CMSC 435/634
Introductory Computer Graphics
Overview

Penny Rheingans
UMBC

Course Staff

- Instructor: Penny Rheingans (ITE 452/355)
- Office Hours: Tues 11:15am-1pm
- by appt
- TA: Elizabeth Baumel (elizaba1@umbc.edu)
- TA OH: Tues/Fri 3-5pm (ITE 352)
- by appt or email

Survey

- Name (what do you wish to be called?)
- Phone
- email
- Major, year (in school), 435/634
- Classes: CS, math, this semester
- Mastery of C++; Favorite computer language
- Something interesting about you

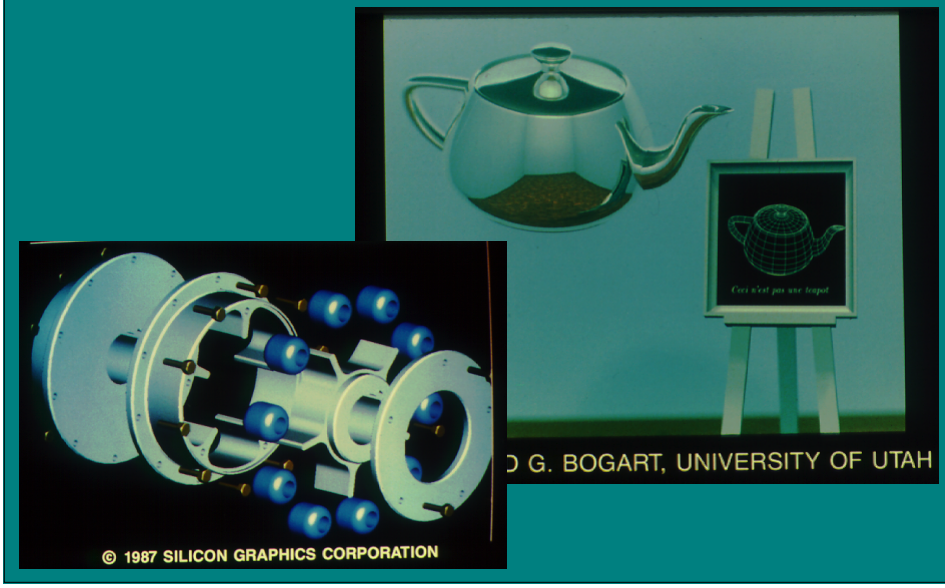
Hold List

- I'll let in as many students as there are seats
- If on hold list:
  - fill out request
  - see me after class
- Decisions by next class

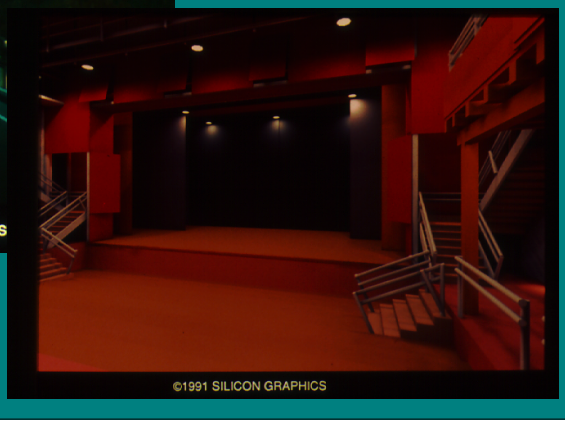
Computer Graphics

- Using computer to generate simulated scenes or worlds
- Requires tricking eye to believe 2D collection of pixels is really a continuous 3D world
- Coding-intensive application with strong basis in creativity and human perception

