Foundations of Computer Graphics
Online Lecture 1: Overview and History

Motivation: Why do we study 3D Graphics?

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

Instructor
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, UC Berkeley since Jan 2009
- http://www.cs.berkeley.edu/~ravir/RavR.wmv
- Have taught Computer Graphics 10+ times

Goals
- **Systems:** Write complex 3D graphics programs (real-time scene viewer 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 like Maya, Alias, DirectX but about the concepts underlying them. You will write programs in OpenGL/GLSL

Why Study 3D Computer Graphics?
- Applications (discussed next)
- Fundamental Intellectual Challenges

Applications
- Movies
- Games
- Computer Aided Design (CAD)
- Lighting Simulation (Interiors, Automobiles, …)
- Visualization (Scientific, Medical)
- Virtual Reality

Image Synthesis Examples
Collage from 2007
Digital Visual Media

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

Why Study 3D Computer Graphics?

- 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, hardware

Foundations of Computer Graphics

Online Lecture 1: Overview and History

Course Outline and Logistics

Ravi Ramamoorthi

3D Graphics Pipeline

Modeling  Animation  Rendering

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

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

HW 3: RayTracer
Realistic images with ray tracing
(two basic approaches: rasterize And raytrace images [HW 2,3])

Assignment Logistics

- HW 0 immediately to check compilation etc.
- Feedback/Grading servers for all HW
- Submit images, compared with originals
  - Program generates images automatically for you
  - Can submit multiple times for feedback
- Skeleton code in C++/OpenGL/GLSL
  - Programming background in C/C++/Java needed
  - No prior knowledge of 3D graphics/OpenGL required
### Workload
- Lots of fun, rewarding but may involve significant work
- 3 programming projects; almost all are time-consuming
- Course will involve understanding of mathematical, geometrical concepts taught (tested on final)
- Prerequisites: Solid C/C++/Java programming.
- Linear algebra (review next lecture) and basic math skills

### A Note on GPU Programming
- Modern 3D Graphics Programming with GPUs
- GLSL + Programmable Shaders in HW 0,1,2
- Should be very portable, but need to set up your environment, compilation framework (HW 0)

### Foundations of Computer Graphics

**Online Lecture 1: Overview and History**  
*Brief History of Computer Graphics*

Ravi Ramamoorthi

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.

### Drawing: Sketchpad (1963)
- Sketchpad (Sutherland, MIT 1963)
- First interactive graphics system
  - http://www.youtube.com/watch?v=mOZqRJzE8xg
- Many of concepts for drawing in current systems
  - Pop up menus
  - Constraint-based drawing
  - Hierarchical Modeling
**Paint Systems**

- Precursor to Photoshop: general image processing

**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: 70s – 80s
- Utah teapot: Famous 3D model
- More recently: Triangle meshes often acquired from real objects

**Rendering: 1960s (visibility)**

- Hidden Line Algorithms: Roberts (63), Appel (67)
- Hidden Surface Algorithms: Warnock (69), Watkins (70)
- Visibility = Sorting Sutherland (74)

**Rendering: 1970s (lighting)**

- Diffuse Lighting (Gouraud 1971)
- Specular Lighting (Phong 1974)
- Curved Surfaces, Texture (Blinn 1974)
- Z-Buffer Hidden Surface (Catmull 1974)

**Rendering (1980s, 90s: Global Illumination)**

- early 1980s - global illumination
  - Whitted (1980) - ray tracing
  - Goral, Torrance et al. (1984) radiosity
  - Kajiya (1986) - the rendering equation
<table>
<thead>
<tr>
<th>History of Computer Animation</th>
</tr>
</thead>
<tbody>
<tr>
<td>- 10 min clip from video on history of animation</td>
</tr>
<tr>
<td>- <a href="http://www.youtube.com/watch?v=LzZwiLUVaKg">http://www.youtube.com/watch?v=LzZwiLUVaKg</a></td>
</tr>
<tr>
<td>- Covers sketchpad, animation, basic modeling, rendering</td>
</tr>
<tr>
<td>- A synopsis of what this course is about</td>
</tr>
</tbody>
</table>