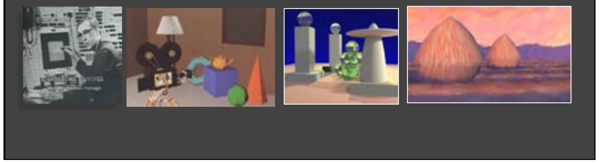
Foundations of Computer Graphics (Fall 2012)

CS 184, Lecture 1: Overview and History

Ravi Ramamoorthi

http://inst.eecs.berkeley.edu/~cs184



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.

Demo: Surreal (HW 4)

Makiko Yasui and Dixon Koesdjojo, Spring 2003

Course Staff

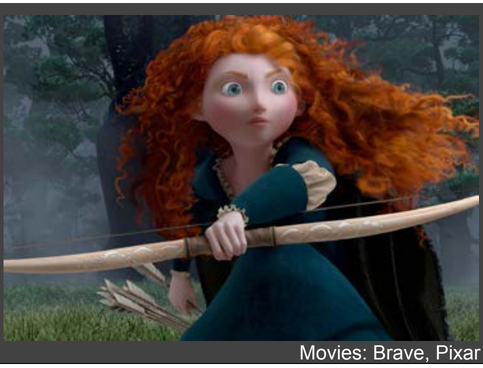
- Ravi Ramamoorthi http://www.cs.berkeley.edu/~ravir
  - PhD Stanford, 2002. PhD thesis developed "Spherical Harmonic Lighting" widely used in games (e.g. Halo series), movies (e.g. Avatar), etc. (Adobe, …)
  - At Columbia 2002-2008, research on rendering/image synthesis, data-driven appearance. Normal Mapping Video
  - At Berkeley since Jan 2009. 3rd time teaching 184. New this semester: modern 3D graphics programs with shaders
- Teaching Assistants: cs184@imail.eecs.berkeley.edu
  - Fu-Chung Huang
  - Brandon Wang
  - Christine Nguyen
  - Nicholas Estorga (grader, feedback servers)

Why Study 3D Computer Graphics?

- Applications (discussed next)
- Fundamental Intellectual Challenges

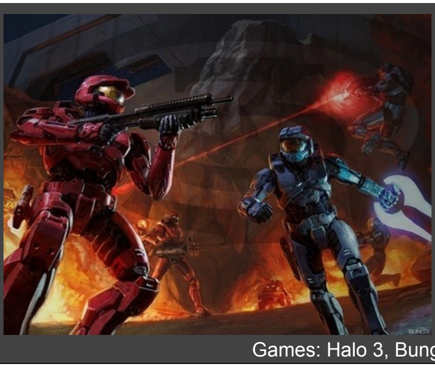
Some content inspired by Pat Hanrahan from Stanford's CS148