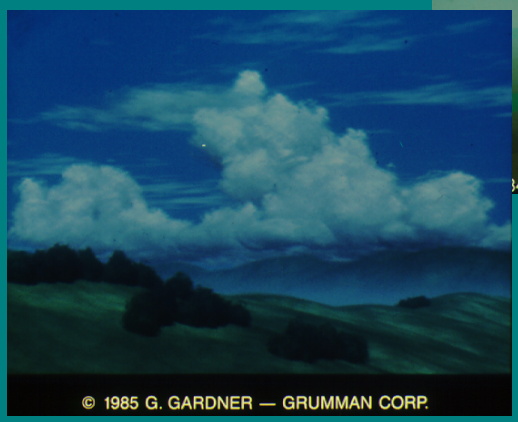
Graphics for Design



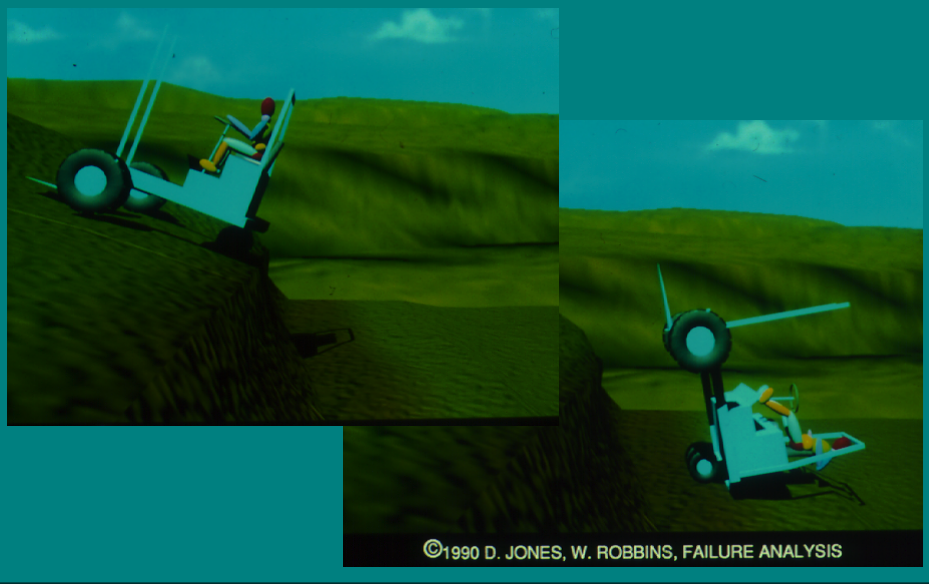
Graphics for Planning



Graphics for Training

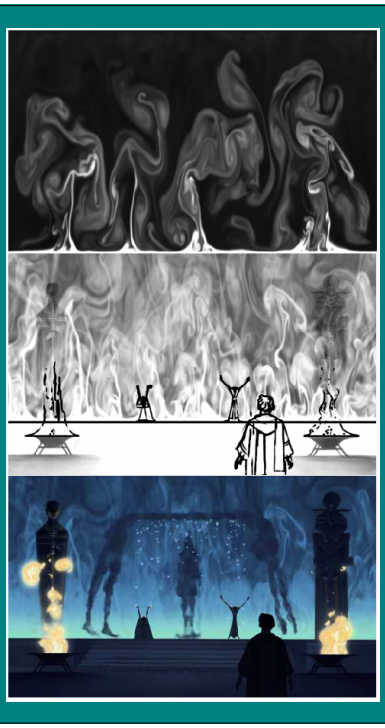


Graphics for Training

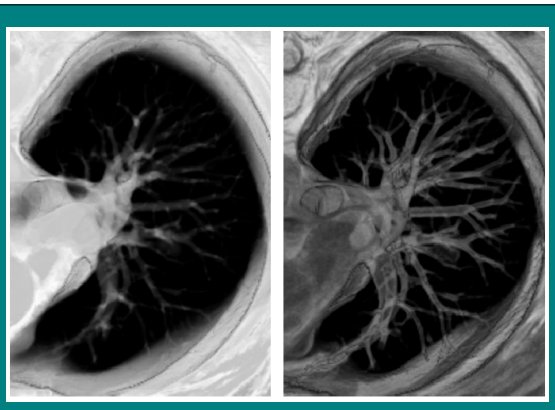


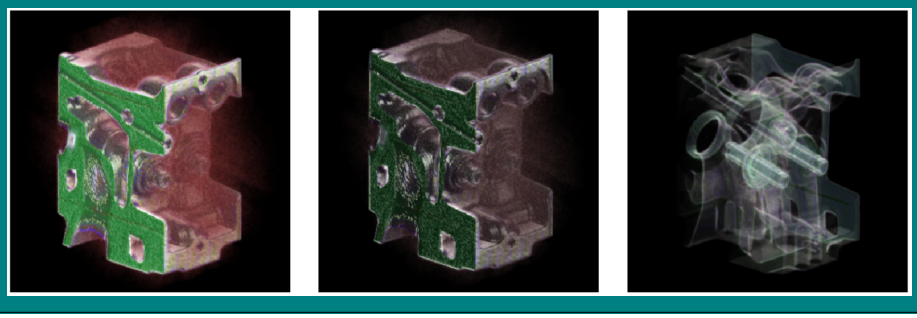
Graphics for Advertising

Graphics for Entertainment

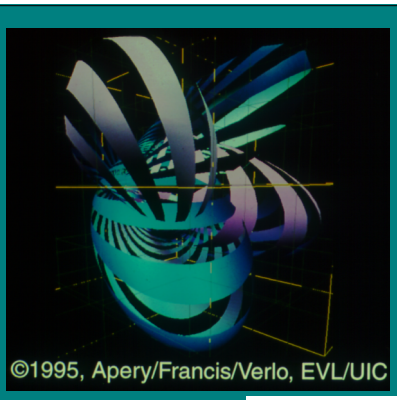


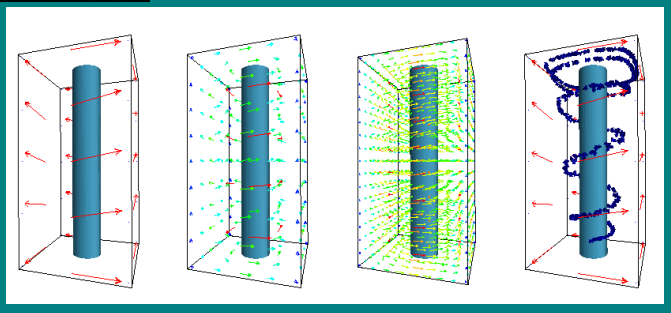
Graphics for Data Display



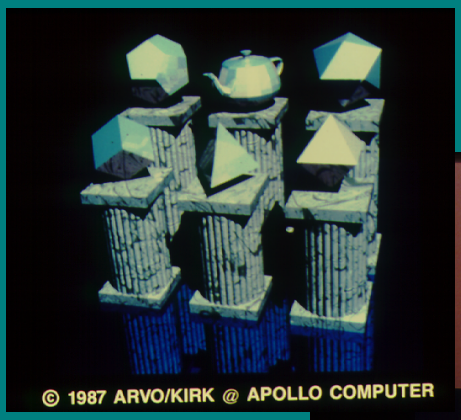


Graphics for Data Display





Graphics for Fun



Five Key Problems

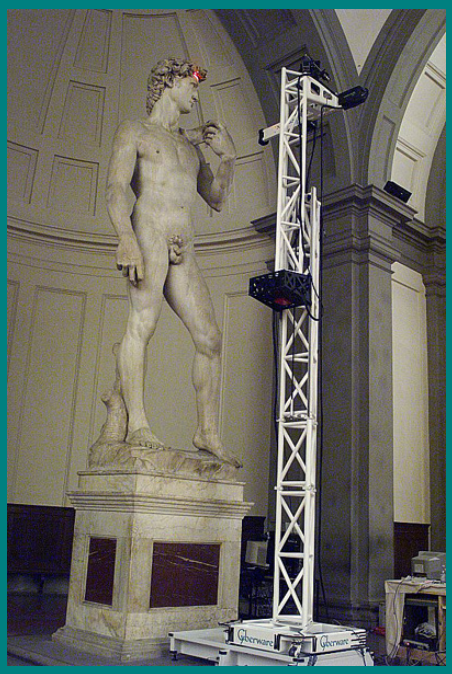
- What do you see?
- What does it look like?
- What shape is it?
- How does it move?
- Why does it have to look like a photograph?

What shape is it?

Modeling Approaches

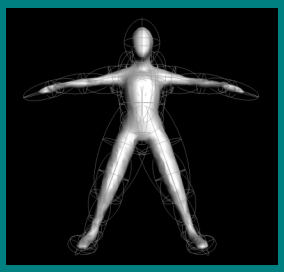
