Computergrafik 1

Andreas Butz
Ludwig-Maximilians-Universität München

Sommersemester 2014
Chapter 1 - Introduction, Motivation, Basics

• About this Class: Organization
• Tutorials
• Why Should I Learn about Computer Graphics?
• Very Brief History of Computer Graphics
• Math Recap: What We Need to Survive...
About this class: Organization

• Mainly Bachelor Medieninformatik, 4th semester
• „Vertiefende Themen“ in Bachelor Informatik + MI
• All others, please check how course can be counted

• Lecture: Andreas Butz
• Thursday, 2-4pm, Hauptgebäude, Room E004
  – Lecture (2 hours) + tutorials (2 hours)
  – Start c.t., break?

• Web page http://www.medien.ifi.lmu.de/lehre/ss14/cg1/
• PDF of the slides: night before class, print out and bring to take notes and **fill in blanks**
• Podcast: night after class
Chapter 1 - Introduction, Motivation, Basics

• About this Class: Organization

• Tutorials

• Why Should I Learn about Computer Graphics?
• Very Brief History of Computer Graphics
• Math Recap: What We Need to Survive...
About the Tutorials: Organization

• Tutorials: Henri Palleis
  – Will start on April 28th
  – Discussion of assignments

• Weekly assignments, in sync with lecture
  – [http://www.medien.ifi.lmu.de/lehre/ss14/cg1/](http://www.medien.ifi.lmu.de/lehre/ss14/cg1/)
  – Submission voluntary (means: No bonus points!)
  – students who did the exercises statistically got better grades!

• Purpose:
  – In-depth understanding of concepts from lecture
  – Gaining some basic practical experience in low-level graphics programming
  – Preparation for written test (best preparation strategy: do the assignments)
  – Please note: Tutorials and assignments are a service for the students
Schedule I

- Tutorial dates:
  - Mon 12-14, Geschw.-Scholl-Pl. 1, Rm A U117
  - Mon 14-16, Amalienstr. 73A, Rm 114
  - Tue 08-10, Amalienstr. 73A, Rm 114
  - Tue 10-12, Amalienstr. 73A, Rm 114
  - Wed 16-18, Amalienstr. 73A, Rm 114
Schedule II

• Registration for the tutorials will open **tonight at 8pm**

• Register via UniWorX:
  [https://uniworx.ifi.lmu.de/](https://uniworx.ifi.lmu.de/)

• First assignment will be published today
• First tutorial on Monday April 28th
Chapter 1 - Introduction, Motivation, Basics

• About this Class: Organization
• Tutorials
• Why Should I Learn about Computer Graphics?
• Very Brief History of Computer Graphics
• Math Recap: What We Need to Survive...
Why should I learn about Computer Graphics?

• Basis for graphical digital media
  – in the heart of your study and many future jobs!

• Basis for recent CG movies and SFX
  – practically no more movies without it!

• Basis for many computer games
  – market bigger than the film industry

source: http://sketchup.google.com
2D vs. 3D graphics vs. Pixels (see „Digitale Medien“)

- **Pixel-based graphics**
  - given resolution, describe color at each pixel
  - basis for digital photography
  - whole research area of image processing

- **2D graphics (aka vector graphics)**
  - uses 2D lines and areas to describe an image
  - 2D drawing programs: Inkscape, Illustrator, Corel Draw, ...

- **3D graphics**
  - describe 3D objects of a scene
  - compute what light would do to these objects
  - compute pixel image from a virtual camera
...so: 3D content on a 2D screen, huh?

• General problem: current screens are 2D
  – for true 3D perception, we need 2 images for the 2 eyes (stereo)
  – this is technically still difficult (need glasses)
  – research area of volumetric or (auto)stereoscopic displays

• Content is 3D, display is 2D: what problems does this bring?
The 3D rendering pipeline (our version for this class)

3D models in model coordinates -> 3D models in world coordinates -> 2D Polygons in camera coordinates -> Pixels in image coordinates

- Scene graph
- Camera
- Animation, Interaction
- Rasterization
- Lights
…this was not the only way to draw this pipeline…
# Lecture Content & Schedule

