

Lecture 0



CS480 Computer Graphics

- Pre-requisites:
 - **Good programming skills in C/C++**
- Policies: No late homework (**-25% a day**)
- Grading:
 - HW Assignments: 60%
 - Midterm: 10%
 - Final Exam: 30%

Books and References

- 3D Computer Graphics, Alan Watt
- Computer Graphics, Foley and Van Dam
- OpenGL Programming Guide
- OpenGL Reference Manual
- <http://www.opengl.org>

Goals for this semester

- Learn the fundamental techniques involved in computer graphics
- Learn the standard computer graphics API, OpenGL
- Get an overview of the various subfields in computer graphics
- Touch on some state-of-the-art trends in Computer Graphics

Lecture 1 (Fundamentals)

- Computer Graphics is making pictures from models using the computer
- Computer Graphics involves a lot of mathematics (Linear Algebra)
- Points, Normals and operations on them, dot product, cross product
- Transformations: Translation, Rotation, Scale (the basic rigid transformations)

Lecture 2 (Rotations)

- Rotations around an arbitrary axis can be represented by a set of rotations around the world axis
- Other types of transformations: Shearing, Tapering, Twisting, Bending (Deformation)

Lecture 3 (Coordinate Systems)

- Transformation between Coordinate systems
- Object, World, Camera, Normalized Device, Window
- Perspective and Orthographic projection
- Midpoint line drawing algorithm

Lecture 4 (Rasterizing Polygons)

- Half-space Test
- Jordan Curve Theorem
- Scan-line Algorithm
- Flood Fill

Lecture 5 (Visibility)

- Clipping
 - Cohen Sutherland
 - Liang Barsky
- Culling
 - Backface Culling

Lecture 6 (Shading)

- Global Illumination
 - Ray tracing/casting
 - Radiosity
- Local Illumination
 - Phong Model
 - Interpolation (of Color, Normals)

Lecture 7 (Visibility 2)

- Hidden surface removal
- Object Space Algorithms
 - Warnock
 - BSP
 - Portals
- Image Space Algorithms
 - Ray casting
 - Painters
 - Z-buffering

Lecture 8 (OpenGL)

- Graphics API standard
- Has an Architectural Review board
- OpenGL is a state machine
- OpenGL is a primitive-based renderer (vertices)
- OpenGL is independent of the windowing system

Lecture 9 (Sampling/Antialiasing)

- Problems with undersampling == aliasing
- Shannon's Sampling Theorem
- Fourier Transforms
- Convolution
- Anti-aliasing Strategies
 - Band-limiting
 - Pre-filtering
 - Super Sampling (uniform/stochastic)

Lecture 10 (Texture Mapping)

- The S map
- The O map
- Mip-mapping (image pyramids)
- Environment map(reflective surface), refraction map, opacity map, illumination map, bump map, displacement map
- 3D Textures (via noise and turbulence)

Lecture 11(Shadows)

- Umbra, Penumbra
- Projected shadows (to a plane)
- Shadow Volumes
- Shadow maps

Lecture 12 (Surfaces)

- Representing smooth surface geometry using a finite set of control points and a set of basis functions
- Bezier Curves, B-Splines, Uniform B-splines, Non-uniform B-splines, Non-uniform Rational B-splines (NURBS), Subdivision curves and surfaces, Implicit Surfaces.

Lecture 13 (Animation)

- The animation loop
- Methods of specifying Animation
 - Scripting
 - Keyframing
 - Inverse kinematics
 - Behavioral animation
 - Dynamics
- Speed Control of interpolating a curve
 - Arc Length for position
 - Slerp (Quaternion) for rotation

Lecture 14 (Deformation)

- Geometric
 - Splines
 - Wires, SSD, FFD, PSD, RBF
- Physically Based
 - Mass-spring
 - FEM

Lecture 15 (Vertex Shading)

- Offshoot from shading languages
- Other uses of Vertex shading
 - Non-graphical uses
 - Advanced lighting models
 - Vertex blending
 - Texture Generation
- Similar to writing Assembly
- Equivalent to writing code to another processor

Lecture 16 (CG)

- Programmable shading splintered the API since it was specific to each graphics card
- Solution: Use a compiler to perform cross-compilation to the different assembly of each graphics cards
- A testbed for the standardization of programmable shading in OpenGL

And who knows what else...

- More to computer graphics than these subjects alone
- A whole slew of new research bringing new ideas to computer graphics applications
- Seeing all this effort trickle into the final product (games and graphics cards)
- Welcome to the world of Computer Graphics