- Modeling problem
  - Define shape, color, and other visual properties
- Modeling solutions
  - Manual primitive creation
  - Scans from physical object
  - Functional descriptions
  - Grammar-based generation
  - Biologically-inspired simulations

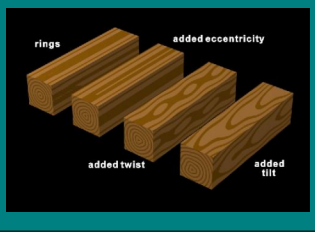
Scanning

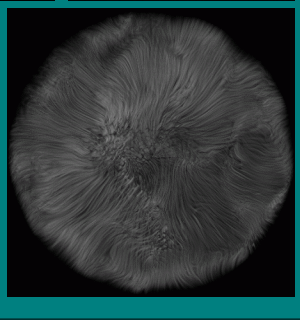


Functional Descriptions

- Define visual attributes with function, defined over space
  - Shape
  - Density
  - Color

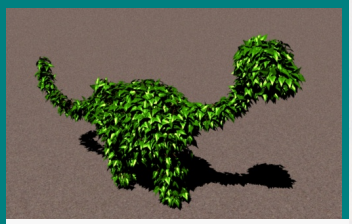


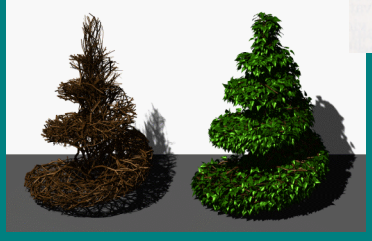




Grammar-based Generation

- Specify structural change over generations

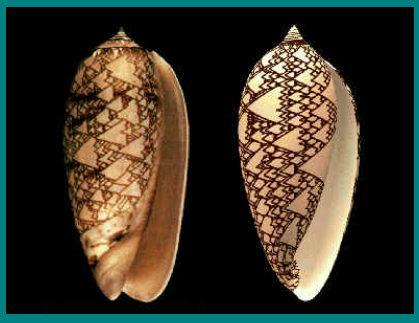




Biological Simulations

Mimic developmental process:

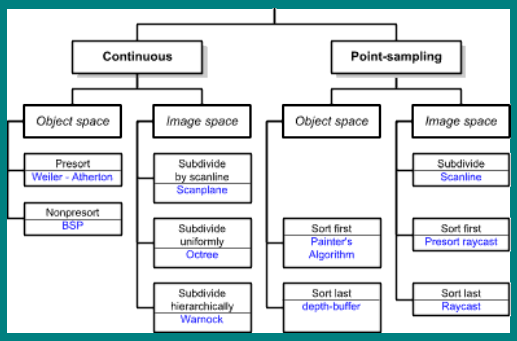
- cellular automata
- reaction diffusion



What do you see?

