

CS 410 Lecture 01: Introduction

August 27, 2019

CS 410: Computer Graphics

Professor:

Ross Beveridge

Room 348 CSB

(970) 491-5877

Office Hours:

Tue 10-11AM

Fri 10-11AM

or by appointment

Ross.Beveridge@colostate.edu

Teaching Assistants :

Ben Sattelberg

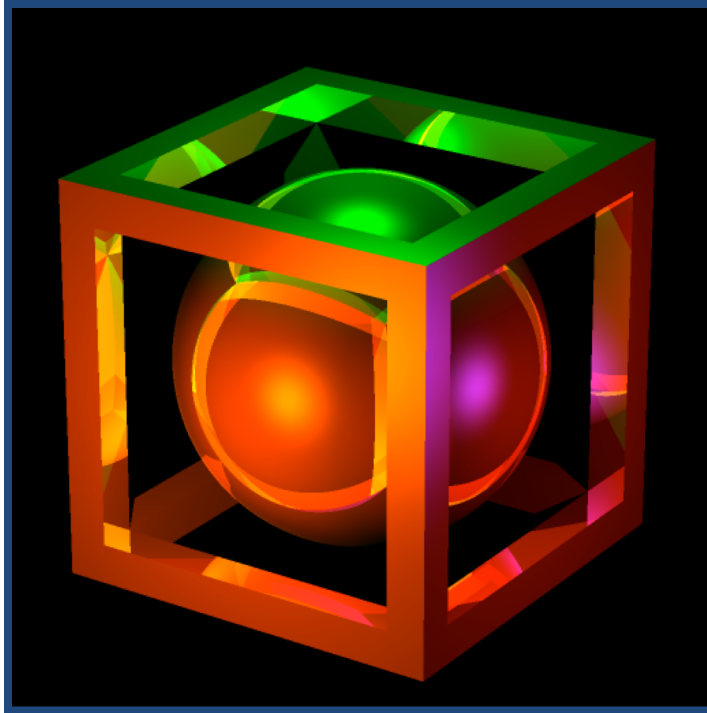
Wen Qin

Office Hours: TBD

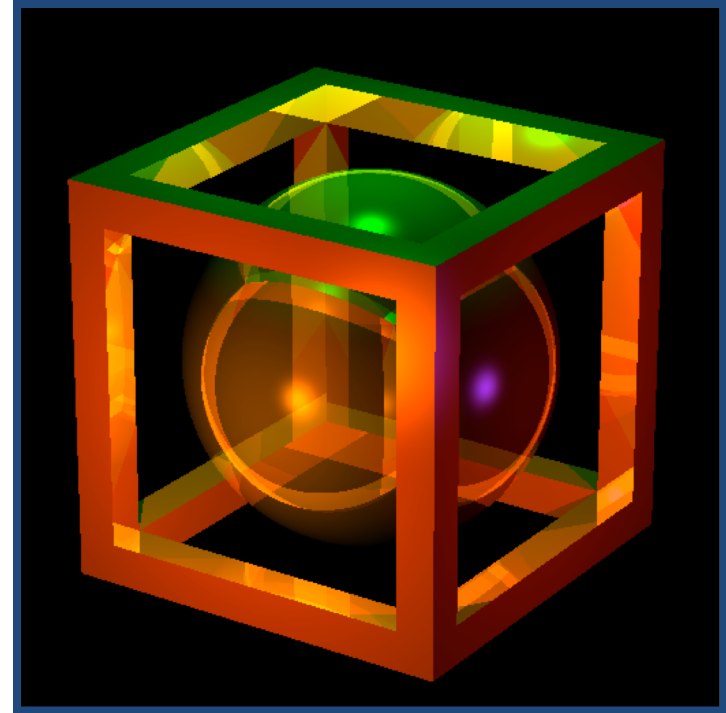
Web Site: <http://www.cs.colostate.edu/~cs410>

1-slide Overview

- This is what you will learn to make



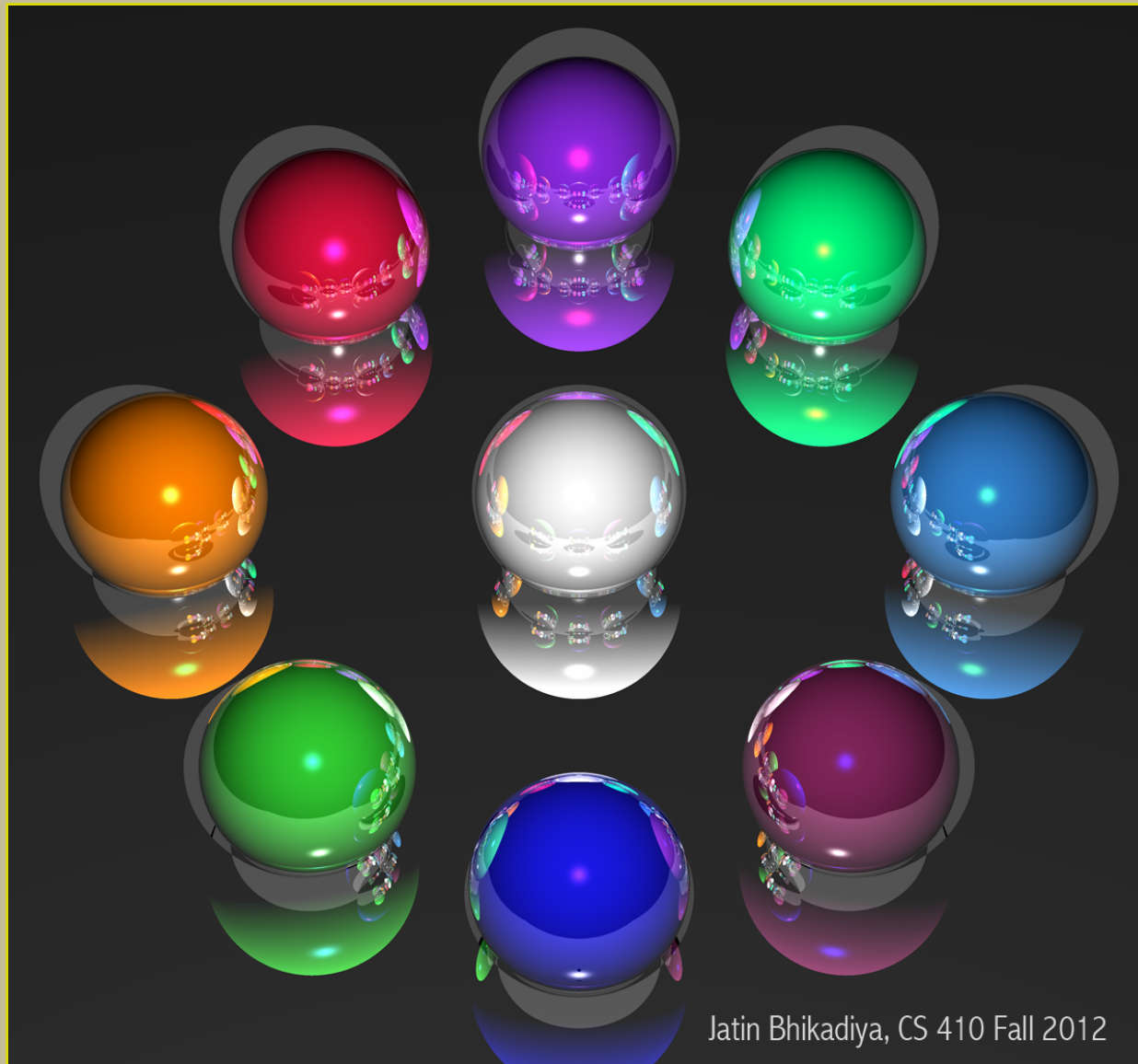
Charumathi Chandrasekaran



Ben Randall

- But you will have to work hard to get there

Another Example – Fall 2012



Jatin Bhikadiya, CS 410 Fall 2012

What can you say about this image?

In other words, what processes are at work – simulated – to create what you are seeing?

