Animation

(Not Even an Introduction)

Chapter 23

Animation

- animated flipbooks and film (19th century)
- principles of animation (1930's and 40's)

- to animate =
  to give life to an inanimate object, image or drawing
  - *anima*: means soul in Latin
- <u>Animation</u>: the art of movement expressed with images that are not taken directly from reality
- In animation, the illusion of movement is achieved by rapidly displaying many still images (or frames) in sequence.

Keyframing and in-betweening

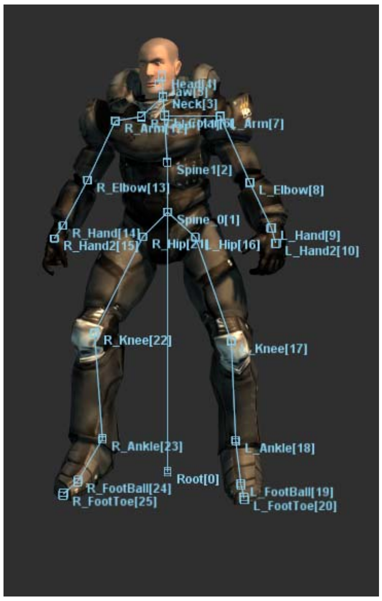
- Keyframing
  - is used to define an animated sequence based on its key moments.
  - Hand-drawn animation: keyframes or extremes
  - Stop-motion animation: key poises
- In-betweening
  - is used once the keyframes have been established and drawn, creates all the transition or in-between drawings that fill the gaps between the keyframes.
  - Interpolation
    - Interpolation of position and orientation
    - Interpolation of shape
    - Interpolation of attributes

Luxo Jr. (1986)

PIXAR

Skinning

- Start with complicated mesh we want to animate
- Rigging: design a geometric hierarchical skeletal structure
- Associate each vertex with one "bone" (RBT node)
- Idea: express the vertex in its bone frame
- Manipulate the skeleton
- Vertices now move along with the bones



Details

- Instead of actually changing the vertex coordinates,
- A rest accumulated matrix represents the relationship between the object and rest bone frame.
- A new accumulated matrix represents the relation after the bone has been moved.
- Use these matrices, in the vertex shader to move the vertex.
- In this context, we can associate a vector of bone-weights and do soft skinning.
  - We use the weights to blend the updates.
- This is used extensively in games and would make a nice project
- Example: Aleka McAdams, Yongning Zhu, Andrew Selle, Mark Empey, Rasmus Tamstorf, Joseph Teran, and Eftychios Sifakis. 2011. Efficient elasticity for character skinning with contact and collisions. In *ACM SIGGRAPH 2011*

Simulation

- Physics uses equations to describe physical processes.
- We can try to simulate these processes computationally.
- Techniques: physics and computational mathematics.
- Some methods are slow and only work for offline animation.
- Some methods can be made real-time
- Hard to control the output

Physically-based modeling

- Techniques that use physics to create realistic animations automatically

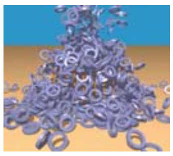
Particle System

[R83]

Flocking System

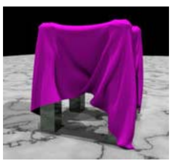
[TT94]

Rigid Body System



[GBF03]

Mass Spring System



[BFA02]

Compressible Fluid Flow

[FSJ01]

Incompressible Fluid Flow

[EMF02]

Particle simulation systems

- Simplest version of physics.
- A large bunch of non-interacting particles.
- Ordinary differential equation (ODE) for the time evaluation of a point:

$$f = ma = m\dot{v} = m\ddot{x} \qquad p = mv$$

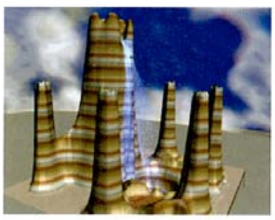
- Force might be gravity or wind
- Can model flowing fall of water particles, or a stream of smoke particles in the air
- Typically each particle is rendered as a semitransparent little blob or surface

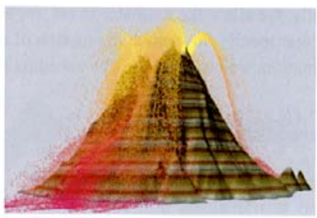
ODE integration

- Starting from an initial condition, we can discretize this ODE and march forward in time using so-called *Euler steps:*

$$x_{t+h} = x_t + v_t h$$
$$v_{t+h} = v_t + a_t h$$
$$a_{t+h} = f(x_{t+h}, t+h)/m$$

- Steps must be small (often need many more than 30/sec).
- There is a whole literature of more sophisticated ways to solve an ODE.