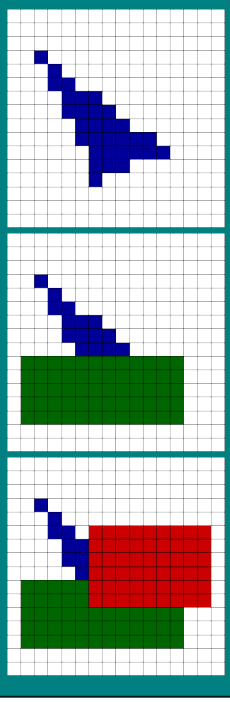
Visibility Approaches

- Visibility problem
  - Determine which objects (or parts of objects) are closest and therefore visible (a sorting problem)
- (Some) visibility solutions
  - Painter's algorithm
  - Zbuffer
  - Ray tracing



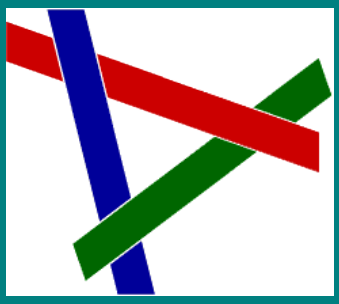
Painter's Algorithm

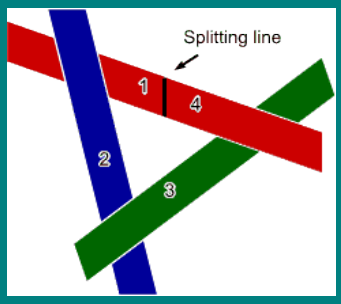
- Basic approach
  - Draw polygons, from farthest to closest
- First polygon:
  - (6,3,10), (11, 5,10), (2,2,10)
- Second polygon:
  - (1,2,8), (12,2,8), (12,6,8), (1,6,8)
- Third polygon:
  - (6,5,5), (14,5,5), (14,10,5),(6,10,5)



Painter's Algorithm: Cycles

- Which to scan first?

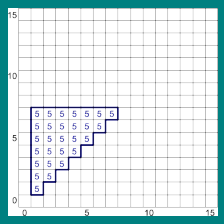


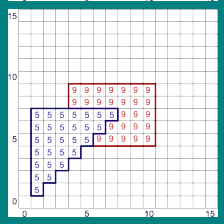


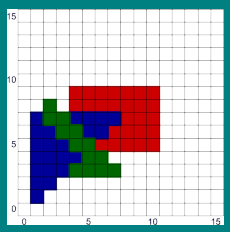
- Split along line, then scan 1,2,3,4 (or split another polygon and scan accordingly)
- Moral: Painter's algorithm is fast and easy, except for detecting and splitting cycles and other ambiguities

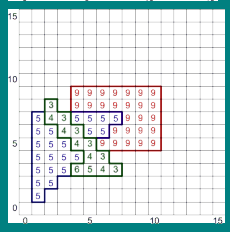
Z-Buffer

- Basic approach
  - Draw polygons, remembering depth of stuff drawn so far
- First polygon
  (1, 1, 5), (7, 7, 5), (1, 7, 5)
- Second polygon
  (3, 5, 9), (10, 5, 9), (10, 9, 9), (3, 9, 9)
- Third polygon
  (2, 6, 3), (2, 3, 8), (7, 3, 3)



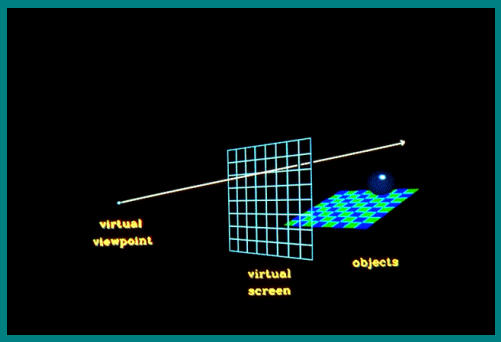


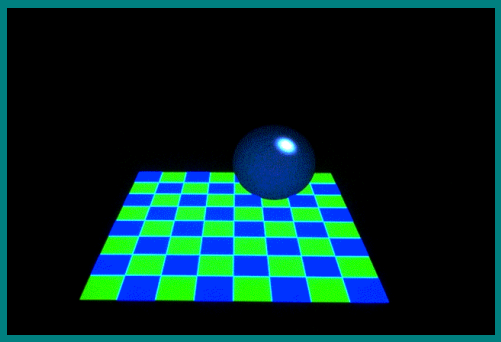




Raytracing

- Basic approach
  - Cast ray from viewpoint through pixels into scene





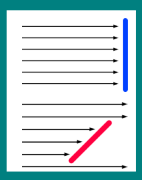
What does it look like?

