Lecture 1:
Introduction and Overview
Welcome!
Who Am I?

- Lingqi Yan
  - Pronunciation: ling—chi—yen
  - Assistant Professor
  - Web: www.cs.ucsb.edu/~lingqi/
    Email: lingqi@cs.ucsb.edu
  - Research: Rendering in Computer Graphics
  - Hobbies: research, video games, piano and NBA
About this Course
What is CS291A about?

Real-Time High Quality Rendering

Intermediate level — connecting basic knowledge and research
What is CS291A about?

- Real-Time High Quality Rendering
  - What is Rendering?

3D scene (meshes, lights, etc.)

Calculating light -> eye

Image
What is CS291A about?

- So, we will not cover 3D modeling or game development using Unreal Engine (where can I learn them?)

Modeling character animation in Maya

CSGO PoV Cam set up in Unreal Engine
[https://www.youtube.com/watch?v=3TQ18SmQSw0]
What is CS291A about?

• And we will not cover physically-based animation/simulation (where can I learn this?)

Adaptive Anisotropic Remeshing for Cloth Simulation, Narain et al.
What is CS291A about?

• **Real-Time** High Quality Rendering
  - Speed: more than 30 FPS (frames per second), even more for Virtual / Augmented Reality (VR / AR): 90 FPS
  - Interactivity: Each frame generated **on the fly**

• **Real-Time** **High Quality** Rendering
  - Realism: advanced approaches to make rendering more realistic
  - Dependability: all-time **correctness**, no tolerance to (uncontrollable) failures
What is CS291A about?

• So, we will not cover expensive (but more accurate) light transport techniques in movies / animations (where can I learn this?)

Manifold Metropolis Light Transport
Jakob et al.

Gradient Domain Path Tracing
Kettunen et al.
What is CS291A about?

- And we will **not** cover pure Computer Vision / Deep Learning topics, e.g. “XYZ"GAN (where can I learn this?)

GAN 2.0: NVIDIA’s Hyperrealistic Face Generator (both are fake)
What is CS291A about?

- But we do talk about successful deep learning approaches in rendering!

Neural BTF Compression and Interpolation
Rainer et al.
More topics in this course
Course Topics

- Shadow and Environment Mapping

Real-Time, All-Frequency Shadows in Dynamic Scenes
Annen et al.
Course Topics

• Interactive Global Illumination Techniques

Micro-Rendering for Scalable, Parallel Final Gathering
Ritschel et al.
Course Topics

- Precomputed Radiance Transfer

Global Illumination with Radiance Regression Functions
Ren et al.
Course Topics

• Sampling and Reconstruction

Fast 4D Sheared Filtering for Interactive Rendering of Distribution Effects
Yan et al.
Course Topics

- Real-Time Ray Tracing
Course Topics

- Image-based Rendering and Light Fields

Composition with input image and textures of virtual objects

Filtering Environment Illumination for Interactive Physically-Based Rendering in Mixed Reality
Mehta et al.
Course Topics

- Participating Media Rendering, Image Space Effects, etc.

Multiple scattering  
Image space reflection
Course Topics

• Non-Photorealistic Rendering

The Legend Of Zelda: Breath Of The Wild's

Animal Crossing: New Horizons
Questions?
Course Logistics
Class Philosophy

• We want a very active class
• Come to class
• Follow, think, discuss and question
Prerequisites

• **Strong interest** in Graphics, Rendering

• Computer Graphics experience
  - CS180, CS280 or CS285 in our department
  - HCI or CV courses will not help
  - A brief survey
  - What if lacking prerequisites? Next slide

• Course will move quickly
  - Need not to fully follow each one
    (but doing so will be most rewarding)
If Lacking Prerequisites

• You may **NOT** have time to catch up with introductory level Computer Graphics knowledge

• But if you have taken a CG course before, you can catch up with basic OpenGL usage soon

• The next lecture will briefly review related topics

• Project 0 (**optional**, will be released after next lecture) will help you warm up quickly

• Practice makes perfect!
Course Logistics

• Course Website
  - Everything’s there!

