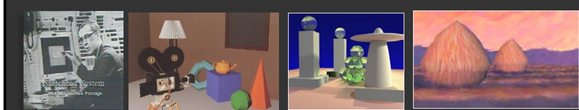
Computer Graphics

CSE 167 [Win 24], Lecture 1: Overview and History

Ravi Ramamoorthi

http://viscomp.ucsd.edu/classes/cse167/wi24



1

Goals

- **Systems:** Write complex 3D graphics programs (real-time scene in OpenGL, offline raytracer)
- **Theory:** Mathematical aspects and algorithms underlying modern 3D graphics systems
- This course is ***not*** about the specifics of 3D graphics programs and APIs like Maya, Alias, DirectX but about the concepts underlying them.

2

Instructor

Ravi Ramamoorthi http://www.cs.ucsd.edu/~ravir

- PhD Stanford, 2002. PhD thesis developed "*Spherical Harmonic Lighting*" widely used in games (e.g. Halo series), movies (e.g. Avatar), etc. (Adobe, …)
- At Columbia 2002-2008, UC Berkeley 2009-2014
- *"Monte Carlo denoising"* inspired raytracing offline, real-time
- At UCSD since Jul 2014: Director, Center for Visual Computing
- Awards for research: White House PECASE (2008), SIGGRAPH Significant New Researcher (2007), ACM Fellow
- https://www.youtube.com/watch?v=qpyCXqXGe7I
- Have taught Computer Graphics 10+ times
- Computer Graphics online MOOC (CSE 167x) has had 100,000+ registrations, 700,000 video views. Finalist for two inaugural edX Prizes. Will use UCSD Online, auto-feedback

3

MOOC Introductory Video

4

Course Staff

- Ravi Ramamoorthi
- Teaching Assistants:
  - Mustafa Yaldiz (myaldiz@ucsd.edu)
  - Nithin Raghavan (n2raghavan@ucsd.edu)
  - Liwen Wu (liw026@ucsd.edu)
  - Zhongyi Wang (zhw039@ucsd.edu)

5

Why Study 3D Computer Graphics?

- Applications (discussed next)
- Fundamental Intellectual Challenges

Some content inspired by Pat Hanrahan from Stanford's CS148

6

Entertainment

Movies: Brave, Pixar 2012

7

Entertainment

Movies: Turning Red, Pixar 2022

8

Entertainment

Games: Halo 3, Bungie 2007

9

Entertainment

Games: Halo Infinite, Bungie 2021

10

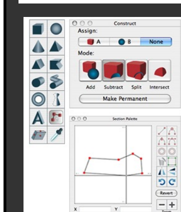
Lighting Simulation

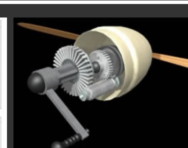
Interior Design

Automobile Visualization

11

Computer Aided Design





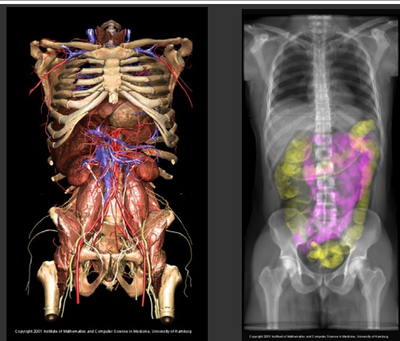
Mechanical CAD
Architectural CAD
Electronics CAD
Casual Users

Interiors Professional

Google Sketchup

12

Visualization: Science and Medicine



Visible Human Project: University of Hamburg

13

Virtual Reality

- VR for design and entertainment
- Simulators: Surgical, Flight, Driving, Spacecraft

14

Virtual Reality: Metaverse

15

Digital Visual Media

- From text to images to video (to 3D?)
- Image and video processing and photography
- Multimedia computers, tablets, phones
- Flickr, YouTube, WebGL
- Real, Virtual Worlds (Google Earth, Metaverse)
- Electronic publishing
- Online gaming
- 3D printers and fabrication

16

NEURAL RADIANCE FIELDS

17

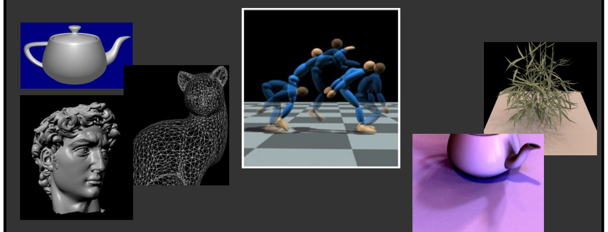
Why Study 3D Computer Graphics?