More Recent Example 2018



Isaac Law CS 410 Fall 2018

In This Class

- You will build a 'ray tracer'.
- Input
 - Scene model (objects, surfaces & materials)
 - Lighting models
 - Camera models
- Output
 - An image showing what the camera would see.

What we will do....

- Program a ray tracer from scratch
- General flow of programming assignments
 - Basics of 3D geometric transformations
 - Basics of 3D object modeling
 - Simple ray casting with global geometry
 - Geometric generality – relative placement
 - Illumination and surface reflectance
 - Multi-bounce tracing with reflection/refraction
- Each assignment builds on the one before
 - 20 – 30 hours each
 - Following good software practices is key

In the Process ...

- Master basics of linear alg. as geometry
 - Dot (i.e. inner) products (of vectors)
 - Cross products (of vectors)
 - Homogeneous coordinates
 - Affine & perspective transformations
 - Matrix multiplication
 - Orthonormal matrices
- Expertise of value beyond Graphics !

... 3-D Geometric Intuition ...

- Expertise requires three things.
 - Linear algebraic objects and operations
 - Visual intuition – “seeing” what your specifying
 - Quick and easy shifting back and forth
- Most obvious examples.
 - Move the camera back and a bit left
 - Place object A on top of object B
 -

... and of course also ...

- Learn about light
 - Color spaces
 - Reflection & refraction
- Surface properties
 - Lambertian (Matte) reflection
 - Specular reflection
 - Hybrid reflection models

Resources (1)

- Lectures – very important
 - General concepts
 - Illustrative Examples
- Optional Textbook – a good reference
 - Details generally presented in a clear exposition.

Resources (2)

Not Secure — www.cs.colostate.edu/~cs410/yr2019

CS 410 : Homepage



Introduction to Computer Graphics

Fall 2019

CS 410 : Homepage



Home Syllabus Progress Assignments Resources Canvas

CS 410 teaches students how to program a computer to generate photo-realistic images. The general idea is that given a scene model, a sensor model and a viewpoint, one should be able to create the same image that a camera would for that scene and viewpoint. Scene models include 3D object models and light sources. Objects models are composed of surfaces, and include both geometry (where is the surface?) and material properties (what is it made of? What color is it?).

This course will emphasize geometric objects and transformations, perspective projections, lighting and reflectance models, shading models, and 3D curves and surfaces. Students will design and implement a ray tracing program from scratch, thereby becoming intimately familiar with the sensor, lighting and object models described above. Perhaps most importantly, students may come to more fully appreciate the power of linear algebra.

Instructor:
[Ross Beveridge](#)
Office: 348 CS Building
Office Hours: Tue, Fri 10 - 11 AM
Email: Ross.Beveridge@colostate.edu

GTA:
Ben Sattelberg
Office:
Office Hours:
Email: bsattelb@rams.colostate.edu

Lecture Time and Place:
3:30-4:45, Tue, Thur, Clark A204

News:
[Quoting Kosh: "So it begins". Welcome. \(Ross 8/20/19\)](#)

The CS410 Logo image created by CS410 Student Harry Houlton in Fall 2016.

Session Time 0 Secs.
Originating IP 10.1.45.193
User: Guest

Apply to CSU | Contact CSU | Disclaimer | Equal Opportunity
Colorado State University, Fort Collins, CO 80523 USA
© 2019 Colorado State University



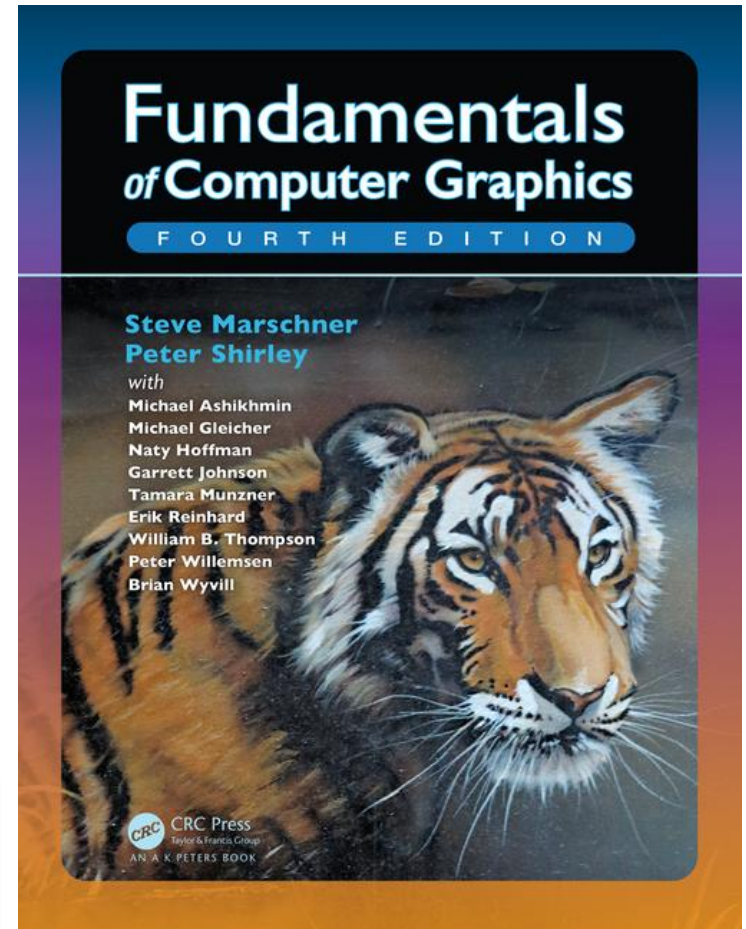
Resources (3)

- Private Website – CANVAS
 - Used for Quizzes, Assignments, Grading ...
- Office hours
 - Mine:
 - Tuesday 10-11AM
 - Friday 10-11AM
 - By appointment
 - GTAs
 - TBD
 - Don't be a stranger!

Optional Textbook

- Adopted in 2011 at CSU
- Virtues: Excellent
 - Focuses on Ray Tracing
 - Consistent terminology
 - Well staged mathematics
 - E-format available
- Either 3rd or 4th edition

But, do not expect lectures to follow textbook specifics nor the textbooks specific mathematical conventions.



Course Rules

- Be on time
 - Class starts at 3:30
- Ask & Answer questions
 - Always give your name in the process
- Professional behavior at all times
 - No non-course related chatter
 - Cell phones on silent
 - Leave the room to answer them
 - Be polite & respect others

Major Activities / Grading

- Quizzes
 - 10% of your grade – there to help you study
- Programming Projects (~5)
 - 50% of your grade total
 - Generally 10% each
- Midterms (2)
 - 20 % of your grade (total)
 - 10 % of your grade (each)
- Final exam
 - 20% of your grade

All quizzes
and exams
administere
d online
using
CANVAS.

Policies

- Assigned work is done alone
 - No joint projects
 - No open note exams
 - No taking code from the internet
 - Follow the department academic integrity policy
- All work done on time
 - No late period – deadlines are deadlines
 - Multiple submissions OK – last is graded
 - When given two (or three) weeks, start right away!
- Exceptions: unforeseeable circumstances

Systems and Tools

- You may be surprised ...
- You will not learn a complex and/or arcane API for a giant graphics package
- Instead, you will build your ray tracer from scratch in C++ or Java (no Python)
- Applying programming techniques from CS253 and CS314.

This course is NOT...

- A course in OpenGL
 - OpenGL shaders are complex & detailed
 - Ray Tracing will become the dominant paradigm
 - Thanks to GPUs and parallel architectures
- A course in using any other Graphics API

This is a course about the mathematics and algorithms underlying Ray Tracing. It should make you a better programmer, improve your linear algebra, and prepare you to study computer vision (which is where the jobs are...)

Systems - Requirements

- Ray Tracer Mechanics
 - Must run on CS Linux Machines
 - Tools for viewing models will be provided
 - Tools for viewing images will be provided
- Test your code in the Department!
- Speed not a priority until
 - Long run-times interfere with grading.

The Pelican

The person with the pelican speaks next.

Why?

- One-way lectures suffer many flaws.
- Instructor can become detached.
- Students less engaged.
- Practice speaking is valuable.
- Why the Pelican?
- Same people always speaking.
- Knowing when your up next.



