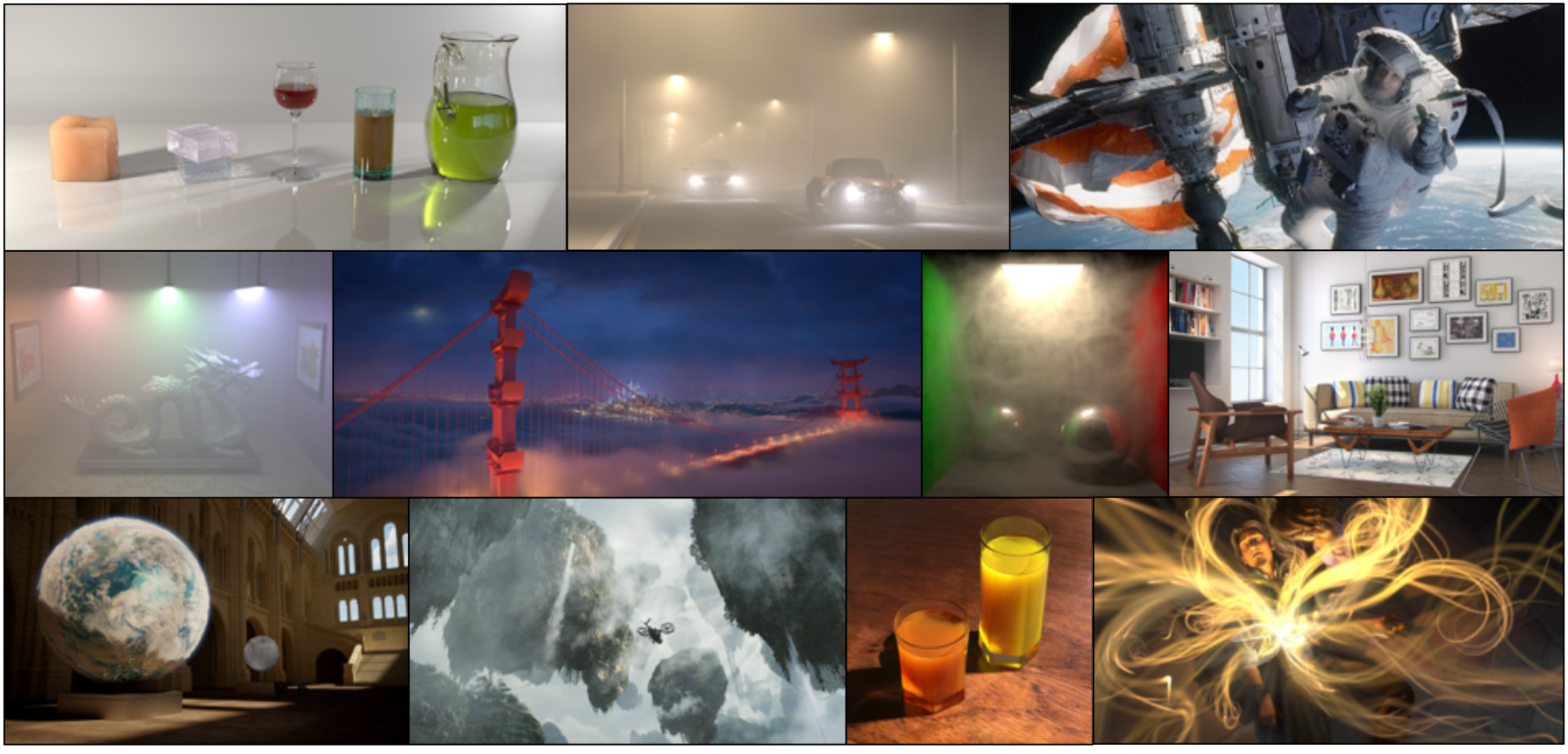
CS 87/187, Spring 2016

RENDERING ALGORITHMS



Prof. Wojciech Jarosz

wojciech.k.jarosz@dartmouth.edu

Dartmouth

Course Staff

Professor

Wojciech Jarosz

wojciech.k.jarosz@dartmouth.edu

Teaching Assistant

Srinath Ravichandran

srinath.ravichandran.gr@dartmouth.edu

Course Details

Time and location

- Lecture:
  - Tuesdays & Thursdays, 14:00 -15:50
    Dartmouth Hall 217 (**this may change**)
- X-hour:
  - Wednesdays, 16:15 - 17:05
    Dartmouth Hall 217 (**this may change**)

Website

- Canvas: https://canvas.dartmouth.edu/courses/13856
- We will also use Piazza for communication

Rendering

The process of converting a description of a 3D scene into an image

- **Non-Photorealistic Rendering (NPR)**

Rendering

The process of converting a description of a 3D scene into an image

- **Photorealistic Rendering**: create an image of a 3D scene that is indistinguishable from a photo
- **Physically based Rendering**: *simulate* the physical behavior of light as closely as possible to *predict* what would be observed

www.pbrt.org/gallery




Relation to curriculum

Digital Arts

Relation to curriculum

Visual Computing

- Lots of programming
- Learn and implement the techniques *behind* the tools
- Can leverage tools, but won't teach them
- Essential skills for R&D

