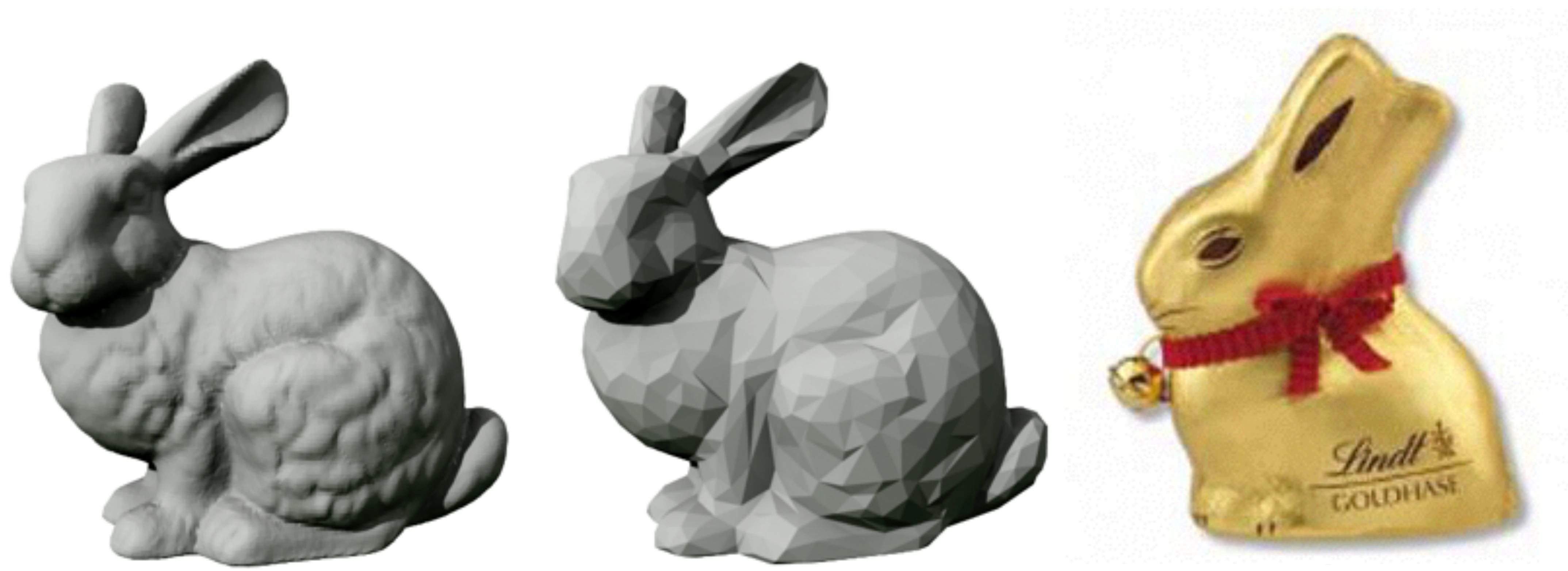
Computer Graphics 1

Ludwig-Maximilians-Universität München

Summer semester 2020

Prof. Dr.-Ing. Andreas Butz

lecture additions by Dr. Michael Krone, Univ. Stuttgart



http://www.wikiwand.com/

Chapter 1 – Introduction, Motivation, Basics

- About this Class: Organization
- Tutorials
- What is Computer Graphics?
- 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,
    also Bachelor "Kunst und Multimedia"
  - All others, please check how this course can be credited
- ~~Tuesday, 10:00 – 12:00, Schellingstr. 3, Room S001~~
  - ~~Lecture (2 hours) + tutorials (2 hours)~~
- Tuesday, 10:00 – 12:00, online in Zoom
  - Video available 1 wk before, zoom meetings for Q&A and for tutorials

- **Asking questions during the zoom meeting is strongly encouraged!**
- Web page: http://www.medien.ifi.lmu.de/lehre/ss20/cg1/
  - PDF of the slides: a week before class
  - Screencast video from 2019: a week before class
  - Access to course material: user “**`cg1`**”, password “**`cg1_sose2020`**”

Chapter 1 – Introduction, Motivation, Basics

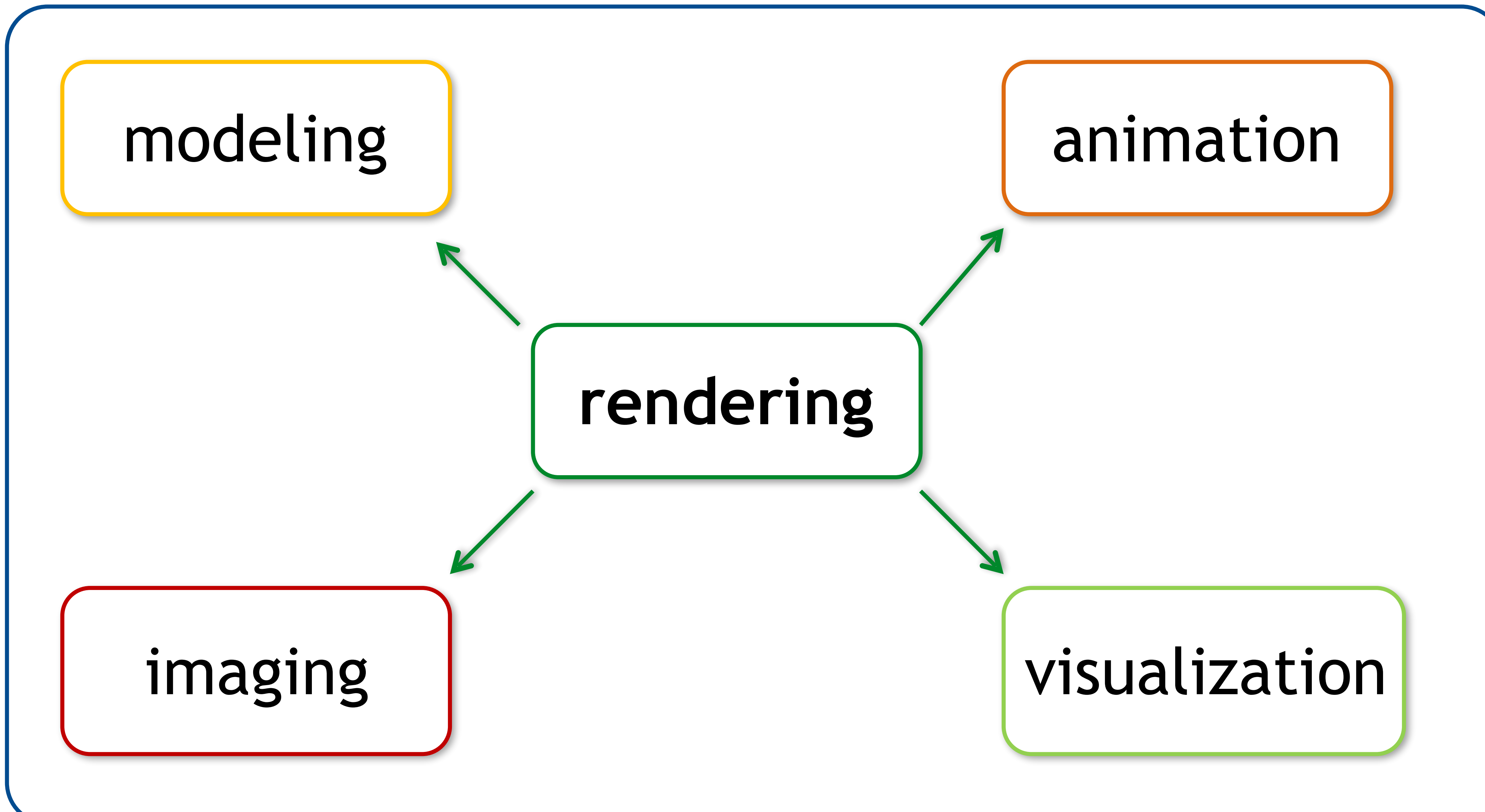
- About this Class: Organization
- Tutorials
- What is Computer Graphics?
- Why Should I Learn about Computer Graphics?
- Very Brief History of Computer Graphics
- Math Recap: What We Need to Survive...

Chapter 1 – Introduction, Motivation, Basics

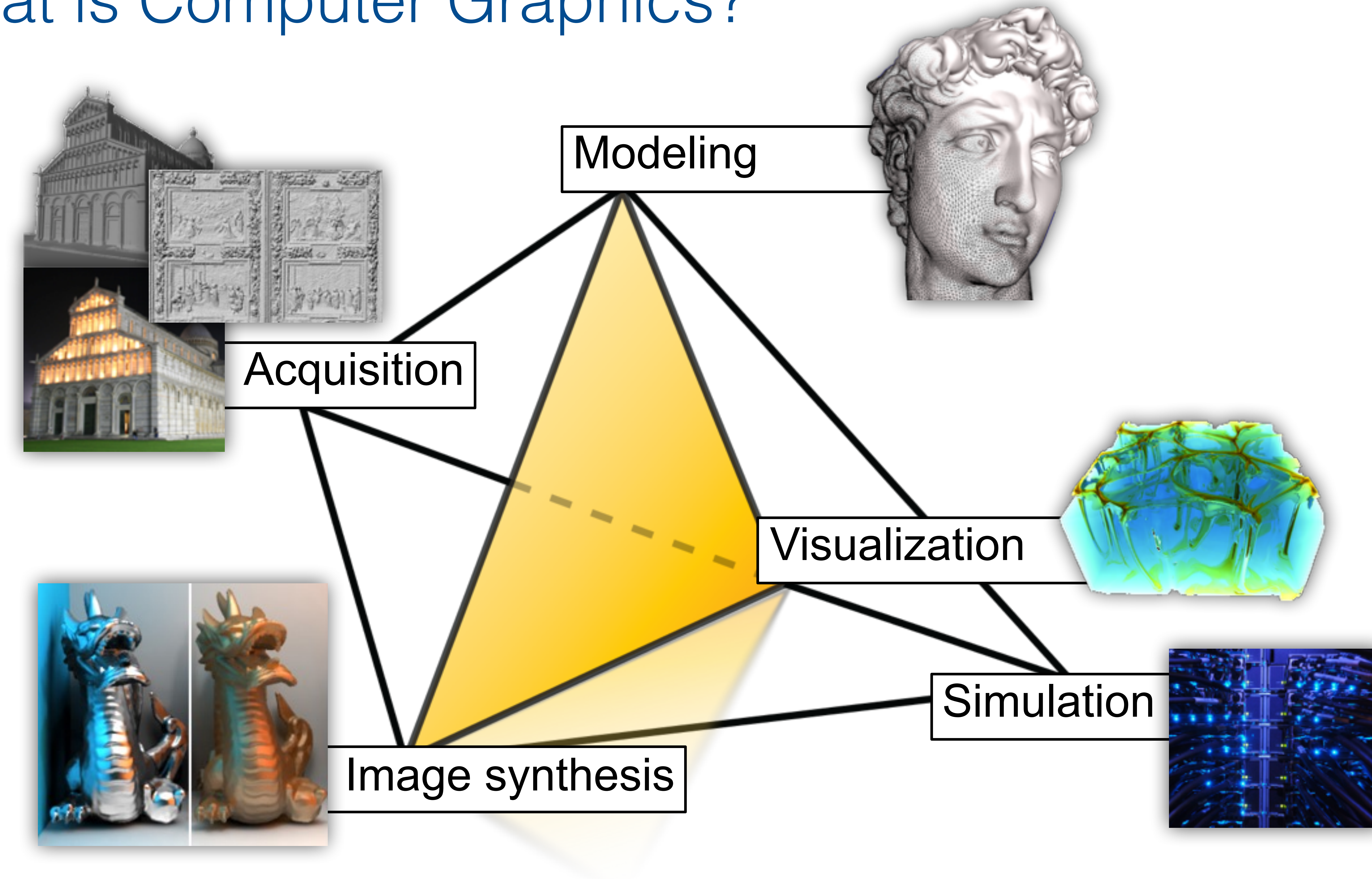
- About this Class: Organization
- Tutorials
- What is Computer Graphics?
- Why Should I Learn about Computer Graphics?
- Very Brief History of Computer Graphics
- Math Recap: What We Need to Survive...