Entertainment



Movies: Brave, Pixar 2012

Entertainment



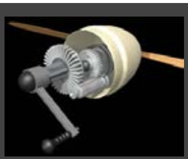
Games: Halo 3, Bungie 2007

Lighting Simulation

Interior Design

Automobile Visualization

Computer Aided Design

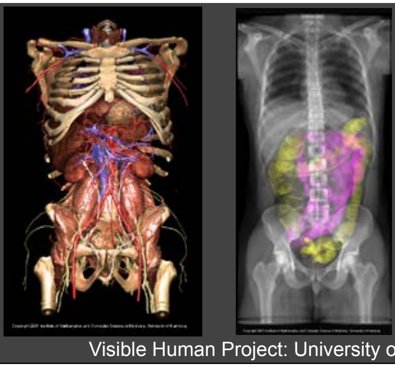


Mechanical CAD
Architectural CAD
Electronics CAD
Casual Users

Interiors Professional

Google Sketchup

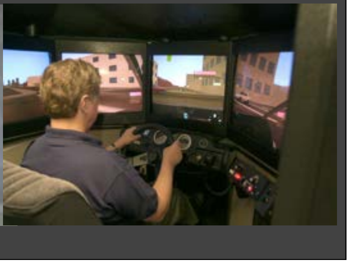
Visualization: Science and Medicine



Visible Human Project: University of Hamburg

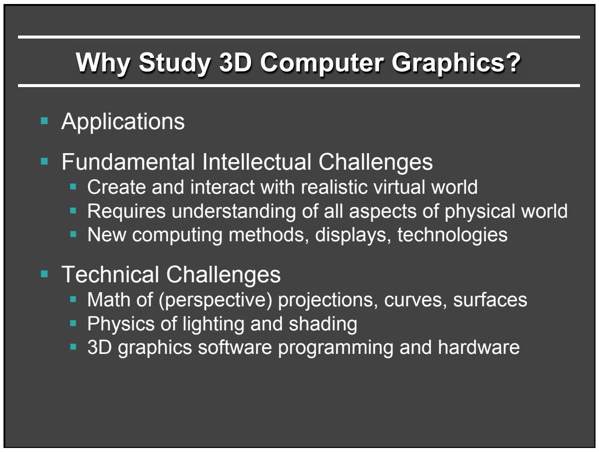
Virtual Reality

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



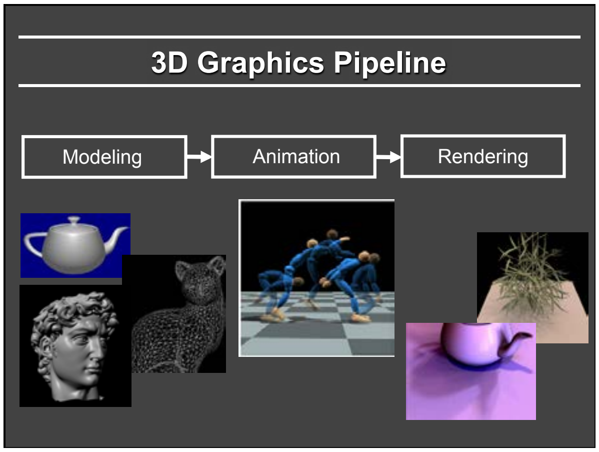
Digital Visual Media

- 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, Second Life)
- Electronic publishing
- Online gaming
- 3D printers and fabrication

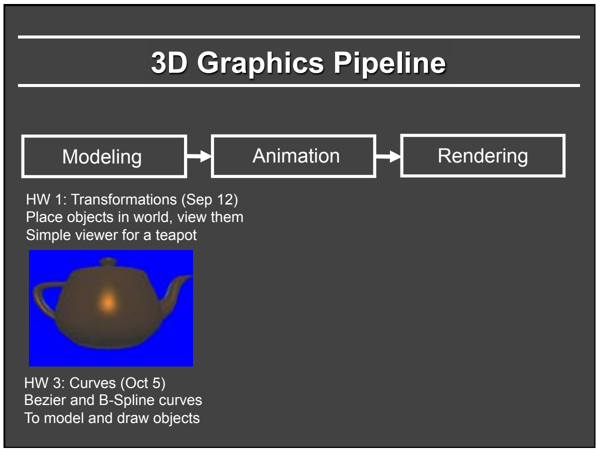


Why Study 3D Computer Graphics?

- 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



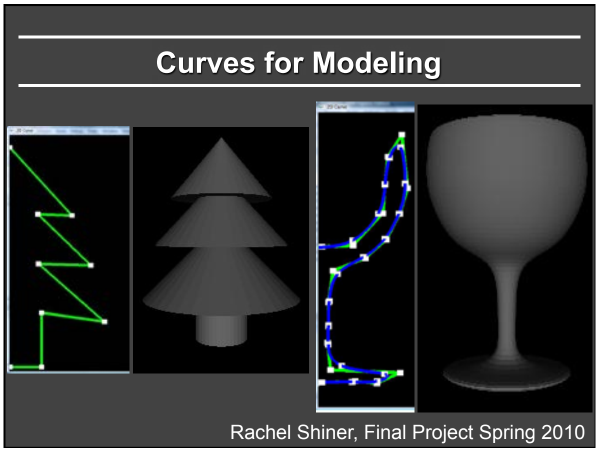
3D Graphics Pipeline



3D Graphics Pipeline

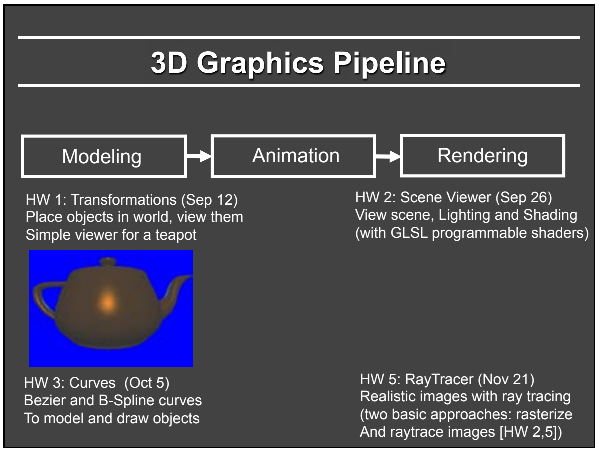
HW 1: Transformations (Sep 12)
Place objects in world, view them
Simple viewer for a teapot