Digital Arts

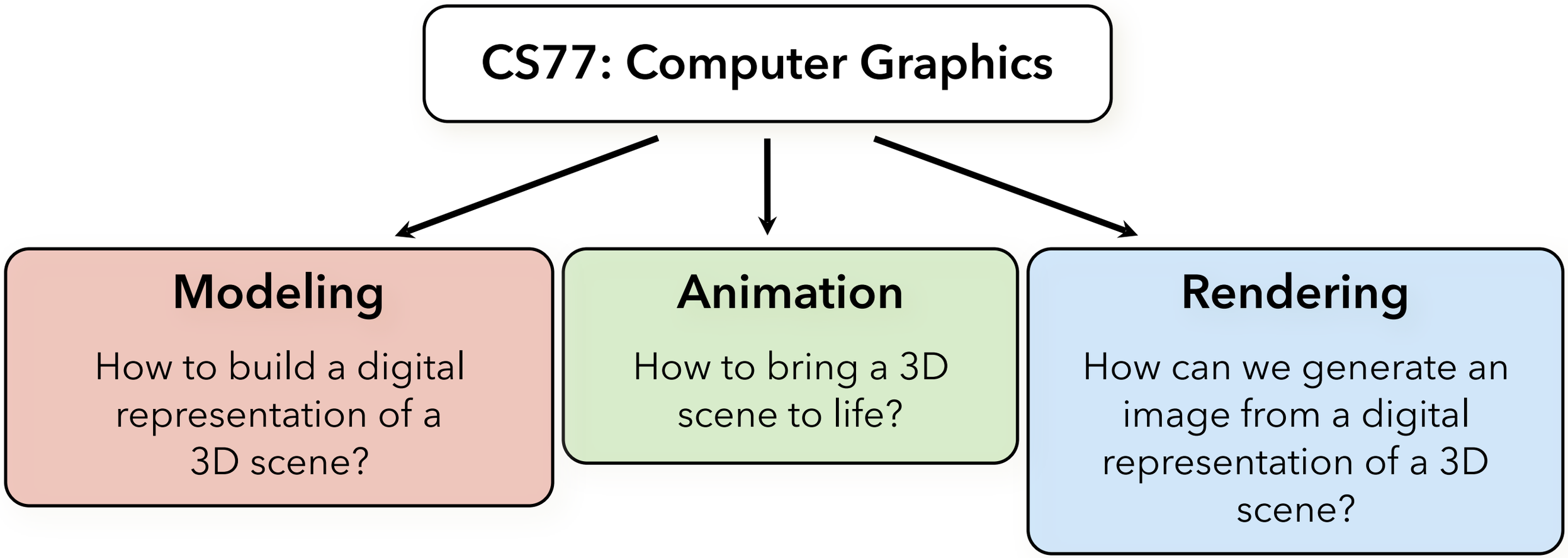
- Less programming
- More focused on art
- Acquiring experience in production tools
- Essential skills for production, TD, etc.

Relation to curriculum

Visual Computing

- CS77: Computer Graphics
- CS87: Rendering Algorithms
- CS89: Fabrication
- CS89: Computational Photo

Relation to curriculum



Relation to curriculum

CS77: Computer Graphics

Modeling

- Surface representations
- Texturing
- Shape modeling algorithms
- Shape modeling interfaces

Animation

- Keyframe animation
- Physics-based animation

Rendering

- Realtime rendering
- Ray tracing
- Ray tracing fast
- Surface reflection

- Physically based rendering
- Appearance modeling
- Light transport
- Monte Carlo simulation
- Volumetric scattering

CS87 (this class)

Motivating questions

How can we generate realistic images?

Why do things look the way they do?

Motivation

Motivation

Motivation

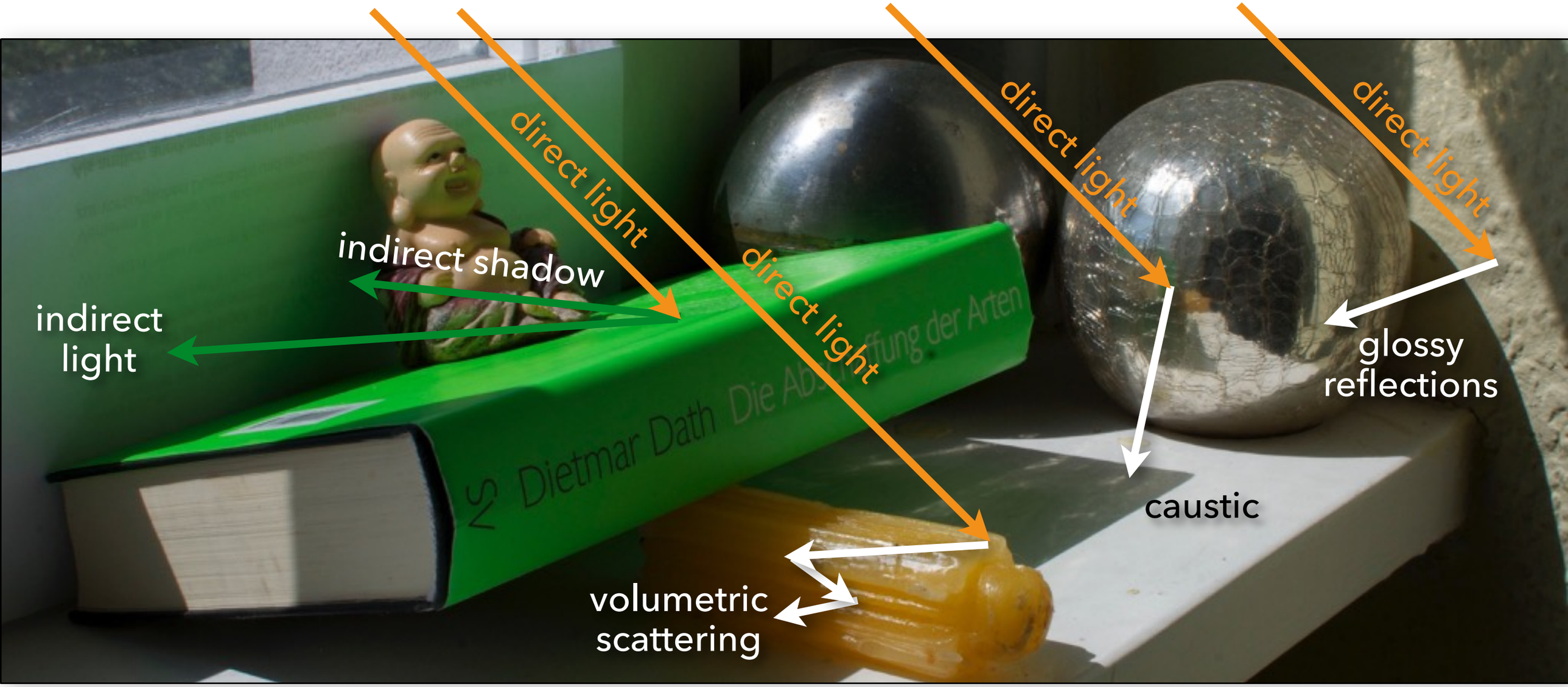
Motivation

Motivation

T.A.Rector (NOAO/AURA/NSF) and the Hubble Heritage Team (STScI/AURA/NASA)