What is Computer Graphics?

- Generation and manipulation of images with computers
- Research areas:

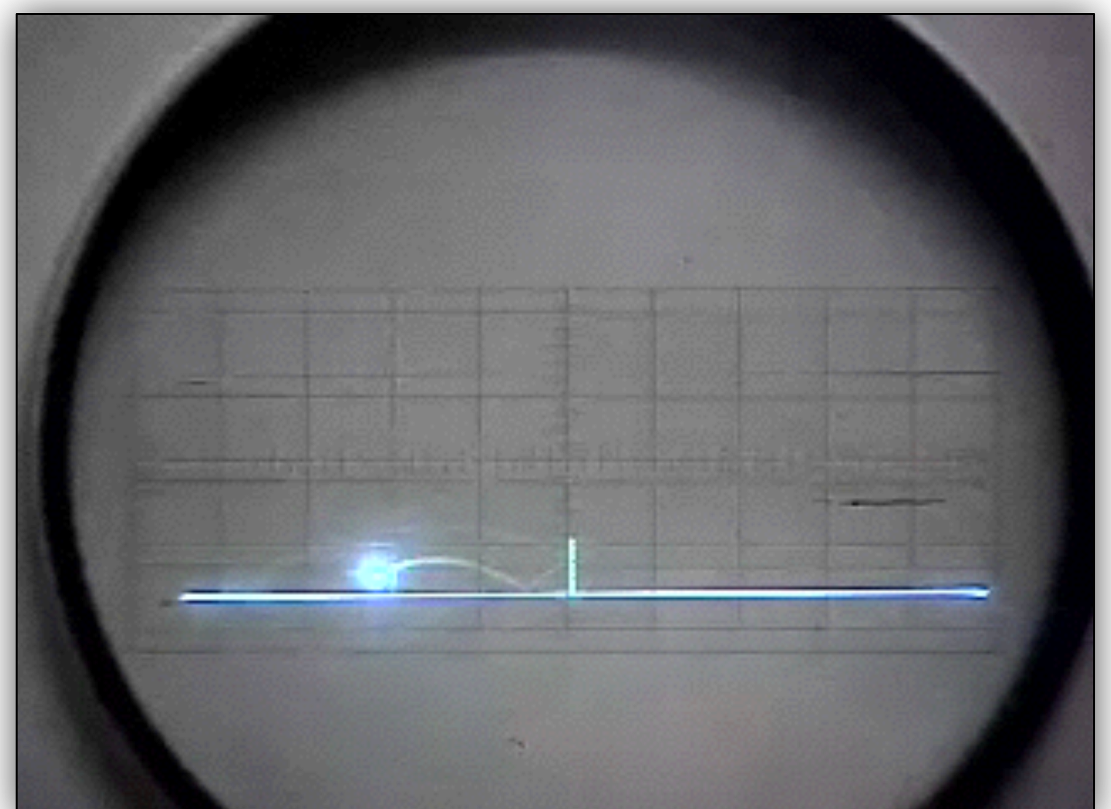


What is Computer Graphics?



