CS 89.15/189.5, Fall 2015

Computational Aspects of Digital Photography

Introduction

Wojciech Jarosz
wojciech.k.jarosz@dartmouth.edu

Dartmouth

CS 89.15/189.5, Fall 2015

COMPUTATIONAL PHOTOGRAPHY

Introduction

Wojciech Jarosz
wojciech.k.jarosz@dartmouth.edu

Dartmouth

The photographic (r)evolution

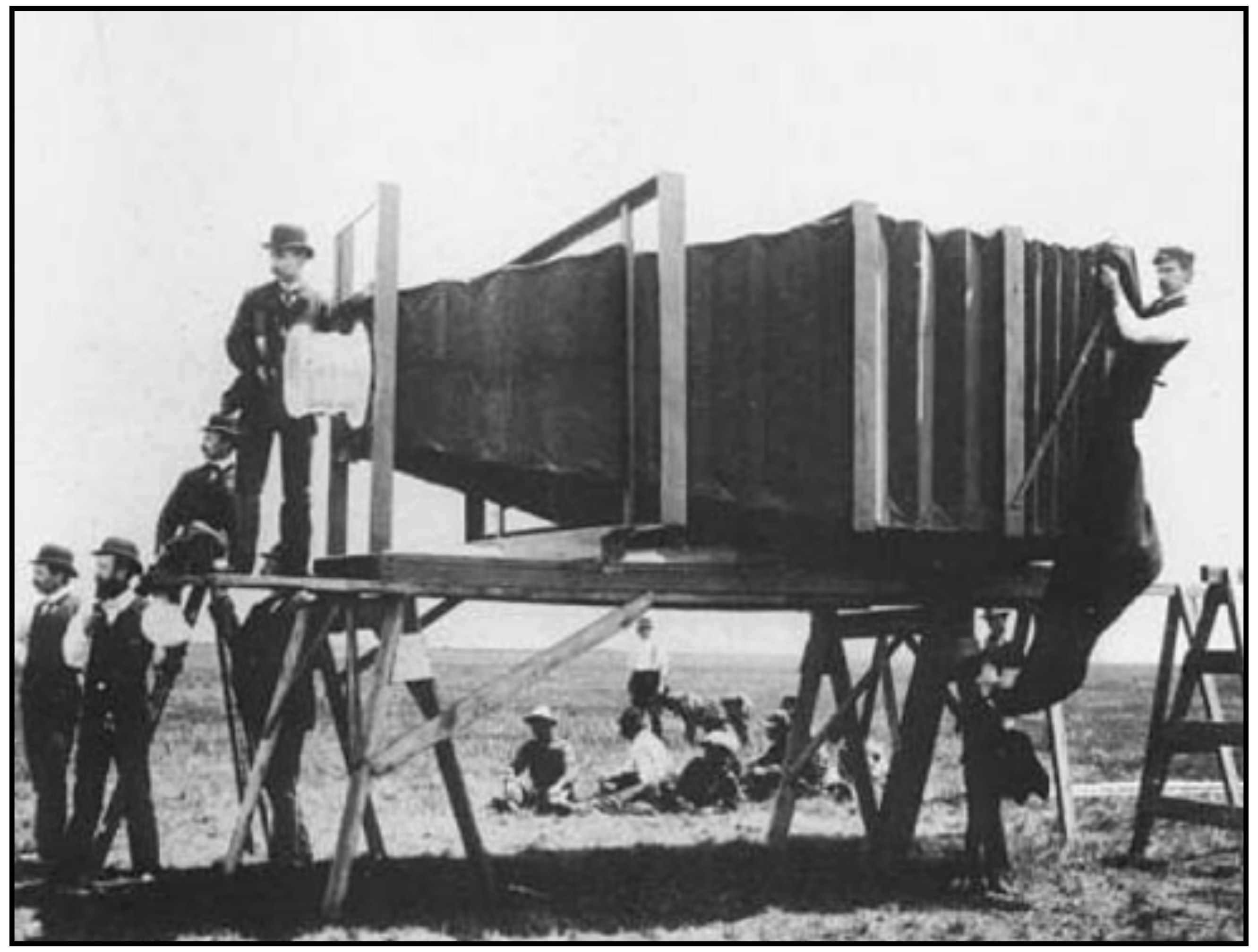
"Measuring light"

Traditional/analog photography:

- optics focus light onto sensor
- chemistry records final image

Digital photography:

- optics focus light onto sensor
- digital sensor records final image



The photographic (r)evolution

Fundamental shift from analog to digital is complete

- digital cameras first outsold film cameras back in 2004
- silicon sensors + digital recording

Today, we (mostly) do what we did with film, but digitally:

- store & transmit images
- share photos as stacks of images
- image processing that replicates darkroom techniques

Tomorrow: what is possible with lots of computation?

Computational photography

More than just digital photography

Arbitrary **computation** between light measurement and final image

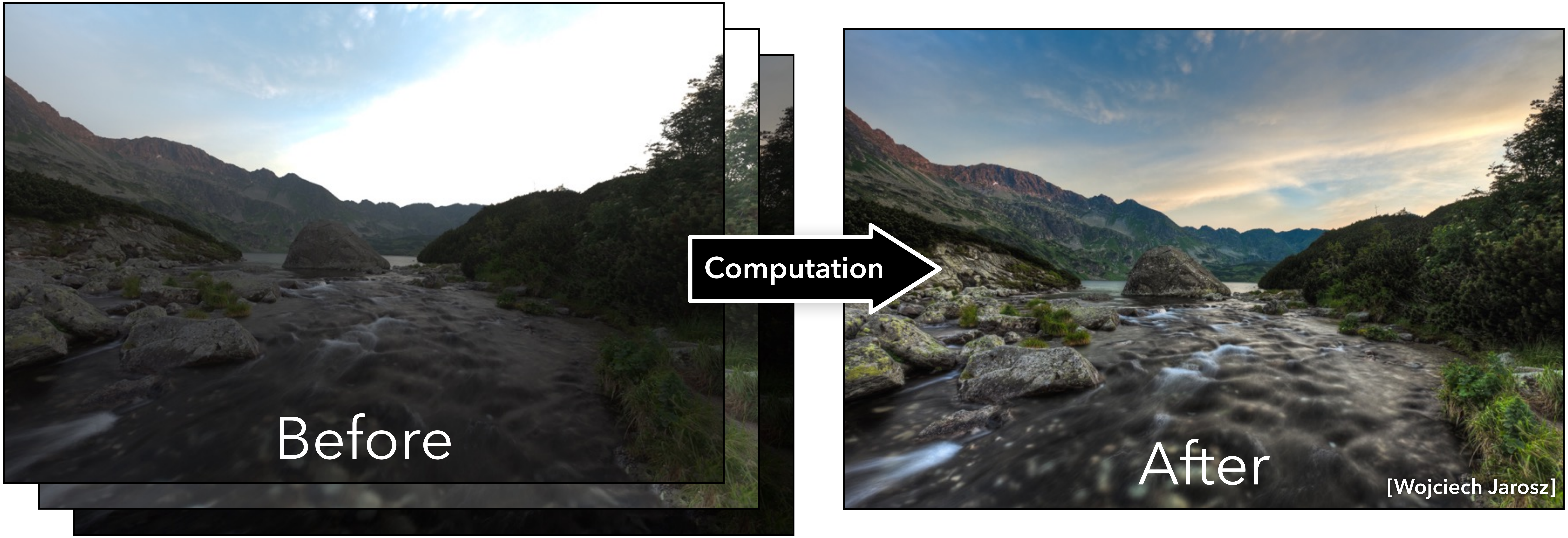
- Light measured on sensor is not the final image
- Computation to enhance and extend capabilities of digital photography

Two types of computation:

1. Post-process after traditional imaging
2. Design new imaging architecture together with computation

Removing sensor/display limitations

High dynamic range images & tone mapping



Removing imaging artifacts

Denoising with detail transfer

Flash + No Flash → Output

Removing imaging artifacts

Denoising & deblurring

Blurry

+

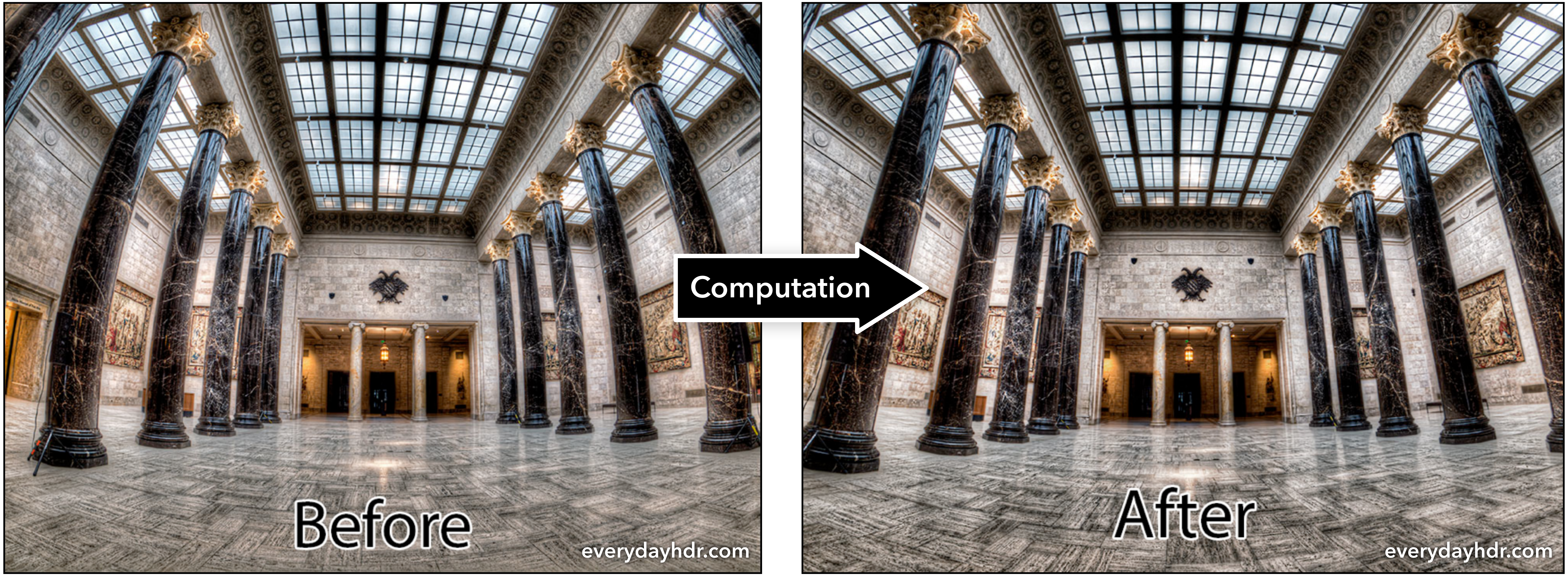
Noisy

Computation

Output

Removing lens limitations

Do lenses have to get everything right?



Removing lens limitations

Do I really need a fish-eye lens?

Removing lens limitations

Do I really need a fish-eye lens?

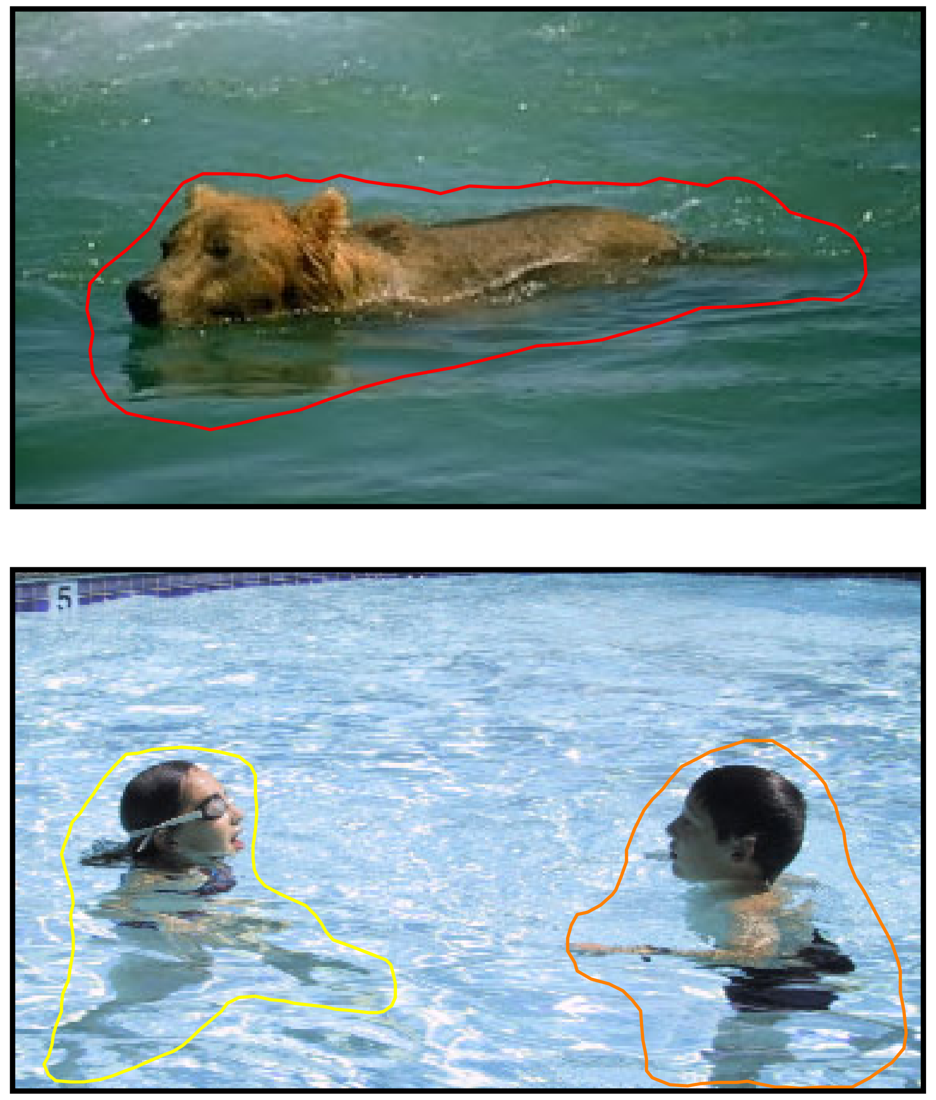
Removing lens limitations

Removing lens limitations


[Wojciech Jarosz]

Advanced image editing tools

Do I really need to put a bear in a swimming pool with my kids?



