6.837
Introduction to Computer Graphics
Plan

- Introduction
- Overview of the semester
- Administrivia
- Iterated Function Systems (fractals)
Team

• Lecturers
  – Frédo Durand
  – Barb Cutler

• Course secretary
  – Bryt Bradley
Why Computer Graphics?

- Movies
- Games
- CAD-CAM
- Simulation
- Virtual reality
- Visualization
- Medical imaging
What you will learn in 6.837

• Fundamentals of computer graphics algorithms
• Able to implement most applications just shown
• Understand how graphics APIs and the graphics hardware work
What you will NOT learn

• Software packages
  – CAD-CAM
  – Photoshop and other painting tools
• Artistic skills
• Game design
• Graphics API
  – Although you will be exposed to OpenGL
Plan

- Introduction
- Overview of the semester
- Administrivia
- Iterated Function Systems (fractals)
Overview of the semester

• Ray Tracing
  – Quiz 1

• Animation, modeling, IBMR
  – Choice of final project

• Rendering pipeline
  – Quiz 2

• Advanced topics
Ray Casting

- For every pixel construct a ray from the eye
  - For every object in the scene
    - Find intersection with the ray
    - Keep if closest
Ray Casting

• For every pixel construct a ray from the eye
  – For every object in the scene
    • Find intersection with the ray
    • Keep if closest
Ray Tracing

- Shade (interaction of light and material)
- Secondary rays (shadows, reflection, refraction)
Ray Tracing

- Original Ray-traced image by Whitted

  Image removed due to copyright considerations.

- Image computed using the Dali ray tracer by Henrik Wann Jensen

  Image removed due to copyright considerations.

- Environment map by Paul Debevec
Overview of the semester

- Ray Tracing
  - Quiz 1
- Animation, modeling, IBMR
  - Choice of final project
- Rendering pipeline
  - Quiz 2
- Advanced topics
Animation: Keyframing

Image adapted from:

Squash & Stretch in Luxo Jr.'s Hop

Image adapted from:
Particle system (PDE)

- Animation
  - Keyframing and interpolation
  - Simulation

Images removed due to copyright considerations.
Rigid body dynamics

- Simulate all external forces and torques

\[ f_1(t) \quad f_2(t) \]

\[ p_1^b(t) \quad p_2^b(t) \quad p_3^b(t) \]

\[ x(t) \quad v(t) \]

\[ f_3(t) \]
Modeling

- Curved surfaces
- Subdivision surfaces

Images removed due to copyright considerations.
Image-based Rendering

- Use images as inputs and representation
  - E.g. Image-based modeling and photo editing
    Boh, Chen, Dorsey and Durand 2001

Input image  |  New viewpoint  |  Relighting
Overview of the semester

- Ray Tracing
  - Quiz 1
- Animation, modeling, IBMR
  - Choice of final project
- Rendering pipeline
  - Quiz 2
- Advanced topics
The Rendering Pipeline

Ray Casting
• For each pixel
  – For each object

Send pixels to the scene

Rendering Pipeline
• For each triangle
  – For each projected pixel

Project scene to the pixels
The Rendering Pipeline

- Transformations
- Clipping
- Rasterization
- Visibility

Courtesy of Leonard McMillan, Computer Science at the University of North Carolina in Chapel Hill. Used with permission
Overview of the semester

• Ray Tracing
  – Quiz 1

• Animation, modeling, IBMR
  – Choice of final project

• Rendering pipeline
  – Quiz 2

• Advanced topics
Textures and shading

For more info on the computer artwork of Jeremy Birn see [http://www.3drender.com/jbirn/productions.html](http://www.3drender.com/jbirn/productions.html)

At what point do things start looking real?

Courtesy of Jeremy Birn. Used with permission.
shadows

Image removed due to copyright considerations.