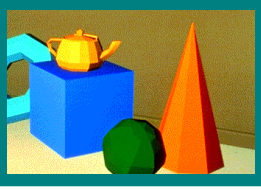
Illumination Approaches

- Illumination problem
  - Model how objects interact with light
- Modeling solutions
  - Simple physics/optics
  - More realistic physics
    - Surface physics
    - Surface microstructure
    - Subsurface scattering
    - Shadows
    - Light transport

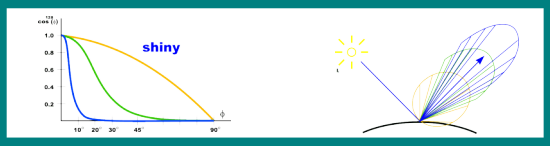
Simple Optics

Diffuse reflection
Reflection of available light in all directions

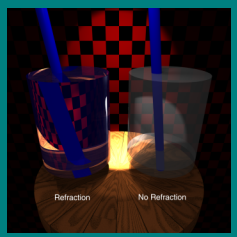


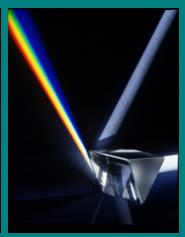


Specular reflection
Reflection in a favored direction



Refraction
Bending of light at material interfaces

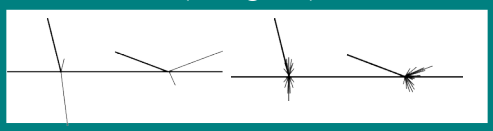




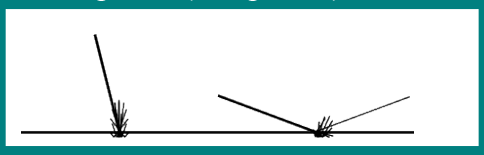
Surface Physics

- Conductor (like metal)

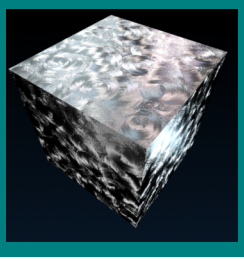
- Dielectric (like glass)

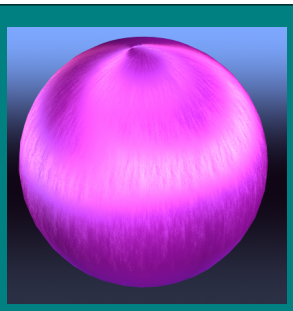


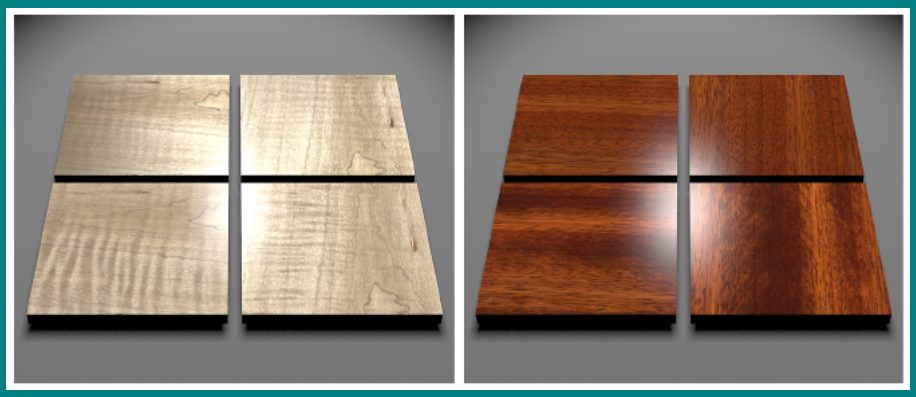
- Composite (like plastic)