sources

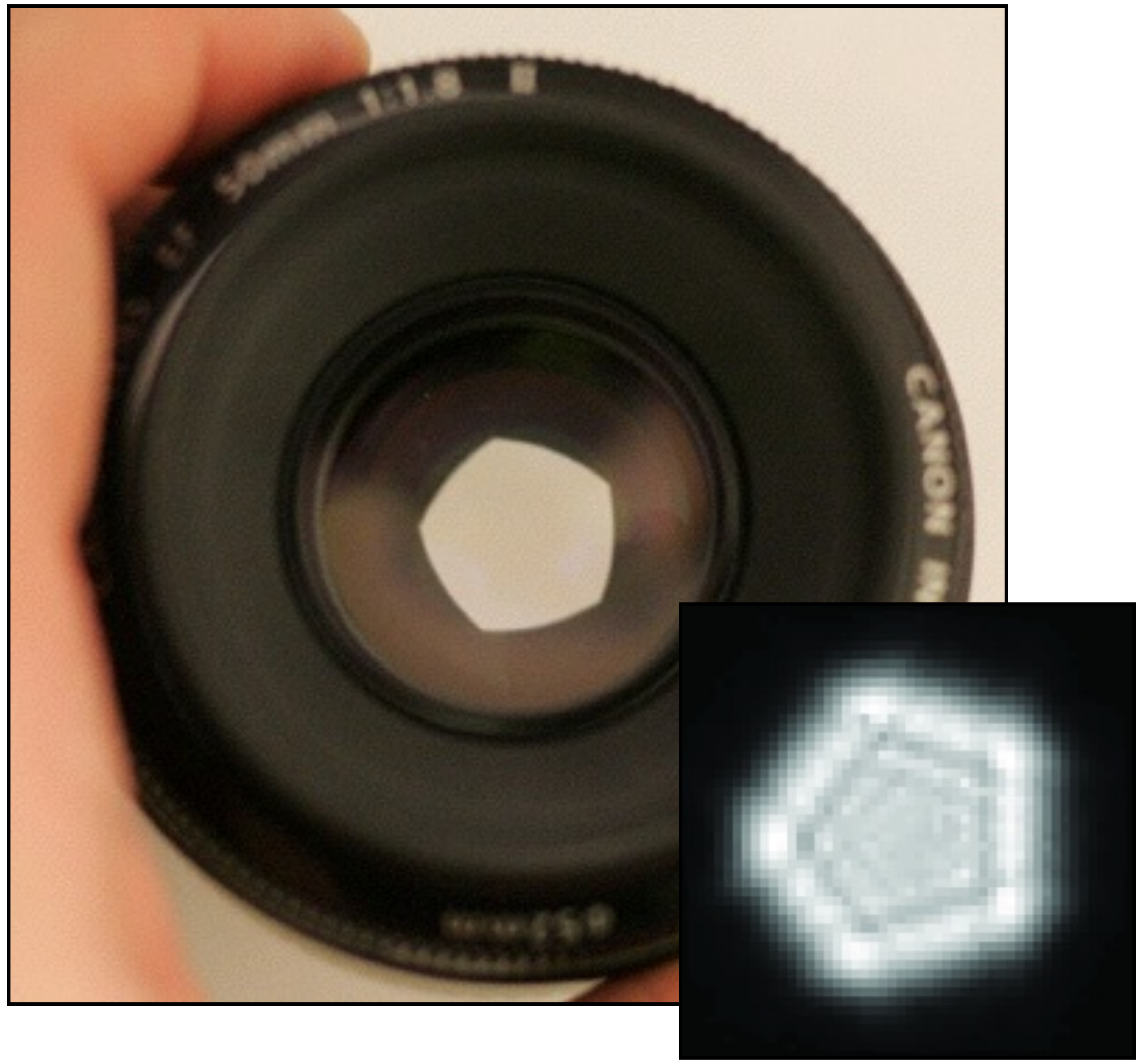
destinations

cloning

seamless cloning

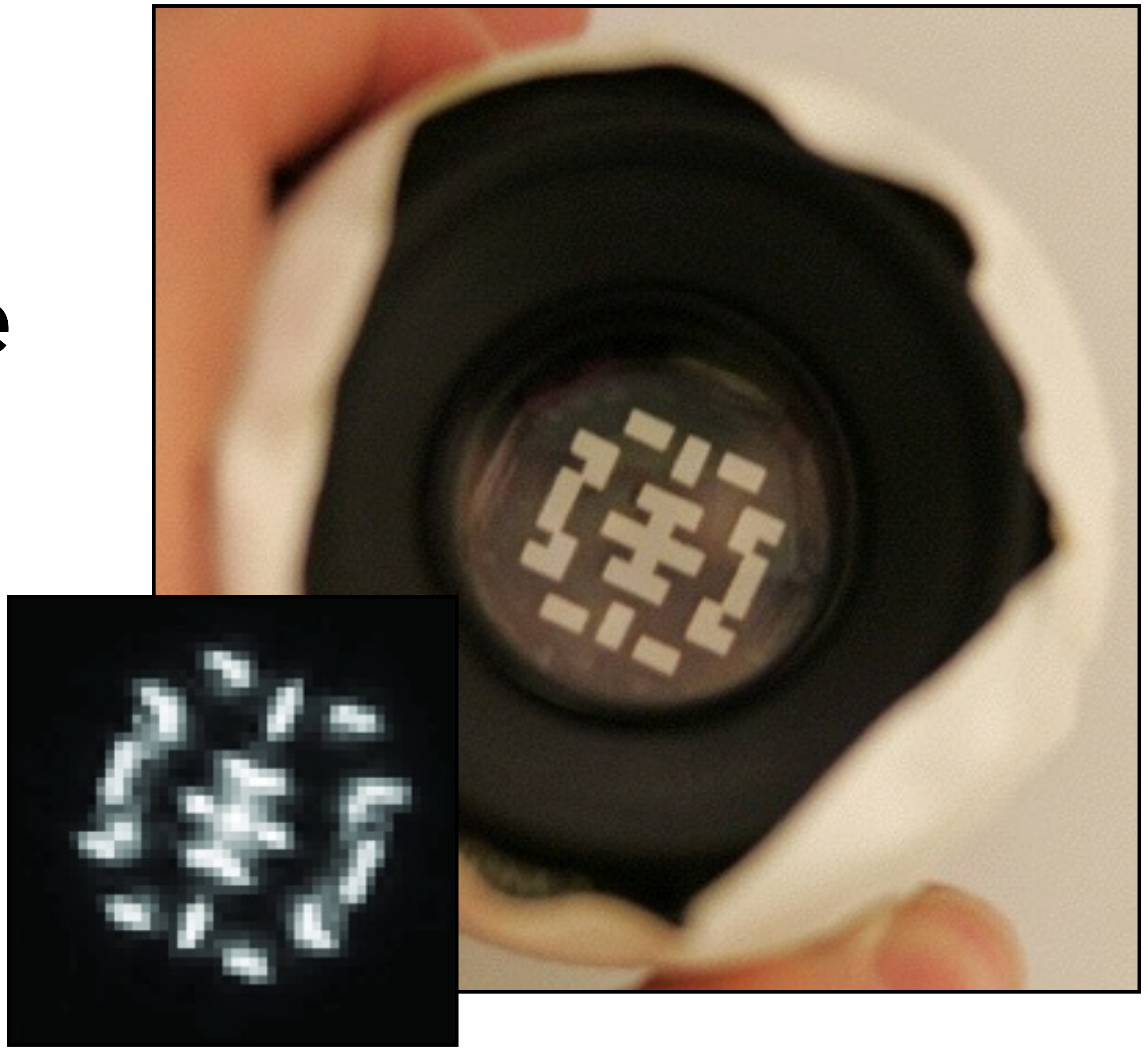
Computational optics

modify lens so you can recover depth & refocus?



lens aperture shape

point spread function



Today

Course administration

Course topics

Programming Assignment 0

- Image formation & representation
- C++ refresher

History of photo technology (if there is time)

Course administration

Instructor: Prof. Wojciech Jarosz

- email: wojciech.k.jarosz@dartmouth.edu
- www: www.cs.dartmouth.edu/~wjarosz
- office hours: TBA, Sudikoff 210 (temporarily)

TA: Rawan Ghofaili

- email: rawan.al.ghofaili.gr@dartmouth.edu
- office hours: TBA

Course administration

Lecture

- Tuesdays & Thursdays, 2:00pm–3:50p
- Sudikoff, Room 214

X-hour

- Wednesday, 4:15pm–5:05pm
- Sudikoff, Room 214
- may sometimes use x-hours to make up missed lectures

Course administration

Class website (www.cs.dartmouth.edu/~wjarosz/courses/cs89-fa15/)

- Syllabus, lecture slides, programming assignments, etc.

Canvas (linked from above)

- primarily for base code and turning in assignments
- register with your full @dartmouth.edu address

Piazza (linked from above)

- for class discussion, asking questions, getting help
- I won't answer technical questions by email
- can be anonymous if you're shy

Required material/equipment

No textbook required

- will post lectures slides
- lots of resources online + links to articles in slides & website

You will need to take some photos

- any digital camera with manual control over shutter speed+ISO (ideally also aperture)
- a recent smartphones with appropriate camera app will do
- no need for a fancy SLR (but it sure is fun!)

Prerequisites

Good programming experience (we will use C++)

- COSC 10 (Java) required
- COSC 77 (C++) and COSC 50 (C) recommended

Some linear algebra (matrix calculations, linear systems of equations, least squares problems)

Coursework & grading (tentative)

60%: Weekly assignments (mostly programming in C++)

25%: Final project

15%: Paper reading, participation, and presentation

Graduate/Extra Credit

- Some assignments will include extra work
- Required for CS 189, extra credit for CS 89
- Though, in general, I'll simply grade grads more strictly

Late submissions & regrading

Assignments will have a **strict deadline** (typically 9pm on Wednesdays)

- I mean it: you get zero if you're 5 minutes late
- upload to Canvas
- special circumstances: ask one week in advance

Regrade request by email within 1 week of grade

Collaboration & academic integrity

You are welcome and encouraged to chat about assignments

All code must be written on your own!

- Don't leave your code on shared computers

Read the full policy on the class website

Assignment turn in (through Canvas)

ZIP file with:

- **readme.txt or webpage**
  - how long it took
  - potential issues with your solution and explanation of partial completion (for partial credit)
  - collaboration acknowledgement (but again, you must write your own code!)
  - what was most unclear/difficult
  - what was most exciting
- **Source code (always!)**
- **Image results (most of the time)**

Programming & lab

We'll be programming in C++

You can develop on whatever platform you want, but…

I must be able to compile/run your code on Mac (preferable) or Linux

- iMac lab available in Sudikoff 003 & 005
- ssh into Linux machines, see available machines here: www.cs.dartmouth.edu/~wbc/suditour/011
    - who needs an account? email me.
- **Don't leave your code on public/shared machines!**

Final project

Similar in style to weekly programming assignments, but should be roughly 3× larger in scope

We can suggest some projects, or you can design your own

Paper reading and presentation

We will read recent research papers on comp. photo.

You will present a research paper

We will discuss the papers together

Questions?

Introductions

Who are you?

What is your experience with photography?

Why did you sign up for the class?

To help me remember your names…

Go on Canvas and record yourself saying your name

- by **Monday, Sep 21**

Computational photography

Topics of this class

- Role of computation, algorithms in digital photography today
- Algorithms to extend and improve capabilities of digital photography in the future

What this class *is not* about

This is not a photography art class!

- little on history of art, photographers
- check CS 29/129 next term, or classes in Studio Art

Not a class about how to use Photoshop/Lightroom

- but how to implement its coolest features!

No medical imaging, tomography, microscopy, radar

No image processing for scientific applications (physics, biology, etc.)