Motivation

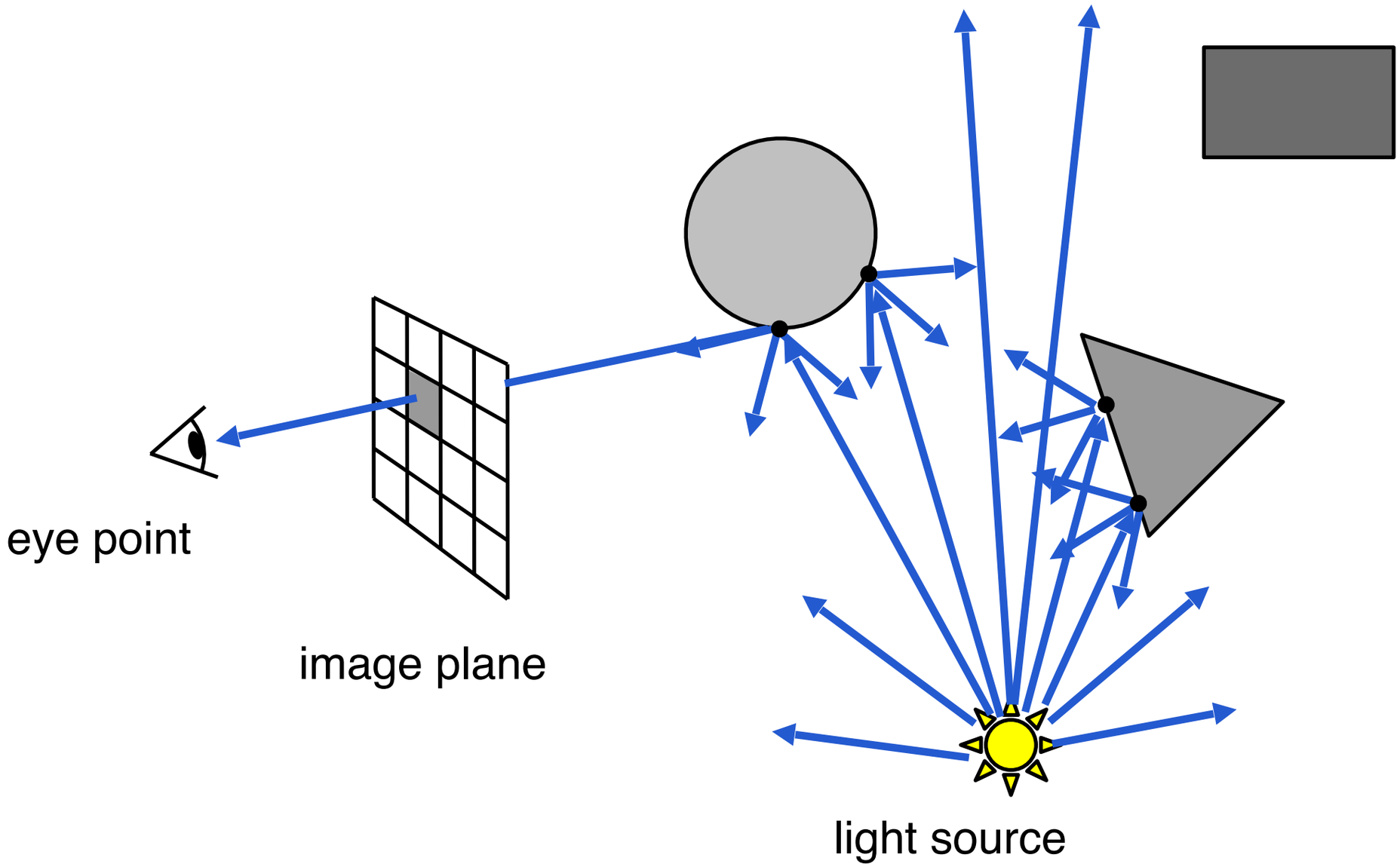
Light transport in the real world



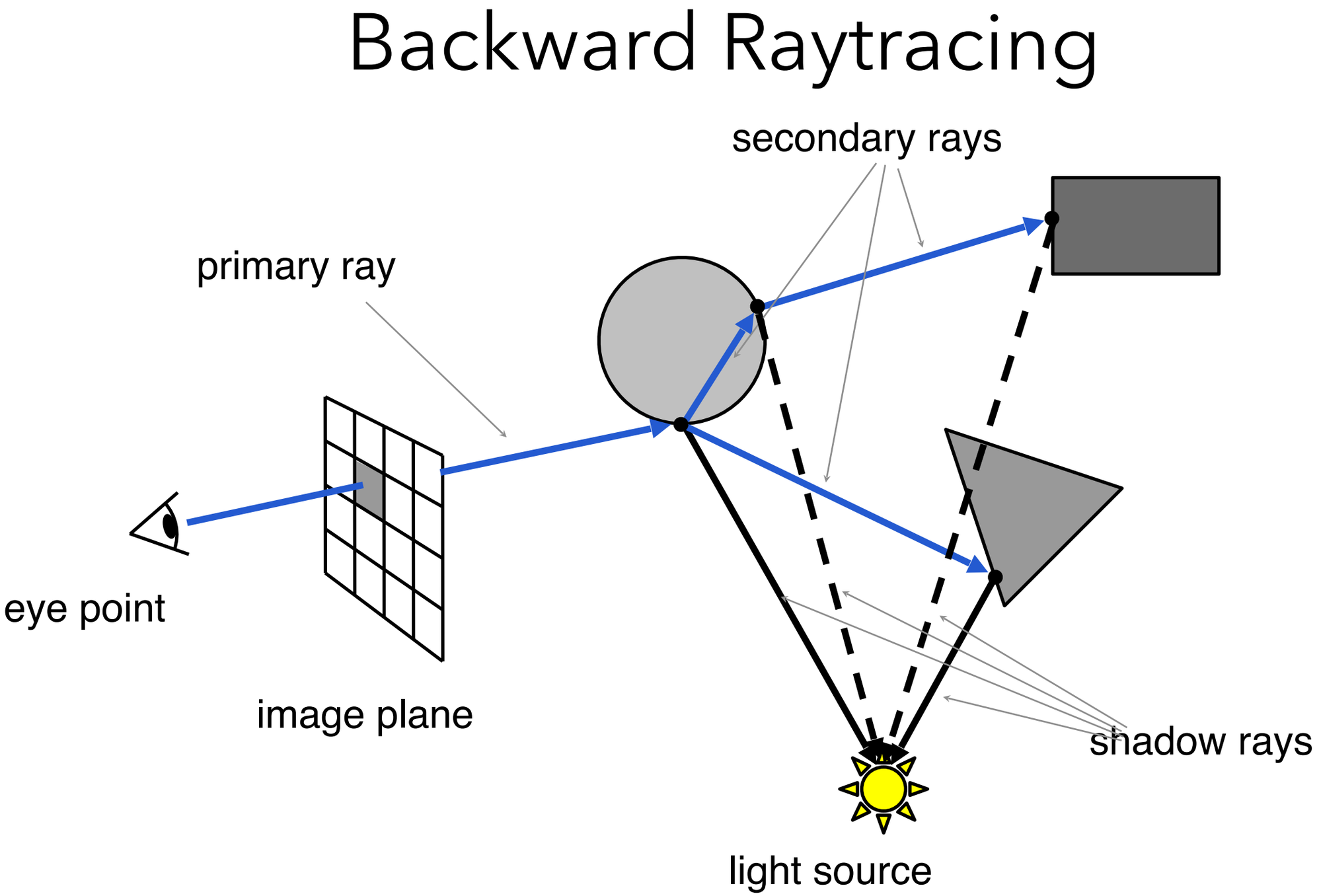
After [Ritschel et al 2011]