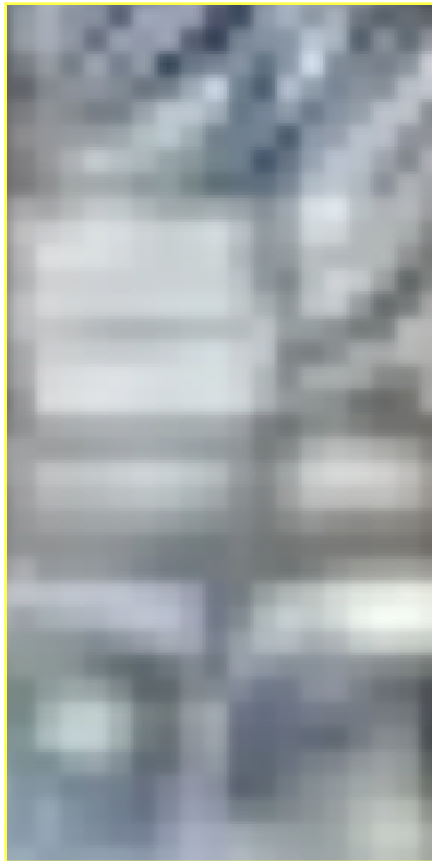
The Main Act – Create an Image

Let us walk quickly through some of the key concepts of this course as a means of setting up a context for what is to come.

What is an image?



What is an image? (cont)



- An array of values
 - Intensities (if gray scale)
 - Triples of red, green & blue (color)
- This image is a part of the previous image
 - Where?

How is an image *formed*?



Step 1: Light Source

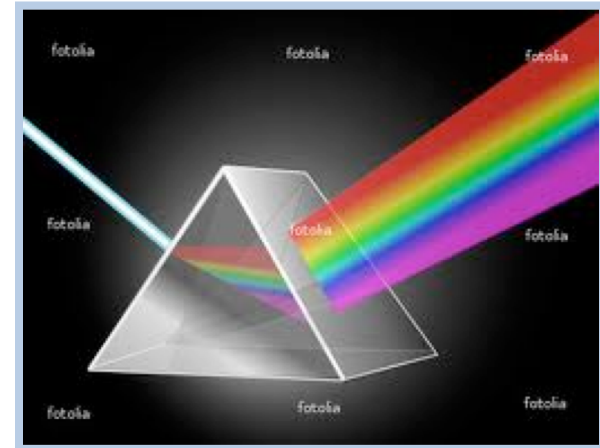
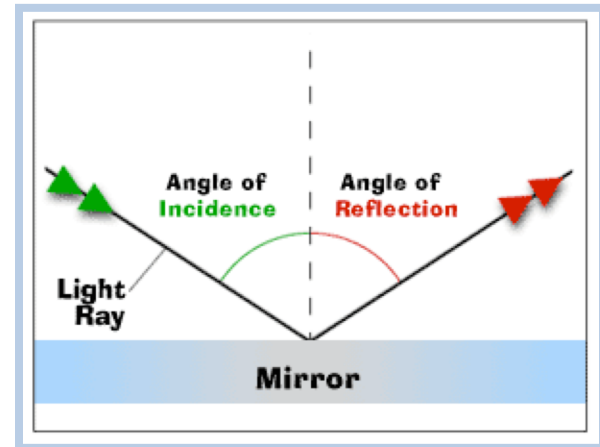
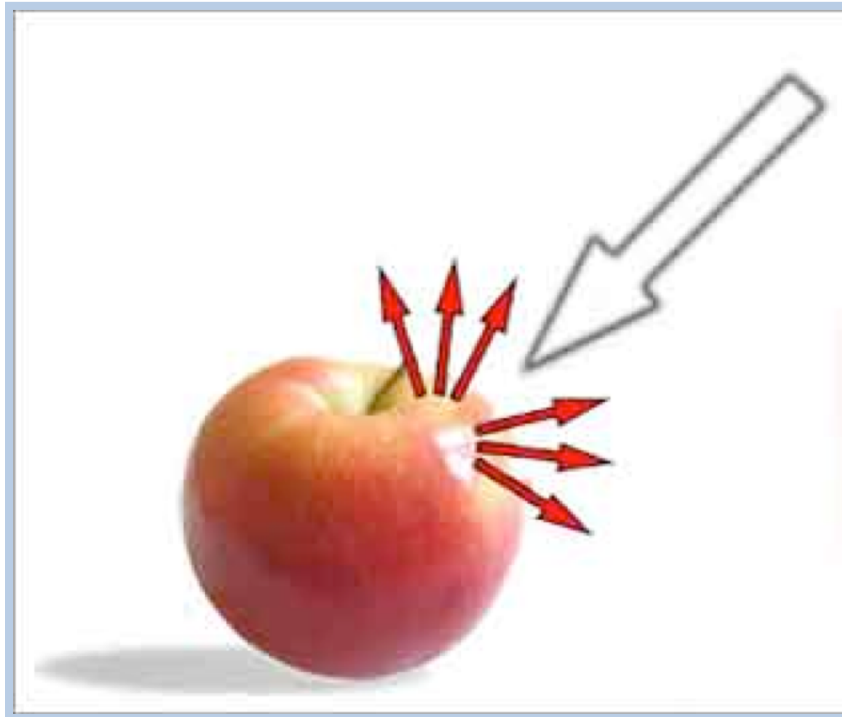


