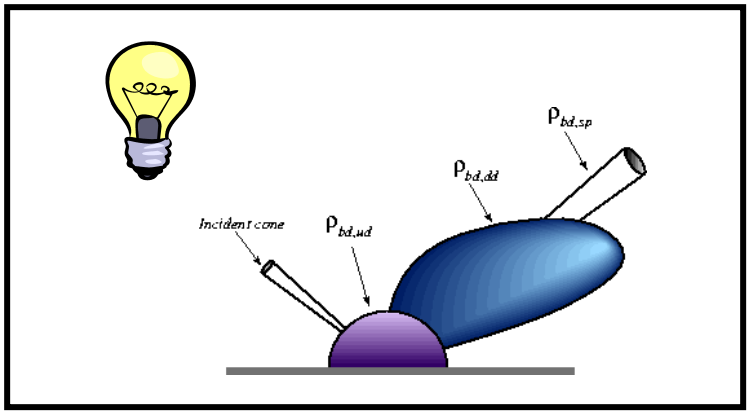
Antonio Torralba & Aude Oliva (2002)

Enter Computer Graphics...

Traditional Computer Graphics



3D geometry



physics

projection

Simulation

GRAPHICS

Modern Computer Graphics

- Amazingly real
- But so sterile, lifeless, *futuristic (why?)*

The richness of our everyday world

Photo by Svetlana Lazebnik

Beauty in complexity

University Parks, Oxford

Which parts are hard to model?

Photo by Svetlana Lazebnik

Creating Realistic Imagery

Computer Graphics

+ great creative possibilities

+ easy to manipulate objects/viewpoint

-Tremendous expertise and effort to obtain realism

Computational Photography

Realism
Manipulation
Ease of capture

Photography

+ instantly realistic

+ easy to aquire

- very hard to manipulate objects/viewpoint