Why Study Computer Science?

Money!  Beautiful Images

What is Computer Graphics?

REEL FX CREATIVE STUDIOS

Elements in This Video

3D Modeling / Geometry
Simulation / Animation / Character Animation
Lighting / Light Transfer
Textures and Color
Post-Processing: Image Processing
Camera position / Optics
What Is Computer Graphics?

Using computers to generate and display images

- Visual Perception
- Modeling
- Rendering
- Animation
- Image Processing

What Else Is Computer Graphics

Scientific Visualization
Illustration
NPR / Art
Computational Photography
Virtual Life

and much more....

Why Do We Care?

Why Do We Care?
Today

Motivation - What is Computer Graphics?

Course Mechanics and Overview
- Who is teaching this class anyway?
- Wiki
- Course Topics
- Assignments - including assignment of assignments #1 and #2

Course Overview

Digital Images

Course Mechanics and Overview

Who am I?
Maneesh Agrawala
- Professor since 2006
- Ph.D. 2002 (from another bay area school)
Research Areas: Graphics, Visualization, HCI

Video Puppetry

Our Approach
Exploded Views

Photomontage

Interactive Digital Photomontage
Asen Agarwala, Mina Dijkstra
Maneesh Agrawala, Sean Drucker, Alain Caffin
Boris Cohen, David Salesin, Michael Cohen

Graduate Student Instructors

Robert Carroll
Ph.D. student
Computational Photography, Graphics, Vision
Office hours: T 3-4pm, F 1-2pm, 514 Soda Hall

Fu-Chung Huang
Ph.D. student
Real-Time Rendering, Animation, Vision, Learning, Modeling
Office hours: W 2-3pm, 535 Soda Hall

Contact With Teaching Staff

Maneesh’s Office Hours: MW 12-12:30pm, T 4-5pm, by appt.

Email: cs184@eecs.berkeley.edu
• Do not email teaching staff directly unless absolutely necessary

Wiki: http://vis.berkeley.edu/courses/cs184-fall-2010
• All class related materials, handouts, schedule, readings posted there
• Will not distribute paper handouts
• Lecture notes posted there (hopefully) before classes

Google group: http://groups.google.com/group/ucb-cs184-fall-2010
• Send email to cs184@eecs.berkeley.edu requesting invitation
• Will post bug announcements, fixes for assignments and lectures there
• Not reading group, bad idea
Computing Resources

Class accounts handed shortly

Can also use CS Labs

- Linux
- Windows
- Mac

Text Book

*Fundamentals of Computer Graphics. 3rd Edition*
by Peter Shirley and Steve Marschner

Today’s readings: 1
Review readings: 2, 5
Weds readings: 21, 22

Supplemental material may also be provided

See other books listed on wiki

Course Topics

- Visual Perception
- Modeling
- Rendering
- Animation
- Image Processing
Grading

Assignments: 40%
• Mix of written and programming (about 5 of them)
• Average 1 or 2 weeks to do them

Final Project: 20%
• Presentation: TBD prior to midterm

Midterm: 20%
• Wednesday, October 13, In class

Final: 20%
• Monday, December 13 8:00-11:00am

Check now for conflicts!

Prerequisites

You must know how to program C or C++
• Big final project, several programming assignments
• No hand holding

Data structures (CS61B)

Math: linear algebra, calculus, trigonometry

Assignments #1 and #2

Assignment #1: Getting Started
• Setup CS184 account and let us know who you are
• Get very simple OpenGL program working
• Due: Sat Sep 4 by 11pm

Assignment #2: Math Review
• Tests math prerequisites
• Due: Fri Sep 10 by 11pm

Assignment #3: Shading

[Ritchie & Cho, F08]
Assignment #4: Ray Tracing

Assignment #5: Parametric Surfaces

Final Project: Open Ended

Your image/animation/game here!

Waitlist

Relax for now... there is lots of space.
Class Participation

Reasons to participate

• More fun for me and you
  • You learn more
  • I won’t give stupid little annoying quizzes in class

How to participate

• Ask questions
  • Make comments

Stupid questions/comments

• That’s okay

Academic Honesty

If you use an external resource cite it clearly!

Don’t do things that would be considered dishonest...

• If in doubt ask

Cheating earns you:

• An ‘F’ in the class and
  • Getting reported to the University
  • No exceptions.

Questions?

Let’s Have Fun!
Images

Something that represents a *pattern of light* that will be perceived by something

Computer representations
- Sampled (pixel based)
- Object based
- Functional/Procedural
Images

Something that represents a *pattern of light* that will be *perceived* by something

Computer representations
- Sampled (pixel based)
- Object based
- Functional/Procedural

Images

Something that represents a *pattern of light* that will be *perceived* by something

Computer representations
- Sampled (pixel based)
- Object based
- Functional/Procedural

This *used* to be an object based representation...

Images

Something that represents a *pattern of light* that will be *perceived* by something

Computer representations
- Sampled (pixel based)
- Object based
- Functional/Procedural

Images

Something that represents a *pattern of light* that will be *perceived* by something

Computer representations
- Sampled (pixel based)
- Object based
- Functional/Procedural
Images

Something that represents a **pattern of light** that will be **perceived** by something

Computer representations
- Sampled (pixel based)
- Object based
- Functional/Procedural

Think about making edits...

Storing Images

Object and Function representations basically arbitrary ...later...

Raster Images
- 2D array of memory
- Pixels store different things
  - Intensity
  - RGB color
  - Depth
  - Others...
- May be mapped to special HW

Storing Images

Object and Function representations basically arbitrary ...later...

Raster Images
- 2D array of memory
- Pixels store different things
  - Intensity (scalar value, e.g., float, int)
  - RGB color (vector value)
  - Depth
  - Others...
- May be mapped to special HW

Discretization

Real world and “object” representations are continuous.
Raster images have discrete pixel **locations** and discrete pixel **values**

We will see problems from this soon...
Monitor Intensity and Gamma

Monitors convert pixel value into intensity level

- 0.0 maps to zero intensity = black (well not quite)
- 1.0 maps to full intensity = white

Monitors are not linear

- 0.5 does not map to “halfway” gray (e.g., 0.5 might map to 0.217)
- Nonlinearity characterized by exponential function
  \[ I = a^\gamma \]
  where \( I \) = displayed intensity and \( a \) = pixel value (between 0 and 1)
- For many monitors \( \gamma \) is near 2 (often between 1.8 and 2.2)

Determining Gamma \( I = a^\gamma \)

Suppose I know displayed intensity of a patch \( I = 0.5 \)

\[ 0.5 = a^\gamma \]

Let viewer adjust pixel value \( a \) of nearby patch until match

\[ \gamma = \frac{\ln 0.5}{\ln a} \]

How do we make a patch of known intensity?

Determining Gamma

How do we make a patch of known intensity?

High Dynamic Range Images

Dynamic range of the human eye >> range of standard monitors

Eye adjusts as we look around
Tone Mapping

Jack Tumblin