Evolution of Computer Graphics in Video Games

- Obviously, CG development was partially motivated by a ludic drive…

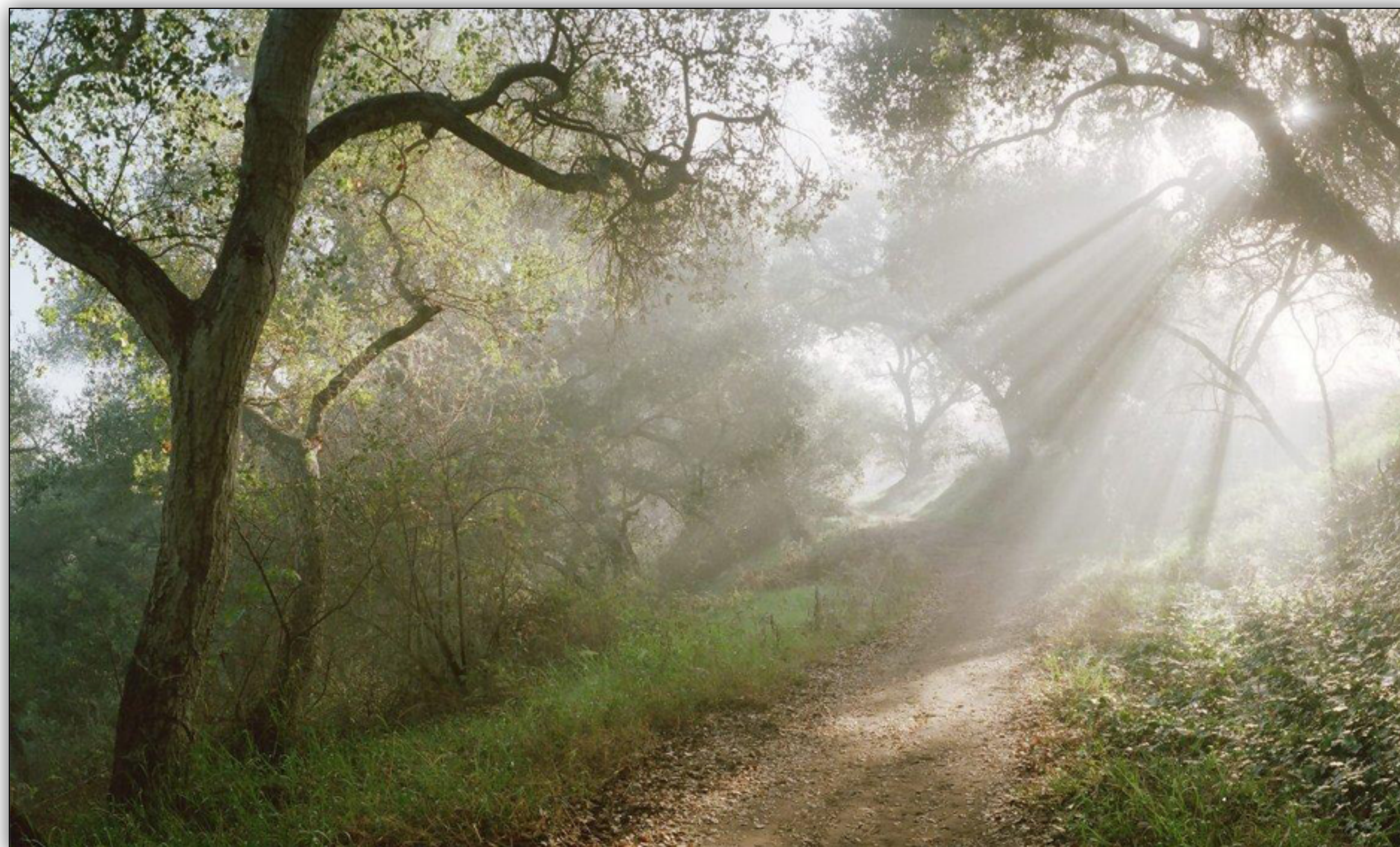


Tennis for Two, 1958
William Higinbotham
Analog computer and oscillograph

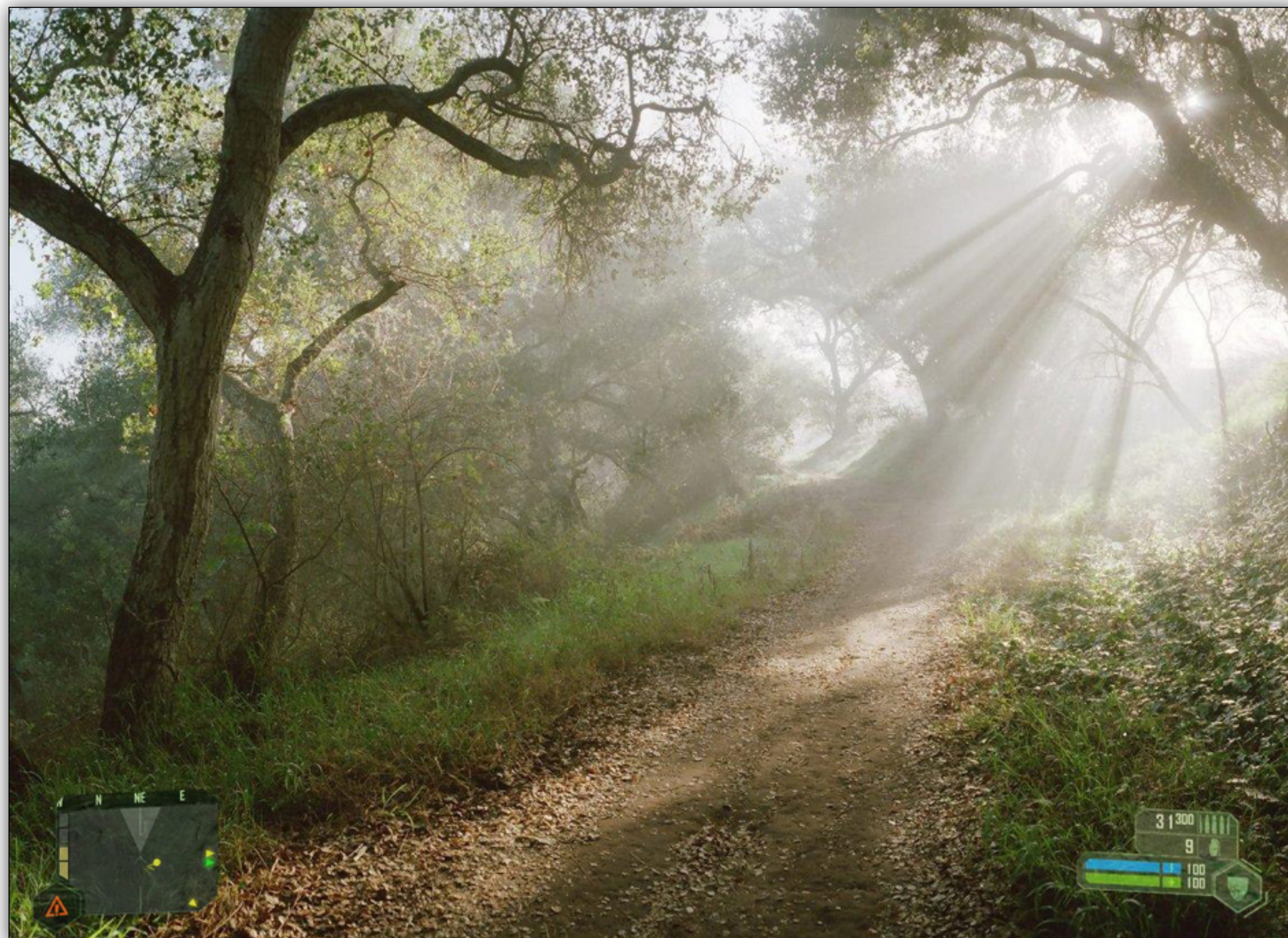
Spacewar!, 1961
MIT Students
DEC PDP-1

Evolution of Computer Graphics in Video Games

When will games reach this degree of realism ?

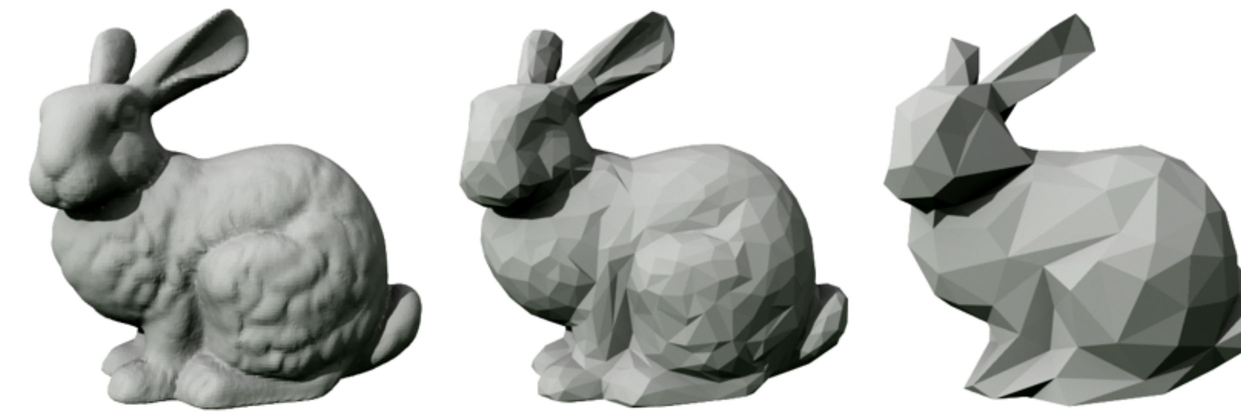


…they already have!

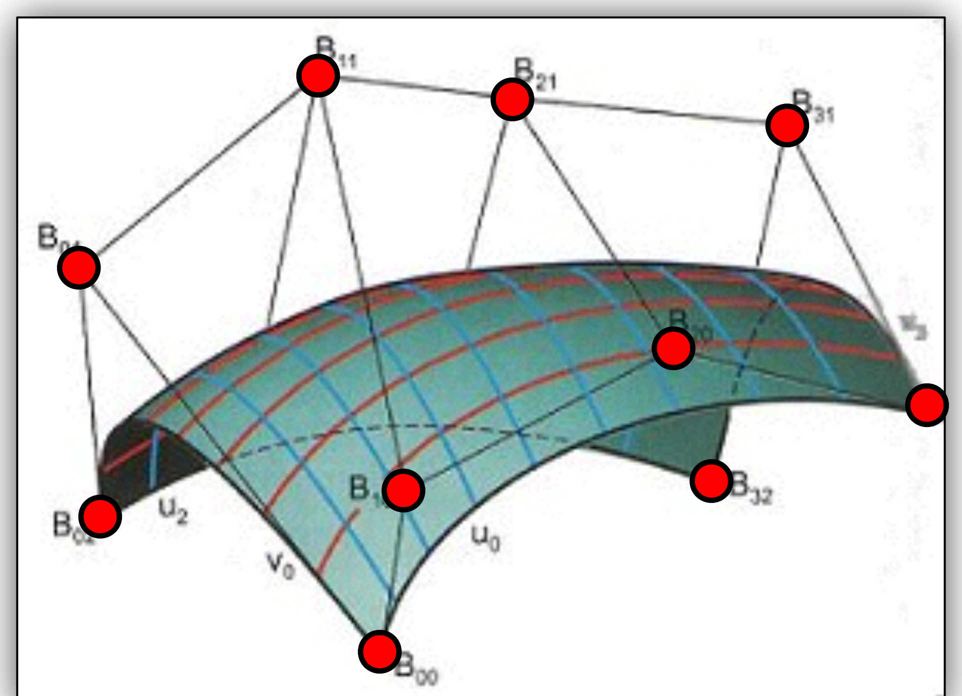


3D Geometry: Description of the shape of objects

- Depiction of the surface
  - Usually via triangles
  - Tessellation (amount/granularity of triangles)

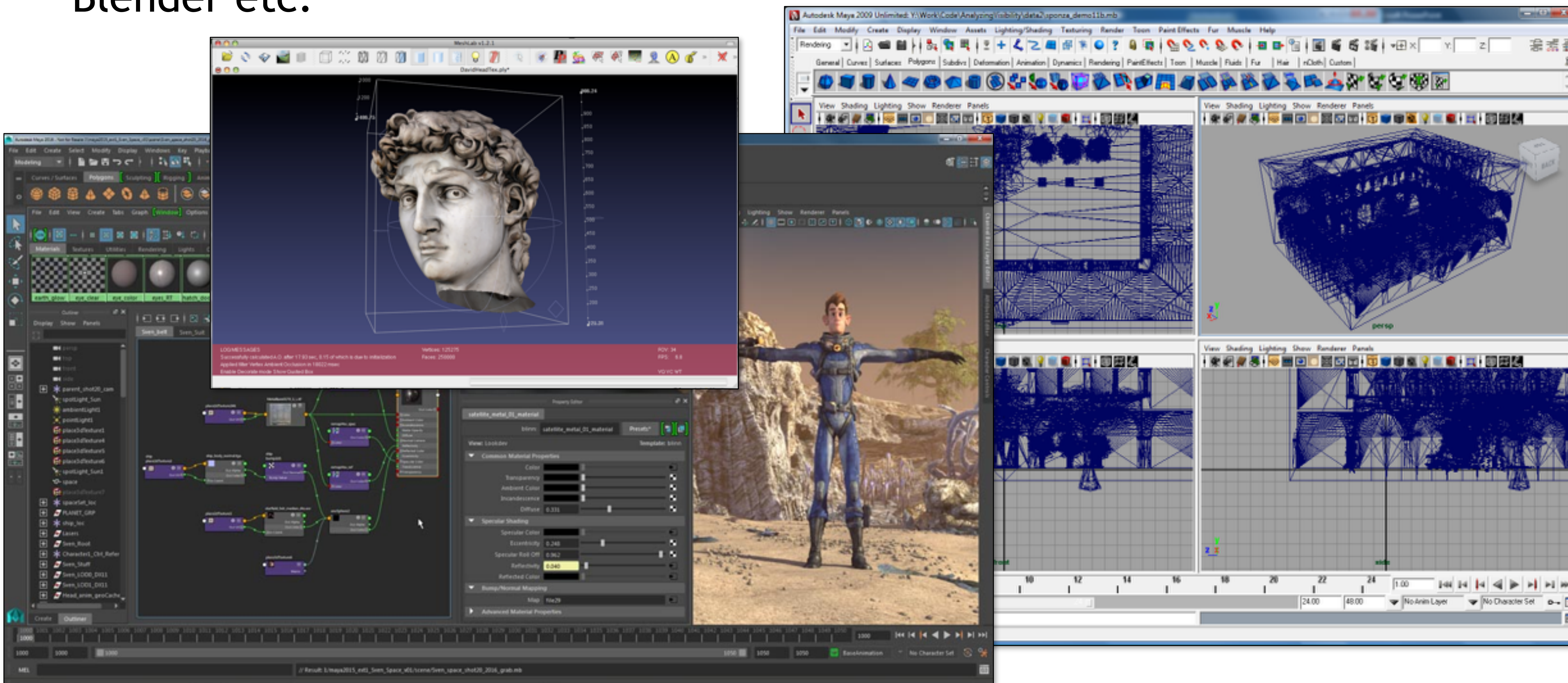


- Free form surfaces
  - Developed independently by Pierre Bézier (Renault) and Paul de Casteljau (Citroën) for the computer-aided construction of car bodies



3D Models

- How are 3D models (triangle meshes) created?
  - Straightforward solution: Explicitly in a modeling tool like Autodesk Maya, Blender etc.