HW 3: Curves (Oct 5)
Bezier and B-Spline curves
To model and draw objects



Curves for Modeling

Rachel Shiner, Final Project Spring 2010



3D Graphics Pipeline

HW 1: Transformations (Sep 12)
Place objects in world, view them
Simple viewer for a teapot

HW 2: Scene Viewer (Sep 26)
View scene, Lighting and Shading
(with GLSL programmable shaders)

HW 3: Curves (Oct 5)
Bezier and B-Spline curves
To model and draw objects

HW 5: RayTracer (Nov 21)
Realistic images with ray tracing
(two basic approaches: rasterize
And raytrace images [HW 2,5])

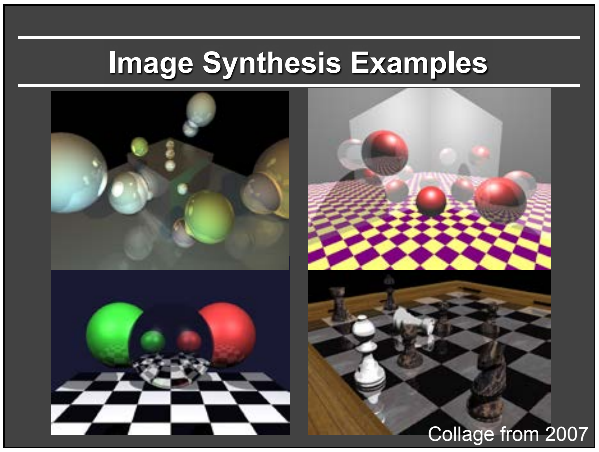
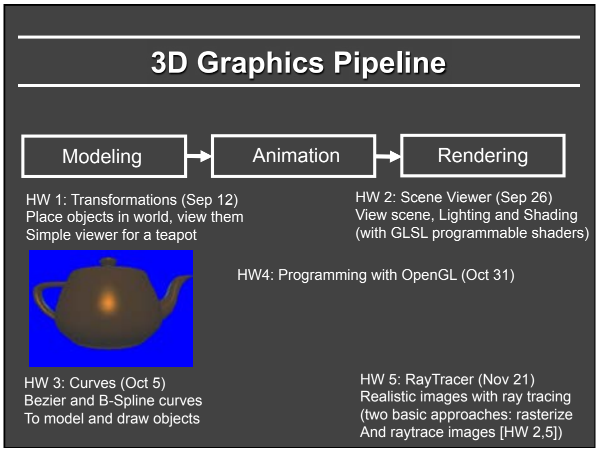


Image Synthesis Examples

Collage from 2007



3D Graphics Pipeline

Modeling → Animation → Rendering

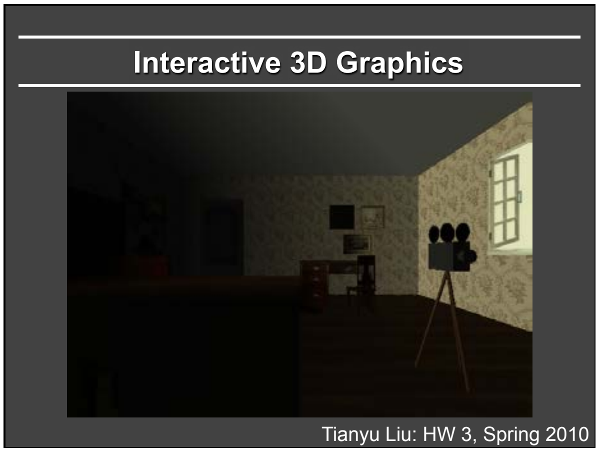
HW 1: Transformations (Sep 12)
Place objects in world, view them
Simple viewer for a teapot

HW 2: Scene Viewer (Sep 26)
View scene, Lighting and Shading
(with GLSL programmable shaders)