Raytracing - Overview

Forward Raytracing



Raytracing - Overview



Visual effects

Visual effects

Animated films

Video games

Architectural visualization

Architectural visualization

Architectural visualization

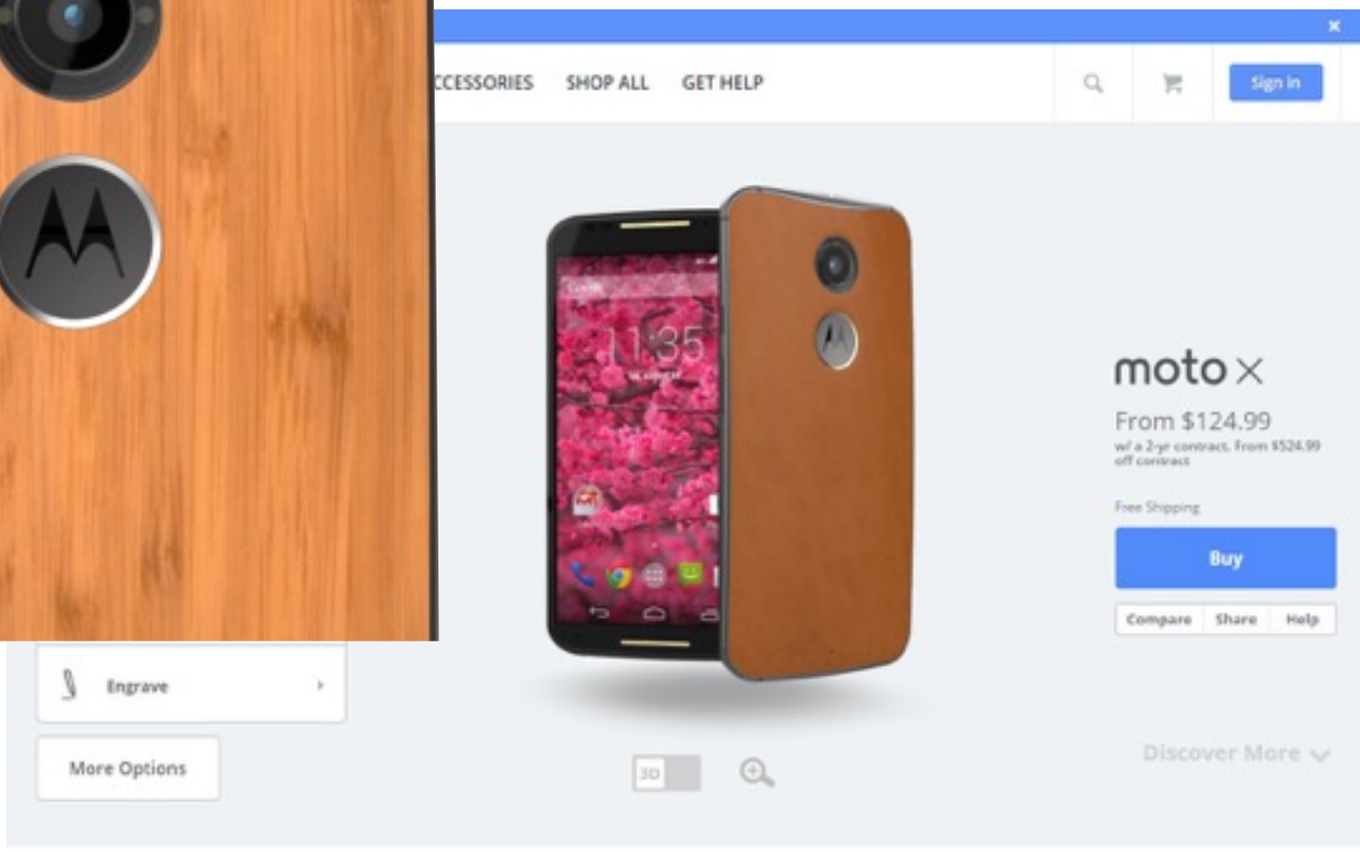
Product visualization

Product visualization

Product visualization

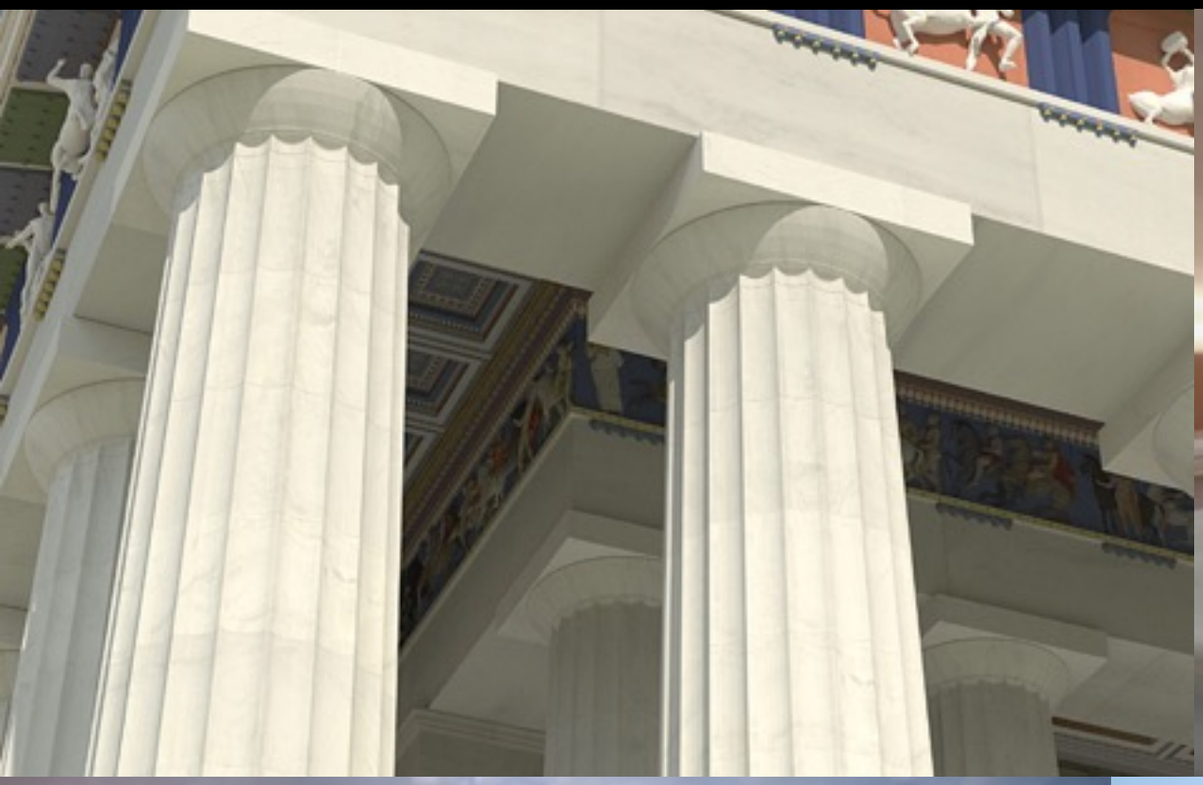
Product visualization

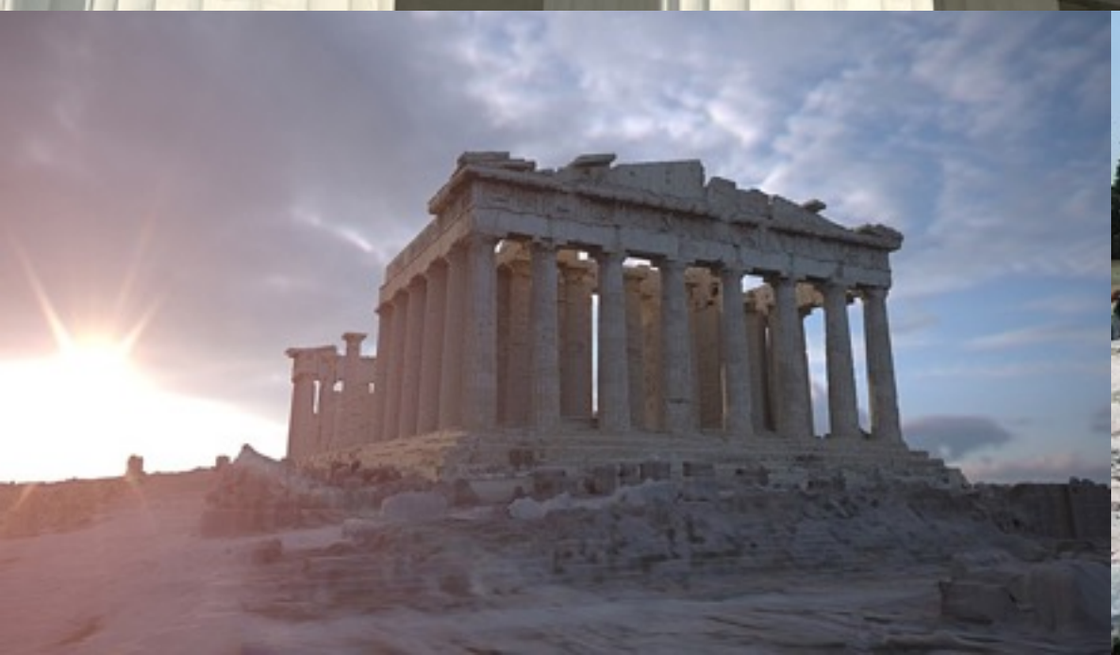
Advertising & E-commerce



Advertising & E-commerce

Cultural heritage





Scientific visualization (& Visual Effects)

Digital Fabrication

What is this class about?

Algorithms for realistic image synthesis

Simulating light transport (global illumination)

Modeling appearance (simulating materials)

Understanding why things look the way they do:

- Why is the sky blue?
- Why is the grass green?
- Why does metal look different than marble?

What you should already know

Basic concepts of algorithms and data structures

Calculus

- you will need to compute some derivatives & integrals

Linear algebra

- should know how to perform basic vector/matrix ops.