Little on hardware

What this class *is* about

Technical basics of photography, light, and color

Software aspects of computational photography

- a bit on hardware, lens technology, optics

Emphasis on applications in consumer domain

- HDR photography, RAW processing, panoramas, morphing…

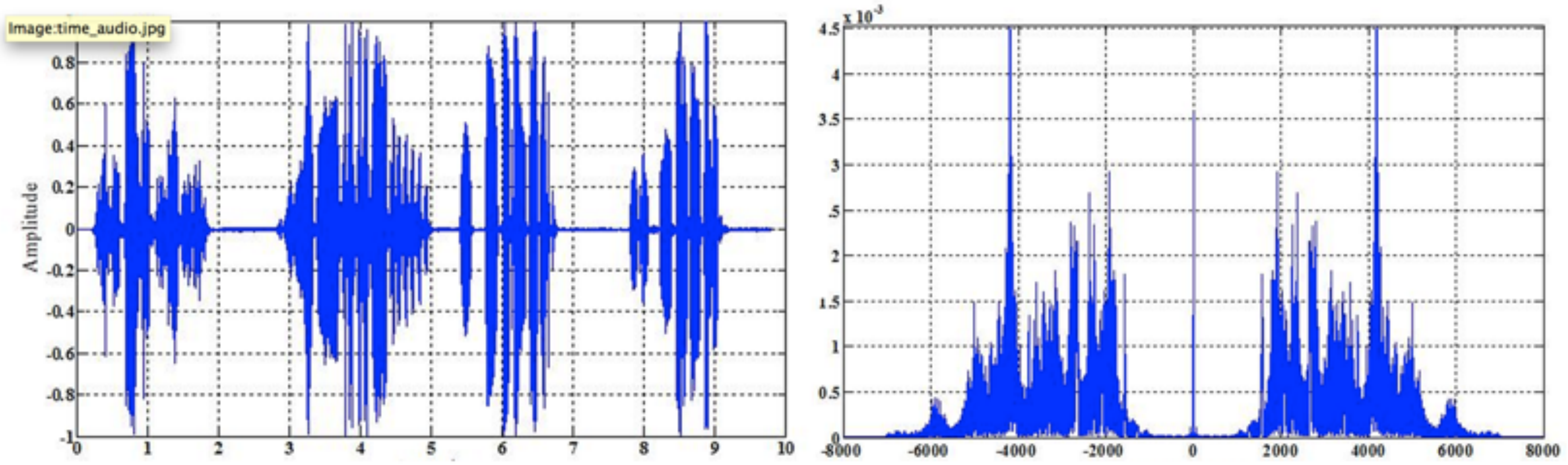
Cool and creative applications of mathematical tools

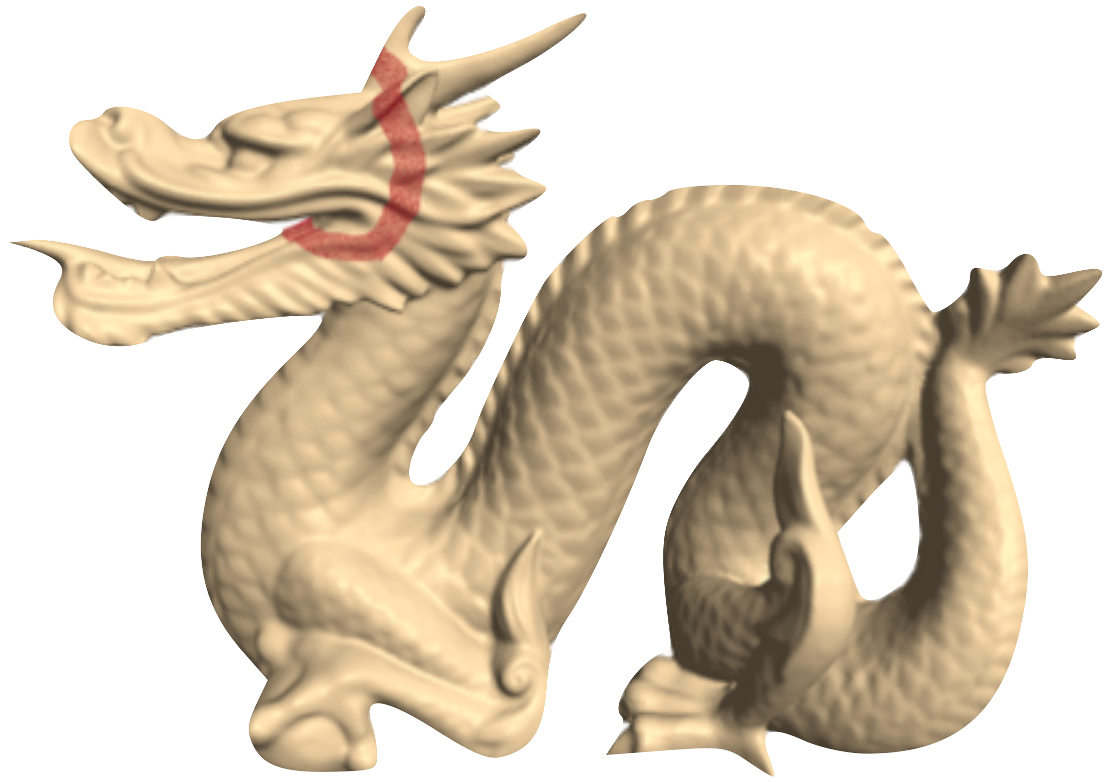
- Linear and non-linear filtering, numerical optimization techniques, probabilistic models…

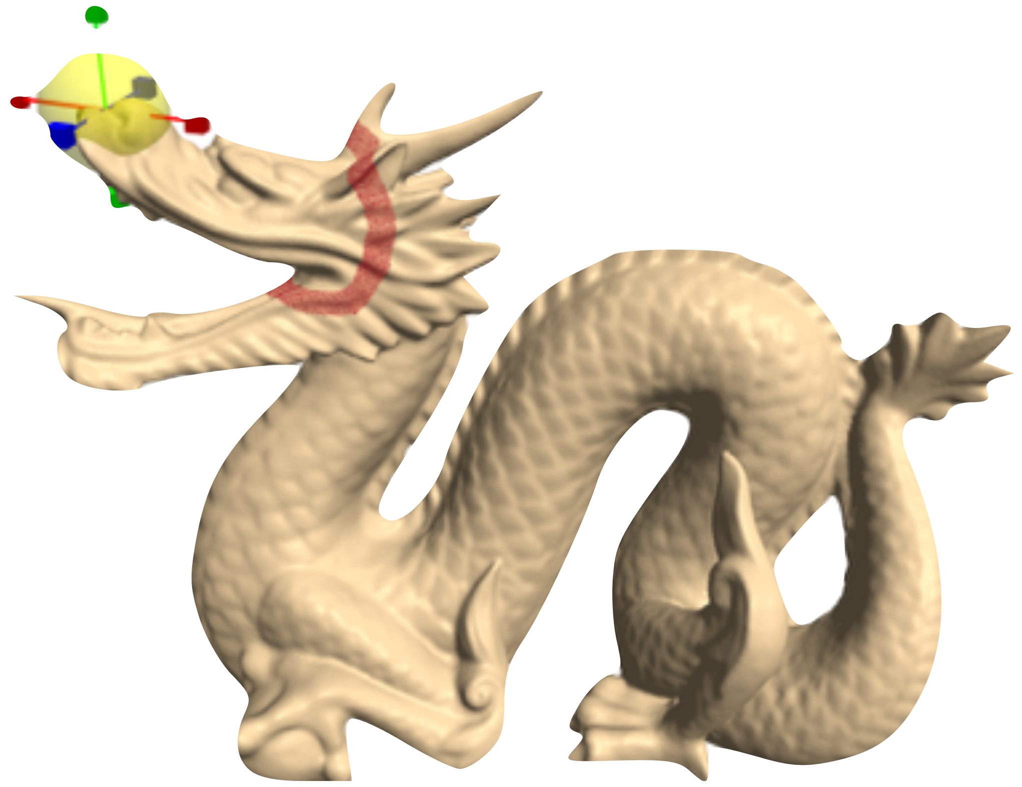
Beyond photography

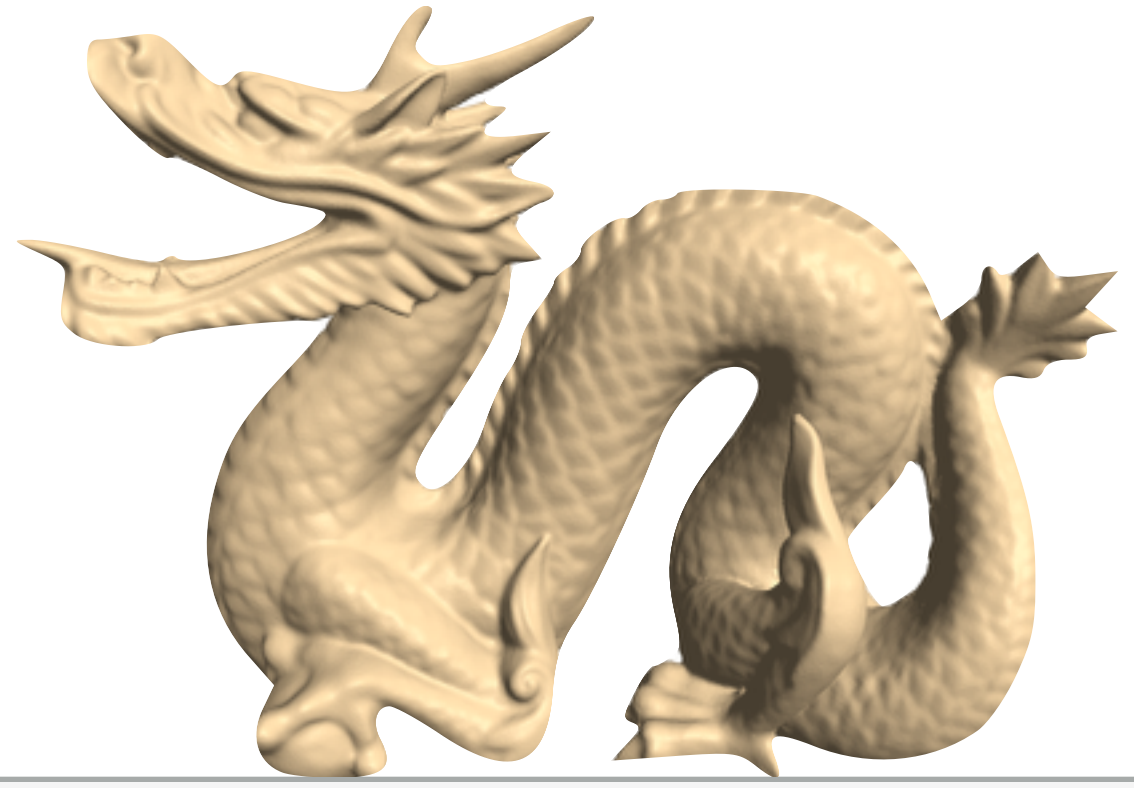
Concepts apply to other domains/types of data:

- audio/speech, motion, geometry







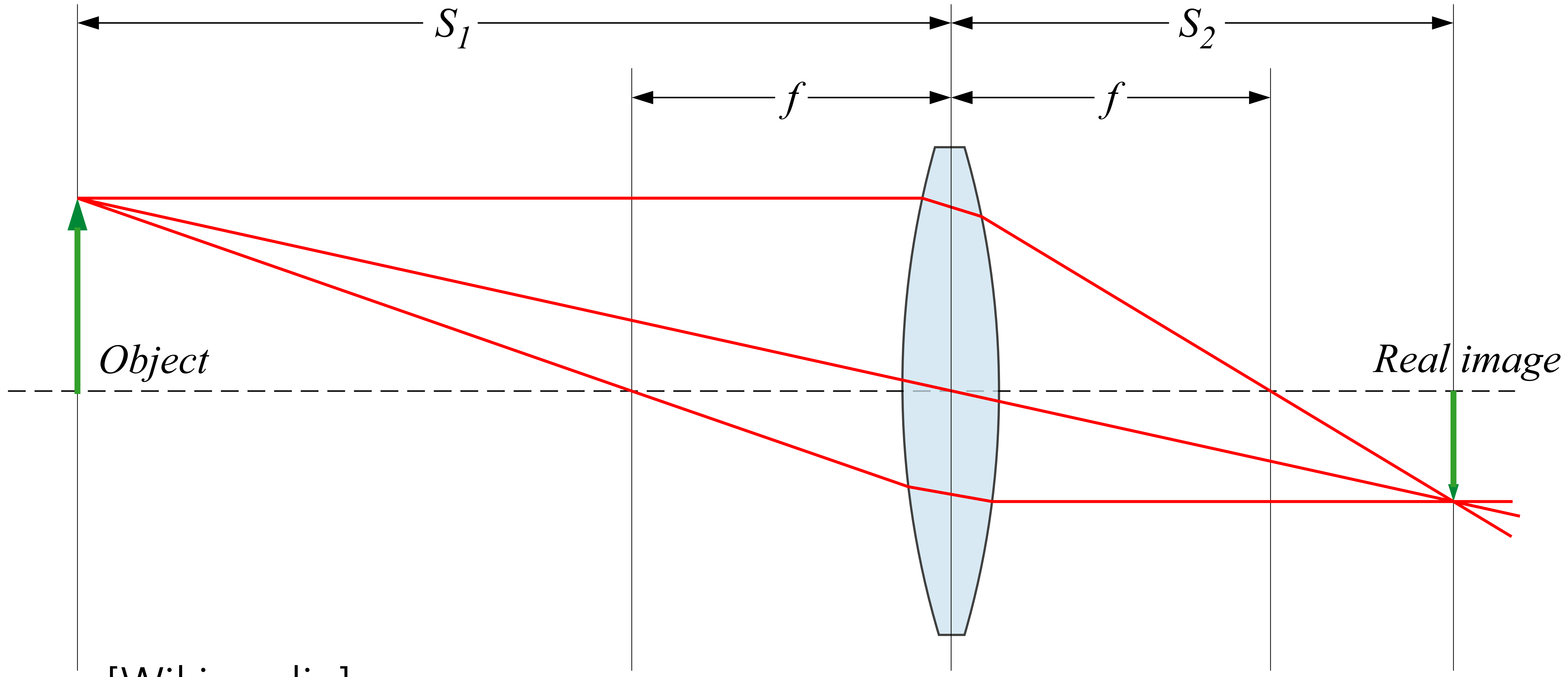


Dragon images from [Sorkin et al. 2004]

Syllabus (tentative)

Syllabus

How does a conventional camera+lens work?



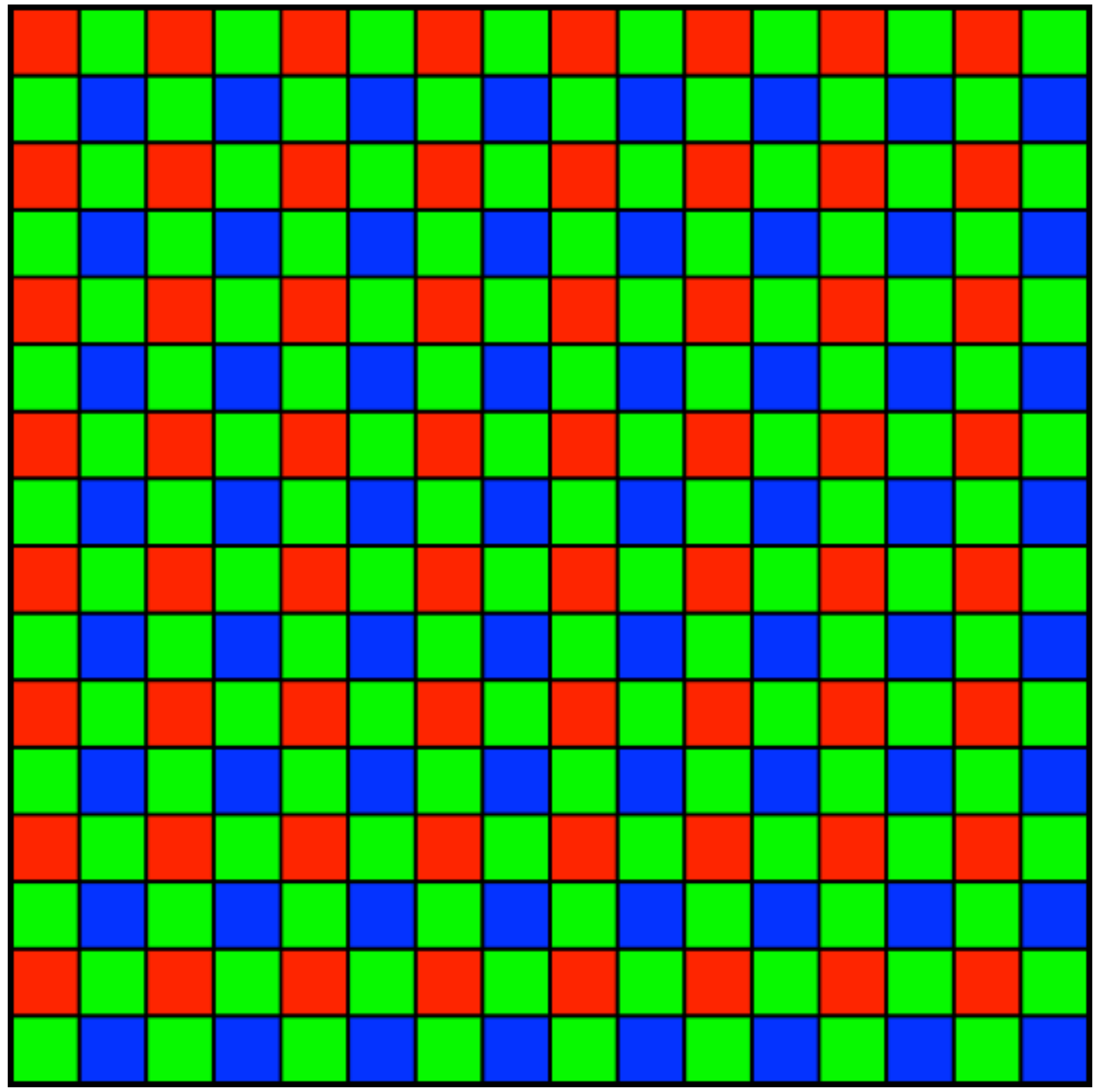
[Wikimedia]