<table>
<thead>
<tr>
<th>Date</th>
<th>Nr</th>
<th>lecture topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.04.14</td>
<td>1</td>
<td>Organization, Motivation, Basics</td>
</tr>
<tr>
<td>17.04.14</td>
<td></td>
<td>Easter holiday</td>
</tr>
<tr>
<td>24.04.14</td>
<td>3</td>
<td>Mathematics for CG1 (HP)</td>
</tr>
<tr>
<td>01.05.14</td>
<td></td>
<td>CHI 2014</td>
</tr>
<tr>
<td>08.05.14</td>
<td>4</td>
<td>3D Modeling</td>
</tr>
<tr>
<td>15.05.14</td>
<td>5</td>
<td>3D Camera, Culling, Rasterization</td>
</tr>
<tr>
<td>22.05.14</td>
<td>6</td>
<td>Scene Graph</td>
</tr>
<tr>
<td>29.05.14</td>
<td></td>
<td>holiday</td>
</tr>
<tr>
<td>05.06.14</td>
<td>7</td>
<td>Light, Materials, Appearance</td>
</tr>
<tr>
<td>12.06.14</td>
<td>8</td>
<td>Shading and Rendering (HP?)</td>
</tr>
<tr>
<td>19.06.14</td>
<td></td>
<td>holiday</td>
</tr>
<tr>
<td>26.06.14</td>
<td>9</td>
<td>Basics of 3D Animation</td>
</tr>
<tr>
<td>03.07.14</td>
<td>10</td>
<td>Interactive 3D Graphics</td>
</tr>
<tr>
<td>10.07.14</td>
<td></td>
<td>final project presentation??</td>
</tr>
</tbody>
</table>
Literature Recommendations and links

• Malaka, Butz, Hussmann: Medieninformatik, Pearson Studium 2009
  – v.a. Kapitel 8: 3D-Grafik

• Bungartz, Griebel, Zenger: Einführung in die Computergraphik, 2. Auflage, Vieweg, 2002


• OpenGL: www.opengl.org

• Three.js: http://threejs.org/
Chapter 1 - Introduction, Motivation, Basics

• About this Class: Organization
• Tutorials
• Why Should I Learn about Computer Graphics?
• Very Brief History of Computer Graphics
• Math Recap: What We Need to Survive...

Based on lecture material by Regina Pohle-Fröhlich

1945-1952: “Whirlwind” computer (Jay Forrester, MIT)
Digital computer using oscilloscope screen displaying real-time aircraft data, later “SAGE” system

“Bouncing ball” (C. Adams)
Using “light pen” for input

1957-1969: “TX-2” computer at MIT Lincoln Lab
Transistor-based computer providing interactive graphic displays
Ivan Sutherland, 1963: Sketchpad

wired.com
design.osu.edu/carlson/history
www.rendering.ovgu.de
research.microsoft.com
computerhistory.org
Theory Development in the 1970s

- 1971: Raster Scan Principle (M. Noll, Bell Labs)
  - Connecting a TV-like display with computer memory
- 1973: First ACM “SIGGRAPH” Conference
- 1977-1978: Shadow computation (Crow, Williams)
- 1975: 3D Model “Utah Teapot” (M. Nevell, U. Utah)
- 1979: Raytracing (mirror reflection, transparency) (Kay, Whitted)
- 1984: Global illumination model “radiosity”
  (Goral et al., Nishita)

Utah Teapot
at Computer History Museum, Boston

wikipedia.org
Computer Graphics Goes to Cinema: 1980s

• 1979: Computer Graphics department of Lucas Film founded (ILM)
• 1980: Demonstration of video “Vol Libre” (L. Carpenter) at SIGGRAPH
• 1980: Computer Animations in movie “Tron”
• 1981: Predecessor of “Renderman” (REYES) by L. Carpenter at Lucas Film
• 1986: “Pixar” founded (Catmull, Smith), split off Lucas Film
• 1988: Movie “The Abyss” (Water creature by Lucas Film ILM)
• 1989: Motion Capturing (Jim Henson)
• 1995: Movie “Toy Story” (Pixar, fully computer-generated)
Chapter 1 - Introduction, Motivation, Basics

• About this Class: Organization
• Tutorials
• Why Should I Learn about Computer Graphics?
• Very Brief History of Computer Graphics
• Math Recap: What We Need to Survive...
Coordinate Reference Frames