- Example:
  - Tobias Pfaff, Nils Thuerey, Jonathan Cohen, Sarah Tariq, and Markus Gross. 2010. Scalable fluid simulation using anisotropic turbulence particles. In ACM SIGGRAPH Asia 2010 papers (SIGGRAPH ASIA '10).

Rigid Bodies

- Upgrade from particles to solid hard finite objects (dice rolling on a table)
- Need to deal with rotational issues
- With to deal with interaction: collision detection
  - Bounding hierarchies
  - Must undo interpenetration
  - Must have the object bounce (this requires hacked physics since real objects slightly deform and undeform
- Must deal with objects resting on objects and not endles sly bouncing

Cloth

- Can be modeled as a grid of particles connected by springs
- Can be modeled as mesh of physical triangular elements
- Need forces to avoid stretching and shearing and oscillation
- Example:

Zhili Chen, Renguo Feng, and Huamin Wang. 2013. Modeling friction and air effects between cloth and deformable bodies. ACM Trans. Graph. 32, 4

Hair

- Hair modeling is also often similarly dealt with as a mass-spring model

- Example:

Aleka McAdams, Andrew Selle, Kelly Ward, Eftychios Sifakis, and Joseph Teran. 2009. Detail preserving continuum simulation of straight hair. In ACM SIGGRAPH 2009 papers (SIGGRAPH '09)

Deformable materials

- Real objects are deformable
- Can be modeled as volumetric objects (mesh of 3D tetrahedra)

- Example:

Chris Wojtan, Nils Thürey, Markus Gross, and Greg Turk. 2009. Deforming meshes that split and merge. In ACM SIGGRAPH 2009 papers (SIGGRAPH '09)

Fire and water

- Special physical equations
- Modeled with combination of equations
- Example

Jeffrey N. Chadwick and Doug L. James. 2011. Animating fire with sound. In ACM SIGGRAPH 2011 papers (SIGGRAPH '11)

Human locomotion

- Not passive objects
- Much harder than previously discussed phenomenon
- Ideas are used from robotics, control, and optimization
- Nowadays motion capture data is relied on heavily, and possibly altered or used as part of the rocket science.
- Example:

Jack M. Wang, David J. Fleet, and Aaron Hertzmann. 2010. Optimizing walking controllers for uncertain inputs and environments. In ACM SIGGRAPH 2010 papers (SIGGRAPH '10)

Introduction to Computer Graphics

CS380

<Lecture 1>

CS380

- This course provides a broad introduction to the field of 3D computer graphics.
- The goal of this course is to learn **how to form images by computer.**
  - We will study the basic methods used to define shapes, materials and lighting when creating computer-generated images for use in film, games and other applications.
  - Covered topics include affine and projective transformations, viewing, shading, lighting, texture mapping, modeling, animation and 3D interactive applications.
- Through a series of OpenGL programming assignments, students will become familiar with interactive 2D and 3D graphical display concepts.

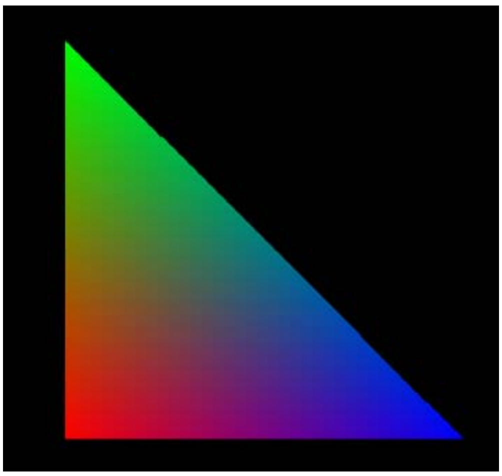
Computer Graphics

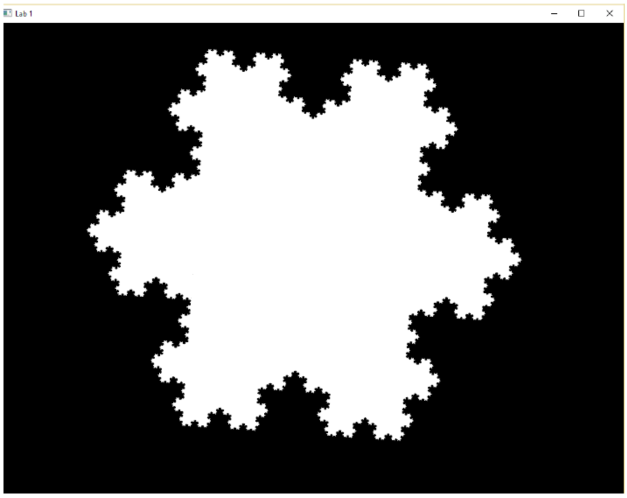
- Concerns with all aspects of producing pictures or images using a computer
- Algorithm for visual simulation
- Applications of Computer Graphics
  - Display of information
  - Design
  - Simulation and animation
  - User interfaces