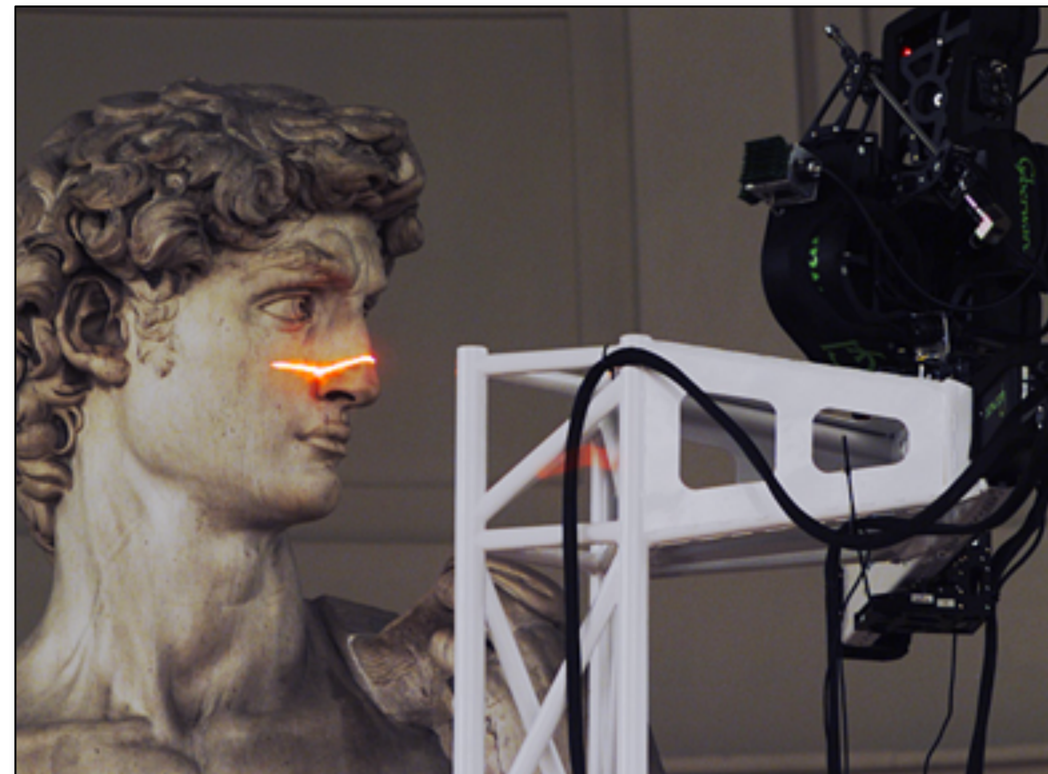
Procedural Models – Example: Rocks

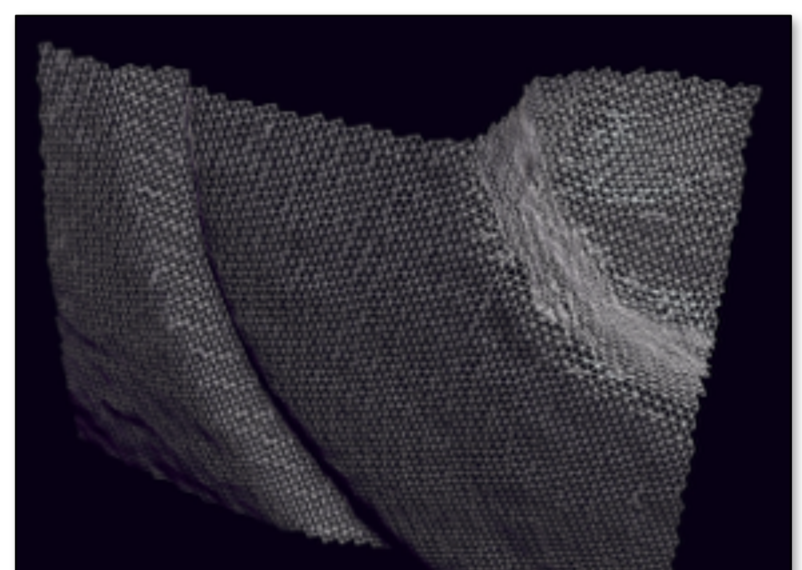
- Generate randomly distributed points and from them, coarse meshes
- Subdivide the triangles and randomly displace their vertices

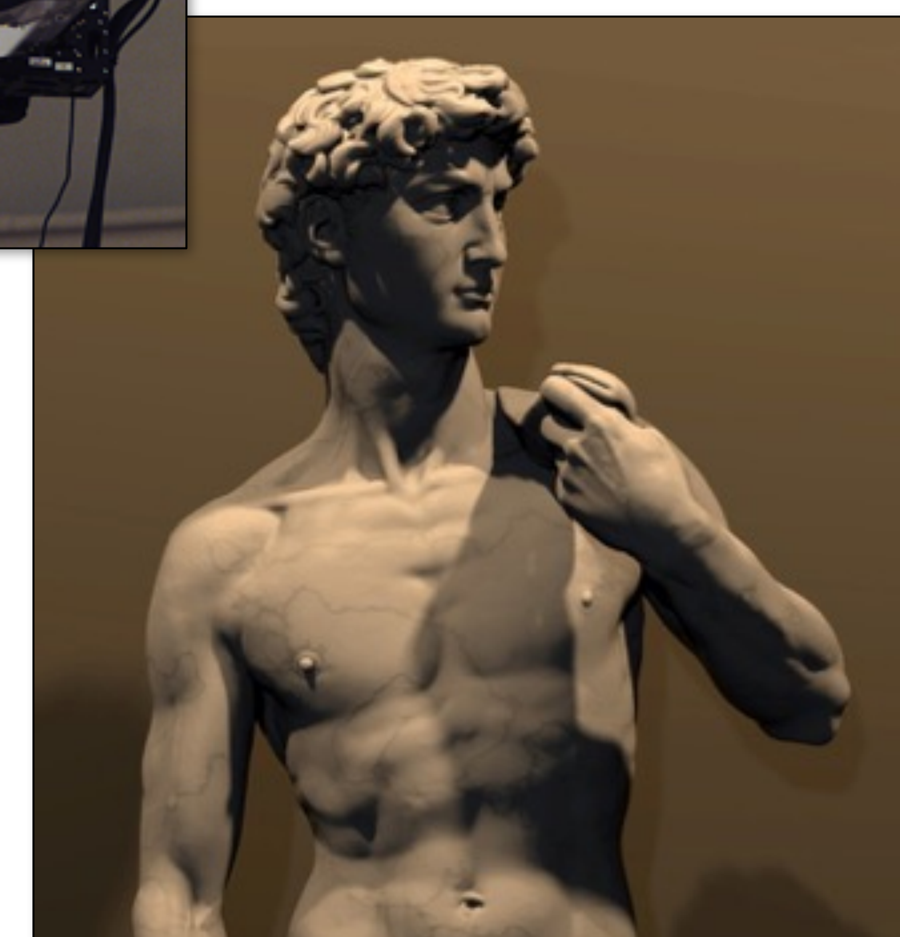
Image: Frank Doassans

Detailed Geometry

- 3D Scanning: Acquisition of surfaces with a laser



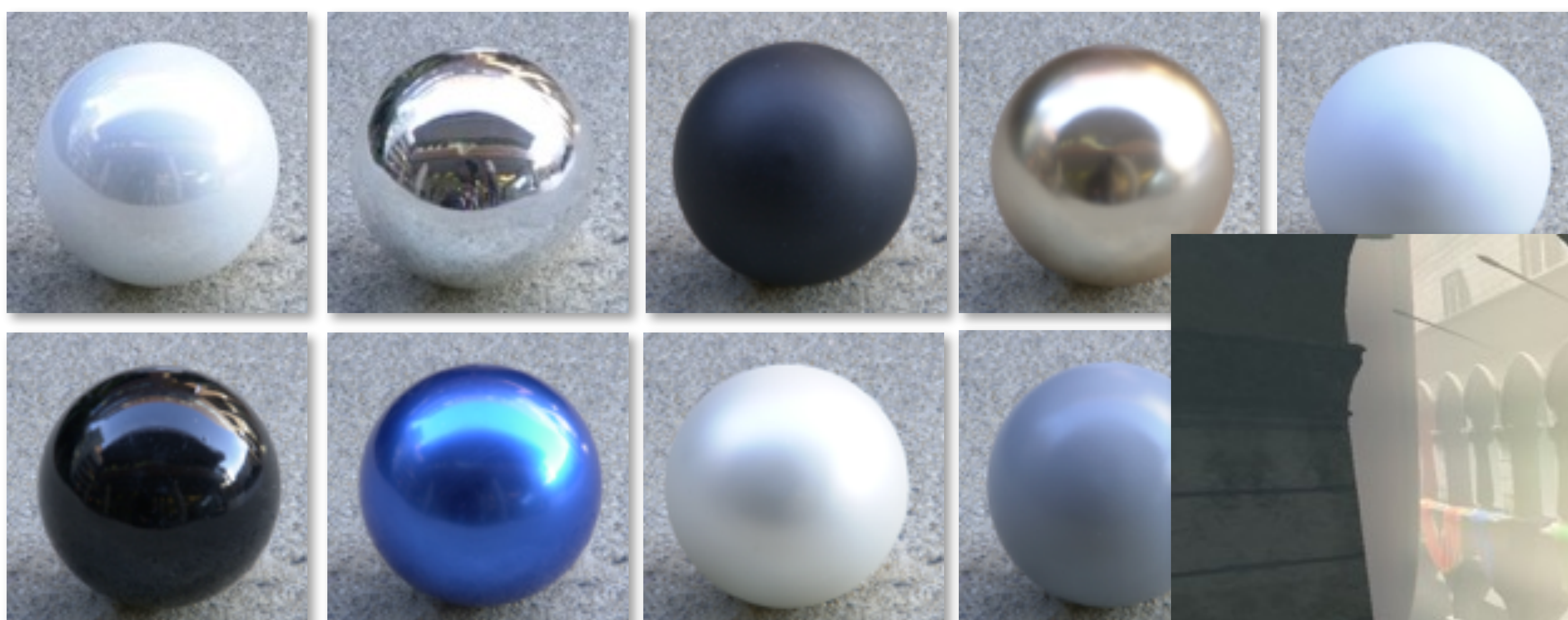




www-graphics.stanford.edu/projects/mich/

What else do we need?

- Material properties (reflectance, opacity etc.)
- Shading, lighting (e.g., photorealistic or illustrative)
- Animation
- …



Chapter 1 – Introduction, Motivation, Basics

- About this Class: Organization
- Tutorials
- What is Computer Graphics?
- 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 scientific visualization
  - Graphical depiction of scientific data
- Basis for most computer games
  - Market bigger than the film industry

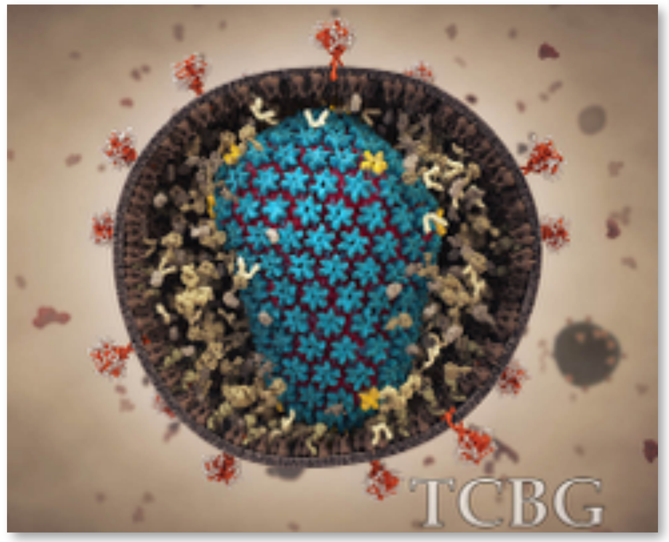


Image source: http://www.ks.uiuc.edu/

Image source: https://www.dirtgame.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

© Maarten Boot

- 2D graphics (aka vector graphics)
  - Uses 2D lines and areas to describe an image
  - 2D drawing programs: Inkscape, Adobe Illustrator, MS PowerPoint, ..