Image removed due to copyright considerations.
Traditional Ray Tracing

Image removed due to copyright considerations.
Ray Tracing+soft shadows

Image removed due to copyright considerations.
Ray Tracing+caustics

Image removed due to copyright considerations.
Global Illumination

Image removed due to copyright considerations.
Antialiasing

Overview

Courtesy of Leonard McMillan, Computer Science at the University of North Carolina in Chapel Hill. Used with permission.
Questions?
Plan

- Introduction

- Overview of the semester

- Administrivia

- Iterated Function Systems (fractals)
Administrivia

• Web:
  http://graphics.csail.mit.edu/classes/6.837/F03/

• Lectures
  – Slides will be online

• Office hours
  – Posted on the web

• Review sessions
  – C++, linear algebra
Prerequisites

• Not enforced
• 18.06 Linear Algebra
  – Simple linear algebra, vectors, matrices, basis, solving systems of equations, inversion
• 6.046J Algorithms
  – Orders of growth, bounds, sorting, trees
• C++
  – All assignments are in C++
  – Review/introductory session Monday
Grading policy

• Assignments: 40%
  – Must be completed individually
  – No late policy. Stamped by stellar.

• 2 Quizzes: 20%
  – 1 hour in class

• Final project: 40%
  – Groups of 3, single grade for the group
  – Initial proposal: 3-5 pages
  – Steady weekly progress
  – Final report & presentation
  – Overall technical merit
Assignments

• Turn in code AND executable
• We will watch code style
• Platform
  – Windows
  – Linux
• Collaboration policy:
  – You can chat, but code on your own
• No late policy
Project

- Groups of 3
- Brainstorming
  - Middle of the semester
- Proposal
- Weekly meeting with TAs
- Report & presentation
Plan

• Introduction

• Overview of the semester

• Administrivia

• Iterated Function Systems (fractals)
IFS: self-similar fractals

• Described by a set of $n$ transformations $f_i$
  – Capture the self-similarity
  – Affine transformations
  – Contractions (reduce distances)
• An attractor is a fixed point

$$A = \bigcup f_i(A)$$

Image removed due to copyright considerations.
Example: Sierpinsky triangle

- 3 transforms
- Translation and scale by 0.5
Rendering

For a number of random input points \((x_0, y_0)\)

For \(j=0\) to big number

Pick transformation \(i\)

\((x_{k+1}, y_{k+1}) = f_i (x_k, y_k)\)

Display \((x_k, y_k)\)

- Probabilistic application of one transformation
Example: Sierpinsky triangle

For a number of random input points \((x_0, y_0)\)

\[
\begin{align*}
\text{For } j=0 \text{ to big number} & \\
\text{Pick transformation } i & \\
(x_{k+1}, y_{k+1}) &= f_i (x_k, y_k) \\
\text{Display } (x_k, y_k)
\end{align*}
\]
Example: Sierpinsky triangle

For a number of random input points \((x_0, y_0)\)

For \(j=0\) to big number

Pick transformation \(i\)

\[
(x_{k+1}, y_{k+1}) = f_i (x_k, y_k)
\]

Display \((x_k, y_k)\)
Example: Sierpinsky triangle

For a number of random input points \((x_0, y_0)\)

For \(j=0\) to big number

Pick transformation \(i\)

\[
(x_{k+1}, y_{k+1}) = f_i (x_k, y_k)
\]

Display \((x_k, y_k)\)
Example: Sierpinsky triangle

For a number of random input points \((x_0, y_0)\)

For \(j=0\) to big number

Pick transformation \(i\)

\[
(x_{k+1}, y_{k+1}) = f_i (x_k, y_k)
\]

Display \((x_k, y_k)\)
Example: Sierpinsky triangle

For a number of random input points \((x_0, y_0)\)

For \(j=0\) to big number

Pick transformation \(i\)

\[
(x_{k+1}, y_{k+1}) = f_i (x_k, y_k)
\]

Display \((x_k, y_k)\)
Example: Sierpinsky triangle

For a number of random input points \((x_0, y_0)\)

For \(j=0\) to big number

Pick transformation \(i\)

\[(x_{k+1}, y_{k+1}) = f_i (x_k, y_k)\]

Display \((x_k, y_k)\)
Example: Sierpinsky triangle

For a number of random input points \((x_0, y_0)\)

For \(j=0\) to big number

Pick transformation \(i\)

\[
(x_{k+1}, y_{k+1}) = f_i (x_k, y_k)
\]