Programming in C++

CS77: Computer Graphics

- basics of ray tracing + acceleration structures
- BRDF basics (diffuse + specular)

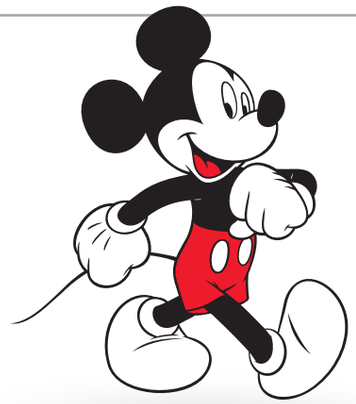
What you will learn

By the end of course you should:

- understand advanced concepts in rendering, light transport, and appearance modeling
- be able to program graphics algorithms for the above topics
- have increased knowledge of good software engineering practices
- have experience designing/extending a large software project/architecture

About me

I do this for a living (and for fun)!



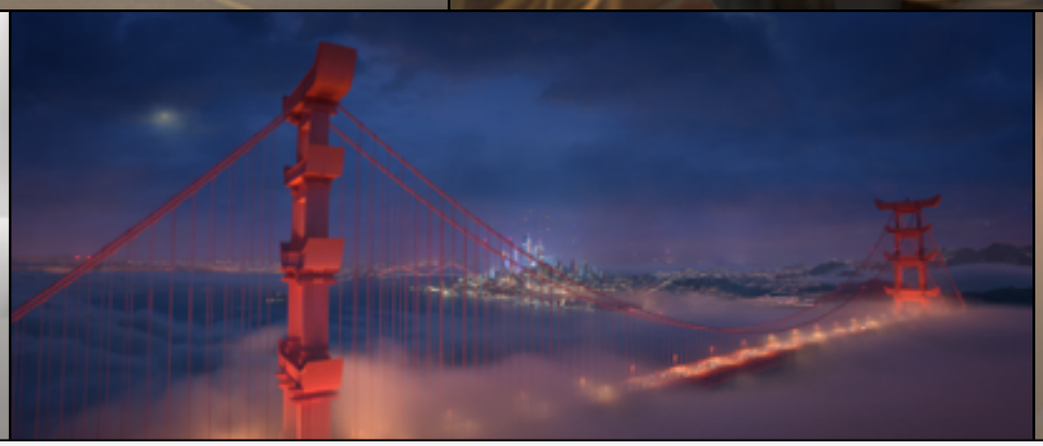
Previously worked at Disney Research developing new rendering techniques and getting them used in production

VISUAL COMPUTING LAB

Dartmouth's Visual Computing Lab





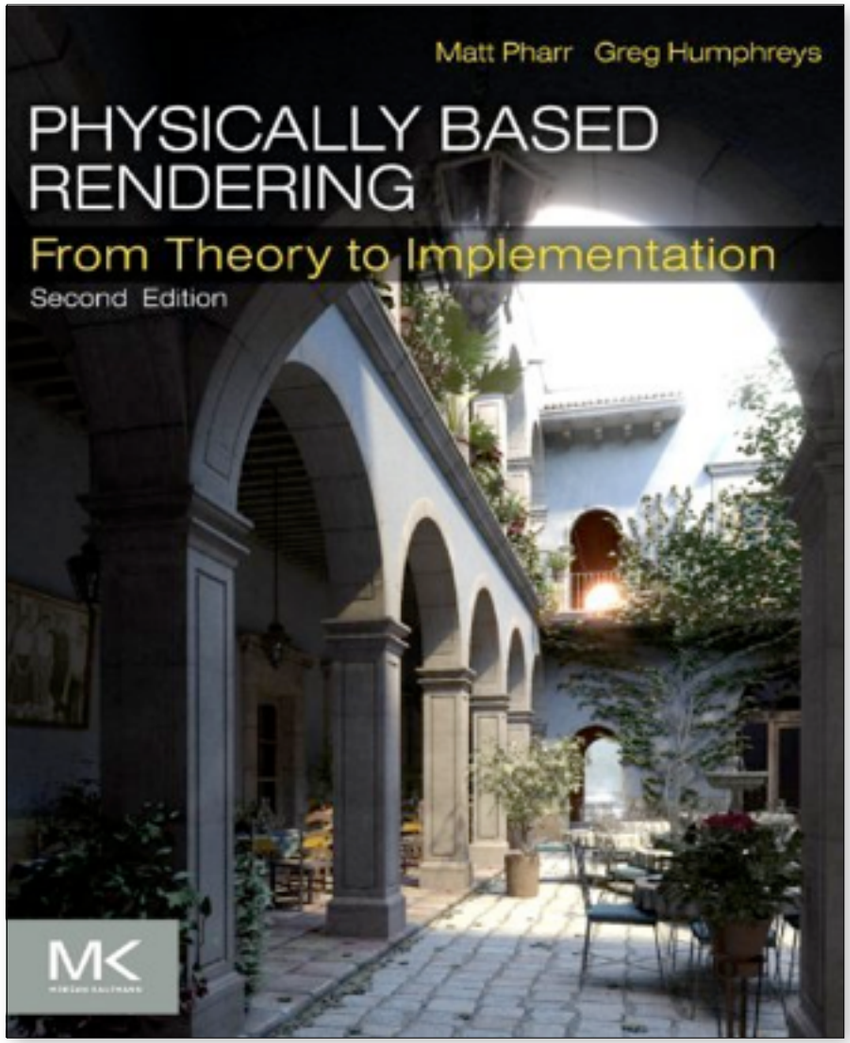




Literature

Required Textbook:

- Pharr & Humphreys. ***Physically Based Rendering: From Theory to Implementation***, 2nd edition (2010).