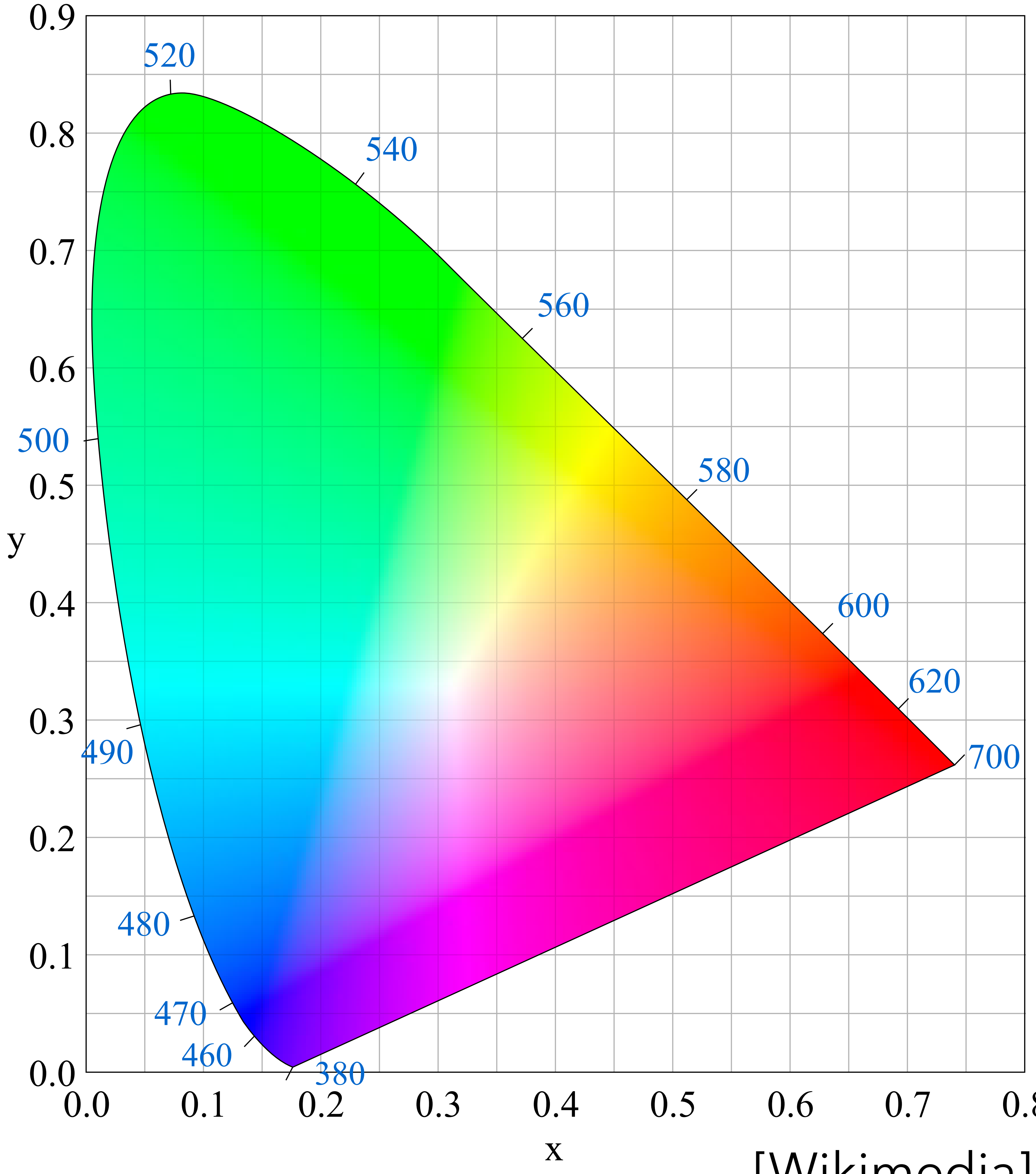
Syllabus

Color & color perception

How do cameras capture color?

Demosaicing





[Wikimedia]

Syllabus

How can we capture the whole intensity range of a scene?

- high dynamic range imaging

How do we display that on screen?

- tone mapping

Images from debevec.org

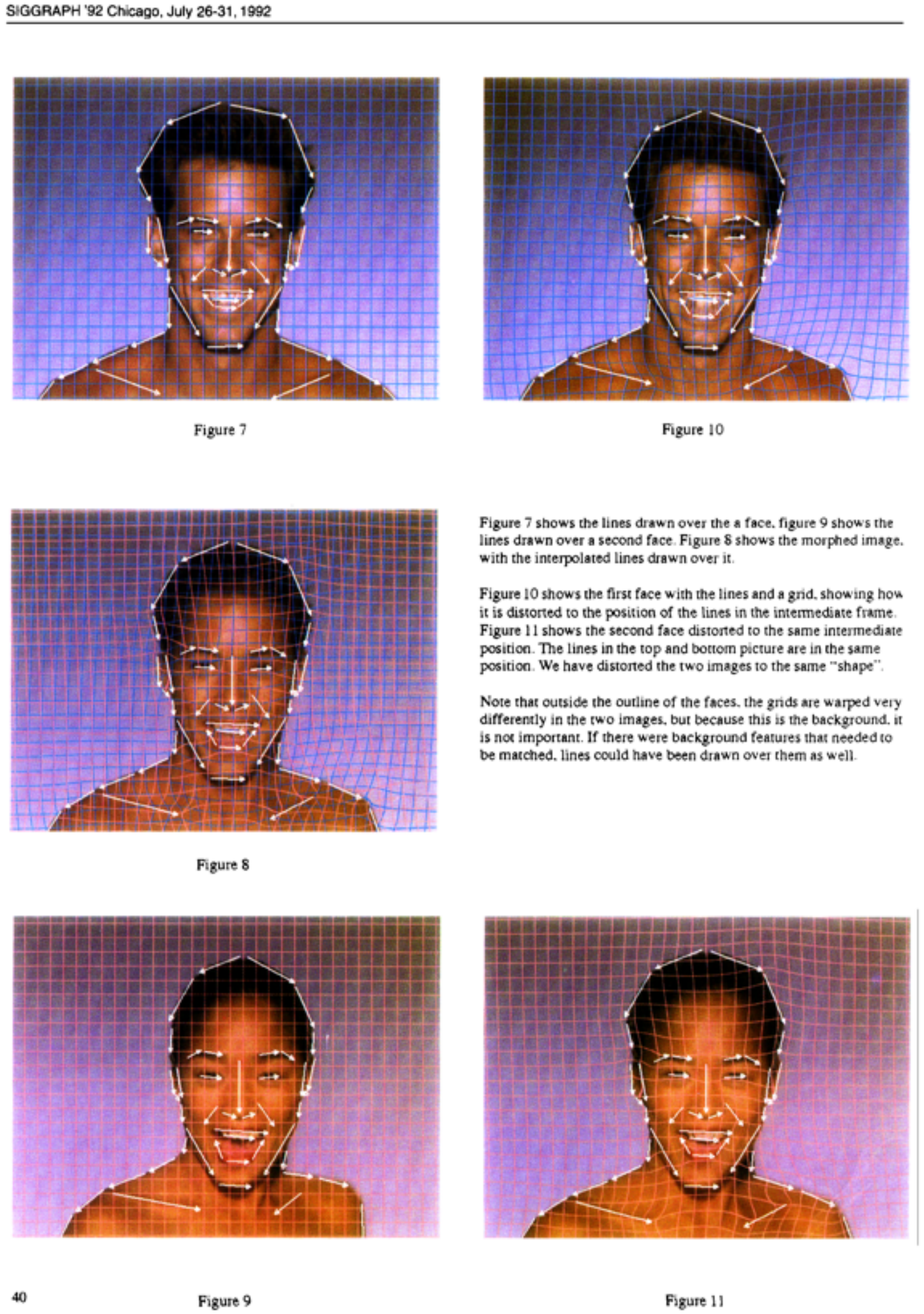
Syllabus

Panoramic imaging, automatic alignment, stitching

Syllabus

Warping the contents of an image

Morphing one image to another



SIGGRAPH '92 Chicago, July 26-31, 1992

Figure 7

Figure 10

Figure 7 shows the lines drawn over the a face. figure 9 shows the lines drawn over a second face. Figure 8 shows the morphed image, with the interpolated lines drawn over it.

Figure 10 shows the first face with the lines and a grid, showing how it is distorted to the position of the lines in the intermediate frame. Figure 11 shows the second face distorted to the same intermediate position. The lines in the top and bottom picture are in the same position. We have distorted the two images to the same "shape".

Note that outside the outline of the faces, the grids are warped very differently in the two images, but because this is the background, it is not important. If there were background features that needed to be matched, lines could have been drawn over them as well.