© KeithByrne (via DeviantArt)

- 3D graphics
  - Describe 3D objects of a scene
  - Compute what light would do to these objects
  - Compute pixel image from a virtual camera

© Crystal Dynamics / Square Enix

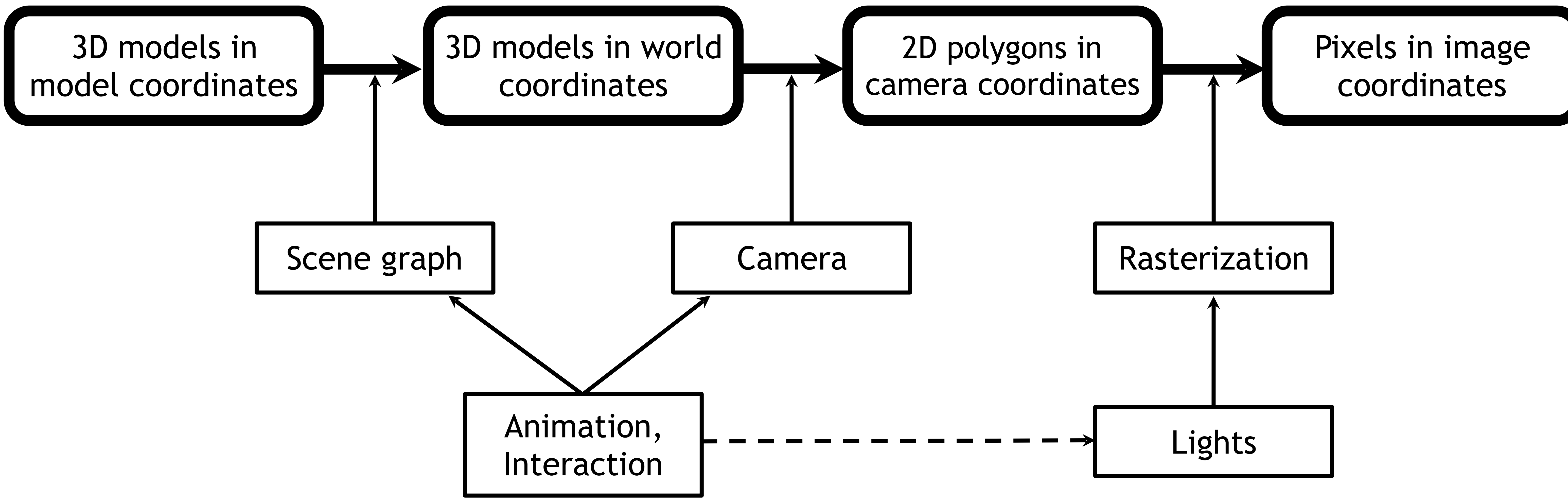
source: http://static.technorati.com/10/01/20/3467/Avatar-movie-Wallpapers.jpg

...so: 3D content on a 2D screen, huh?

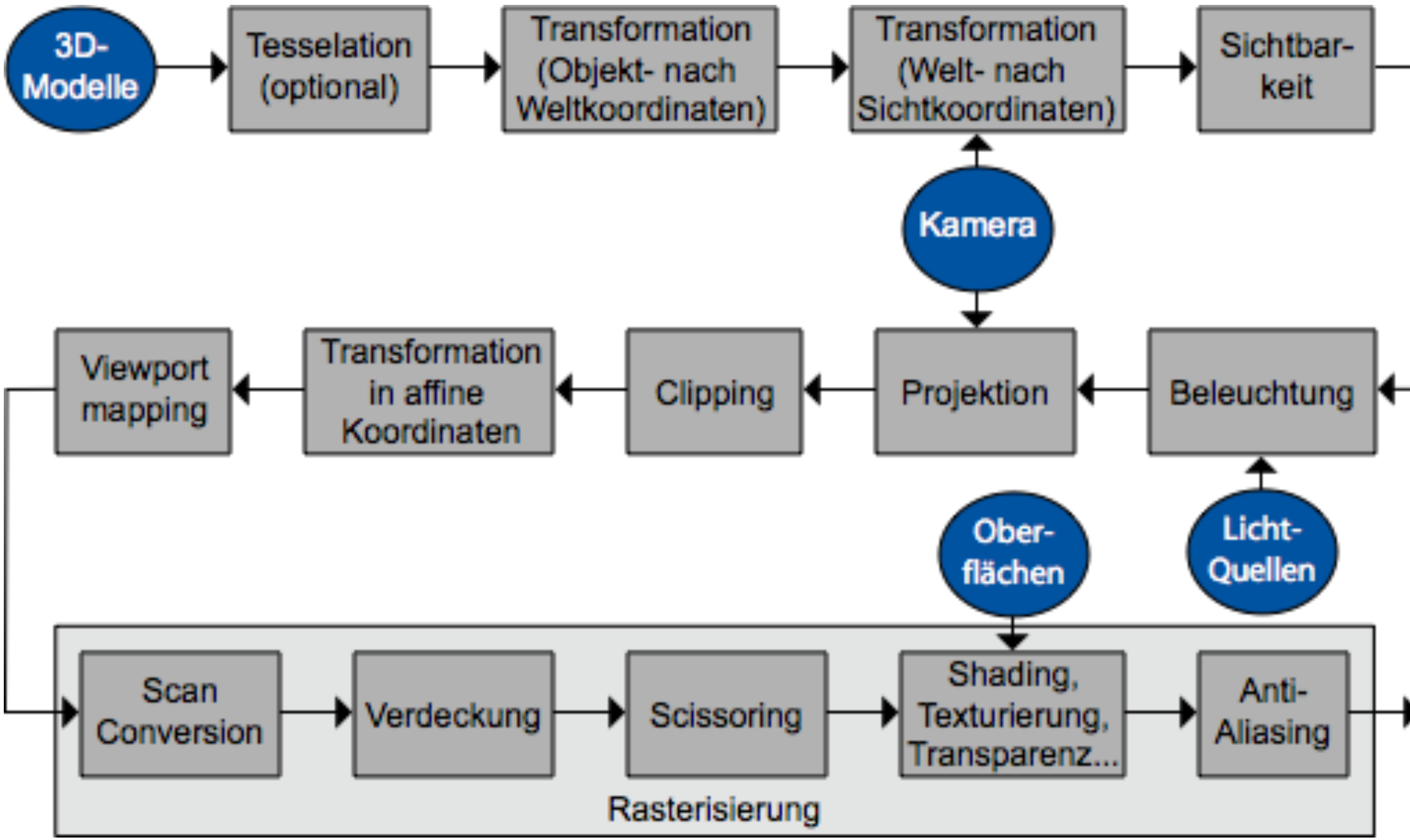
- General problem: current screens are 2D
  - For true 3D perception, we need 2 images for the 2 eyes (stereo)
  - This is technically still difficult (need glasses, e.g., 3D movies in cinema or on modern TV)
  - Research area of volumetric or (auto)stereoscopic displays
  - Alternative: use head-mounted display (Oculus Rift, HTC Vive, Google Cardboard…)

- Content is 3D, display is 2D: what problems does this bring?
  -
  -
  -
  -
  -
  -

The 3D rendering pipeline (our version for this class)

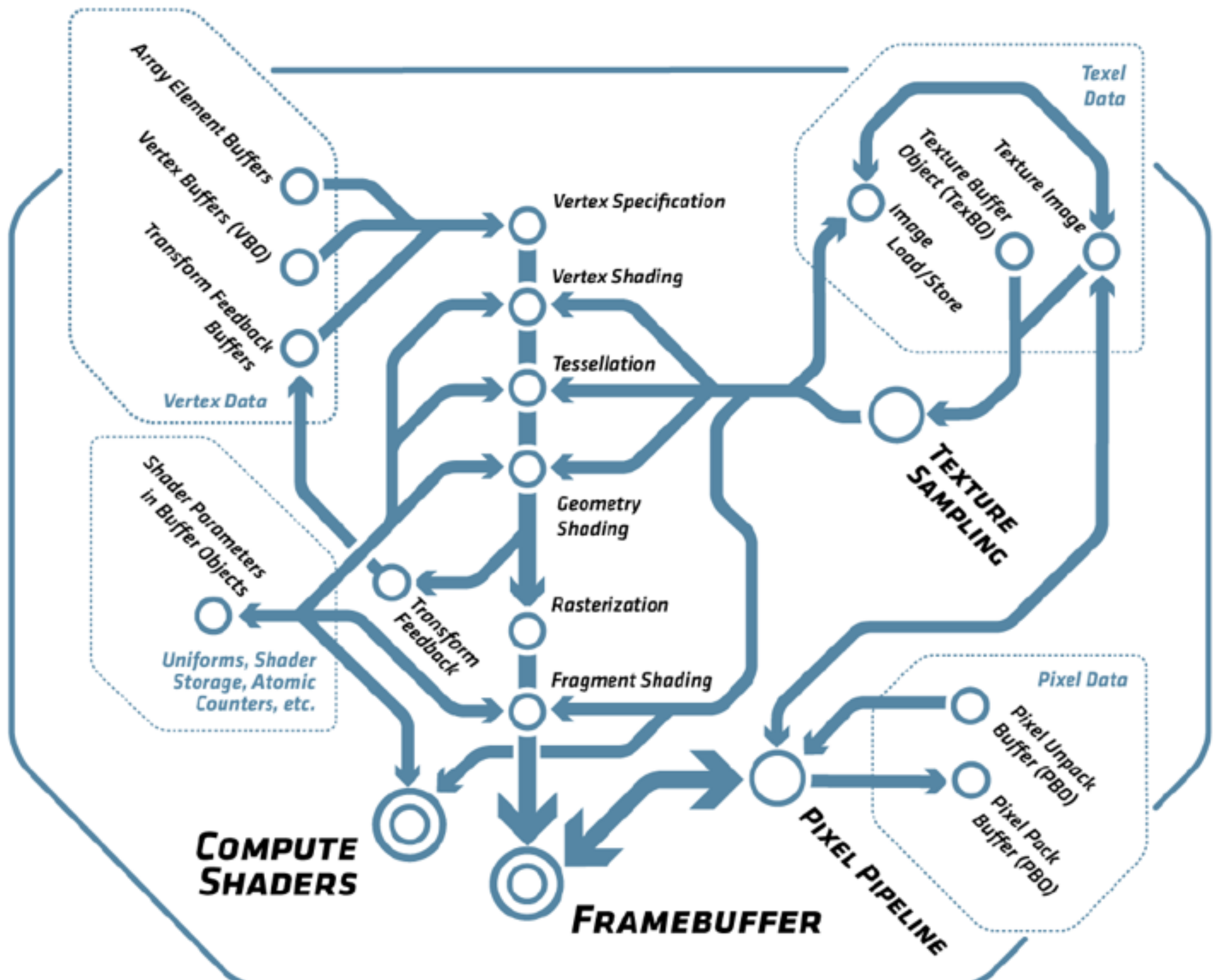


...this was not the only way to draw this pipeline...