• Dimensionality
  – We will meet: 2, 3 and 4 dimensions

• Types of coordinate systems
  – Cartesian (rectilinear): Pairwise orthogonal axes with (identical) linear scale
  – Non-cartesian (curvilinear): Many other systems
    • e.g. polar/spherical coordinates: angle plus distance
2D Cartesian Coordinate Reference Frames

Device-independent commands of graphics packages:
Varying schemata: origin may be in lower-left corner, center, upper-left corner

Device coordinates:
- Scan lines on cathode ray tubes, printers:
  origin in upper left corner, y axis points downwards
- Other devices: Origin in lower-left corner

Normalized device coordinates: Range from 0.0 to 1.0 (real number)
Physical device coordinates: Integers (pixel address)
Standard 3D Cartesian Coordinate Reference Frames

- Most frequently used “world coordinates” (e.g. in OpenGL): “Right handed” system, often depicted as looking from z axis

- “Left handed” system used in special cases (e.g. 2D screen positions with additional depth information)

Pictures: euclidienspace.com, cornell.edu
Points and Vectors

• **Point**
  – Position specified with coordinate values in some reference frame
  – e.g. in 3D Cartesian coordinates: \((p_x, p_y, p_z)\)

• **Vector**
  – Tuple of real numbers, considered as element of a vector space
  – Often written vertically (column vector)
  – In CG, people are sloppy about the difference between row and column vectors!

• Difference between two positions gives a vector
• Position can be specified by vector from origin in Cartesian system
• Vectors can be multiplied with a real number pointwise
• Two vectors of same length can be added pointwise
Properties of Vectors

• Magnitude (length)

\[
a = \left( a_x, a_y, a_z \right) \quad \|a\| = \sqrt{a_x^2 + a_y^2 + a_z^2}
\]

• Direction angles

\[
\cos \delta_x = \frac{a_x}{\|a\|} \quad \cos \delta_y = \frac{a_y}{\|a\|} \quad \cos \delta_z = \frac{a_z}{\|a\|}
\]
Scalar Product (Dot Product)

- The *scalar product* computes a real (scalar) value from two coordinate vectors of equal length

\[
\begin{pmatrix}
  a_x \\
  a_y \\
  a_z
\end{pmatrix}
\cdot
\begin{pmatrix}
  b_x \\
  b_y \\
  b_z
\end{pmatrix}
= a_x b_x + a_y b_y + a_z b_z
\]

- Application: Computation of angle between two coordinate vectors

\[
a \cdot b = ||a|| \cdot ||b|| \cdot \cos \alpha
\]
Cross Product (Vector Product)

- The *cross product* of two coordinate vectors is a vector which is perpendicular to both given vectors
  - Direction: Right-hand rule
  - Magnitude: Equals spanned parallelogram

\[
\begin{pmatrix}
a_x \\
a_y \\
a_z \\
\end{pmatrix} \times \begin{pmatrix}
b_x \\
b_y \\
b_z \\
\end{pmatrix} = \begin{pmatrix}
a_y b_z - a_z b_y \\
a_z b_x - a_x b_z \\
a_x b_y - a_y b_x \\
\end{pmatrix}
\]
Matrices

- A matrix is an \((m \times n)\) arrangement of real numbers \((m\ \text{rows}, \ n\ \text{columns})\)
- Used in CG for expressing computations on coordinate vectors
- A matrix can be multiplied with a real number pointwise
- Two matrices of identical dimensions can be added pointwise
- Multiplying matrices:
  \((m \times p)\)-matrix \(A\) multiplied by \((p \times n)\)-matrix \(B\) gives \((m \times n)\)-matrix \(C\)

\[
C_{i,j} = \sum_{k=1}^{p} A_{ik} \cdot B_{kj} \quad 1 \leq i \leq m, \\
1 \leq j \leq n
\]
Multiplying a Matrix and a Vector

• Special case of matrix multiplication
• \((m \times p)\)-matrix \(A\) multiplied with vector \(v\) of length \(p\) gives vector \(w\) of length \(m\)

\[
    w_j = \sum_{k=1}^{p} A_{ik} \cdot v_k
\]