HW4: Programming with OpenGL (Oct 31)

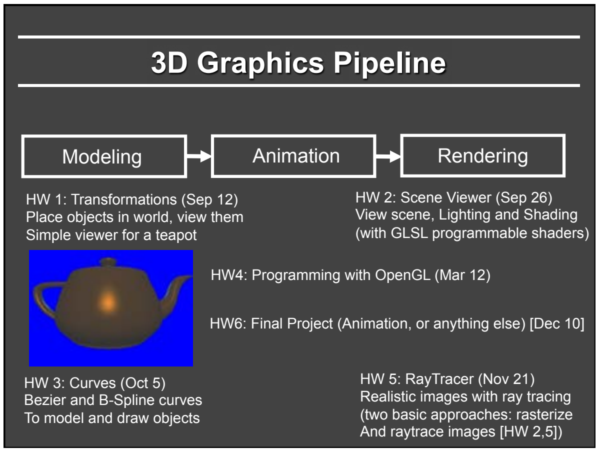
HW 3: Curves (Oct 5)
Bezier and B-Spline curves
To model and draw objects

HW 5: RayTracer (Nov 21)
Realistic images with ray tracing
(two basic approaches: rasterize
And raytrace images [HW 2,5])



Interactive 3D Graphics

Tianyu Liu: HW 3, Spring 2010



3D Graphics Pipeline

Modeling → Animation → Rendering

HW 1: Transformations (Sep 12)
Place objects in world, view them
Simple viewer for a teapot

HW 2: Scene Viewer (Sep 26)
View scene, Lighting and Shading
(with GLSL programmable shaders)

HW4: Programming with OpenGL (Mar 12)

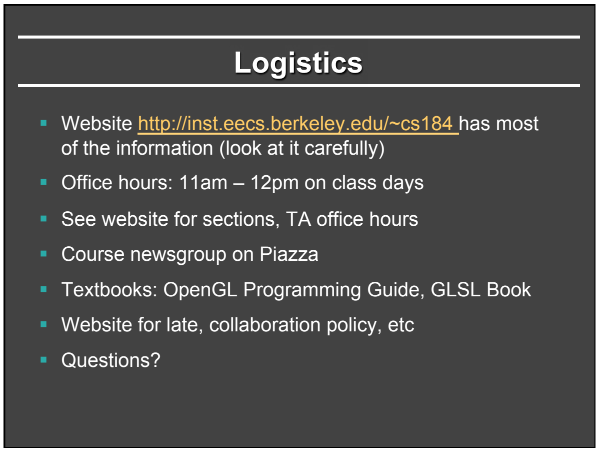
HW6: Final Project (Animation, or anything else) [Dec 10]

HW 3: Curves (Oct 5)
Bezier and B-Spline curves
To model and draw objects

HW 5: RayTracer (Nov 21)
Realistic images with ray tracing
(two basic approaches: rasterize
And raytrace images [HW 2,5])

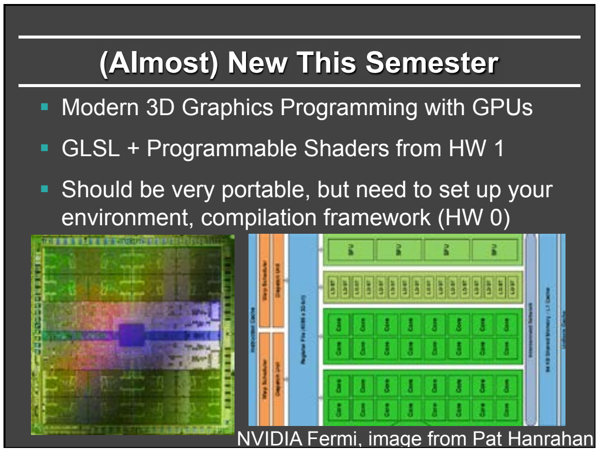
Final Project

John Ng and Andrea Goh, Spring 2010



Logistics

- Website http://inst.eecs.berkeley.edu/~cs184 has most of the information (look at it carefully)
- Office hours: 11am – 12pm on class days
- See website for sections, TA office hours
- Course newsgroup on Piazza
- Textbooks: OpenGL Programming Guide, GLSL Book
- Website for late, collaboration policy, etc
- Questions?