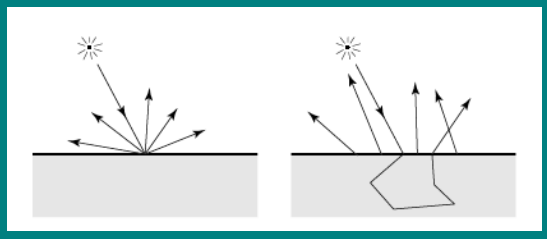
Surface Microstructure



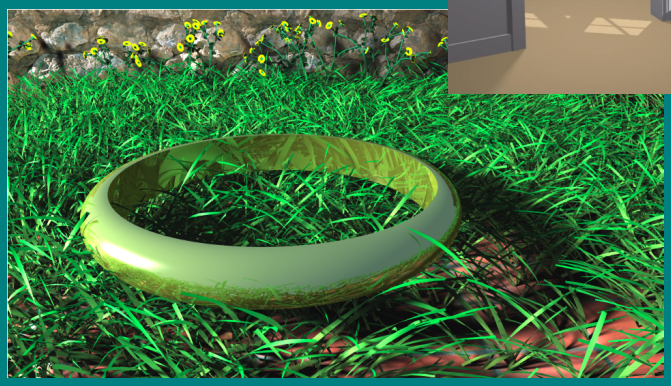




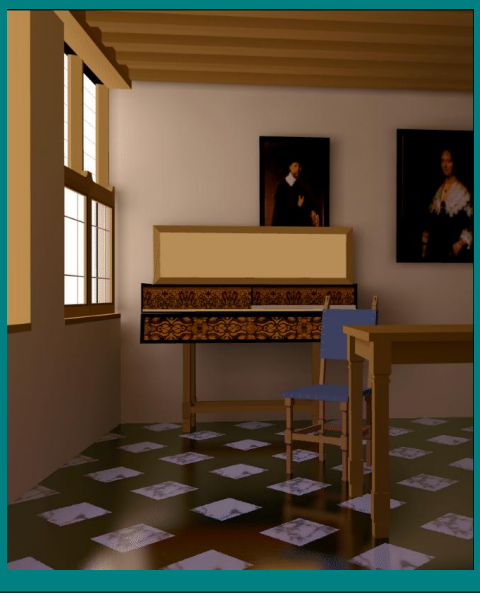
Subsurface Scattering



Shadows



Light Transport

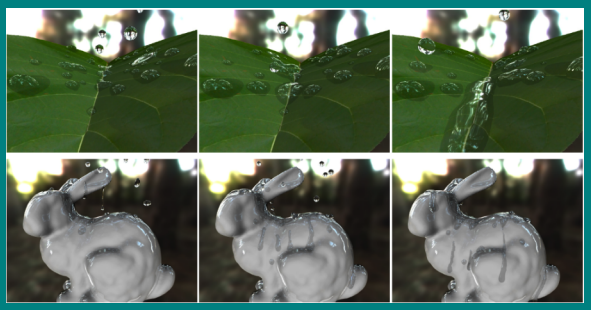


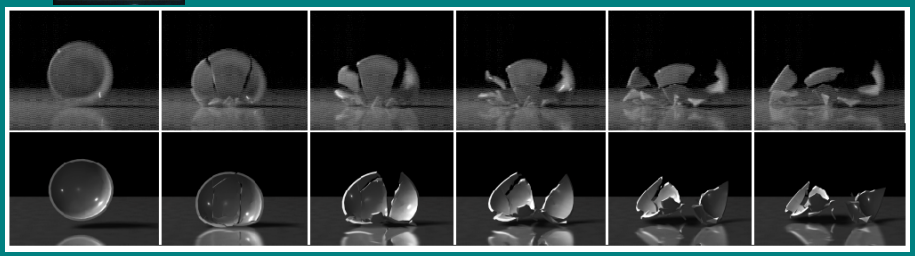
How does it move?

Motion Dynamics Approaches

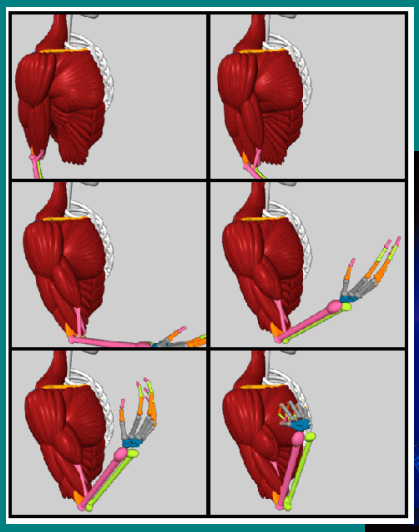
- Motion dynamics problem
  - Define geometric movements and deformations of objections under motion
- Dynamics solutions
  - Simulate physics of simple objects
  - Model structure and constraints
  - Capture motion from reality
  - Simulate group dynamics
  - Use your imagination