- Applications
- Fundamental Intellectual Challenges
  - Create and interact with realistic virtual world
  - Requires understanding of all aspects of physical world
  - New computing methods, displays, technologies
- Technical Challenges
  - Math of (perspective) projections, curves, surfaces
  - Physics of lighting and shading
  - 3D graphics software programming and hardware

18

3D Graphics Pipeline

Modeling → Animation → Rendering



19

3D Graphics Pipeline

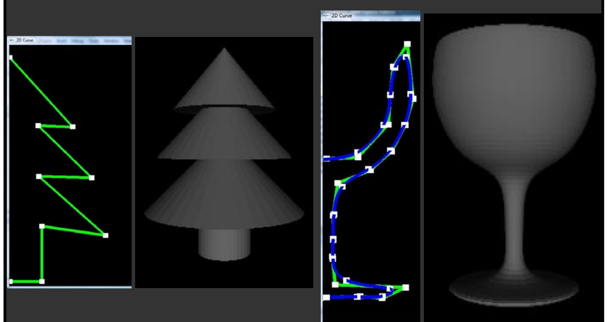
Modeling → Animation → Rendering

HW 1: Transformations (Jan 26)
Place objects in world, view them
Simple viewer for a teapot

HW 3: Curves (Feb 28)
Bezier and B-Spline curves
To model and draw objects

20

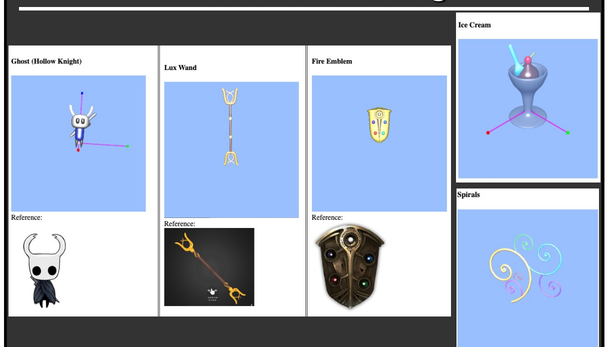
Curves for Modeling



Rachel Shiner, Final Project Spring 2010

21

Curves for Modeling



James Zhao. HW 3 Extra Credit Winter 2023

22

3D Graphics Pipeline

Modeling → Animation → Rendering

HW 1: Transformations (Jan 26)
Place objects in world, view them
Simple viewer for a teapot

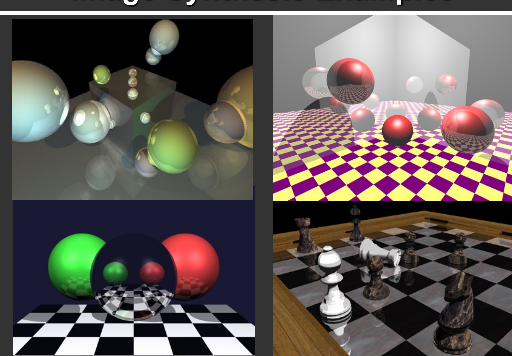
HW 3: Curves (Feb 28)
Bezier and B-Spline curves
To model and draw objects

HW 2: Scene Viewer (Feb 14)
View scene, Lighting and Shading
(with GLSL programmable shaders)

HW 4: RayTracer (Mar 19)
Realistic images with ray tracing
(two basic approaches: rasterize
And raytrace images [HW 2,4])

23

Image Synthesis Examples



Collage from 2007

24

Logistics

- Website http://viscomp.ucsd.edu/classes/cse167/wi24 has most of the information (look at it carefully)
- *We will be leveraging MOOC infrastructure in a SPOC*
  - Please sign up for account at UCSD online, join course: **DEMO**
  - UCSD online compulsory for most assignments, feedback systems
  - Optional for video lectures (class may differ a bit, more), problems
  - *Must still submit "official" CSE 167 assignment (see website)*
  - Please do ask us if you are confused; we are here to help
  - No required texts; OpenGL programming guide, GLSL optional
- Office hours: Tu/Thu
  - See website for sections, TA office hours.
- Course newsgroup on Piazza
- Website for late, collaboration policy, etc
- Questions?