Main Theme

- Imaging
  - Representing 2D images
- Modeling
  - Representing 3D objects
- Rendering
  - Constructing 2D images from 3D models
- Animation
  - Simulating changes over time

Homework & Lab

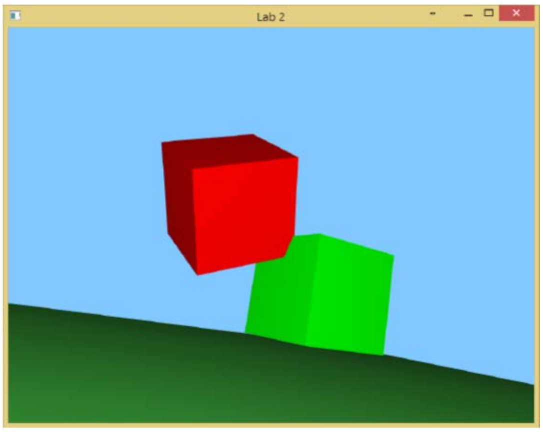


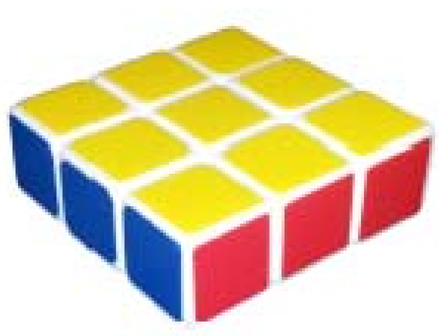


- Snowflake 2D animation

Homework & Lab

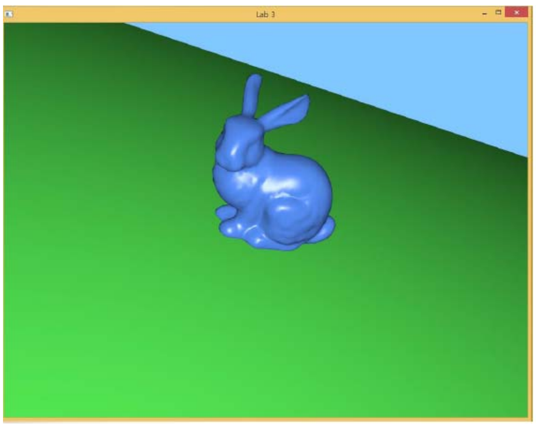
- Object Manipulation with Arcball Interface
- 3x3x1 Floppy Cube Manipulation with Arcball Interface

