CSE167
Introduction to Computer Graphics

Matthias Zwicker
University of California, San Diego
Fall 2007

Course staff
Instructor
• Matthias Zwicker
• Assistant professor at UCSD CSE since 2006
Teaching Assistant
• Iman Sadeghi
• Second year CSE PhD student

Today
• Course organization
• Course overview
• Linear algebra review

Course organization
Instructor
• Matthias Zwicker (mzwicker@cs.ucsd.edu)
Teaching Assistant
• Iman Sadeghi (sadeghi@gmail.com)

Course organization
Lecture
• Tue/Thur, 9:30am-10:50am, CENTR 212
Lab hours
• Wed, 1:30pm-4:30pm, EBU3B B210
• Fri, 1:30pm-4:30pm, EBU3B B210
Office hours
• Fri, 4:30pm-5:30pm, EBU3B 4114, by appointment
**Course organization**

**Class web page**
- [http://graphics.ucsd.edu/courses/cse167_f07](http://graphics.ucsd.edu/courses/cse167_f07)

**Webboard**
- [http://webboard.ucsd.edu/](http://webboard.ucsd.edu/)
- Use your network user/PID

---

**Programming Projects**

- Assignments on class webpage
- Base code (for Windows platform) and documentation on class webpage
- Use EBU3B 2xx labs or your own PC
- Wednesdays lab hours: turn-in by demonstration to TA
- Friday lab hours: presentation and discussion of new project
- No group projects

---

**Programming Projects**

Build your own 3D rendering engine
- **Project 1**: Matrices, Vectors, and Coordinate Transformations
- **Project 2**: Interactive Viewing
- **Project 3**: Rasterization
- **Project 4**: Lighting and Texturing
- **Project 5**: Scene Graphs
- **Project 6**: Shader Programming
- **Final Project**

---

**Tests**

**Midterm**
- In class
- Tentatively Thu 11/08

**Final**
- Thursday, Dec 13, 08:00a - 10:59a
- Place TBD

---

**Grading**

- Project 1-6: 10% each
- Final project: 15%
- Midterm: 10%
- Final exam: 15%

---

**Prerequisites**

**Basic familiarity with**
- Linear Algebra
- C++ (if you know Java you’ll be able to adapt)
- Object oriented programming
...and you’ll get more comfortable with them as we proceed!
### Questions?

<table>
<thead>
<tr>
<th>Today</th>
</tr>
</thead>
</table>
| • Course organization  
• Course overview  
• Linear algebra review |

<table>
<thead>
<tr>
<th>What is computer graphics?</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Applications</strong></td>
</tr>
</tbody>
</table>
| Movie, TV special effects  
Video games  
Scientific visualization  
GIS (Geographic Information Systems)  
Medical visualization  
Industrial design  
Simulation  
Communication  
Etc. |

<table>
<thead>
<tr>
<th>What is computer graphics?</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Research areas</strong></td>
</tr>
</tbody>
</table>
| Image Synthesis (rendering)  
3D Modeling  
Animation  
Simulation  
Interaction  
Image processing  
Vision  
Perception  
Hardware design  
Etc. |
### 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
- Different objectives
  - Photorealistic
  - Interactive
  - Artistic

### Photorealistic rendering

- Physically-based simulation of light, camera
- Shadows, realistic illumination, multiple light bounces
- Slow, minutes to hours per image
- Special effects, movies
- CSE168

### Interactive rendering

- Produce images within milliseconds
- Using specialized hardware, graphics processing units (GPUs)
- Standardized APIs (OpenGL, DirectX)
- Often “as photorealistic as possible”
- Hard shadows, fake soft shadows, only single bounce of light
- Games
- CSE167

### Artistic rendering

- Stylized
- Artwork, illustrations, data visualization
Artistic rendering

Modeling

- Creating 3D geometric data
  - The “model” or the “scene”
- By hand
  - Autodesk (Maya, AutoCAD), LightWave 3D, ...
- Free software
  - Blender
- Not as easy to use as Notepad...