(Almost) New This Semester

- Modern 3D Graphics Programming with GPUs
- 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

(Almost) New: Feedback Servers

- Feedback/Grading servers for HW 0,1,2,3,5
- Submit images and/or code, compare to original
  - Program generates difference images, report url
  - Can get feedback multiple times; submit final url
- “Feedback” not necessarily grading
  - Can run extra test cases, look at code, grade fairly
  - But use of feedback servers is mandatory
- Will test out immediately with HW 0 images
  - HW 1 - 3 will have both code and image feedbacks

Demo of HW 0 Feedback (Nick)

- Instructions posted on website and on Piazza

New: Online Lectures

- Online lectures and screencasts for first half: http://inst.eecs.berkeley.edu/~cs184/fa12/onlinelectures.html
- Three main goals for online screencasts
  - Review for CS 184 (but still have regular classes)
  - For general interest (share with non-CS 184 students)
  - Hope to teach an online class on EdX soon
- Currently view screencasts as complementary
  - Hence, viewing them optional (but recommended)
  - May be minor differences from in-class lectures
- Subscribe to YouTube channel for updates

Workload

- Lots of fun, rewarding but may involve significant work
- 6 programming projects; almost all are time-consuming (but you have groups of two for later projects 4,5). START EARLY !!
- Course will involve understanding of mathematical, geometrical concepts taught (tested on midterm, final)
- Prerequisites: Solid C/C++/Java programming background. Linear algebra (review on Tue) and general math skills. No knowledge of graphics/OpenGL needed.
- Should be a difficult, but fun and rewarding course

To Do

- Look at website
- Various policies for course. E-mail if confused.
- Skim assignments if you want. All are ready
- Assignment 0, Due Aug 29(a), 31(b) next week (see website). Compilation and Photo [both are essential, counts for total of 20 points]
- Set up compilation framework in HW 0, feedback
- Any questions?

History

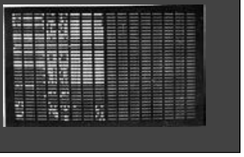
- Brief history of significant developments in field
- End with a video showcasing graphics

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)

How far we' ve come: TEXT

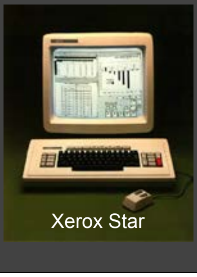
Manchester Mark I

Display →

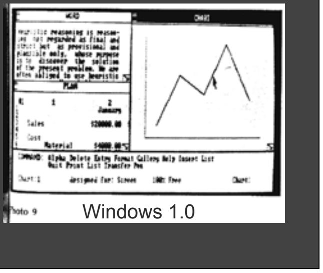


From Text to GUIs

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



Xerox Star



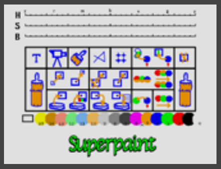
Windows 1.0

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

Paint Systems

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



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

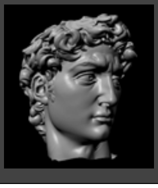
Image Processing

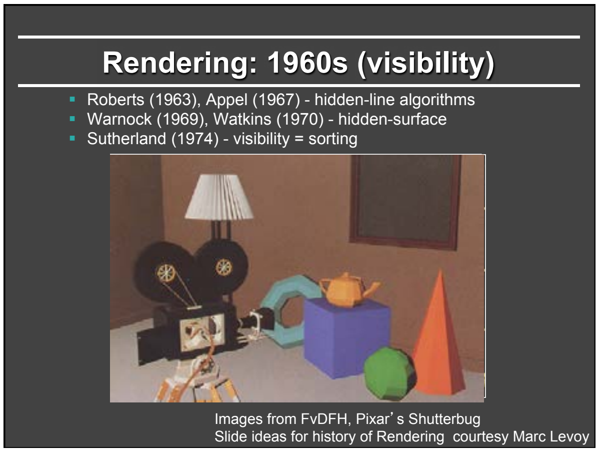
- Digitally alter images, crop, scale, composite
- Add or remove objects
- Sports broadcasts for TV (combine 2D and 3D processing)