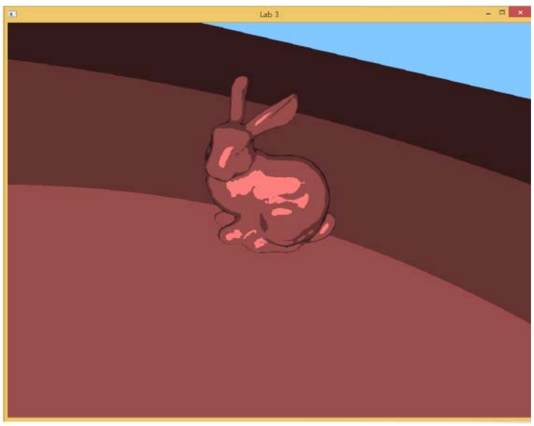




Homework & Lab

- Multiple Lights Animation with Multiple Shaders

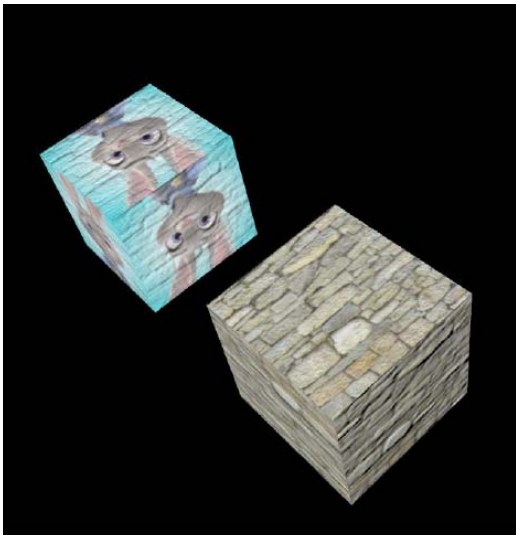


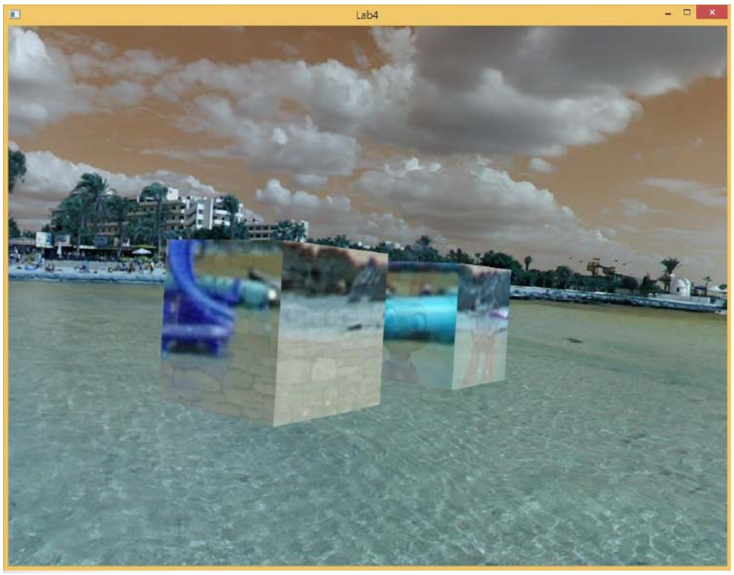


Homework #5 Due June 14 (Midnight)
Late submission allowed only upto *June 21*.

Homework & Lab

- AWESOME virtual 3D Rubik's cube





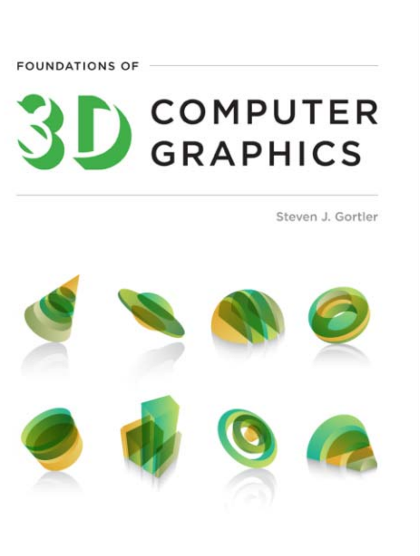
Contest

<optional but highly-recommended!>

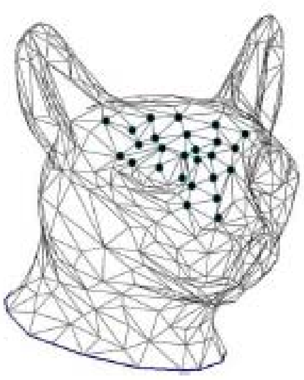
- Best image of a virtual floppy cube
  - Created by you
  - 512 x 512 (any standard image format is fine)
  - High quality rendered output
  - Title & short description will be nice to have
  - Enter by June 15 Midnight to KLMS
  - **Winner(s)** will be announced at the final on June 21 and receive a prize!

3D Computer Graphics

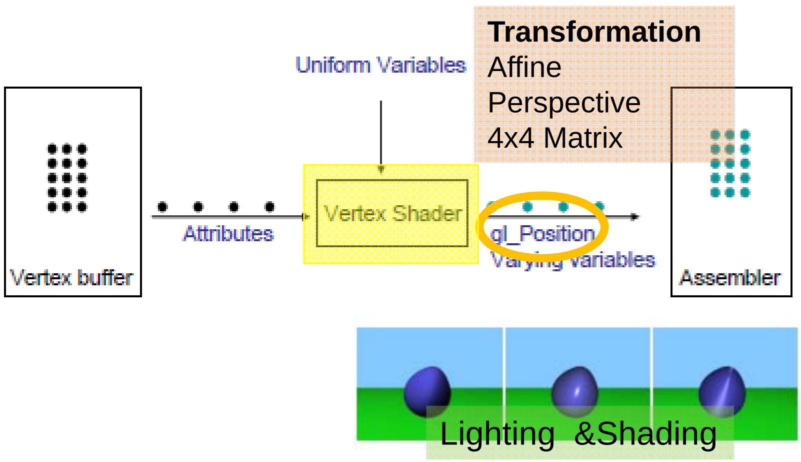
- I. Getting Started
  - **Introduction** / Linear / Affine / Respect
  - **Frames in graphics** / Hello World 3D
- II. Rotations and Interpolation
  - Quaternions / **Balls: Track and Arc**
  - Smooth interpolation
- III. Cameras and Rasterization
  - **Projection / Depth**
  - From vertex to pixel / Varying variables
- IV. Pixels and Such
  - **Materials / Texture mapping**
  - Sampling / Reconstruction / Resampling
- V. Advanced Topics
  - Color / **What's ray tracing? /** Light
  - Geometric modeling / Animation
- Appendix
  - Hello World 2D
  - Affine functions