- Light Sources have
 - Intensities
 - Color
 - Positions

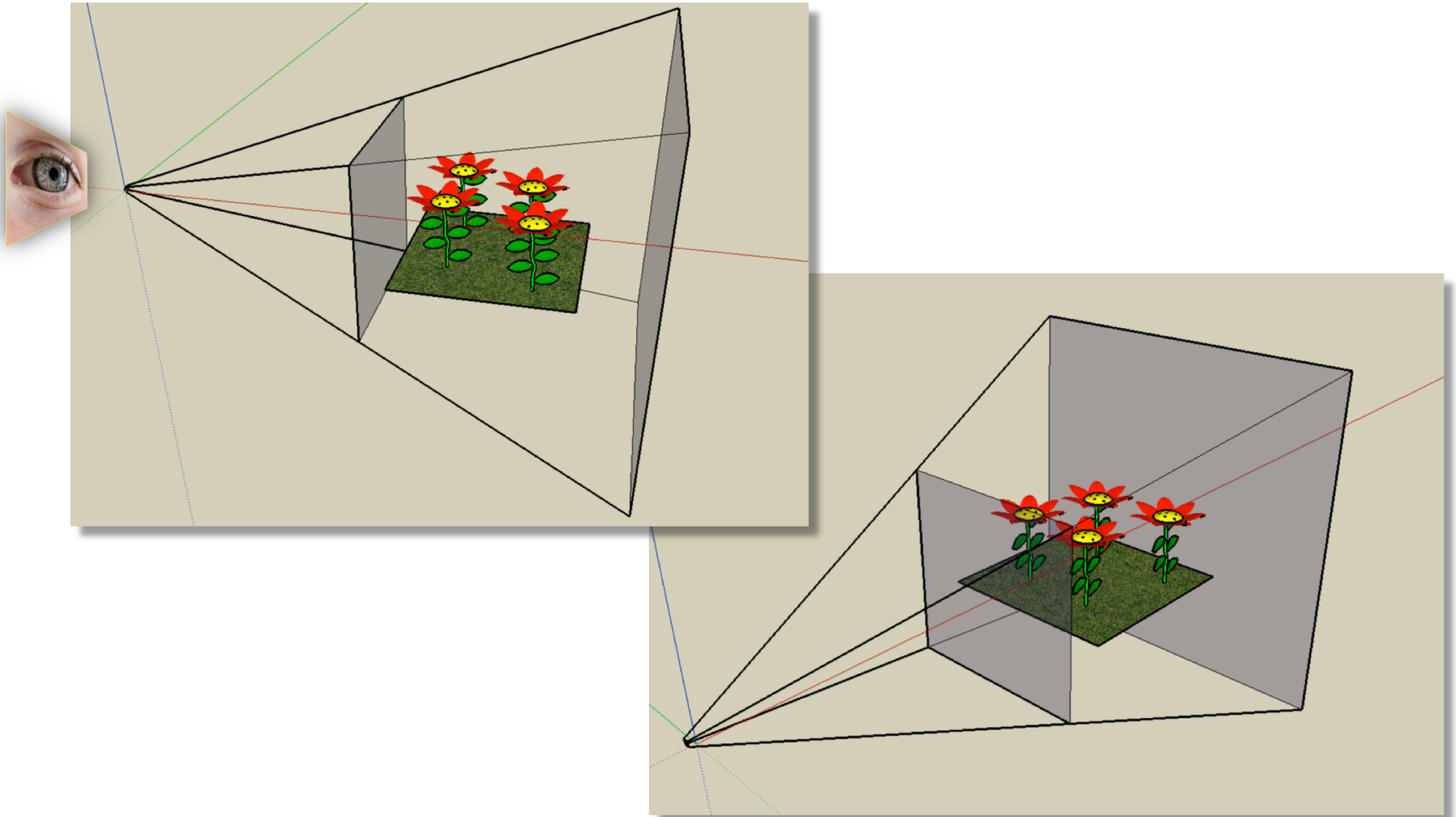


- Bonus: atmospheric effects

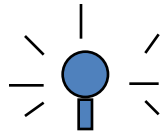
Step 2: Reflection



Step 3: Projection

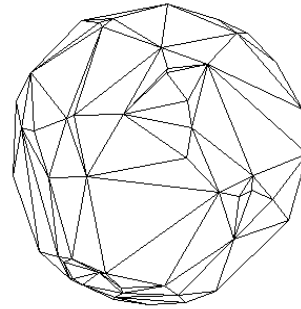
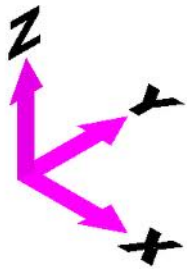


How (artificial) images are formed

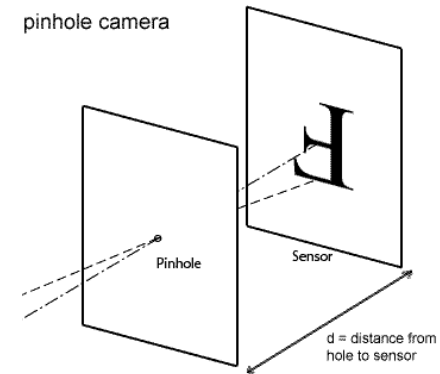


Light Source

Coordinate System

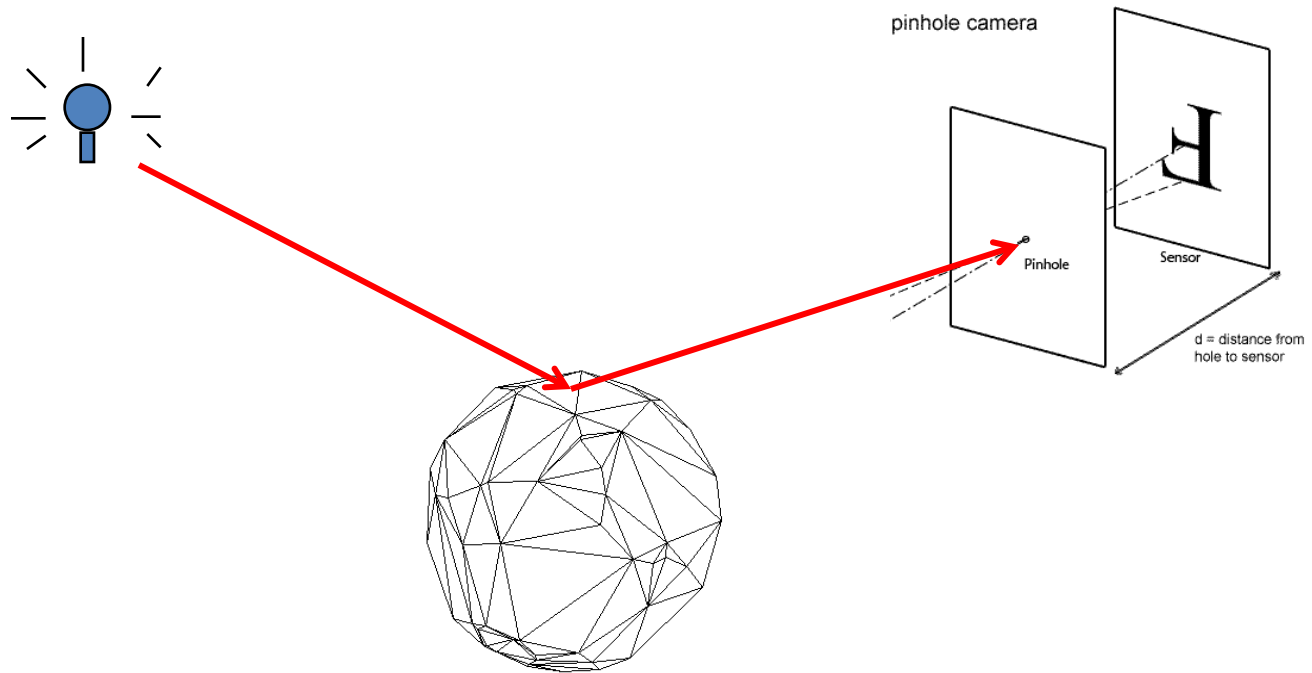


Geometric Objects



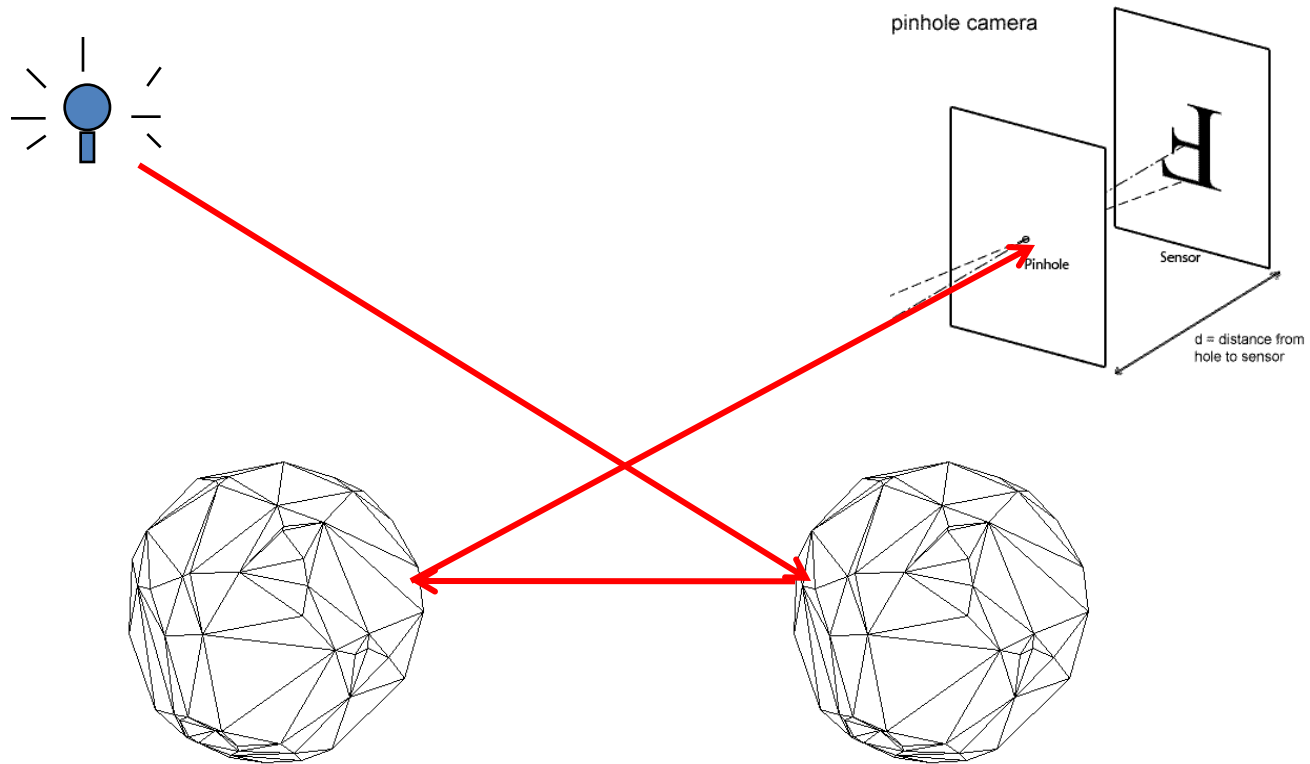
Sensor

Now mimic light...