25

This is a Modernized Course

- Modern 3D Graphics Programming with GPUs
  - Modern OpenGL (3+), GLSL 330 core
  - Real-time feedback servers for all homeworks
- GLSL + Programmable Shaders from HW 1
- Should be very portable, but need to set up your environment, compilation framework (HW 0)

NVIDIA Fermi, image from Pat Hanrahan

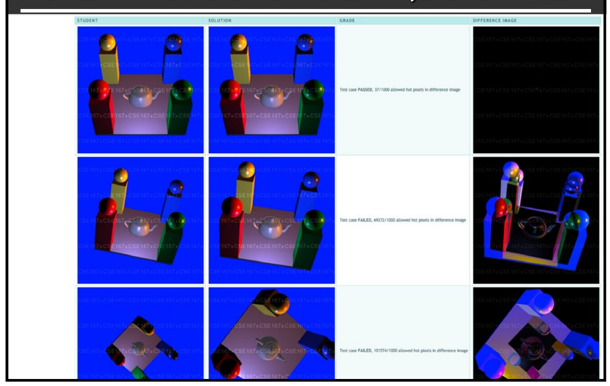
26

Innovation: Feedback Servers

- Feedback/Grading servers for all homeworks
- Submit images and/or code, compare to original
  - Program generates difference images, report url
  - Can get feedback multiple times; submit final url
  - All (except curves homework 3) run on UCSD Online
- "Feedback" not necessarily grading
  - Can run extra test cases, look at code, grade fairly
  - But use of feedback servers/UCSD Online is mandatory
- Will test out immediately with HW 0 images
  - HW 1 - 2 will have both code and image feedbacks
  - Can use any (laptop/desktop) computer. We also try to have the basement labs fully set up.

27

Demo of edX/UCSD Online, Feedbacks



28

Online Lectures

- Online lectures and screencasts for most course:
  - http://viscomp.ucsd.edu/classes/cse167/wi24/index.html
  - (with English *and Chinese!* Subtitles [courtesy XuetangX])
  - Review for CSE 167 (but still have regular classes)
  - For general interest (share with non-CS 167 students)
- Originally recorded in 2012 for MOOC offering
  - CAVEAT: Does not include all material (curves)
  - Was updated in 2017 for more recent OpenGL
  - Same as video lectures on UCSD online (some errata)
- Currently view lectures as complementary
  - Hence, viewing them optional (e.g. miss a class)
  - Please note caveats; "official" CSE 167 is in class
- UCSD Canvas lectures from last year

29

Workload

- Lots of fun, rewarding but may involve significant work
- 4 programming projects (+HW 0); almost all are time-consuming (individual except HW 4). START EARLY !!
- Course will involve understanding of mathematical, geometrical concepts taught (tested on midterm)
  - *No final; will do a take-home small assignment instead*
- Grade mostly programming, weights on website
  - Ignore weighting on UCSD online site; we weight as on CSE 167 site
- Prerequisites: Solid *C*/C++/Java/Python programming background. Linear algebra (review on Thu) and general math skills. No knowledge of graphics/OpenGL needed.
  - Should be able to pick up C/C++, and look up some OpenGL
- Should be a difficult, but fun and rewarding course

30

CSE 167 is only a first step

- *If you enjoy CSE 167 and do well*:
- In Spring: I teach CSE 168 (Rendering continues)
- Next winter: CSE 169 (Animation)
- CSE 165 (VR)
- Graduate: CSE 274 (Topics), many 291s, 273 (Computational Photography)

31

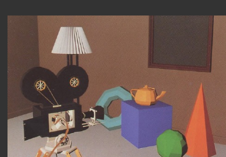
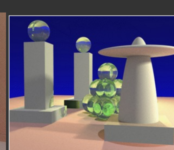
To Do

- Look at website
- Various policies for course. E-mail if confused.
- Sign up for UCSD Online, Piazza, etc.
- Skim assignments if you want. All are ready
- Assignment 0, Due Jan 17 next week (see website). [both parts needed, total 10 points]
- Set up compilation framework in HW 0, feedback
  - Also test HW1, HW 2, HW 3 (curves)
  - Make sure you can compile multi-file from scratch (HW4)
- Any questions?

