CSE 167: Introduction to Computer Graphics

Jürgen P. Schulze, Ph.D.
University of California, San Diego

Fall Quarter 2017

Today

- Course organization
- Course overview

Course Staff

Instructor

- Jürgen Schulze, Ph.D.
  Adjunct Professor in CSE
  Research Scientist at Qualcomm Institute

Assistants

- Teaching Assistant:
  - Jean Choi
- Tutors:
  - Yining Liang
  - Kaiser Pister
  - Timothy Wang
  - Glynn Williams
  - Jiasheng Zhu
  - Hoang Tran (volunteer)

Weekly Schedule

Lecture

- Tue/Thu, 9:30-10:50am, Center Hall 119

Homework Discussion

- Mondays, 8:00-8:50pm, Center Hall 216

Homework Grading

- Due dates are on Fridays at 2:00pm
- Turn in by demonstration in CSE lab 260 or 270 and submission to TritonEd

Office Hours

Instructor

- Tue 11-noon, Atkinson Hall, room 2125

TA and Tutors

- Tutoring hours held in basement lab 260
- Times to be posted on Piazza

Prerequisites

Expected is familiarity with:

- C++
- Object oriented programming concepts
- CSE 100: Advanced Data Structures
  - Advanced data structures in C++, e.g., graphs
  - Data structure analysis
  - Reason about appropriate data structures to solve problems
  - C++ with STL
  - GIT for code management

Course Web Site

- URL: http://http://ivl.calit2.net/wiki/index.php/CSE167F2017
- Provides:
  - Course schedule
  - Lecture slides
  - Textbook recommendations
  - Homework assignments
  - Grading + exam information

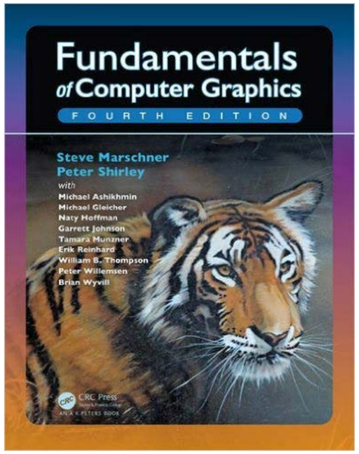
TritonEd

- For homework and exam grades
  - Check your grades regularly
  - Let us know if a grade is missing or incorrect
- Upload source code
  - Only ASCII (text) files

Textbooks

Recommended textbooks:

- Peter Shirley: *Fundamentals of Computer Graphics*, Fourth Edition
  - Earlier editions useful but using older OpenGL
  - Google Books has full text version of edition 3



Programming Projects

- 5 programming assignments
  - First four projects are individual projects, final project is team project
- Find assignments and due dates on home page
  - Due dates every other week
- Starter code is also on home page
- Use CSE basement labs or your own PC/laptop
  - Windows, Mac or Linux
- Individual assistance by TAs/tutors during office hours
- Turn in by demonstration to course staff during homework grading hours on Fridays
  - Demonstration can be done on lab PC or personal laptop
  - Grading from 2pm until at least 3:15pm
  - Required: submit source code by 2pm
- All programming projects have extra credit option

If you can’t come to grading

- Submit source code by 2pm on due date as usual
- Email instructor:
  - Reason of absence
  - When you can demo instead (in TA/tutor office hours)

Written Examinations

- Two in-class written exams
- Closed book
- No notes or cheat sheets
- Allowed:
  - pen, pencil, ruler, eraser
  - blank scratch paper
- Dates posted on course schedule

Grading

- Homework Assignments 1-4: 15% each
- 2 midterm exams: 10% each
- Final project: 20%
- Late submission policy for homework projects:
  - Allowed within 1 week of due date, with 25% penalty
    - Example: for perfect score of 110 points (including extra credit), when submitted late you will get 83 points)
  - No partial penalty if submitting earlier
  - No points if submitting later than 1 week after due date

Grading Key

Final Score	Letter Grade
100+	A+
95+	A
90+	A-
85+	B+
80+	B
75+	B-
70+	C+
65+	C
60+	C-