Figure 8

40

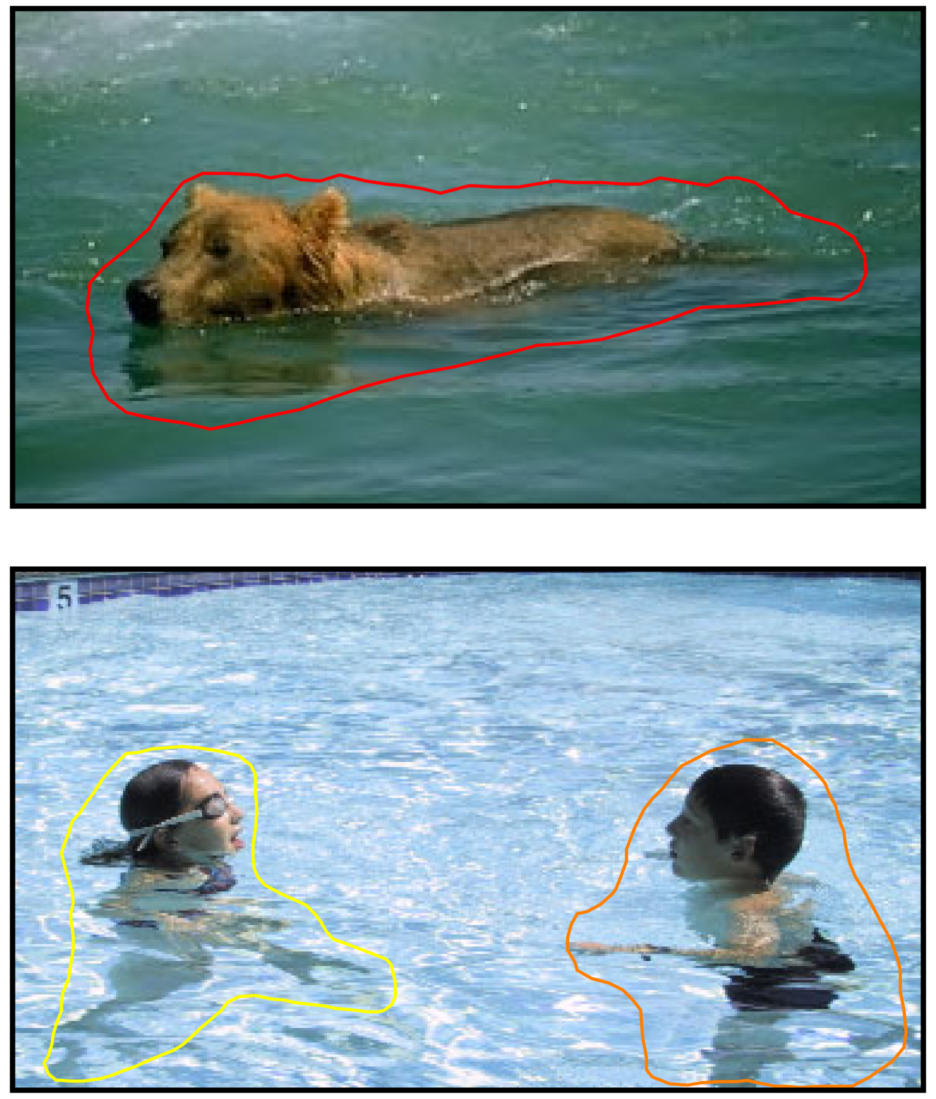
Figure 9

Figure 11

Syllabus

Gradient-domain manipulation

Optimization-based manipulation



sources

destinations

cloning

seamless cloning

Images from [Pérez et al. 2003]

Syllabus

Modifying cameras and capturing more information

Stanford Multi-Camera Array

Assignments (tentative)