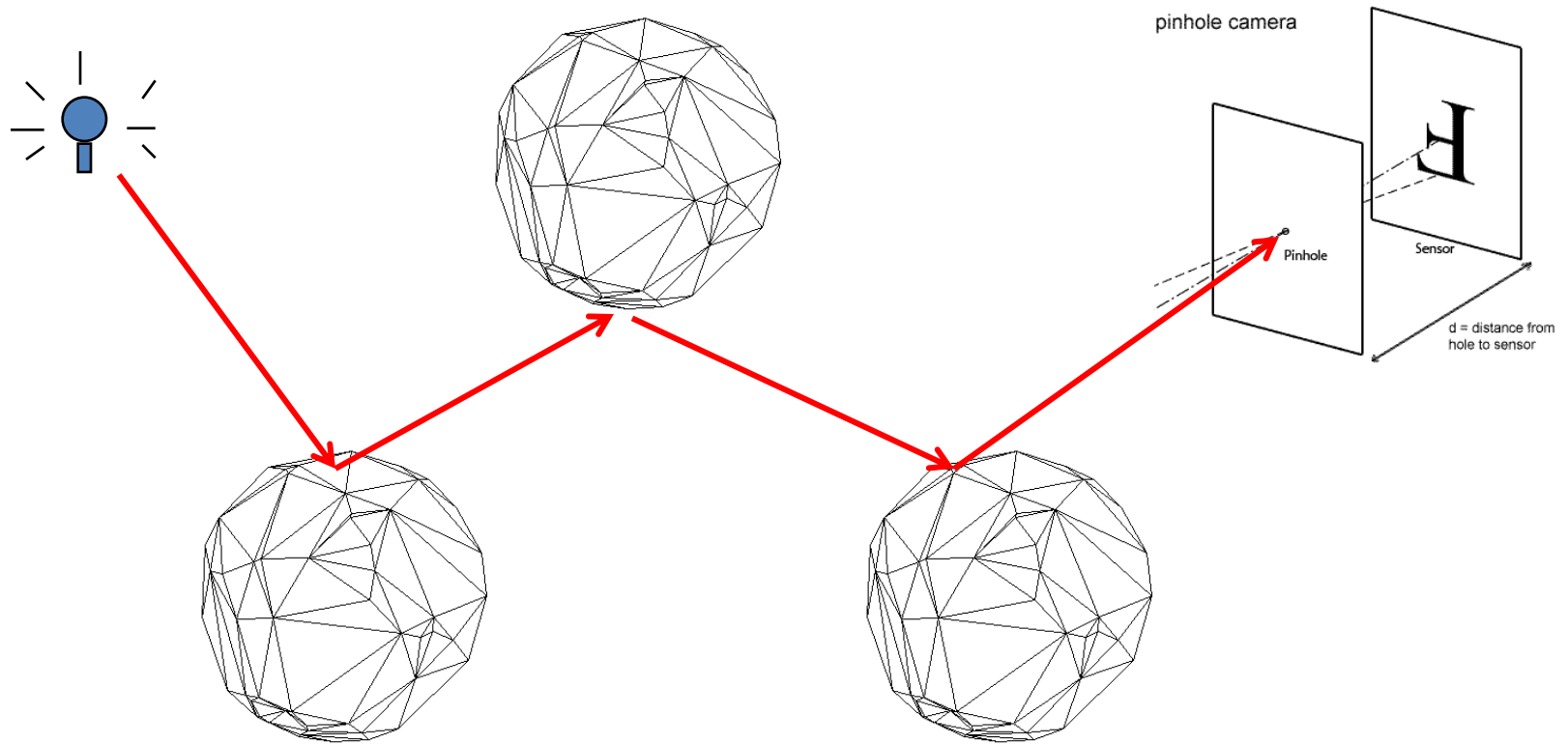
From source to object to sensor

... Or ...



...from source to object to object to sensor

... or ...



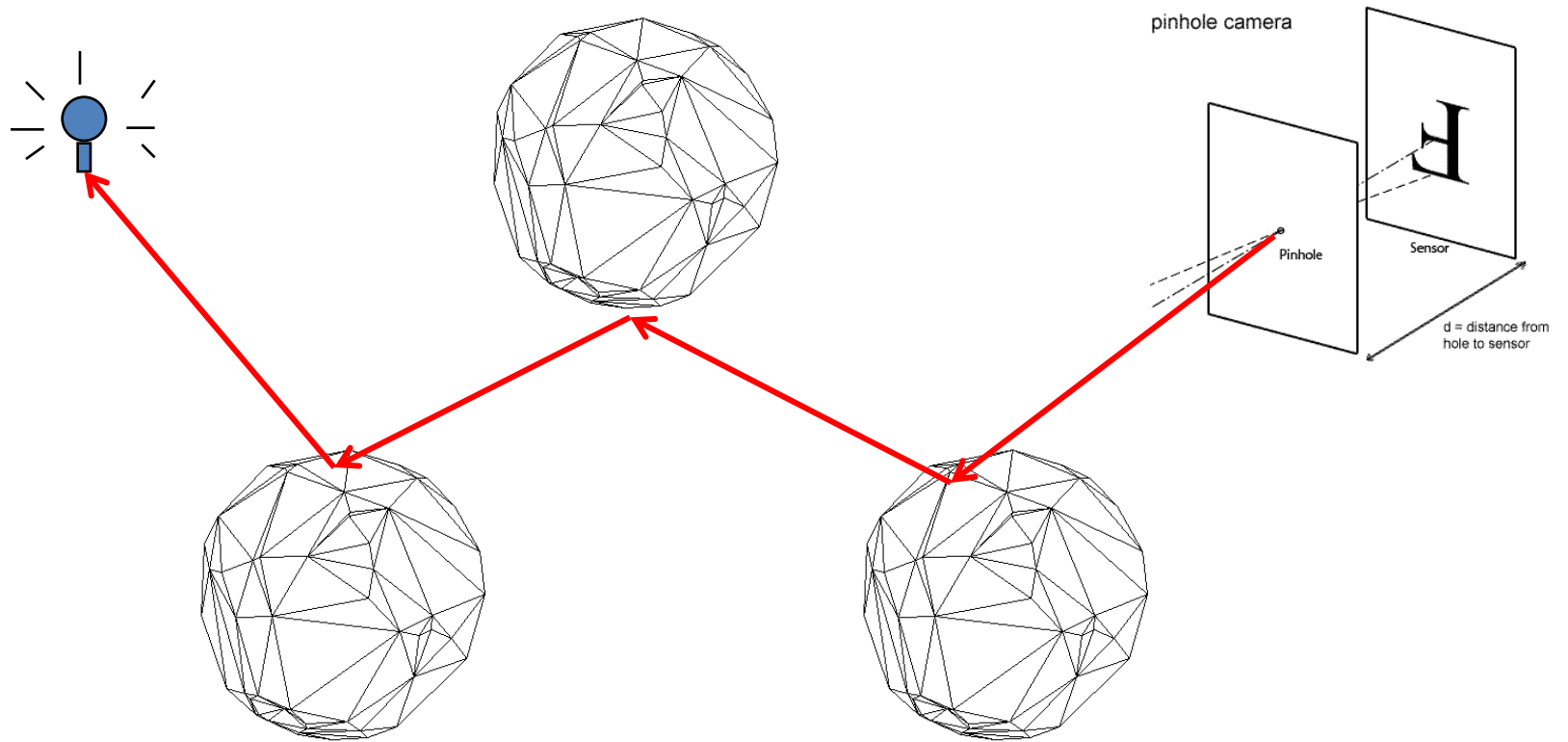
...you get the point.