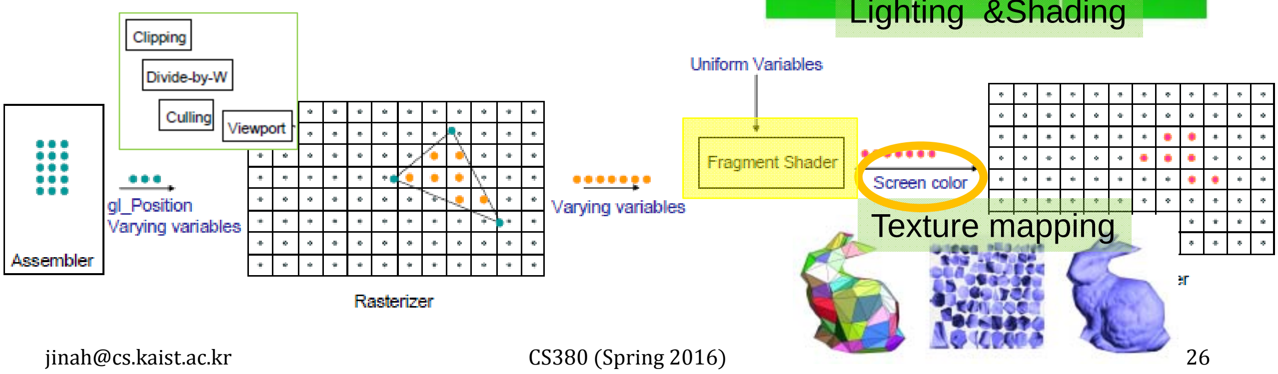


Ch.1 Introduction

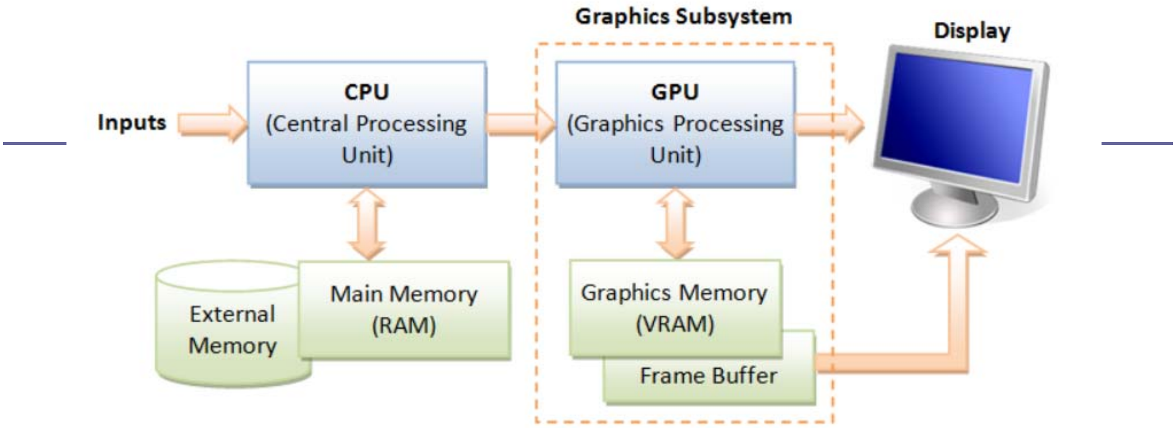


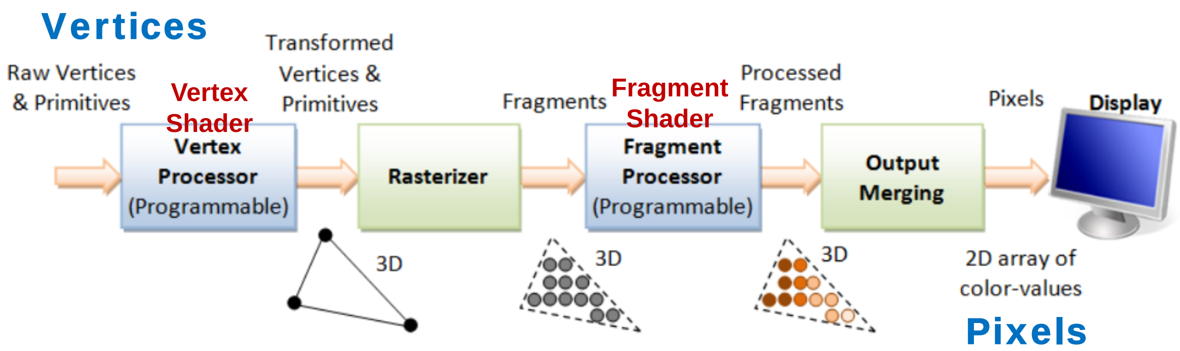
$$\tilde{p} = \vec{\mathbf{o}}^t \mathbf{c} = \vec{\mathbf{w}}^t O \mathbf{c}$$
$$= \vec{\mathbf{e}}^t E^{-1} O \mathbf{c}$$