Today

- Course organization
- Course overview

Rendering

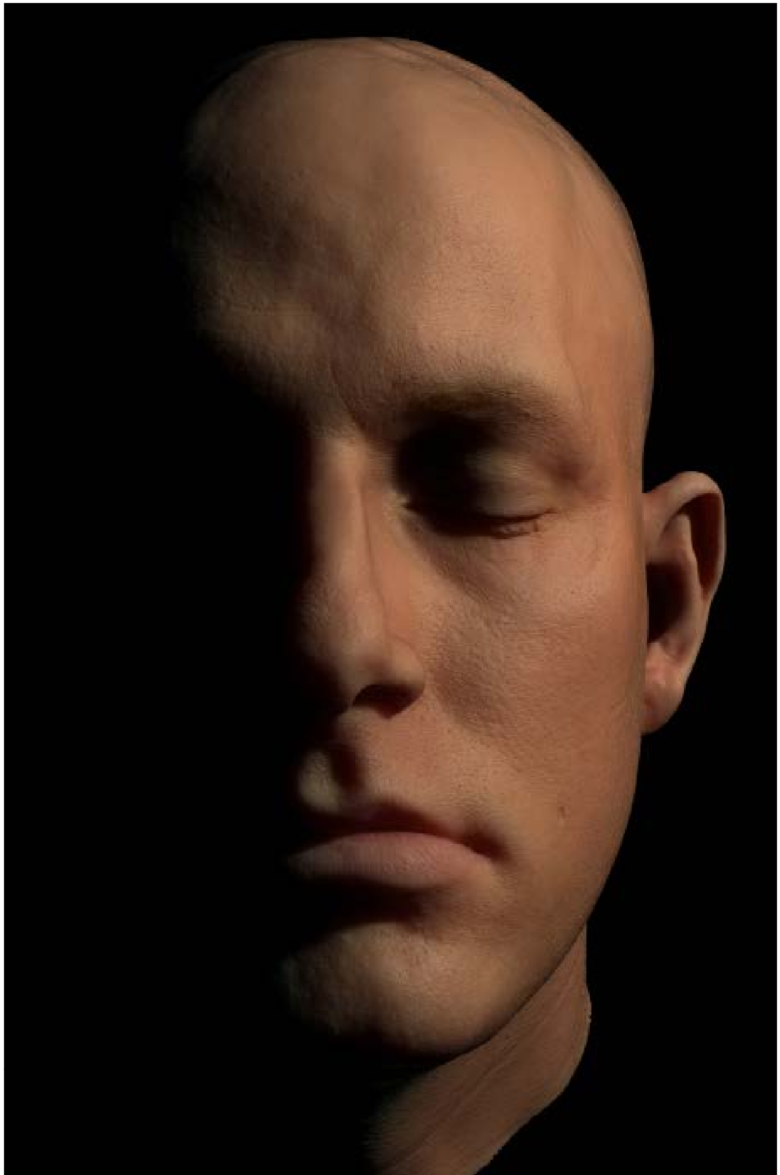
- Synthesis of a 2D image from a 3D scene description
  - Rendering algorithm interprets data structures that represent the scene in terms of geometric primitives, textures, and lights
- 2D image is an array of pixels
  - Red, green, blue values for each pixel
- Objectives
  - Photorealistic
  - Interactive

Photorealistic rendering

- Physically-based simulation of light, camera
- Shadows, global illumination, multiple bounces of light
- Slow, can take minutes or hours to render an image
- Used in movies, animation
- Covered in CSE168: Rendering Algorithms