• No required textbooks
  - Related papers will be available online before lectures
  - Lecture slides (in PDF) will be available right after class
  - The book “Real-Time Rendering (3rd ed or later)” by Moller and Haines may be helpful (we will not follow it, though)
Course Logistics

• No TA for this class
  - Sign up on Piazza for student discussion

• Office hour
  - Mondays 4PM - 5PM, same zoom address
  - I probably can’t help you with debugging

• Academic integrity
  - Work alone, including the final project (no copy-pasting from others)
  - Do not publish your code (on Github, etc.) (results are welcome)
  - Other details on the website, strictly enforced
Assignments and Grading

• Assignments
  - **Choose 1** paper to present in class (20%)
  - **Choose 2** from 4 potential projects (20% each, topics are given)
  - Final project (40%, still by yourself)
  - NO EXAMS!

• Grading
  - Submit your project by 11:59PM on/before the due dates by sending me emails (MD5 is fine)
  - Each late day = 10% off
Assignments and Grading

• More words about projects
  - Getting a “working example” is the most important
    - Minimize distractions w.r.t. C++ syntax and OpenGL usage, and focus on core shader implementations
    - Software engineering is not necessary
    - Hard code is definitely acceptable
  - START EARLY!
    - You’ll have 2 weeks 3 weeks for each project (if you take it)
Zoom Policies

• First of all
  - These policies only apply to this course

• Do not distribute our zoom link to anyone who’s not enrolled

• Audio / Video
  - Please keep your microphone muted unless you are speaking
  - But feel free to turn on/off your video at any time

• Recording
  - You are NOT allowed to record and/or publicize lectures from this course
  - Again, PDF slides will be available after class
Enrolling Policies

- Dept. policy
  - Cannot enroll more when the course is full

- Procedure
  - Ask me for approval by providing your prerequisites
  - Contact Karen (our graduate student advisor) for a petition form
  - Ask me to sign for it
Questions?
Today’s Lecture
Outline

• Motivation

• Evolution of real-time rendering

• Technological and algorithmic milestones
  - Programmable graphics hardware
  - Precomputation-based methods
  - Interactive Ray Tracing
Motivation

- Today, Computer Graphics is able to generate photorealistic images
  - Complex geometry, lighting, materials, shadows
  - Computer-generated movies/special effects (difficult or impossible to tell real from rendered…)

[Artist: Teruyuki and Yuka]
[Artist: Hyun Kyung]
Motivation

• But accurate algorithms (esp. ray tracing) are very slow
  – So they are called offline rendering methods
  – Guess how long does it take to render one frame in Zootopia?

Zootopia, Disney Animation
Motivation

• With proper approximations, we can generate plausible results but runs much faster

Toyota 2000GT, from TurboSquid (offline rendering)

Need for Speed: Payback (real-time rendering)
Evolution of Real-Time Rendering

• Interactive 3D graphics pipeline as in OpenGL
  - Earliest SGI machines (Clark 82) to today
  - Most of focus on more geometry, texture mapping
  - Some tweaks for realism (shadow mapping, accum. buffer)

![SGI Reality Engine 93 (Kurt Akeley)]
Evolution of Real-Time Rendering

• 20 years ago
  - Interactive 3D geometry with simple texture mapping, fake shadows (OpenGL, DirectX)

Evolution of Real-Time Rendering

- 20 -> 10 years ago
  - A giant leap since the emergence of programmable shaders (2000)
  - Complex environment lighting, real materials (velvet, satin, paints), soft shadows

Assassin’s Creed II (2009)

Resident Evil 5 (2009)
Evolution of Real-Time Rendering

• Today
  - Extended to Virtual Reality (VR) and even movies
  - “Stunning graphics”

Beat Saber, VR Game

Zafari, animation series rendered completely using Unreal game engine
Evolution of Real-Time Rendering

• Today
Evolution of Real-Time Rendering

• Today

(2018) - Real-Time Ray Tracing Demo, NVIDIA
Evolution of Real-Time Rendering

• In the future

The Matrix (1999 movie)
Evolution of Real-Time Rendering

- In the future

Ready Player One (2018 movie)
Thank you!