Modeling

- Basic 3D models consist of array of triangles
- Each triangle stores 3 vertices
- Each vertex contains
  - xyz position
  - Color
  - Etc.

Modeling

- Procedurally, i.e. by writing programs
- Scanning real-world objects

Animation

- Deforming or editing the data
- Change over time
- Faces, articulated characters, ...
- CSE169
What you’ll learn in this class

The Basics...
- Rendering 3D models
  - Camera simulation
  - Lighting, shading
- Modeling
  - Triangle meshes
  - Smooth surfaces
- Experience with Linear Algebra, C++, OpenGL
- Background for advanced topics (CSE168, CSE169)

Questions?

Today
- Course organization
- Course overview
- Linear algebra review

Linear algebra review
Why linear algebra?
- Need to describe 3D scenes
  - Position, orientation, motion of objects
  - Relation of objects to virtual camera
  - Projection of scene onto image plane
- Linear algebra provides mathematical tools to do so
Vectors
- Direction and length in 3D
- Vectors can describe
  - Difference between two 3D points
  - Speed of an object
- Vectors are in bold-face

Multiplication by scalar

Addition

Linear combination

Linearly dependent vectors
- A set of vectors \( \mathbf{a}_i, i = 1 \ldots n \) is linearly dependent if there exist scalars \( s_i \) such that
  \[
  \mathbf{a}_j = \sum_{i=1,i\neq j}^{n} s_i \mathbf{a}_i
  \]
  - Otherwise, they are linearly independent
Coordinate systems
- Describe any vector with respect to three basis vectors \( x, y, z \)
- The basis vectors form a coordinate system

Coordinate systems
- Any three vectors that are linearly independent could be used as a basis
  - Different lengths
  - Not perpendicular to each other
- Why linearly independent?
- Why exactly three vectors?
- Other coordinate systems?

Coordinate systems
- Any three vectors that are linearly independent could be used as a basis
  - Different lengths
  - Not perpendicular to each other
- Why linearly independent?
- Why exactly three vectors?
- Other coordinate systems?

Euclidean coordinate systems
- Basis vectors
  - Have unit length
  - Are perpendicular to each other
- Orthonormal

Vector arithmetic using coordinates
\[
\begin{align*}
a &= \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} & b &= \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \\
\end{align*}
\[
\begin{align*}
a + b &= \begin{bmatrix} a_1 + b_1 \\ a_2 + b_2 \\ a_3 + b_3 \end{bmatrix} & a - b &= \begin{bmatrix} a_1 - b_1 \\ a_2 - b_2 \\ a_3 - b_3 \end{bmatrix} \\
-s &= \begin{bmatrix} -a_1 \\ -a_2 \\ -a_3 \end{bmatrix} & sa &= \begin{bmatrix} sa_1 \\ sa_2 \\ sa_3 \end{bmatrix}
\end{align*}
\]
Questions?

Vector Magnitude
• The magnitude (length) of a vector is:
  \[ |\mathbf{v}|^2 = v_x^2 + v_y^2 + v_z^2 \]
  \[ |\mathbf{v}| = \sqrt{v_x^2 + v_y^2 + v_z^2} \]
• A vector with length=1.0 is called a unit vector
• We can also normalize a vector to make it a unit vector
  \[ \frac{\mathbf{v}}{|\mathbf{v}|} \]
• Unit vectors are often used as surface normals

Dot Product
• The dot product is a scalar value that tells us something about the relationship between two vectors
• Angles between vectors
• Lengths of vectors

Next class
• Matrices and transformations
• No lab tomorrow
• Lab Wednesday: introduction to the base code, project 1

\[ \mathbf{a} \cdot \mathbf{b} = \sum a_i b_i \]
\[ \mathbf{a} \cdot \mathbf{b} = a_x b_x + a_y b_y + a_z b_z \]
\[ \mathbf{a} \cdot \mathbf{b} = |a||b| \cos \theta \]