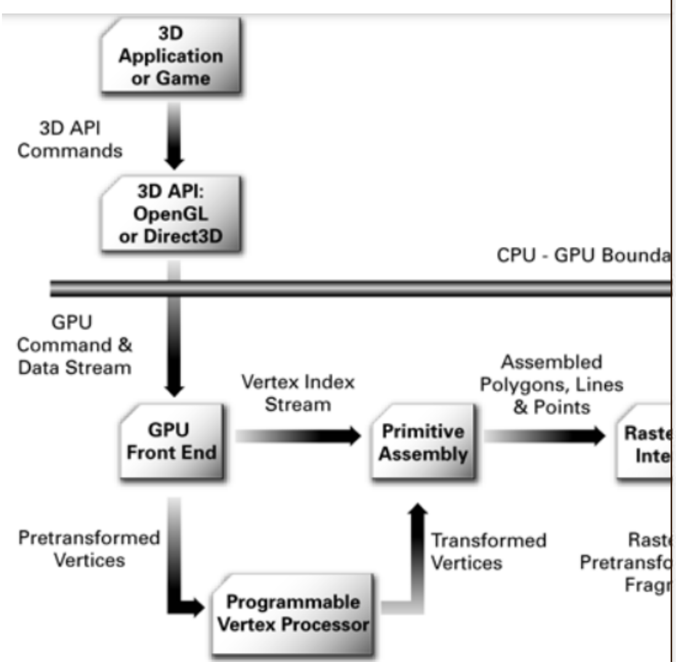


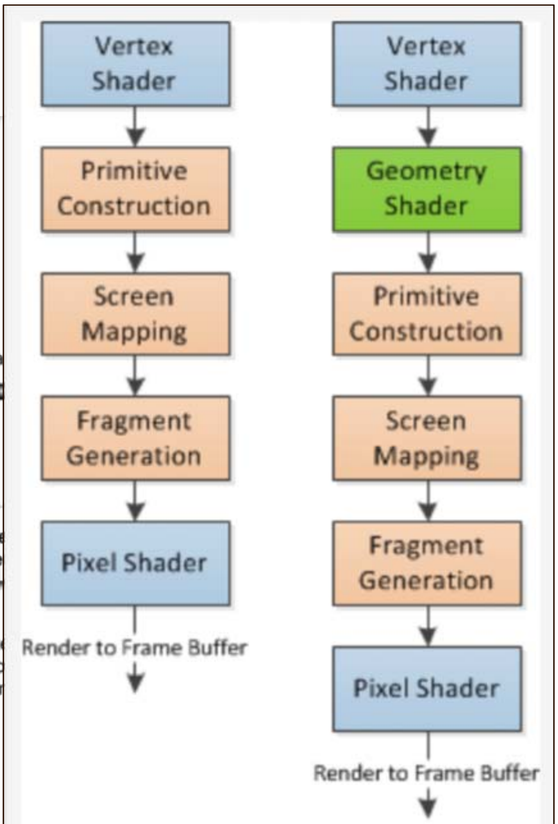


https://www.ntu.edu.sg/home/ehchua/programming/opengl/CG_BasicsTheory.html



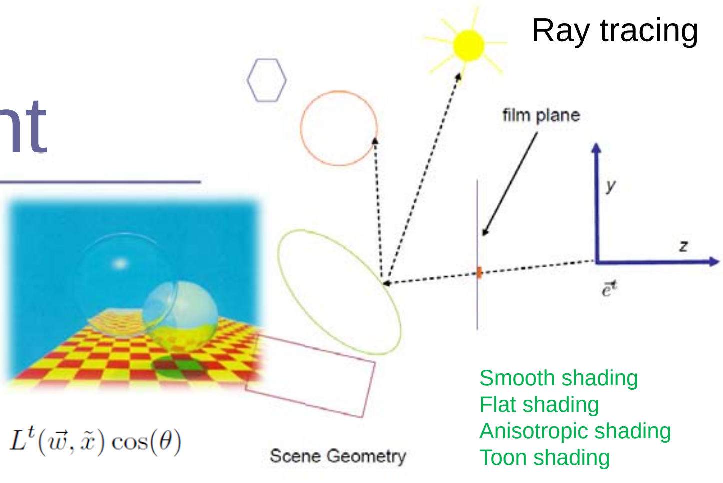






The geometry shader as part of the graphical pipeline.