True – But Backwards!

- Light sources emit a LOT of light rays
 - Most of them never strike your camera
- And you might have multiple light sources
 - In fact, you usually do, and the light intermingles
- So following every light ray is expensive

Instead, work backwards: start at the surface of the lens, and figure out where the light came from ... all the way back to the light source.

In other words



What can this approach do?

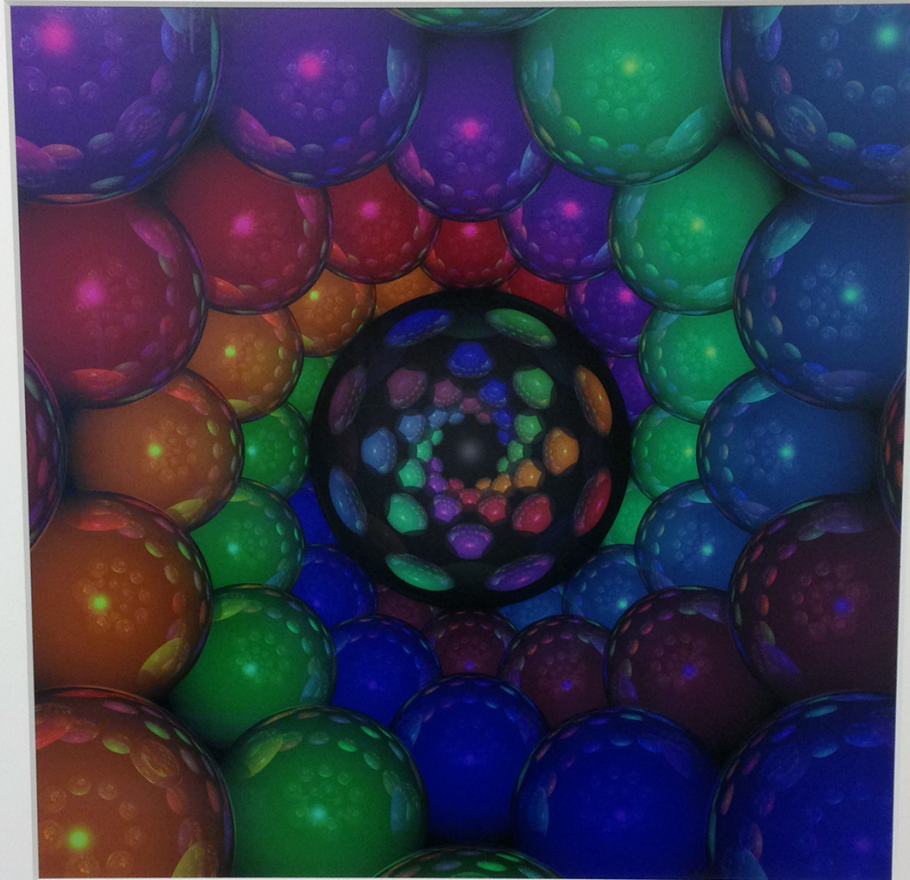


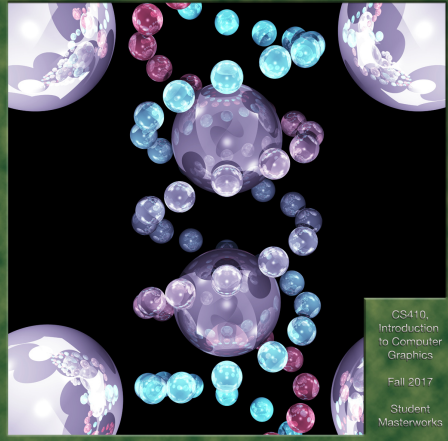
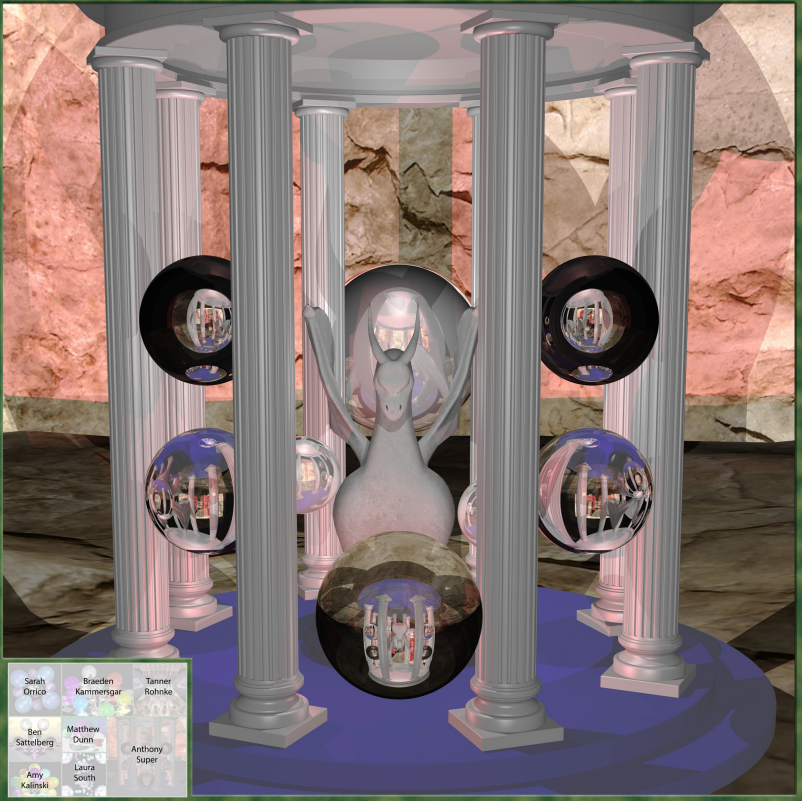
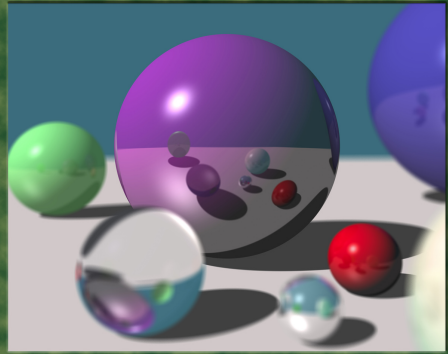
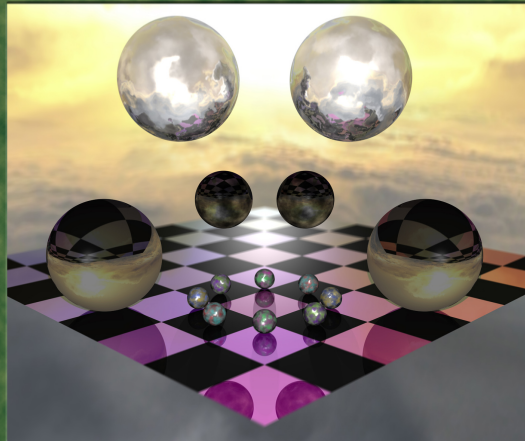
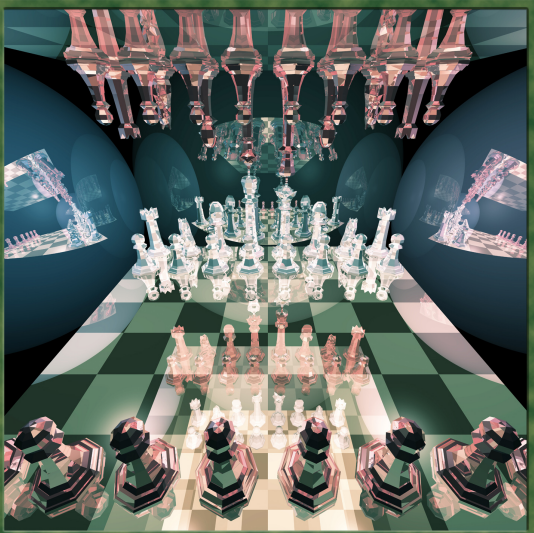
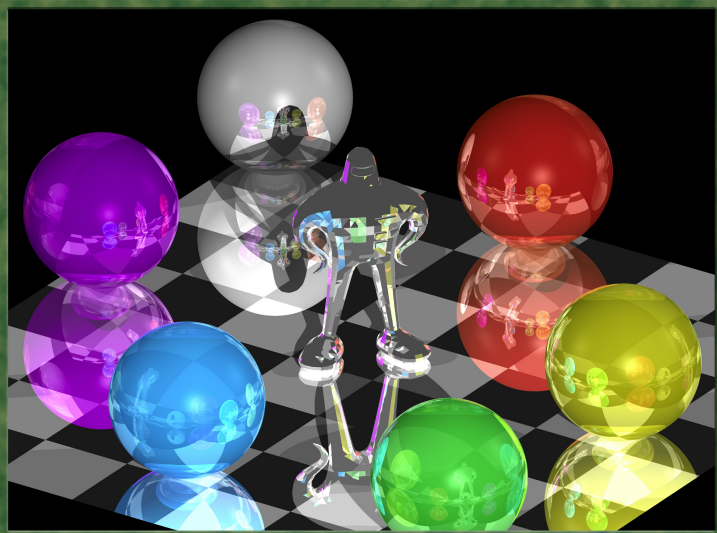
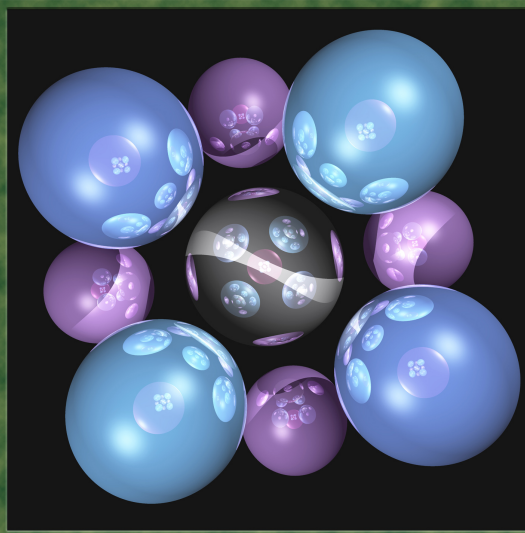
A photorealistic rendered image created by using POV-Ray 3.6
Image from [Wikipedia entry on rendering](#).