...this was not the only way to draw this pipeline...

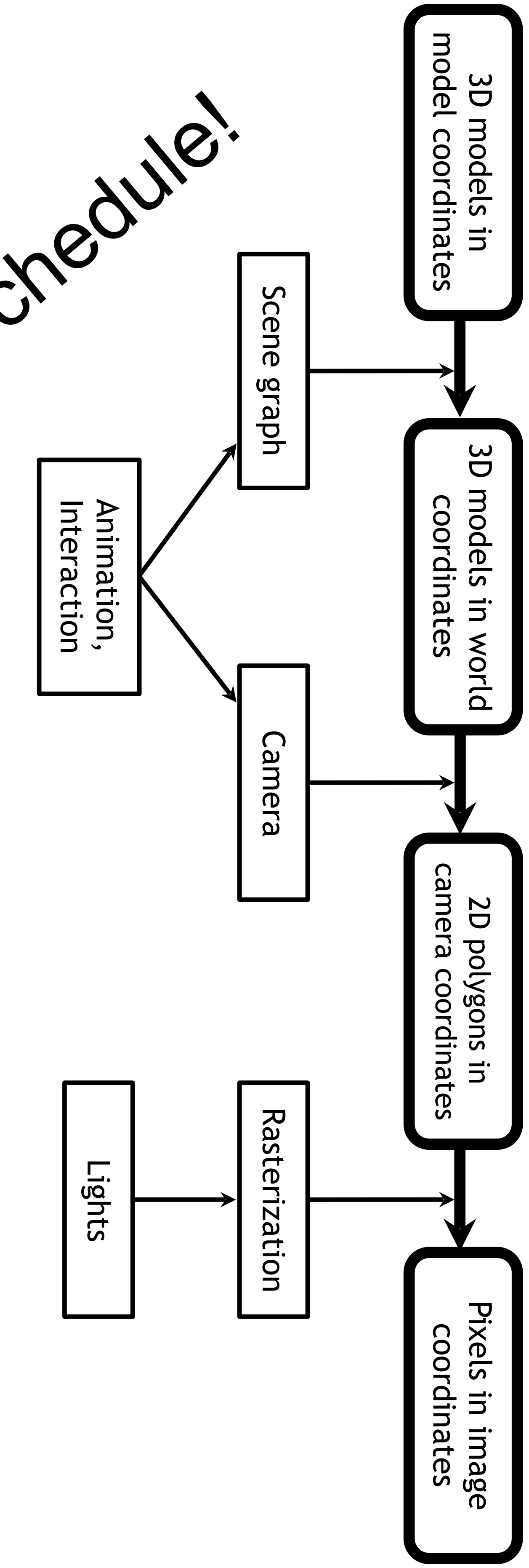
- OpenGL 4.5 Core Profile Specification



Lecture Content & Schedule (as planned)

Date	Chapter	Topics
23.04.19	1	Introduction, Motivation, Basics
30.04.19		AB absent, no class
07.05.19	2	Transformations & Scene Graphs
14.05.19	3	3D Modeling
21.05.19	5	3D Camera & Rasterization
28.05.19	6	Light, Materials & Appearance
04.06.19	7a	Shading and Rendering
11.06.19	4	OpenGL (David Englmei
18.06.19		AB absent, no class
25.06.19	7b	Shading and R onte Carlo Methods
02.07.19	8	Animation
09.07.19	9	Interaction
16.07.19	10	Volume Rendering & Scalar Field Visualization
23.07.19	G	Guest: Markus Groß (head of Disney research Zürich)

Exact dates are outdated, since from 2019.
Please see lecture web page for up-to-date schedule!



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
- Hearn, Baker, Carithers: *Computer Graphics with OpenGL*, 4th edition, Pearson 2011
- Foley, Van Dam, Feiner: *Computer Graphics - Principles and Practice*, 3rd edition, Addison-Wesley, 2013
- Watt, A. et al.: *Advanced Animation and Rendering Techniques.: Theory and Practice*, Addison Wesley, 1992
- OpenGL: http://www.opengl.org/
- Three.js: http://threejs.org/ *(→ WebGL framework used in tutorials!)*

Chapter 1 – Introduction, Motivation, Basics

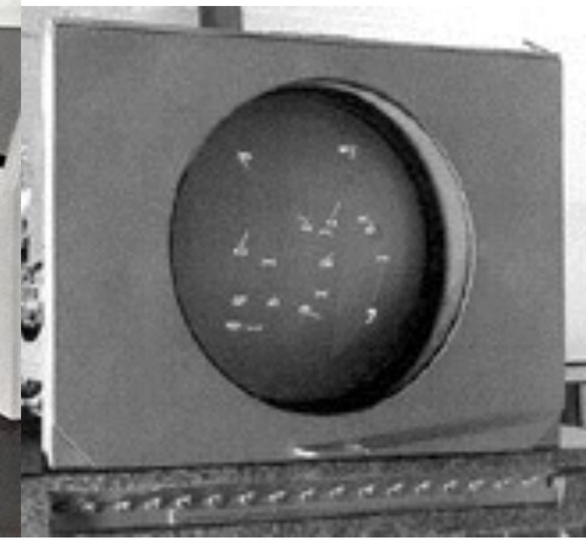
- About this Class: Organization
- Tutorials
- What is Computer Grapics?
- Why Should I Learn about Computer Graphics?
- Very Brief History of Computer Graphics
- Math Recap: What We Need to Survive...

Based on lecture material by Regina Pohle-Fröhlich

First Steps Towards Computer Graphics 1945 – 1963

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

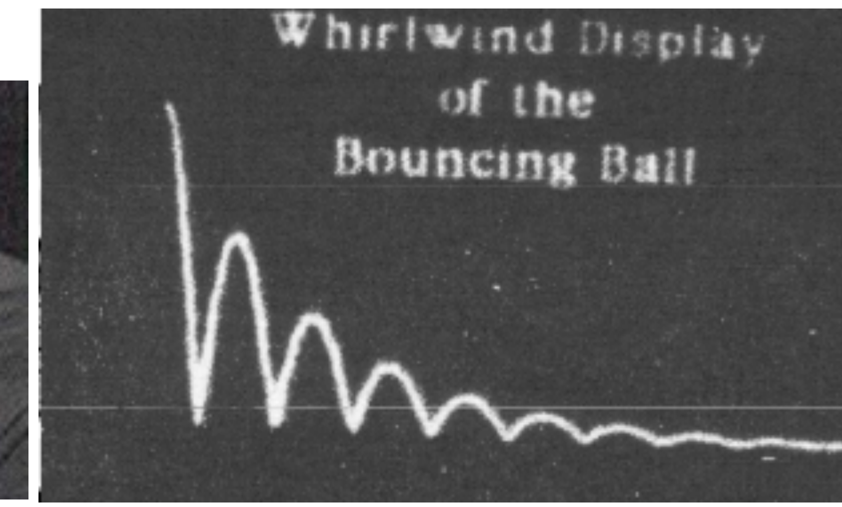
wired.com



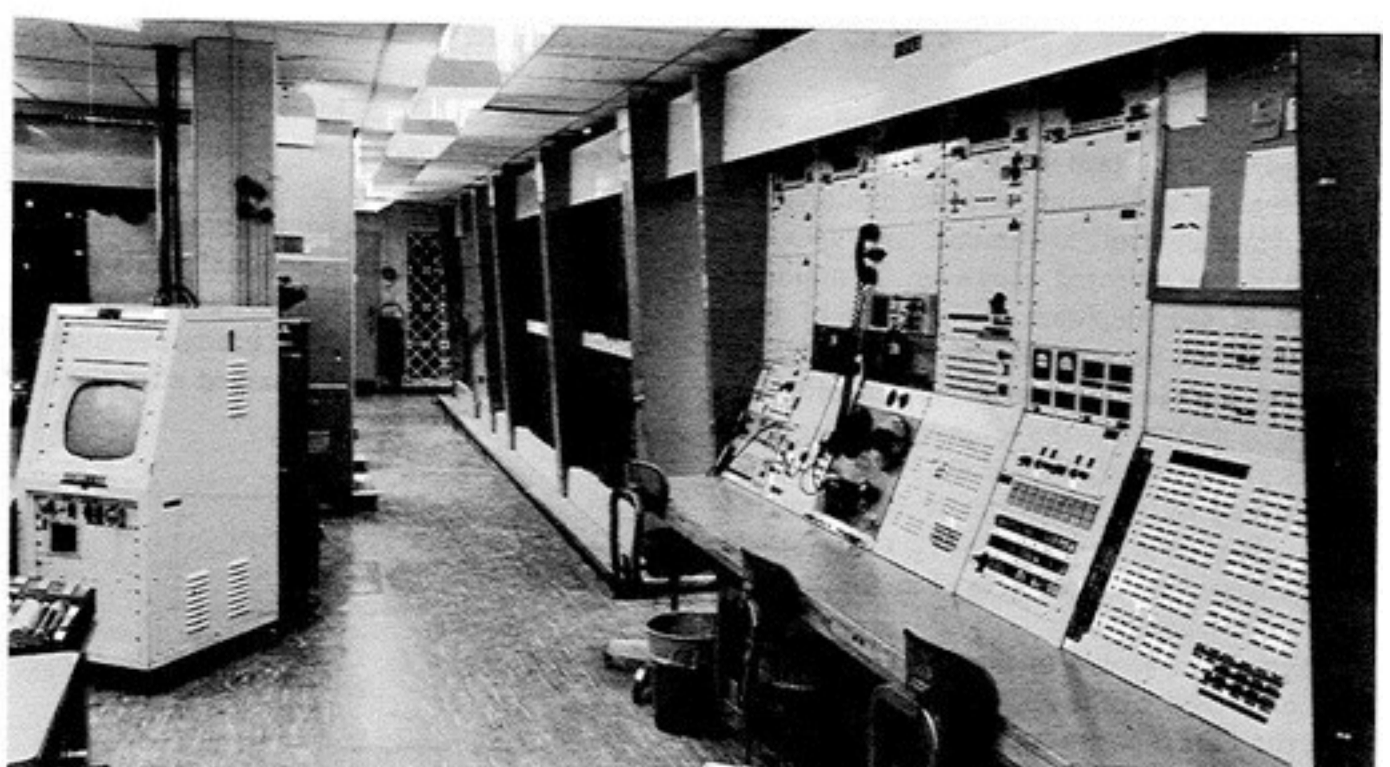
Using “light pen” for input

design.osu.edu/carlson/history

“Bouncing ball” (C. Adams)



www.rendering.ovgu.de



research.microsoft.com

1957-1969: “TX-2” computer at MIT Lincoln Lab
Transistor-based computer providing interactive graphic displays
L.G. Roberts, 1962: 3D Graphics
Ivan Sutherland, 1963: Sketchpad