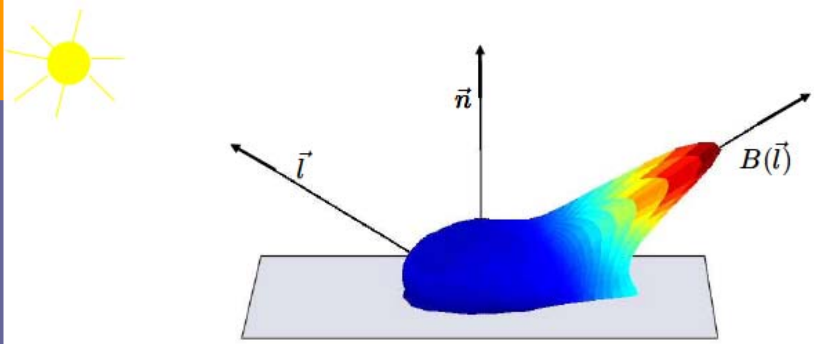
Material / Light



Rendering equation

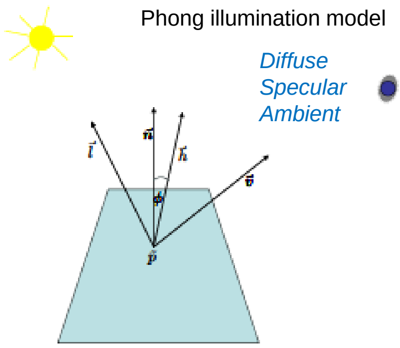
$$L^t(\tilde{x},\vec{v}) \;=\; L^e(\tilde{x},\vec{v}) + \int_H dw\; f_{\tilde{x},\vec{n}}(\vec{w},\vec{v})\; L^t(\vec{w},\tilde{x})\cos(\theta)$$