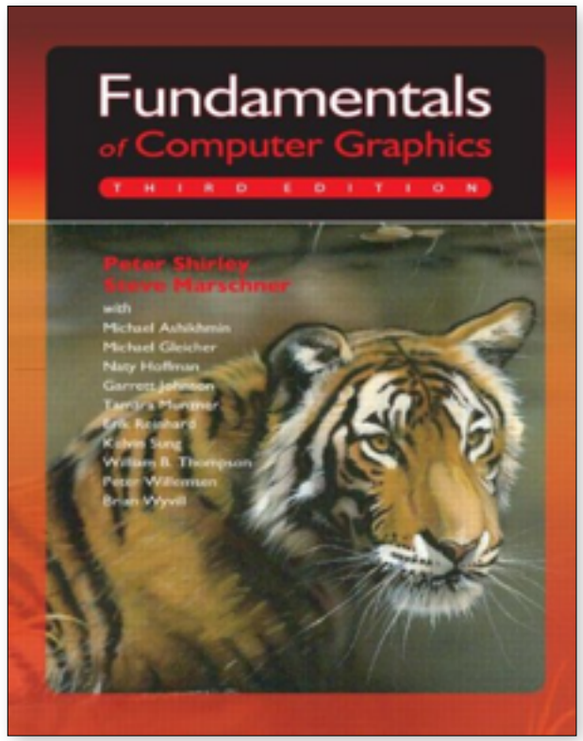
- digital e-book version available from Dartmouth IPs



Literature

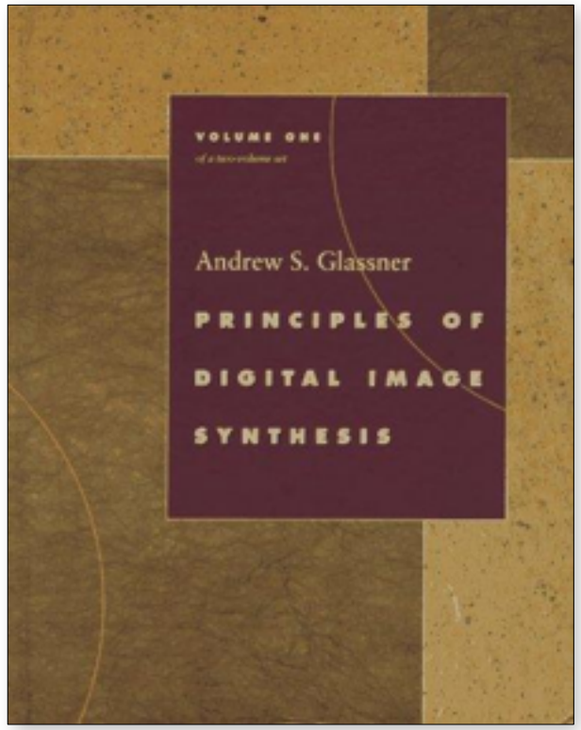
Optional Textbook 1:

- Shirley, Marschner et al.
  Fundamentals of Computer Graphics,
  3rd Edition, 2009.



Optional Textbook 2:

- Andrew S. Glassner.
  Principles of Digital Image Synthesis,
  1995.

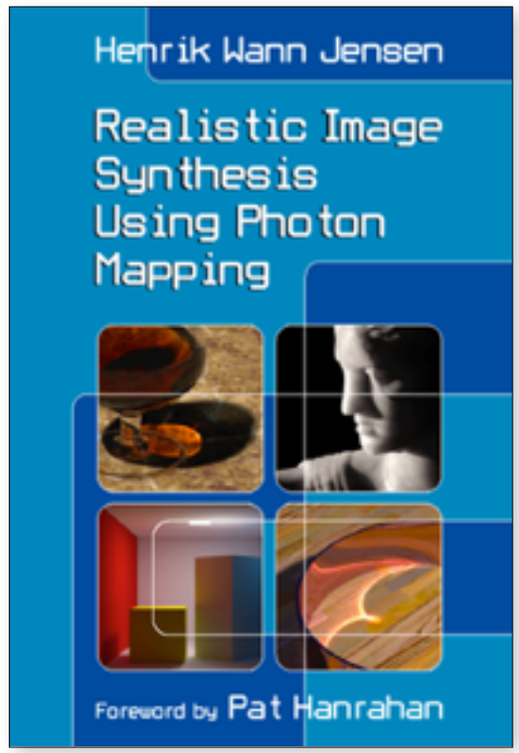


- PDF version available under creative commons

Literature

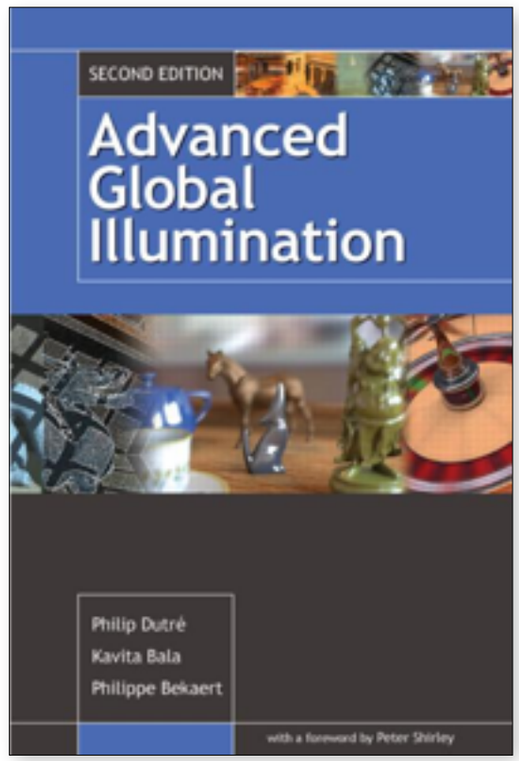
Optional Textbook 3:

- Henrik Wann Jensen.
  Realistic Image Synthesis Using Photon Mapping, 2001.



Optional Textbook 4:

- Dutre, Bala, and Bekaert.
  Advanced Global Illumination,
  2nd Edition, 2006.



Tentative Schedule

Week	Tuesday	Thursday	Homework Due
1	Intro; Light + Matter	The BRDF & Appearance Modeling	PA0: Prelim + Intros
2	Monte Carlo I	Monte Carlo II	PA1: Getting started
3	Direct Illumination I	Direct Illumination II	PA2: Sampling
4	Global Illumination I - Path tracing	Global Illumination II - Many-lights	
5	Global Illumination III - Photon mapping	Participating Media I	PA3: MC ray tracing
6	Participating Media II	Subsurface Scattering	
7	Markov Chain Monte Carlo	Image-based Denoising	PA4: Global illum.
8	**Exam**	Monte Carlo Variance Analysis	
9	Misc topics	Misc topics	
10			**Rendering comp.**

Tentative Grading Scheme

5%: Participation

45%: Programming assignments

20%: Exam

30%: Final project

Programming assignments

Late policy:

- 1 day late: max 75%
- 2 days late: max 25%
- 3 days late: 0%
- only exceptions: ask me >1 week in advance

Zero tolerance for cheating!