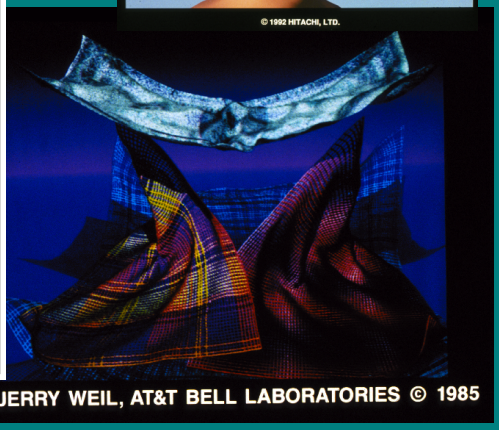
Simulate Physics



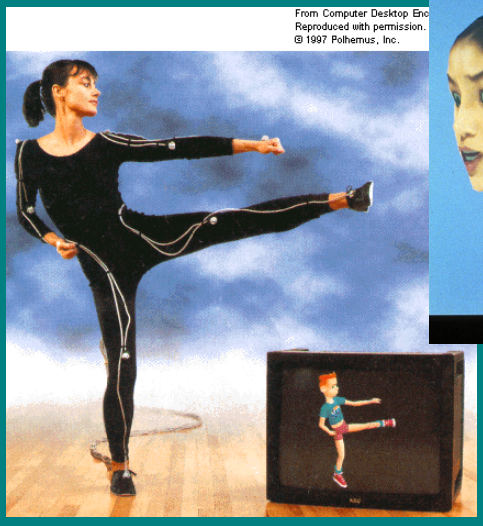


Model Structure

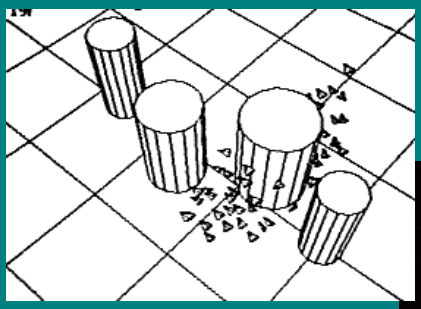


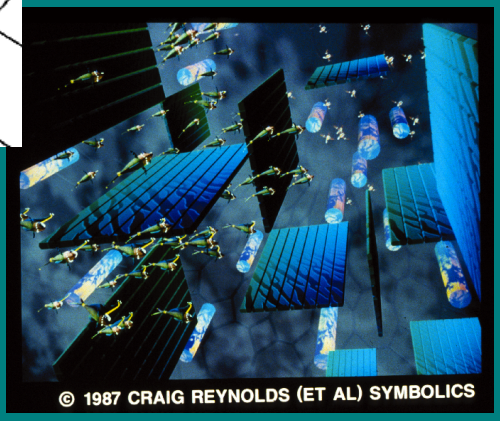


Motion Capture



Behavioral Simulation

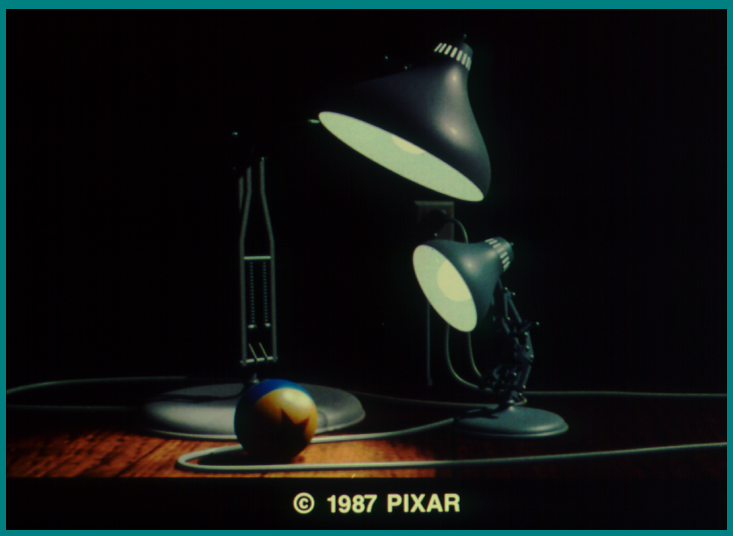




Use Traditional Animation

John Lasseter

Play

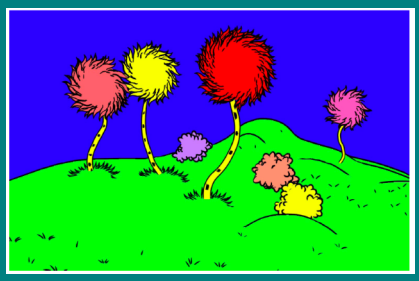


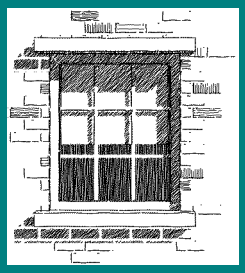
Why does it have to look like a photograph?