Smooth shading
Flat shading
Anisotropic shading
Toon shading
…



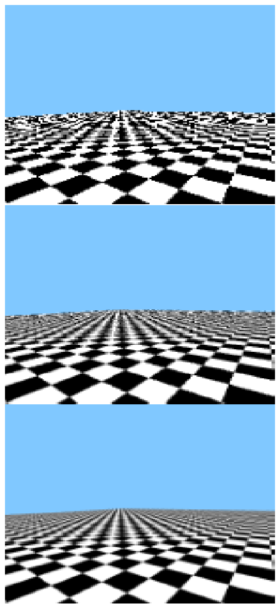
Phong illumination model

Diffuse
Specular
Ambient

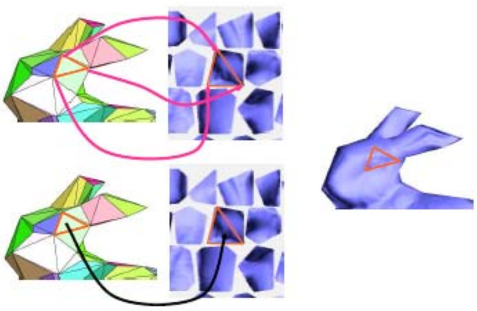


Texture mapping

Sampling problems

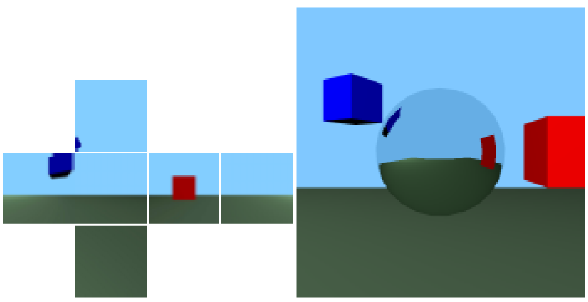


Normal map
Bump map

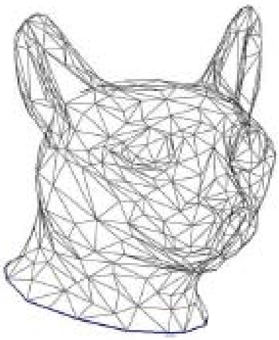


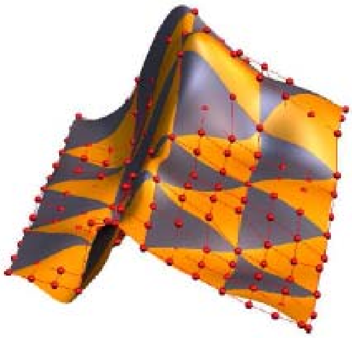
Shadow map

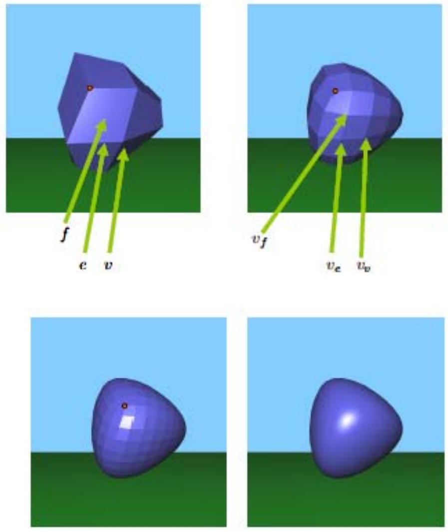
Environment map



Modeling

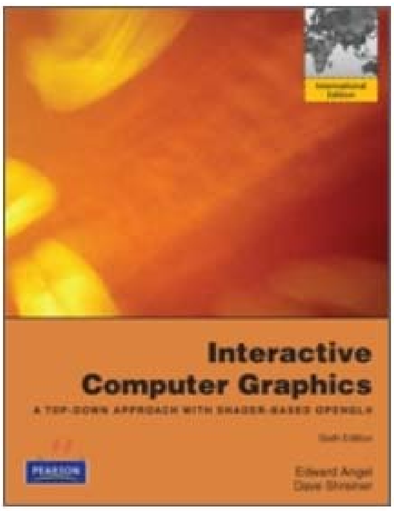


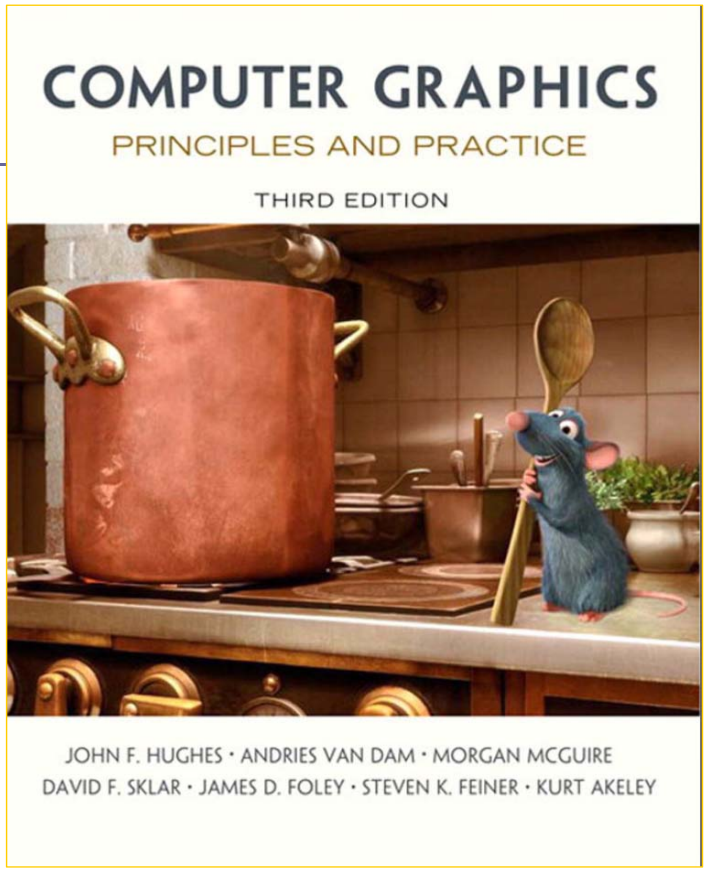




Book Recommendation

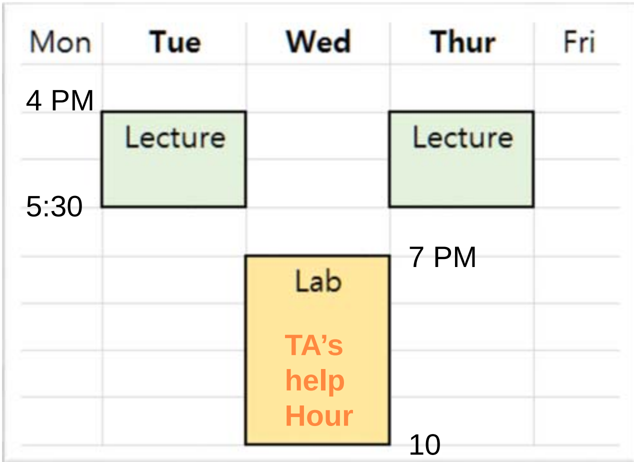
- Traditional topics
- OpenGL





Grading

- Attendance and Participation (including Labs) 10%
- Homework Assignments 40%
- Midterm Exam 25%
- Final Exam 25%



Final Exam

- **Tuesday June 21** 4PM E3-1, #1501
- Reading: all chapters covered
  - omit 19, 23 (Color, Animation)
  - You may bring 1 page help sheet for your own use during the exam.
    - Hand written only on 1 side
    - A4-size paper
    - Will hand-in with your exam

- Announcement
  - Homework #5 Due June 14 (Midnight)
    - Late submission allowed only upto *June 21*.
  - Contest entry by June 15 Midnight to KLMS