Photorealistic rendering

RENDERED USING DALI - HENRIK WANN JENSEN 2000



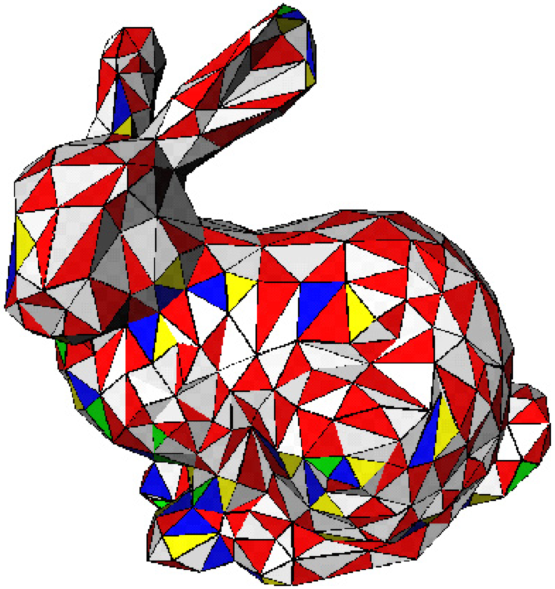
Interactive rendering

- Produce images within milliseconds
- Using specialized hardware, graphics processing units (GPUs)
- Standardized APIs (OpenGL, DirectX)
- Often “as photorealistic as possible”
- Hard shadows, only single bounce of light
- Used in games
- Covered in this course

Interactive rendering

What to render?