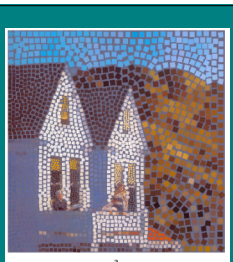
Artistic Rendering Approaches

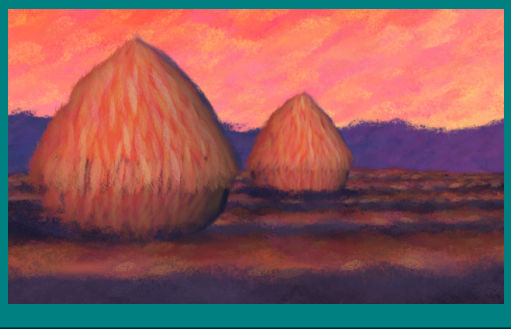
- Artistic rendering problem (NPR)
  - Produce images from geometric models that are more expressive or mimic alternative media
- Artistic rendering solutions
  - Mimic characteristics of media
  - Physically simulate media
  - Break rules

Mimic media

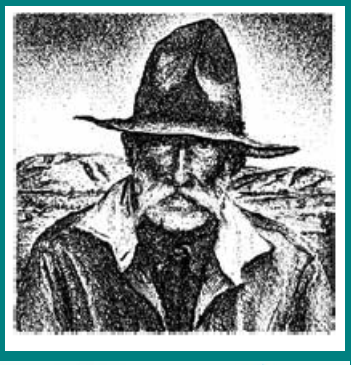


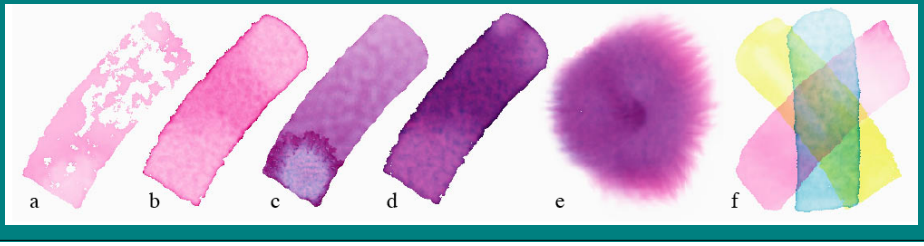




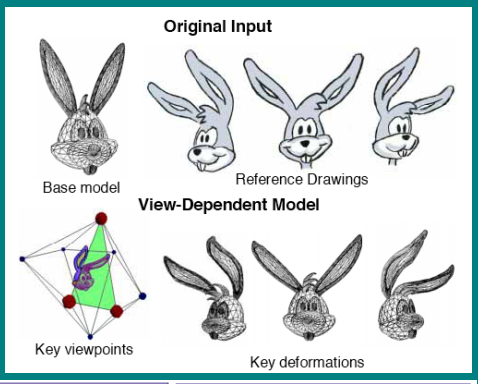


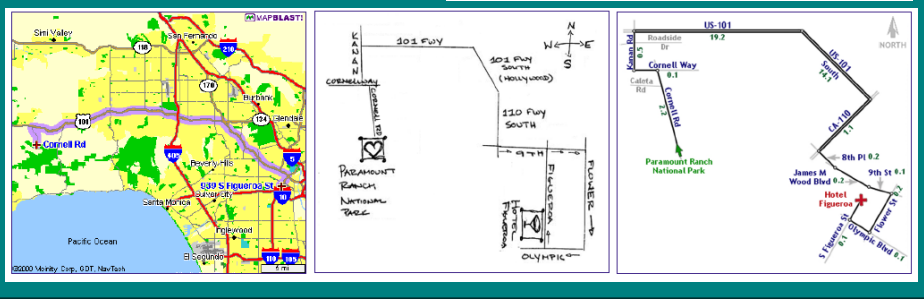
Physical Media Simulation





Break Rules





Objectives

- Understand the foundations of computer graphics.
- Implement key components of the rendering pipeline, understand the rest.
- Use a high-level scene description language to model and render complex, realistic scenes.
- Become acquainted with some advanced topics in computer graphics.

Expected Background

- Expecting Jr/Sr/Grad CS Majors, true?
- Data Structures (CMSC341): arrays, pointers, data structures, basic algorithms
- Programming in C/C++ (CMSC313): language, compilation (make), debugging
- Math (MATH221) vector, matrices
  - Also assorted HS math, esp trig

Reading

- Texts
  - *Fundamentals of Computer Graphics, 3nd edition*. Peter Shirley, AK Peters, 2009. Required: fundamental graphics textbook.
  - *OpenGL Programming Guide*. Mason Woo, Jackie Neider, Tom Davis, and Dave Shreiner, Addison-Wesley, any modern version. Recommended: useful for assignments.
  - Effective C++, Scott Meyers, 2005. Recommended: useful for assignments.
- Read before class
- Expect quizzes: vocabulary and basic concepts

Assignments

- Types
  - Programming Examples: implementing guts of rendering pipeline, in C++ with OpenGL basis
  - Rendering Exercises: scene creation using Renderman, C++
  - Homework Problems: concepts and exercises
- Due
  - Programming/Rendering: at midnight, submit electronically
  - Problems: in class, turn in paper
- Late assignments penalized 20% (up to a week)
  - One free late (up to a week), requested in writing on due date
- Do your own work;
  - document any help (or none) in header of program;
  - projects without statement about level of help will be returned ungraded
- START EARLY!!!

Grades

- Programming assts 50%
- Homework problems 10%
- Midterm exam & quizzes 15%
- Final exam 25%