Today

Introductions
Why Computational Photography?
Overview of the course
Administrative stuff
A bit about me

Alexei (Alyosha) Efros

Moved (back) to UC Berkeley in 2013, after decade at CMU
(also work with colleagues in Paris)

Teaching

The plan is to have fun and learn cool things, both you and me!
Social warning: I don’t see well

Research

PhD 2003 on Texture and Action Synthesis

Inspired Photoshop’s “Context-aware Fill” and Microsoft “Smart Erase” buttons:

© Antonio Criminisi
Other works...
im2GPS
(using 6 million GPS-tagged Flickr images)
Amateur Photographer
GSIs

Rachel Albert
- PhD candidate in Vision Science + MS in Computer Science
- Expert in human vision, graphics, and photography

Weilun Sun
- PhD candidate in Computer Science
- Hard-core rendering geek
- [http://sunweilun.github.io/](http://sunweilun.github.io/)
Why Computational Photography?

A Brief History of the Visual Media
Depicting Our World: The Beginning

Prehistoric Painting, Lascaux Cave, France
~ 13,000 -- 15,000 B.C.
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
Still Life, Louis Jaques Mande Daguerre, 1837
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

Single viewpoint

David Hockney, Place Furstenberg, 1985

Alyosha Efros, Place Furstenberg, 2009
Depicting Our World: Ongoing Quest

Enter Computer Graphics...
Traditional Computer Graphics

3D geometry

physics

Simulation

projection

GRAPHICS
Modern Computer Graphics

- Amazingly real
- But so sterile, lifeless, *futuristic* (why?)
The richness of our everyday world
Beauty in complexity

University Parks, Oxford
Which parts are hard to model?
People

From “Final Fantasy”
Faces / Hair

From “Final Fantasy”
Hyper-humans
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 acquire
- - very hard to manipulate objects/viewpoint
Computational Photography

How can I use computational techniques to capture light in new ways?

How can I use computational techniques to breathe new life into the photograph?

How can I use computational techniques to visualize, organize, and navigate the captured visual world?
Relationship to Vision and Graphics

Computer Graphics: Models to Images
Computer Vision: Images to Models
Comp. Photography: Images to Images
Google Photosphere

https://www.youtube.com/watch?v=ZIsRPqcv0Cw
WordLens / Google Translate

Выход в город
Fyuse

https://www.youtube.com/watch?v=9rTjaCcwX6o
Virtual Real World

Campanile Movie
http://www.debevec.org/Campanile/
Course objectives

1. You will have new abilities for visual creation.
Course objectives

2. You will get a foundation in computer vision.
Course objectives

3. You’ll better appreciate your own visual ability.

Is that a queen or a bishop?
Seeing less than you think…
Seeing less than you think…
Video by Antonio Torralba (starring Rob Fergus)
But actually…

Video by Antonio Torralba (starring Rob Fergus)
Course objectives

4. You will get a more intuitive understanding of important mathematical and computational concepts
   - Convolutions, filtering
   - Gradients
   - Change of basis, interpolation, extrapolation, PCA
   - Fourier Transforms
   - Dynamic programming, recursion
   - …
Course objectives

5. You’ll have fun doing cool stuff!
Programming Project #1

Prokudin-Gorskii’s Color Photography (1907)
Programming Project #1
Programming Project #1

• How to compare R,G,B channels?
• No right answer
  • Sum of Squared Differences (SSD):
    \[ ssd(u, v) = \sum_{(x, y) \in N} [I(u + x, v + y) - P(x, y)]^2 \]
  • Normalized Correlation (NCC):
    \[ ncc(u, v) = \frac{\sum_{(x, y) \in N} [I(u + x, v + y) - \bar{I}] [P(x, y) - \bar{P}]}{\sqrt{\sum_{(x, y) \in N} [I(u + x, v + y) - \bar{I}]^2 \sum_{(x, y) \in N} [P(x, y) - \bar{P}]^2}} \]
Project 2: Building a Camera Obscura
Project 3: Fun with frequencies
Project 3: Fun with frequencies

Prof. Christos Papadimalik
Project 4: Gradient Domain Editing

sources/destinations

cloning

seamless cloning
Project 4: Gradient Domain Editing
Project 5: Face morphing and caricatures
Project 5: Face morphing and caricatures

The Beatles

Polyjuice Potion

© Rachel Albert
Project 6: Playing with Lightfields

© Riyaz Faizullahbhoys
Project 7: Automatic Mosaic Stitching

Photo Mosaics

Full screen panoramas (cubic): http://www.panoramas.dk/
Image warping with homographies

image plane in front

black area where no pixel maps to
Project 7(g)

Tour Into the Picture
Paper Pop-up

Step 1: define planes

Step 2: rectify each plane

Step 3: compute 3D box coords
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)
Sample Project
For each project:

Implement stuff from scratch and apply it to your own photos

Every person does their own project (except final projects and camera obscura)

Reporting via web page (plus submit code)

Afterwards, vote for class favorite(s)!

Programming Language:
- Matlab (or Python)
- you can use other languages, but you are on your own
Textbook

http://szeliski.org/Book/
General Comments

Prerequisites

- Linear algebra!!!
- Good programming skills (at least CS61B)
- Some computer graphics, vision, or image processing is useful, but not required.

Emphasis on programming projects!
- Building something from scratch

Graduate Version:
- Need to do more on each project, plus a final paper

“No Screens” Policy:
- No laptops, no cell phones, no smartphones, etc.
Getting help outside of class

Course Web Page
- http://inst.eecs.berkeley.edu/~cs194-26/

Discussion board:
- piazza.com

Office hours
- TBA… see webpage and piazza
Administrative Stuff

Grading
- Programming Project (60%)
- Exam (20%)
- Final Project (20%)
- Class Participation: priceless

Late Policy
- Five emergency late days for semester, to be spent wisely
- Max 10% of full credit afterwards

Cheating
- Let’s not embarrass ourselves
Academic Integrity

Can discuss projects, but don’t share code

Don’t look up code or copy from a friend

If you’re not sure if it’s allowed, ask

Acknowledge any inspirations

If you get stuck, come talk to us
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

• Need time to think, not just hack
  • Creativity is a class requirement

• 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 create pictures, unleash your creative potential
• Interested in grad school? 😊
Questions?