Assignments will be turned in through Canvas

Grading will happen face-to-face with the TA during grading sessions (typically the following week)

Exam

1-hour written "midterm" exam

- Roughly 3/4 way through the class

No final exam

- but a final project presentation

Final Project/Rendering Competition

Render a realistic image (of your choosing) using the software you create.

Extend your renderer with advanced functionality

- Be creative and ambitious
- Use photographs, images online as inspiration
- Starting thinking about ideas early!

During our final examination period (June 5th)

Participation is mandatory

We will have guest judges & best images will win prizes!

Past Rendering Competitions

Other universities have similar courses with rendering competitions:

ETH, e.g.:

- https://graphics.ethz.ch/teaching/imsynth14/competition/competition.php
- https://graphics.ethz.ch/teaching/imsynth15/competition/competition.php
- https://graphics.ethz.ch/teaching/cg15/www-nori/index.html#project

Stanford, e.g.:

- http://graphics.stanford.edu/courses/cs348b-competition

UC San Diego, e.g.:

- http://graphics.ucsd.edu/courses/rendering/

Simon Kallweit - ETH (2015)

Tizian Zeltner - ETH (2015)

Benedikt Bitterli - ETH (2014)

Simon Kallweit - ETH (2014)

Romain Prévost - ETH (2012)

A. Guzman & J. Schwarzhaupt (2011)

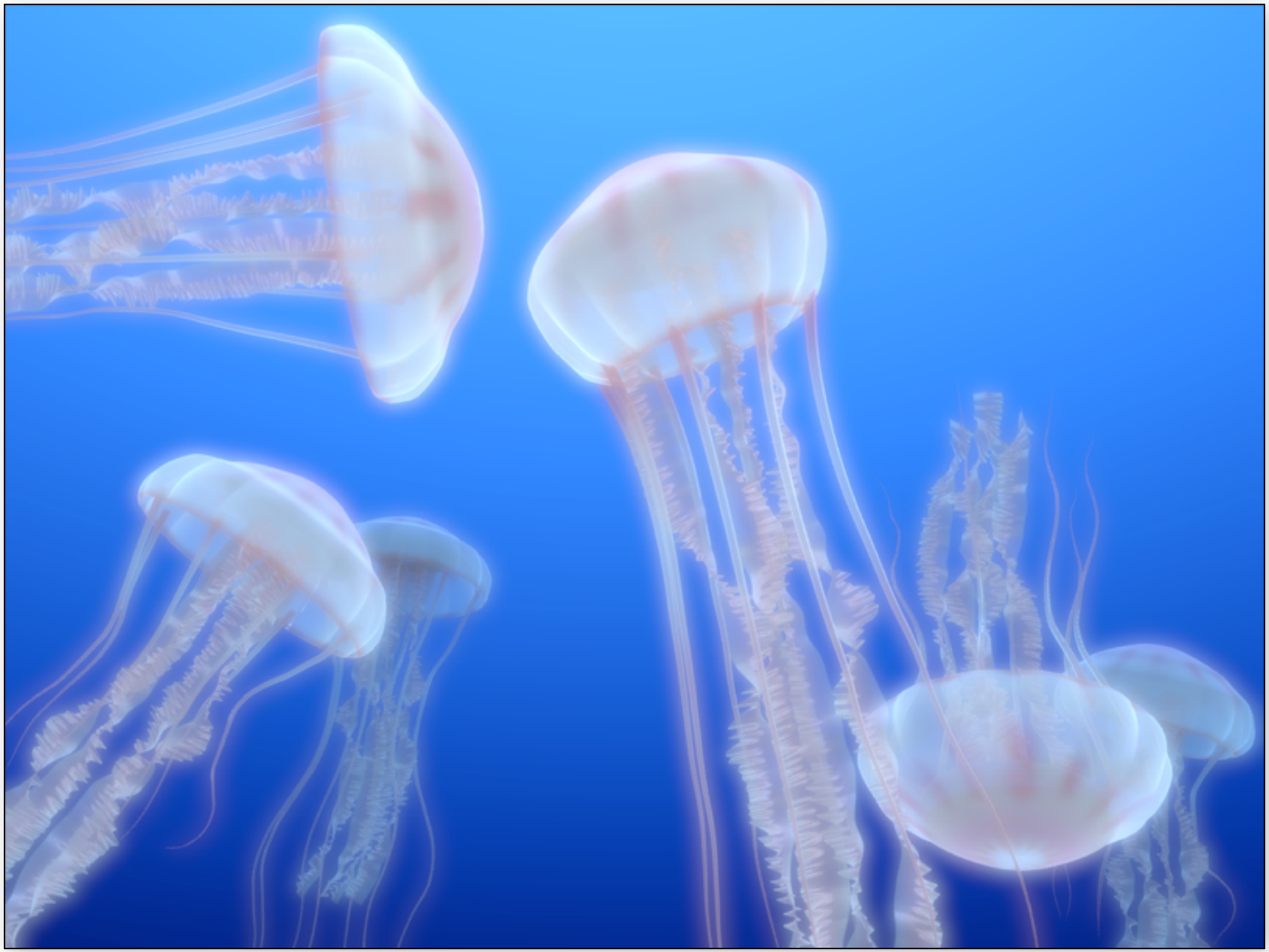
Iman Sadeghi - UC San Diego (2007)

Alex Kozlowski - UC San Diego (2005)

Josh Wills - UC San Diego (2003)

Tom Brow & Ranjitha Kumar - Stanford

Kayvon Fatahalian & Tim Foley - Stanford



First 2 assignments due this Friday!

Shouldn’t take you more than 30 minutes

1. Introductions (so I can learn you names)
2. Git/CMake/Nori tutorial

There is more

First x-hour session (**IMPORTANT!**)

- Tomorrow @ 4:15 pm
- Introduction to our framework

Questions?