Basics

Brightness, constrast, black & white

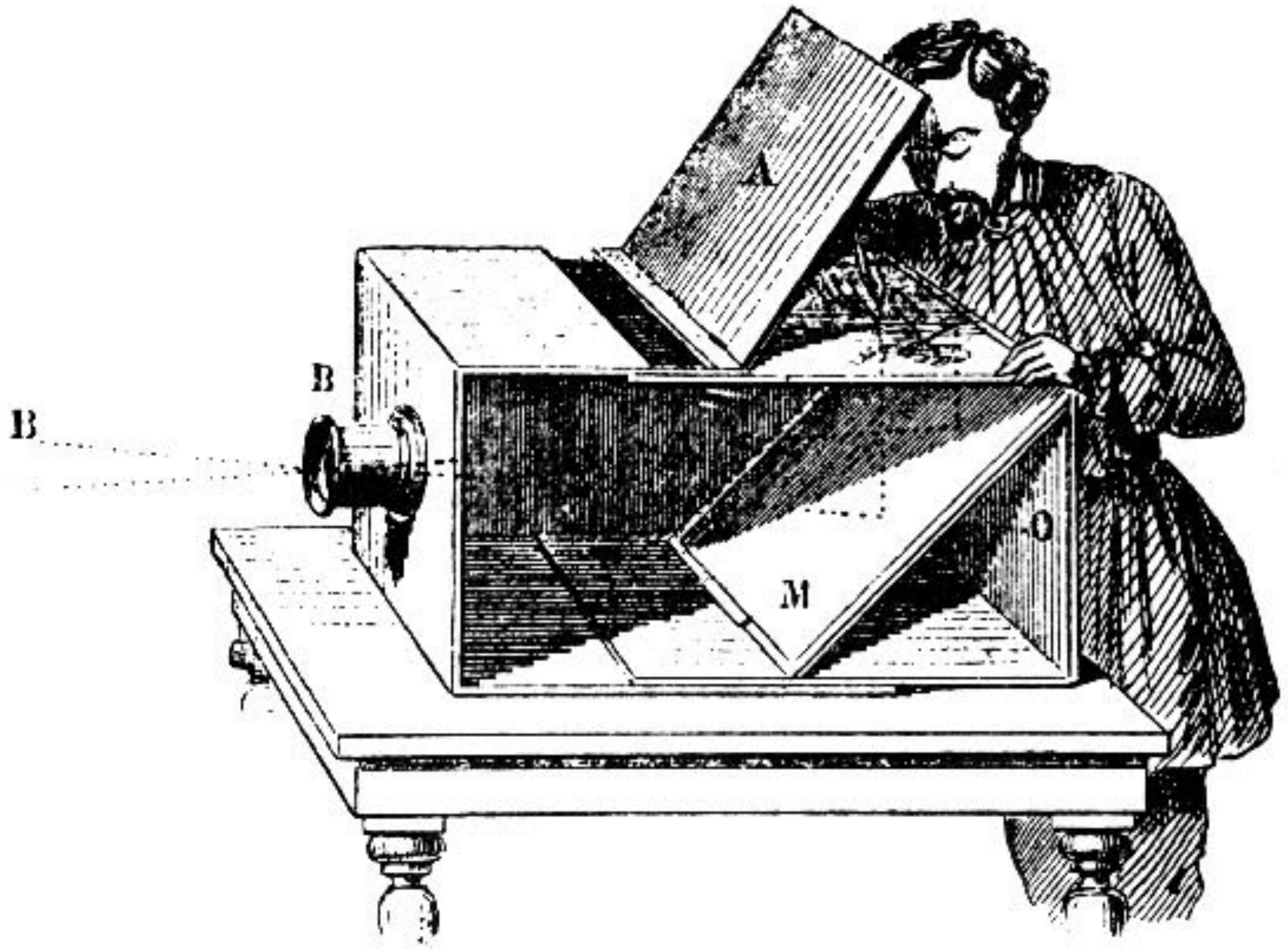
Color spaces

Spanish Castle illusion

Histograms & histogram matching

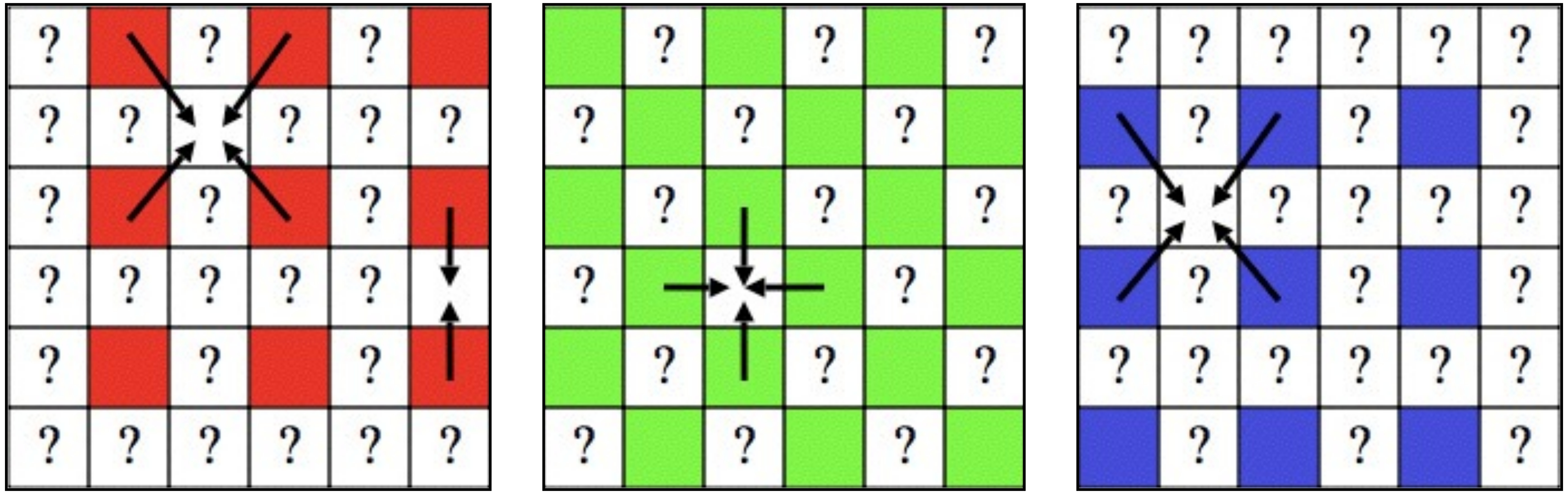
Analog Instagram filter

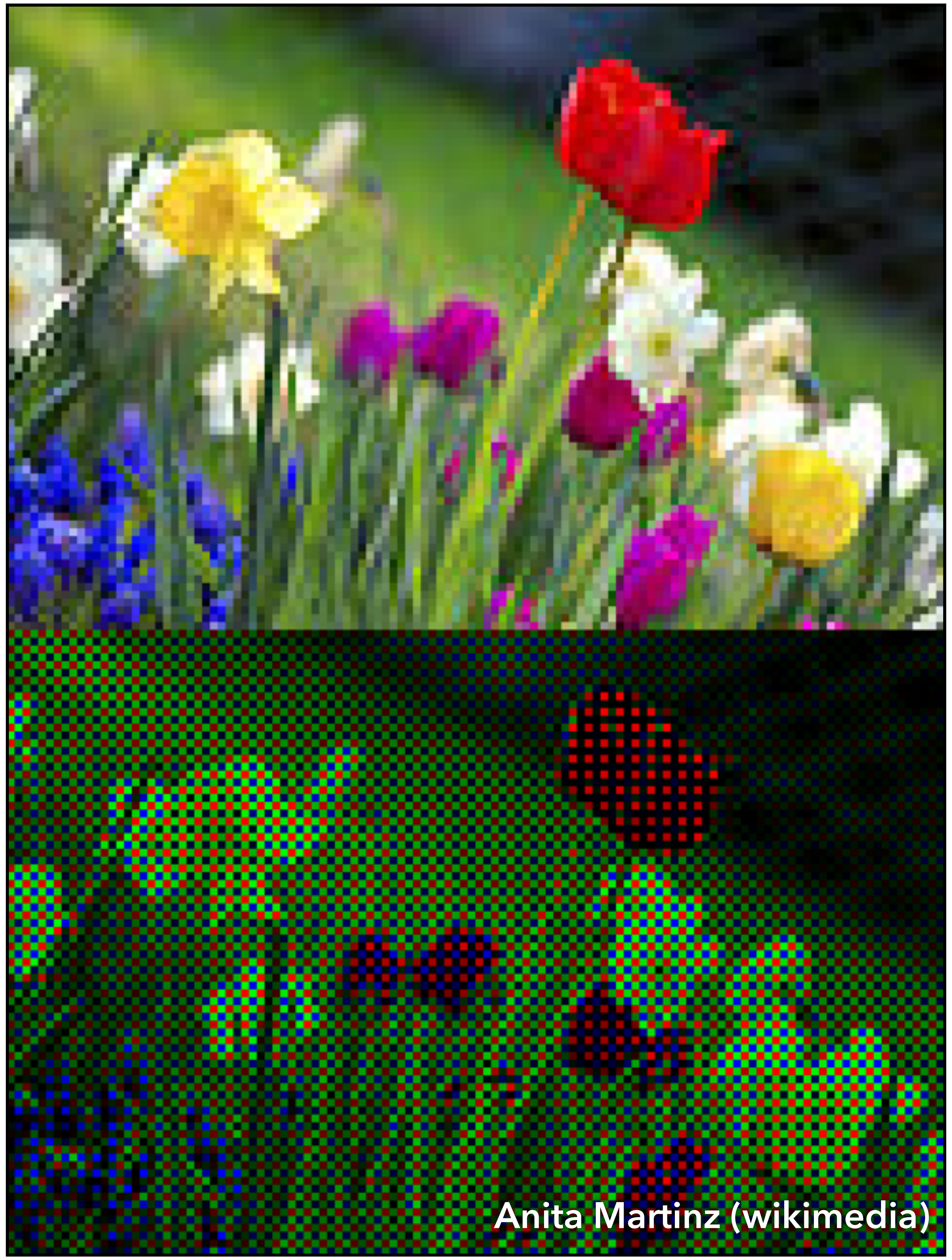
Build your own pinhole camera



Demosaicing

Reconstruct full color image from RAW mosaiced sensor data





Convolution & denoising

Blur, unsharp mask

Denoising with the bilateral filter

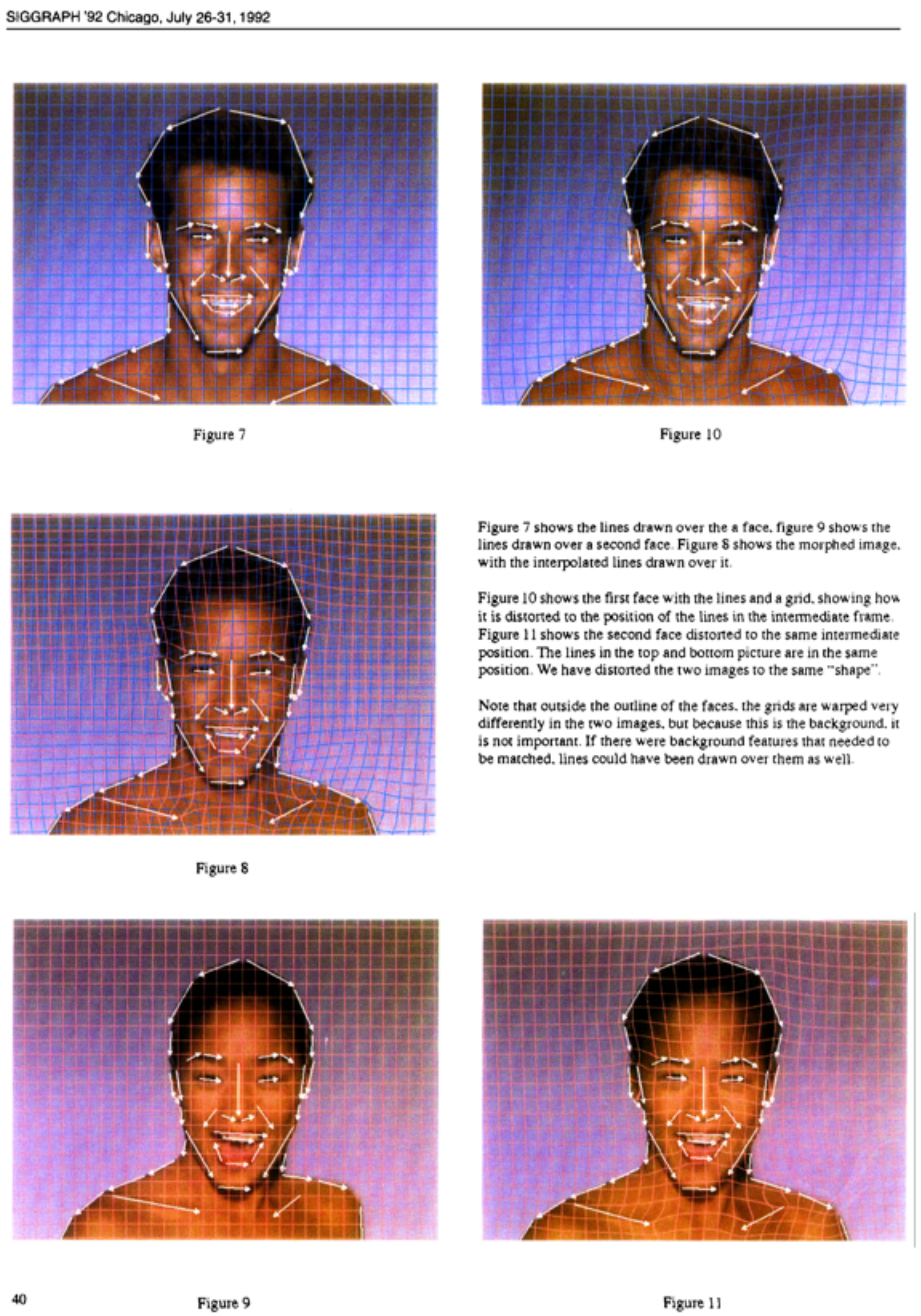
HDR imaging & tone mapping

merge multiple exposures for greater intensity range

Resampling, warping & morphing

Image rescaling & warping

Morphing from one face to another



SIGGRAPH '92 Chicago, July 26-31, 1992

Figure 7

Figure 10

Figure 7 shows the lines drawn over the a face. figure 9 shows the lines drawn over a second face. Figure 8 shows the morphed image, with the interpolated lines drawn over it.

Figure 10 shows the first face with the lines and a grid, showing how it is distorted to the position of the lines in the intermediate frame. Figure 11 shows the second face distorted to the same intermediate position. The lines in the top and bottom picture are in the same position. We have distorted the two images to the same "shape".

Note that outside the outline of the faces, the grids are warped very differently in the two images, but because this is the background, it is not important. If there were background features that needed to be matched, lines could have been drawn over them as well.

Figure 8

40

Figure 9

Figure 11

Final project

A project of your choosing, or, some pre-defined suggestions

Immediate TODOs

If you believe you'll use Linux servers, email me within 24 hours:

- dartmouth email address
- two desired usernames

Go on Canvas and record an intro by **Monday, Sep 21**

First programming assignment due **Tuesday, Sep 22**

Next…

Programming assignment 0

Slide credits

Frédo Durand

Matthias Zwicker

Steve Marschner