32

History

- Brief history of significant developments in field
- End with a video showcasing graphics

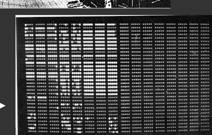



The term Computer Graphics was coined by William Fetter of Boeing in 1960
First graphic system in mid 1950s USAF SAGE radar data (developed MIT)

33

How far we've come: TEXT

Manchester Mark I

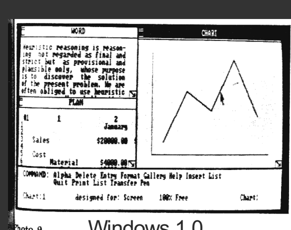


Display

34

From Text to GUIs

- Invented at PARC circa 1975. Used in the Apple Macintosh, and now prevalent everywhere.



35

Drawing: Sketchpad (1963)

- Sketchpad (Sutherland, MIT 1963)
- First interactive graphics system (VIDEO)
- Many of concepts for drawing in current systems
  - Pop up menus
  - Constraint-based drawing
  - Hierarchical Modeling

36

Paint Systems

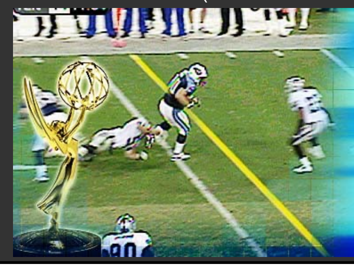
- SuperPaint system: Richard Shoup, Alvy Ray Smith (PARC, 1973-79)

- Nowadays, image processing programs like Photoshop can draw, paint, edit, etc.

37

Image Processing

- Digitally alter images, crop, scale, composite
- Add or remove objects
- Sports broadcasts for TV (combine 2D and 3D processing)

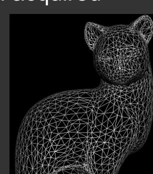


38

Modeling

- Spline curves, surfaces: 70s – 80s
- Utah teapot: Famous 3D model

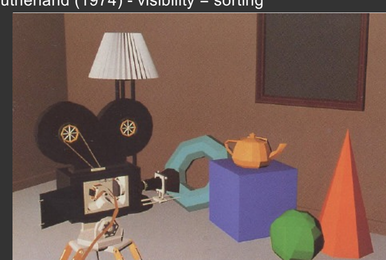
- More recently: Triangle meshes often acquired from real objects

39

Rendering: 1960s (visibility)

- Roberts (1963), Appel (1967) - hidden-line algorithms
- Warnock (1969), Watkins (1970) - hidden-surface
- Sutherland (1974) - visibility = sorting



Images from FvDFH, Pixar's Shutterbug
Slide ideas for history of Rendering courtesy Marc Levoy

40

Rendering: 1970s (lighting)

1970s - raster graphics

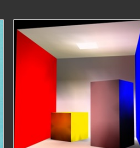
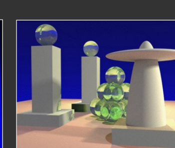
- Gouraud (1971) - diffuse lighting, Phong (1974) - specular lighting
- Blinn (1974) - curved surfaces, texture
- Catmull (1974) - Z-buffer hidden-surface algorithm

41

Rendering (1980s, 90s: Global Illumination)

early 1980s - global illumination

- Whitted (1980) - ray tracing
- Goral, Torrance et al. (1984) radiosity
- Kajiya (1986) - the rendering equation

42

History of Computer Animation

- 10 min clip from video on history of animation
- http://www.youtube.com/watch?v=LzZwiLUVaKg

- Covers sketchpad, animation, basic modeling, rendering
- A synopsis of what this course is about
- (watch offline if short on time)

43