Display \((x_k, y_k)\)
Example: Sierpinsky triangle

For a number of random input points \((x_0, y_0)\)

For \(j=0\) to big number

Pick transformation \(i\)

\[
(x_{k+1}, y_{k+1}) = f_i (x_k, y_k)
\]

Display \((x_k, y_k)\)
Example: Sierpinsky triangle

For a number of random input points \((x_0, y_0)\)

For \(j=0\) to big number

Pick transformation \(i\)

\((x_{k+1}, y_{k+1}) = f_i (x_k, y_k)\)

Display \((x_k, y_k)\)

Overview
Example: Sierpinsky triangle

For a number of random input points \((x_0, y_0)\)

For \(j=0\) to big number

Pick transformation \(i\)

\[
(x_{k+1}, y_{k+1}) = f_i (x_k, y_k)
\]

Display \((x_k, y_k)\)
Example: Sierpinsky triangle

For a number of random input points \((x_0, y_0)\)

For \(j=0\) to big number

Pick transformation \(i\)

\[(x_{k+1}, y_{k+1}) = f_i (x_k, y_k)\]

Display \((x_k, y_k)\)
Example: Sierpinsky triangle

For a number of random input points \((x_0, y_0)\)

For \(j=0\) to big number

Pick transformation \(i\)

\[
(x_{k+1}, y_{k+1}) = f_i (x_k, y_k)
\]

Display \((x_k, y_k)\)

Overview

52
Example: Sierpinsky triangle

For a number of random input points \((x_0, y_0)\)

For \(j=0\) to big number

Pick transformation \(i\)

\[(x_{k+1}, y_{k+1}) = f_i (x_k, y_k)\]

Display \((x_k, y_k)\)
Example: Sierpinsky triangle

For a number of random input points \((x_0, y_0)\)

For \(j = 0\) to big number

Pick transformation \(i\)

\((x_{k+1}, y_{k+1}) = f_i (x_k, y_k)\)

Display \((x_k, y_k)\)
Example: Sierpinsky triangle

For a number of random input points \((x_0, y_0)\)

For \(j=0\) to big number

Pick transformation \(i\)

\[
(x_{k+1}, y_{k+1}) = f_i(x_k, y_k)
\]

Display \((x_k, y_k)\)
Example: Sierpinsky triangle

For a number of random input points \((x_0, y_0)\)

For \(j=0\) to big number

Pick transformation \(i\)

\((x_{k+1}, y_{k+1}) = f_i (x_k, y_k)\)

Display \((x_k, y_k)\)
Example: Sierpinsky triangle

For a number of random input points \((x_0, y_0)\)

For \(j=0\) to big number

Pick transformation \(i\)

\((x_{k+1}, y_{k+1}) = f_i (x_k, y_k)\)

Display \((x_k, y_k)\)
Example: Sierpinsky triangle

For a number of random input points \((x_0, y_0)\)

For \(j=0\) to big number

Pick transformation \(i\)

\[
(x_{k+1}, y_{k+1}) = f_i (x_k, y_k)
\]

Display \((x_k, y_k)\)
Example: Sierpinsky triangle

For a number of random input points \((x_0, y_0)\)
  
  For \(j=0\) to big number
    
    Pick transformation \(i\)
    
    \((x_{k+1}, y_{k+1}) = f_i (x_k, y_k)\)
    
    Display \((x_k, y_k)\)
Example: Sierpinsky triangle

For a number of random input points \((x_0, y_0)\)

For \(j=0\) to big number

Pick transformation \(i\)

\[
(x_{k+1}, y_{k+1}) = f_i (x_k, y_k)
\]

Display \((x_k, y_k)\)
Other IFS

• The Dragon
Application: fractal compression

• Exploit the self-similarity in an image
Assignment: IFS

• Write a C++ IFS class
• Get familiar with
  – vector and matrix library
  – Image library
• Due Wednesday at 11:59pm

• Check on the web page
http://graphics.lcs.mit.edu/classes/6.837/F03/
Review/introduction session: C++

- Monday 7:30-9
Questions?