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
- 1971-1975: Shading algorithms (Gouraud 1971, Phong 1975)
- 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)

ACMSIGGRAPH

Utah Teapot
at Computer History Museum, Boston

wikipedia.org

Computer Graphics goes to Cinema: 1980s

- 1979: CG department of Lucas Film founded (ILM)
- 1980: Demonstration of video "Vol Libre" at SIGGRAPH (L. Carpenter)
- 1980: Computer Animations in movie "Tron"
- 1981: REYES – Predecessor of "Renderman" (by L. Carpenter at Lucas Film)
- 1986: "Pixar" founded (Catmull, Smith), (split off Lucas Film)
- 1988: Movie "The Abyss" (Cameron, water creature by ILM)
- 1989: Motion Capturing (Jim Henson)
- 1995: Movie "Toy Story" by Pixar (first feature-length fully computer-generated film)
- 2009: Movie "Avatar" (J. Cameron; 60% CG; >2.7 billion revenue; special 3D cameras, started 3D boom)

Vol Libre atariarchives.o

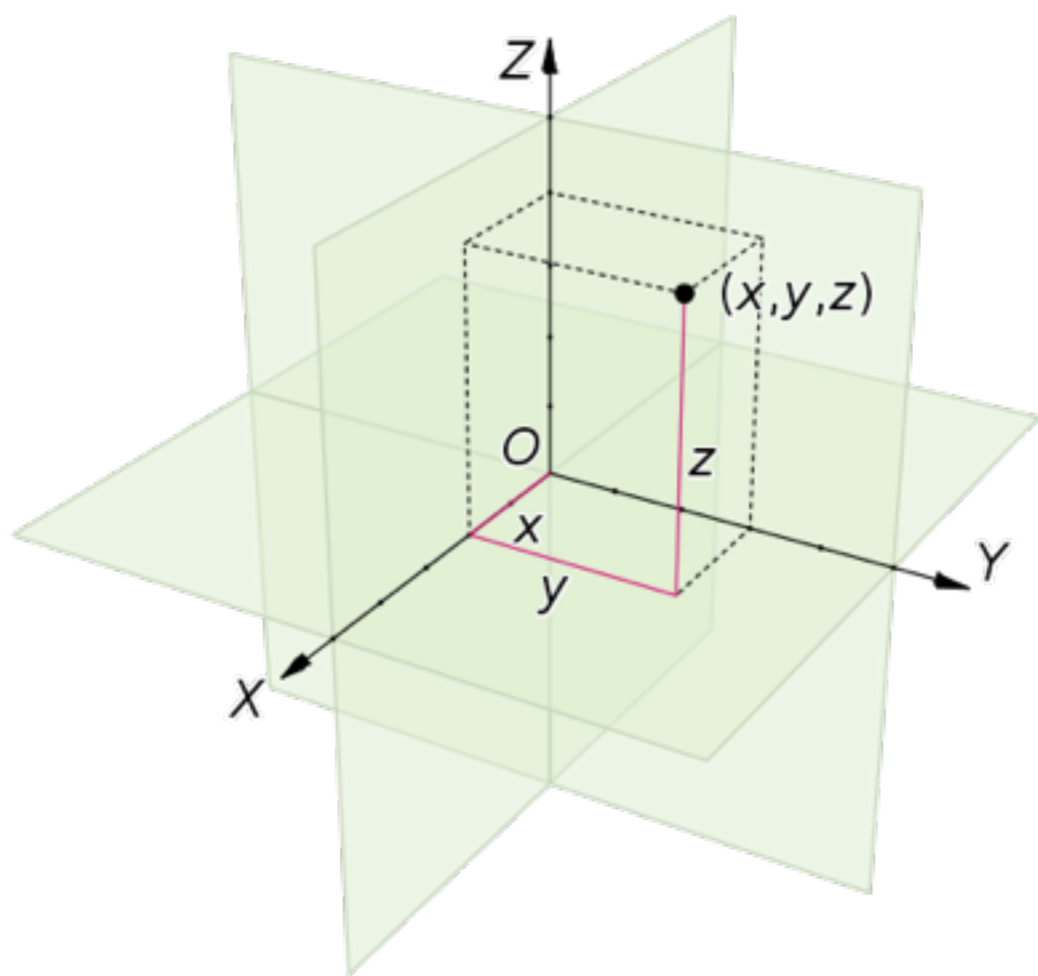
Abyss empireonline.com

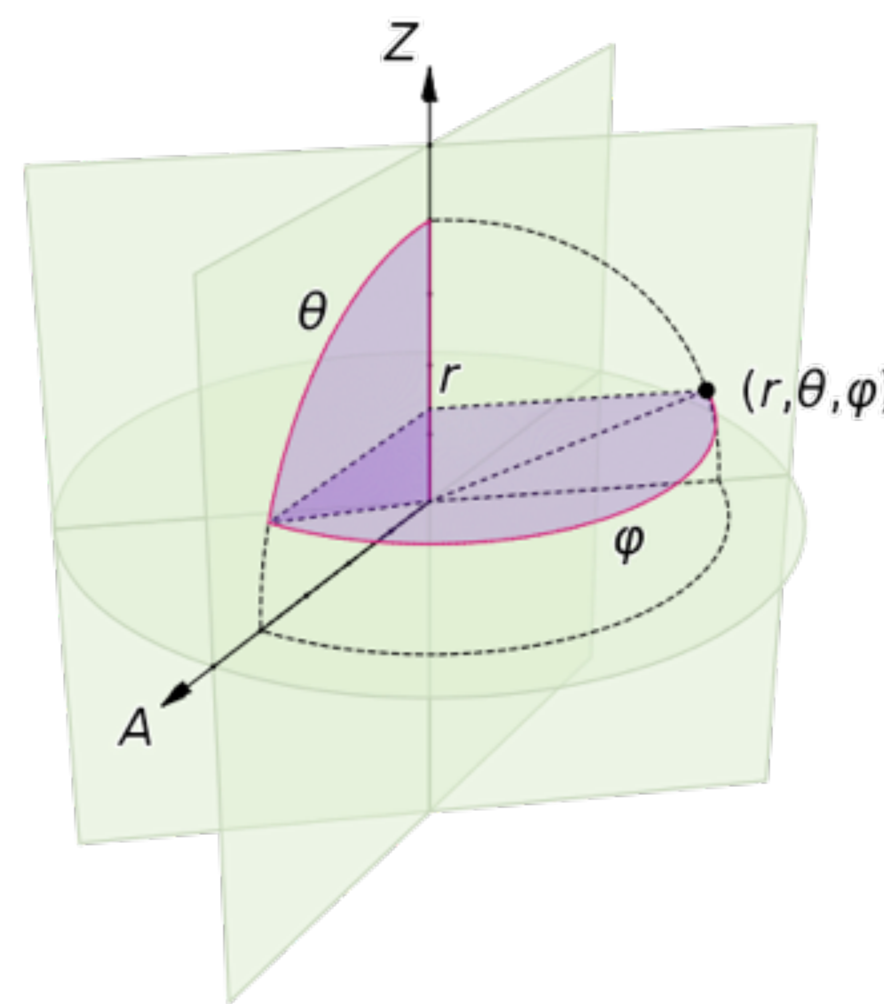
Chapter 1 – Introduction, Motivation, Basics

- About this Class: Organization
- Tutorials
- What is Computer Graphics?
- 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 (non-rectilinear): Many other systems
    - e.g. polar/spherical coordinates: angle plus distance