CS 410 Project done
by Kyle Olson in Fall
of 2014.

This particular
project went a bit
beyond what is
required - that is
why it is framed on
my office wall.

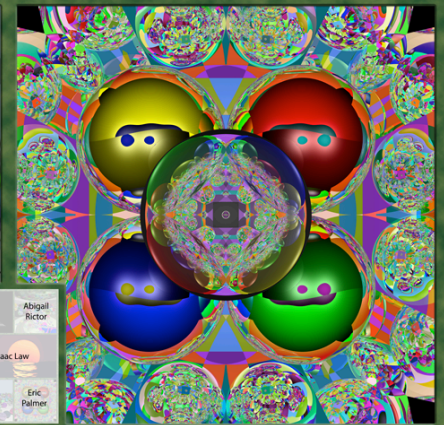
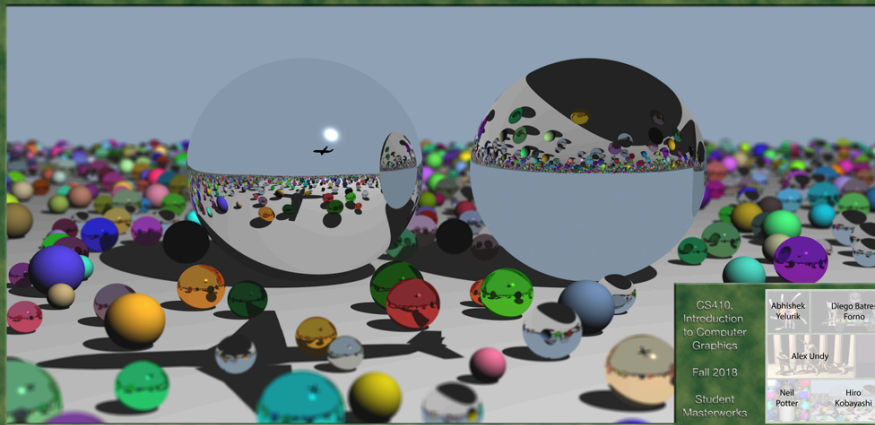
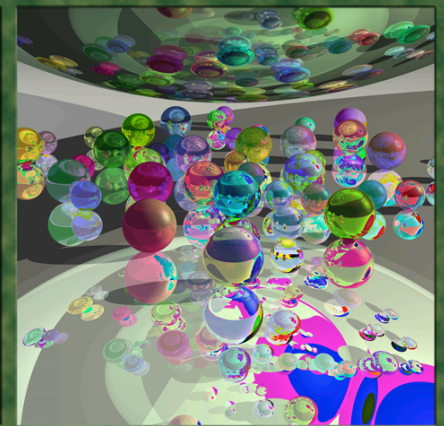
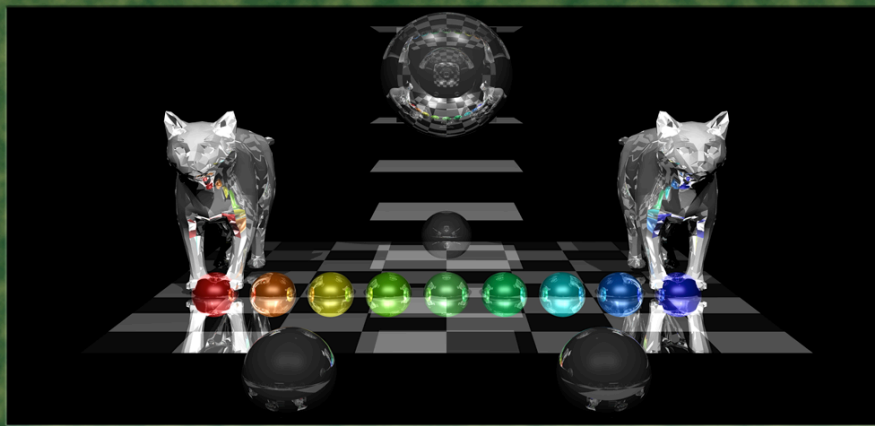
Still, this is what
you can do if you
put your mind to it.





CS410,
Introduction to Computer
Graphics
Fall 2017
Student
Masterworks


Sarah Drisco	Braeden Kammersgar	Tanner Robrke
Ben Sattelberg	Matthew Dunn	Anthony Super
Amy Kaliniski	Laura South	



CS410,
Introduction
to Computer
Graphics
Fall 2018
Student
Masterworks

Abhishek Tefarik	Diego Batres Fornig	Abigail Rictor
Alex Undy	Isaac Law	
Neil Potter	Hiro Kobayashi	Eric Palmer

Real-time Ray Tracing 2012



The screenshot shows a YouTube video player interface. The video content displays a 3D scene with ray-traced glass and destruction. A green NVIDIA logo is visible on a glass surface. The video title is "Nvidia's Kepler real-time raytracing and destruction (1080p) - The Verge". The video has 45,000 views, 268 likes, and 8 comments. The channel name is "The Verge".

www.youtube.com/watch?v=w9SH8xlgzol

Nvidia's Kepler real-time raytracing and destruction (1080p) - The Verge - YouTube