Modeling

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

- More recently: Triangle meshes often acquired from real objects

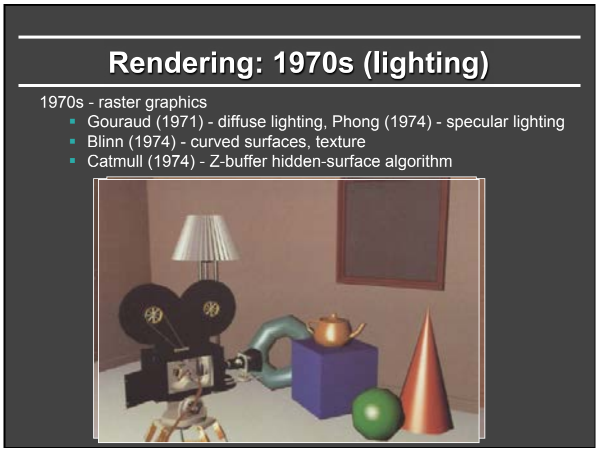




Rendering: 1960s (visibility)

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

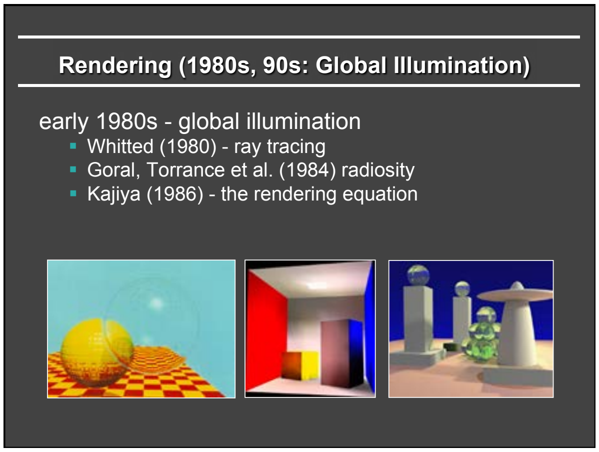
Images from FvDFH, Pixar's Shutterbug
Slide ideas for history of Rendering courtesy Marc Levoy



Rendering: 1970s (lighting)

1970s - raster graphics

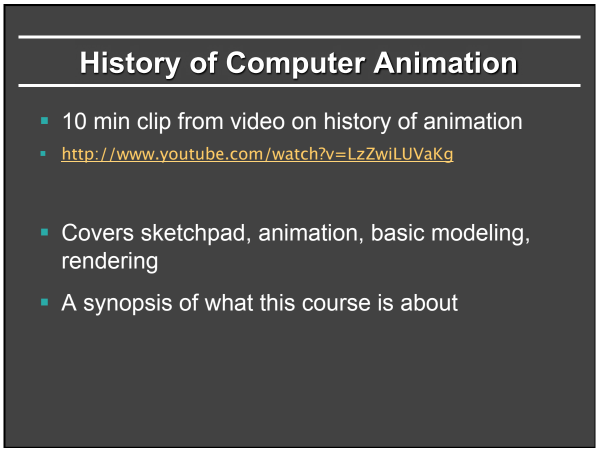
- Gouraud (1971) - diffuse lighting, Phong (1974) - specular lighting
- Blinn (1974) - curved surfaces, texture
- Catmull (1974) - Z-buffer hidden-surface algorithm



Rendering (1980s, 90s: Global Illumination)

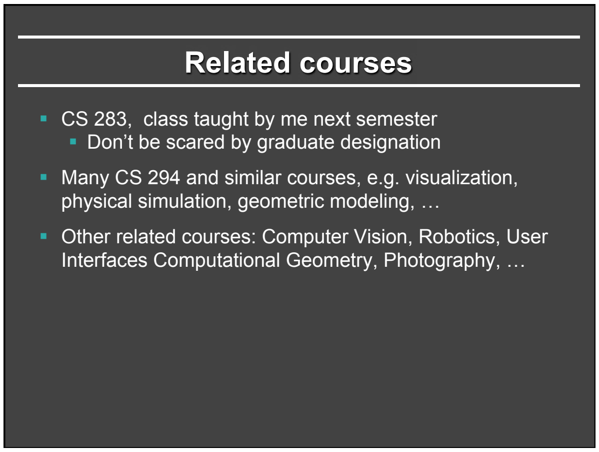
early 1980s - global illumination

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



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



Related courses

- CS 283, class taught by me next semester
  - Don't be scared by graduate designation
- Many CS 294 and similar courses, e.g. visualization, physical simulation, geometric modeling, …
- Other related courses: Computer Vision, Robotics, User Interfaces Computational Geometry, Photography, …