- 3D models
- Basic 3D models consist of array of triangles

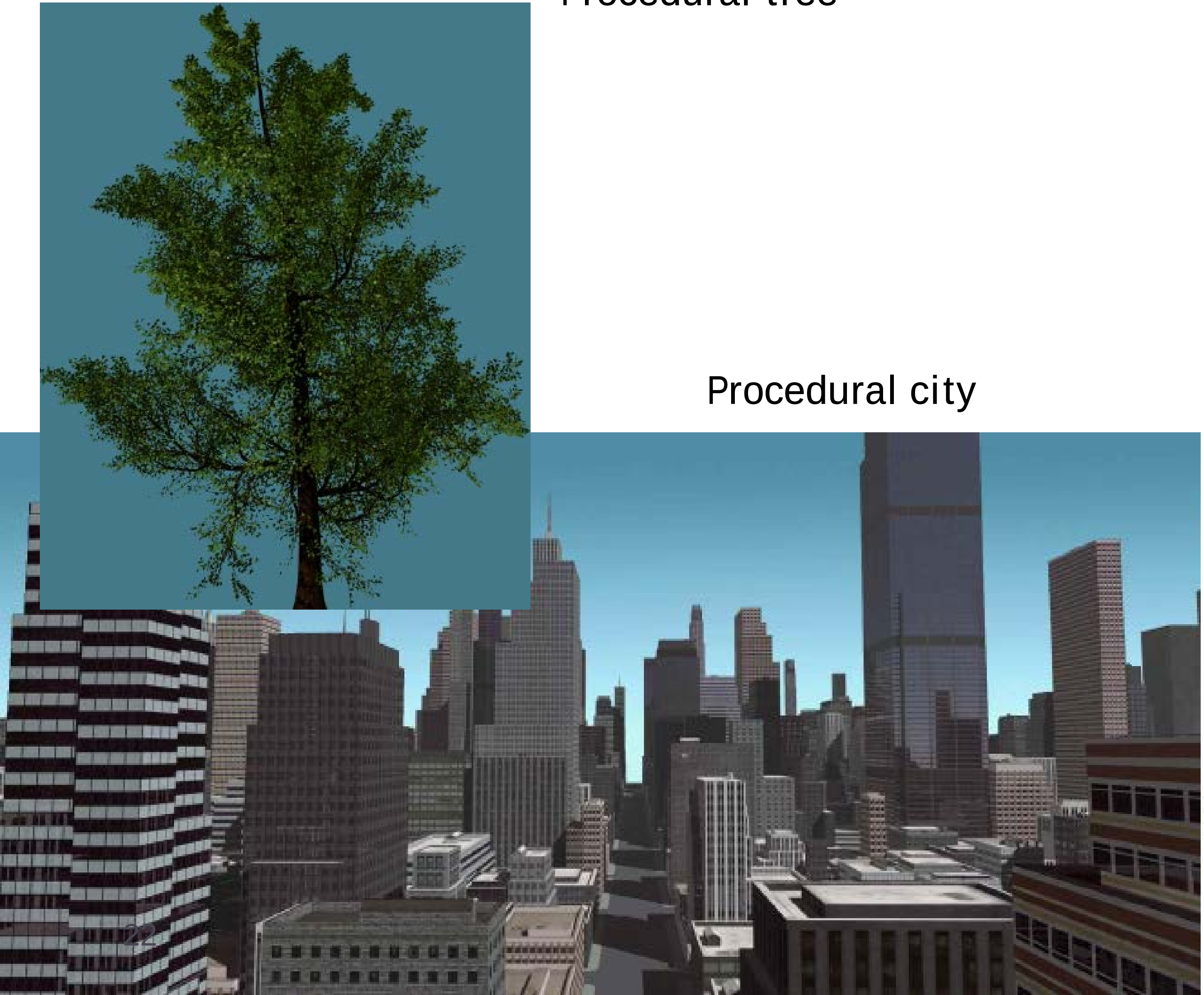


- 3D model sources:
  - Created with 3D modeling tool
  - Loaded from files
  - Procedurally generated: by code you write
  - Created by scanning real-world objects

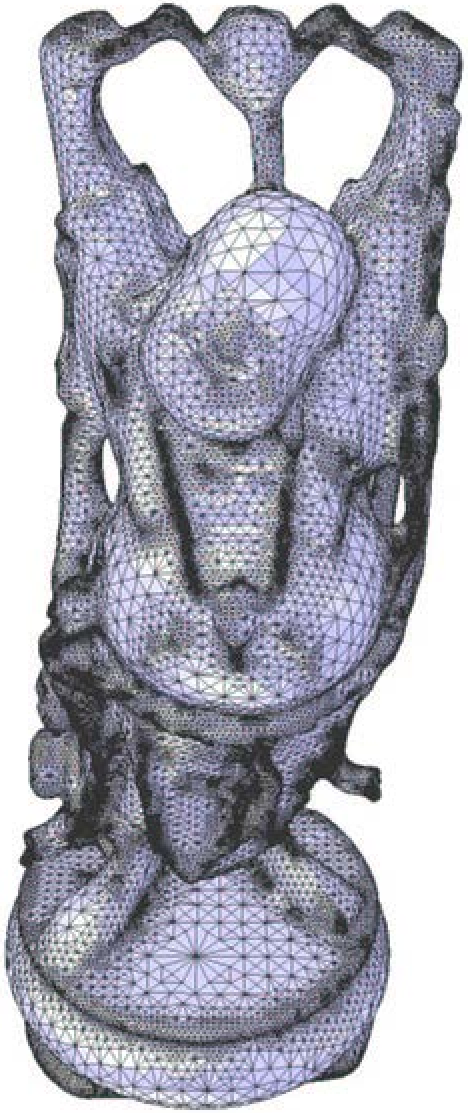
Modeling

Procedural tree

Procedural city



Scanned statue



Topics

- Basic skills:
  - Vector and matrix mathematics
  - Coordinate system transformations
  - 3D to 2D projection
  - Rasterization

Topics

- OpenGL:
  - Lighting
  - Texturing
  - Shading
  - GL Shading Language (GLSL)

Topics

- High Level Concepts:
  - Scene Graph
  - Culling
  - Parametric Curves and Surfaces
  - Procedural Modeling

Topics

- Visual Effects:
  - Environment Mapping
  - Shadows
  - Deferred Rendering

Final Projects from Fall 2016

- Galaxy Battles: Episode V - The Imperium's Counter Attack
  - Austin Sun, Timothy Wang
  - https://www.youtube.com/watch?v=XNSKA9WjGP4
- Propeller Airplane
  - Xinghang Li , Jiasheng Zhu
  - https://www.youtube.com/watch?v=5cYCupSyFBU
- Slim Shading
  - Zhiyao Yan, Dylan Knutson
  - https://www.youtube.com/watch?v=RvOcD09-hac
- Green Island
  - Varanon Austin Pukasamsombut, Joshua Tang
  - https://www.youtube.com/watch?v=arQL7ayVPCk

Announcements

- Homework discussion on Monday 8pm