YouTube Search SIGN IN

SUBSCRIBE

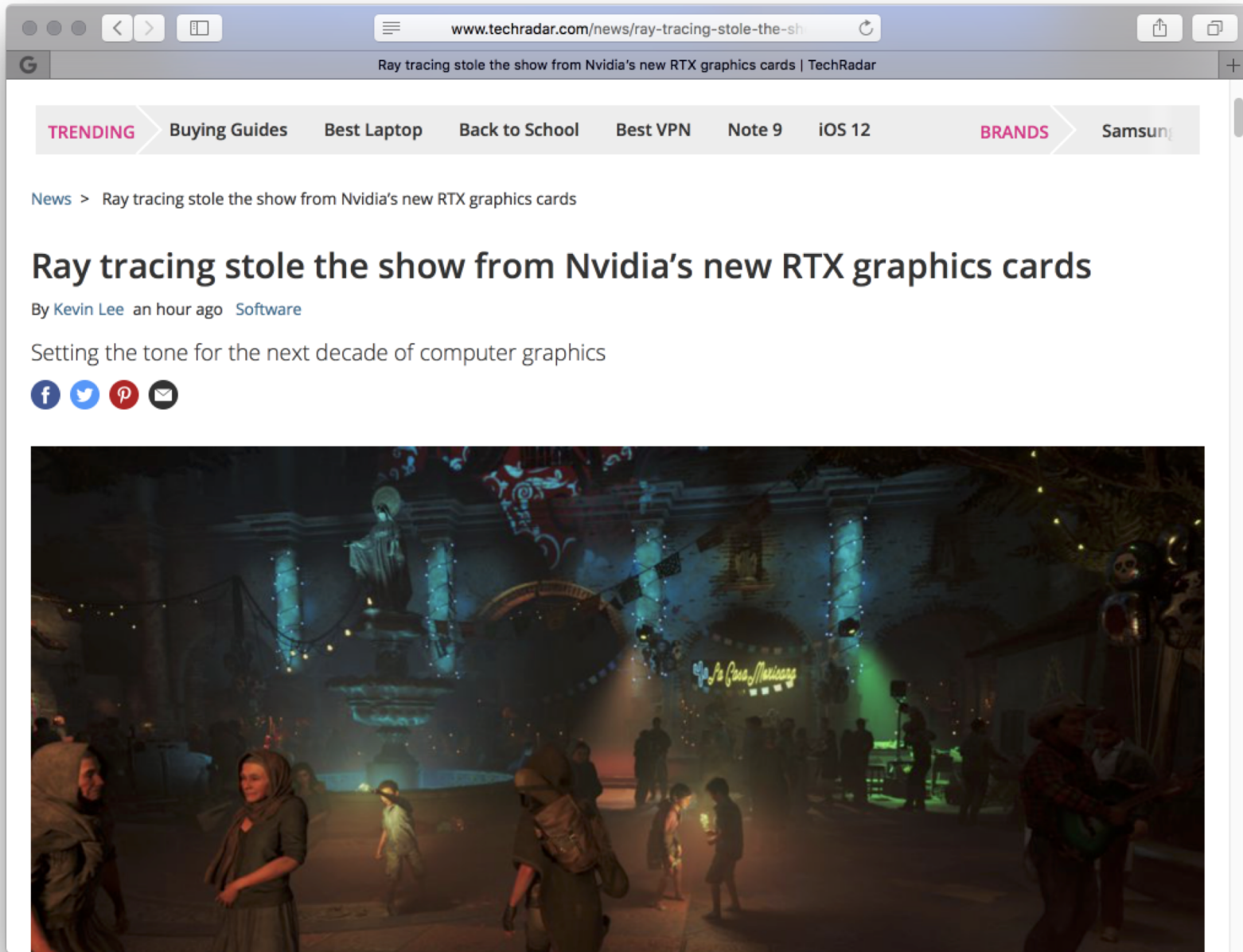
Nvidia's Kepler real-time raytracing and destruction (1080p) - The Verge

45,000 views 268 8 SHARE ...

The Verge

<https://www.youtube.com/watch?v=w9SH8xlgzol>

Real-time Ray Tracing - 2018




The image is a screenshot of a web browser displaying a TechRadar article. The browser's address bar shows the URL www.techradar.com/news/ray-tracing-stole-the-sh. The page title is "Ray tracing stole the show from Nvidia's new RTX graphics cards | TechRadar". A navigation bar includes categories like "TRENDING" (Buying Guides, Best Laptop, Back to School, Best VPN, Note 9, iOS 12) and "BRANDS" (Samsung). The article title is "Ray tracing stole the show from Nvidia's new RTX graphics cards" by Kevin Lee, published an hour ago in the Software category. The sub-headline reads "Setting the tone for the next decade of computer graphics". Below the text are social media sharing icons for Facebook, Twitter, Pinterest, and Email. The main image is a high-quality ray-traced scene of a night festival. It features a large, ornate fountain with a statue in the center, surrounded by people in traditional Middle Eastern attire. The scene is lit with vibrant blue and green lights, and a sign in the background reads "La Casa Mexicana". The overall aesthetic is highly detailed and realistic, showcasing the capabilities of real-time ray tracing.

Real-Time Ray Tracing 2019

YouTube real-time ray tracing in games

JUSTICE different forms of realistic combat in a completely PvP game world.



RTX ON 实时光线追踪反射

6:03 / 8:18

All 11 Games That Support NVIDIA's RTX Ray Tracing Technology

24,522 views

158 24 SHARE SAVE

AllAboutPerspective
Published on Feb 14, 2019

SUBSCRIBE 39

What skills will we need?

- Lots of Math
 - Linear Algebra (Matrices, Vectors, Dot Products)
 - Discrete representations (Images)
- A little Physics
 - Lighting/Energy Transfer
 - Color reflections/refraction
- A little Mechanical Engineering
 - CAD/CAM, Solid Models
- A whole bunch of Computer Science

What do I expect?

- Familiarity with vectors and matrices
 - Mechanics, e.g. multiplication
 - Basic ideas (perhaps rusty)
- Programming ability
 - You know how to design, implement & test thousand line plus programs
 - Test-first development
 - Design patterns
 - Object-oriented designs
- Willingness to work (hard)

Quick Who Does What?

- I will describe graphics principles
 - Mathematics (e.g. projection)
 - Physics (e.g. reflections)
- I will describe some algorithms
 - E.g. ray/polygon intersection
- We will discuss/practice concepts.
- You will convert ideas into code.

One More Thing

The screenshot shows the SageMath website homepage. At the top left is the SageMath logo, which consists of a geometric polyhedron icon and the word "sage" in a stylized font. To the right of the logo are navigation links: "RSS · Blog · Trac · Wiki · Questions? · Donate". Below these are links for "Online: CoCalc · SageCell or Download, Source Code" and a version indicator "v8.0 (2017-07-21)" with social media icons for GitHub, Facebook, and Twitter. A "Language" dropdown menu is also present. A horizontal navigation bar contains links for "Home", "Tour", "Help", "Library", "Download", "Development", and "Links". The main content area features a paragraph describing SageMath as a free open-source mathematics software system licensed under the GPL, built on top of packages like NumPy, SciPy, matplotlib, SymPy, Maxima, GAP, FLINT, and R. It also states the mission: "Creating a viable free open source alternative to Magma, Maple, Mathematica and Matlab." Below this is a call to action box: "Do you want to learn how to use SageMath? Download and read Sage for Undergraduates by Gregory Bard or Calcul mathématique avec Sage (in French)." At the bottom, there are two prominent buttons: "CoCalc (SageMathCloud) or: SageMathCell" and "Download 8.0". The "Download 8.0" button includes links for "Changelogs · Source 8.0 · Packages · Git".

www.sagemath.org

RSS · Blog · Trac · Wiki · Questions? · Donate

Online: CoCalc · SageCell or Download, Source Code

v8.0 (2017-07-21) · GitHub · Facebook · Twitter · Language

Home Tour Help Library Download Development Links

SageMath is a free [open-source](#) mathematics software system licensed under the GPL. It builds on top of many existing open-source packages: [NumPy](#), [SciPy](#), [matplotlib](#), [SymPy](#), [Maxima](#), [GAP](#), [FLINT](#), [R](#) and [many more](#). Access their combined power through a common, Python-based language or directly via interfaces or wrappers.

Mission: *Creating a viable free open source alternative to Magma, Maple, Mathematica and Matlab.*

Do you want to learn how to use SageMath?
Download and read [Sage for Undergraduates](#) by Gregory Bard or [Calcul mathématique avec Sage](#) (in French).

CoCalc (SageMathCloud)
or: SageMathCell

Download 8.0
Changelogs · Source 8.0 · Packages · Git

The End