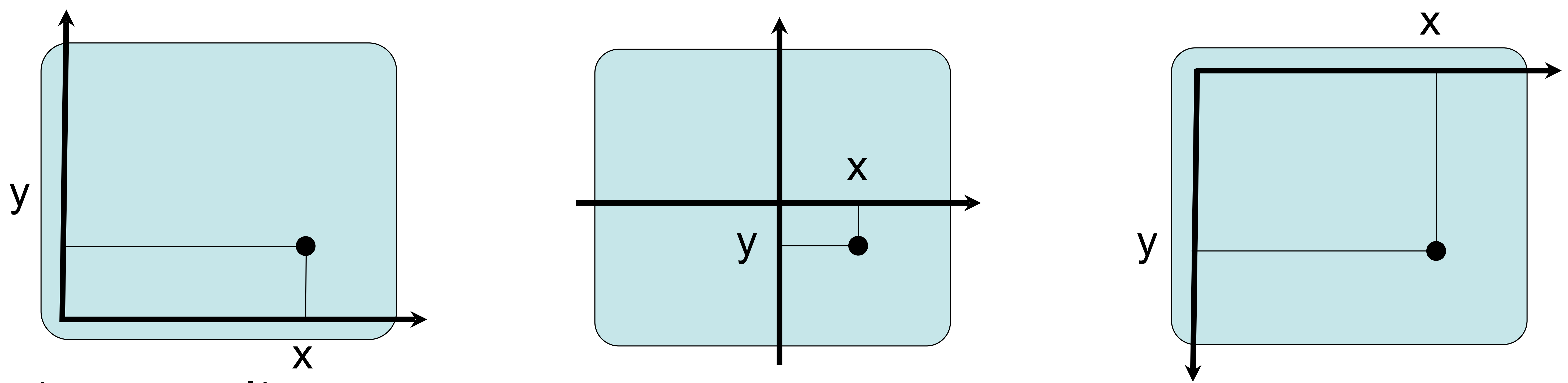




Images: Wikipedia

2D Cartesian Coordinate Reference Frames

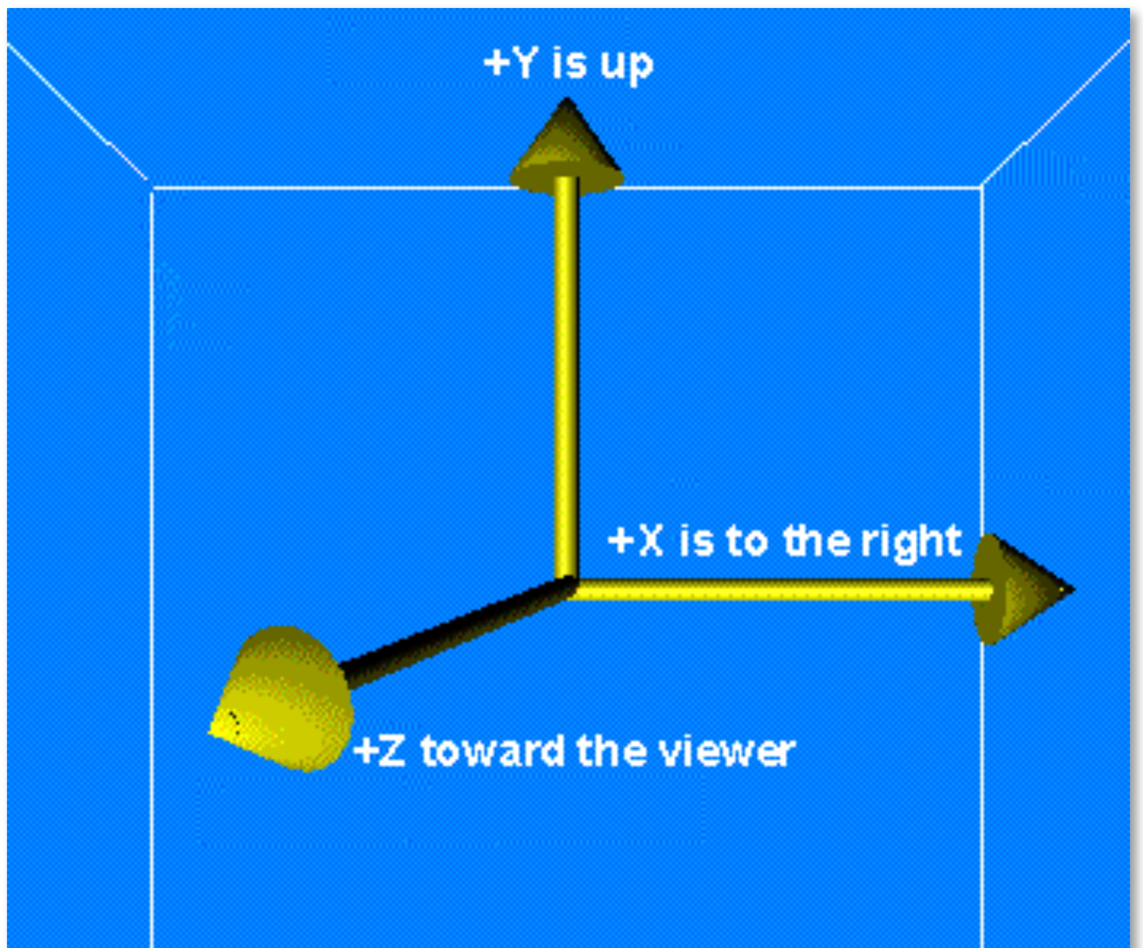
- Device-independent commands of graphics packages:
  - Varying schemata: origin may be in lower-left corner, center, upper-left corner



- Device coordinates
  - Example: Scan lines on cathode ray tubes, printers: origin in upper left corner, y axis points downwards (other devices may have the origin in lower-left corner)
- Normalized device coordinates: Range from 0.0 to 1.0 (real number)
- Physical device coordinates: Pixel addresses of a display (integers)

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



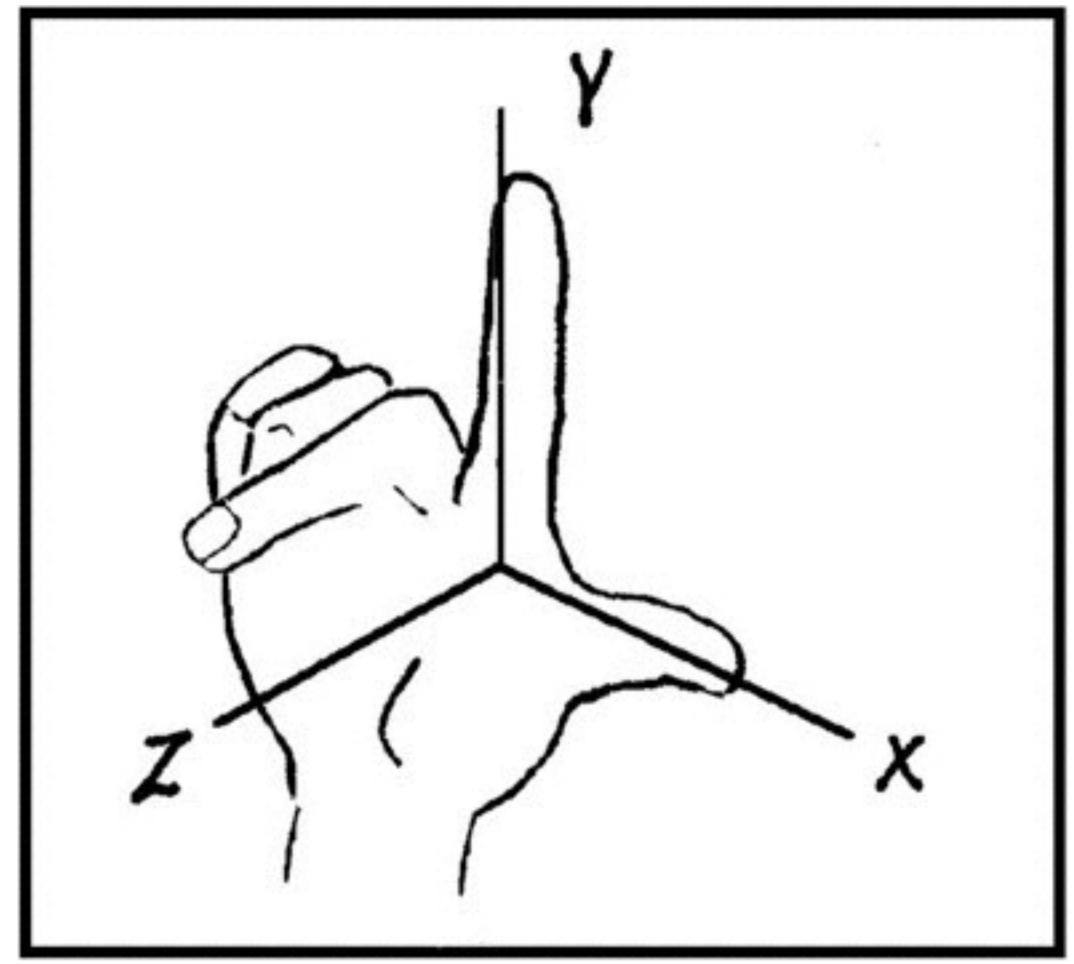


Image sources:
euclidianspace.com,
cornell.edu

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

Points and Vectors

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

$$P = \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix}$$

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

$$\boldsymbol{v} = \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix}$$

$$\boldsymbol{v} = (v_x, v_y, v_z)$$

- Difference between two positions is 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 (i.e., dimension) can be added pointwise

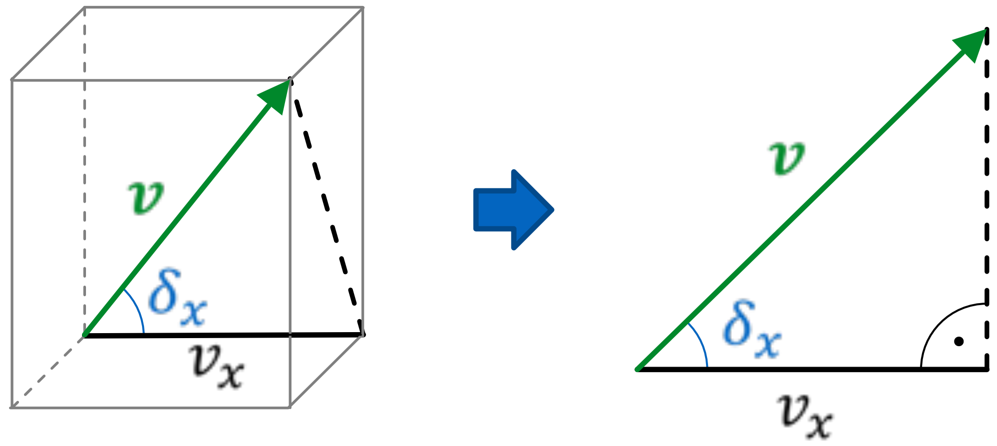
Properties of Vectors

- Magnitude (length)

$$\boldsymbol{v} = (v_x, v_y, v_z) \qquad \|\boldsymbol{v}\| = \sqrt{{v_x}^2 + {v_y}^2 + {v_z}^2}$$

- Direction angles

$$\cos \delta_x = \frac{v_x}{\|\boldsymbol{v}\|} \qquad \cos \delta_y = \frac{v_y}{\|\boldsymbol{v}\|} \qquad \cos \delta_z = \frac{v_z}{\|\boldsymbol{v}\|}$$



Scalar Product (Dot Product)

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

$$\boldsymbol{a} \bullet \boldsymbol{b} = \begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} \bullet \begin{pmatrix} b_x \\ b_y \\ b_z \end{pmatrix} = a_x b_x + a_y b_y + a_z b_z$$

- Application: Computation of angle between two coordinate vectors

$$\boldsymbol{a} \bullet \boldsymbol{b} = \|\boldsymbol{a}\| \cdot \|\boldsymbol{b}\| \cdot \cos\theta$$

- Application: Scalar projection of vector A in direction B

$$\boldsymbol{a}_b = \boldsymbol{a} \bullet \frac{\boldsymbol{b}}{\|\boldsymbol{b}\|} = \|\boldsymbol{a}\| \cdot \cos\theta$$

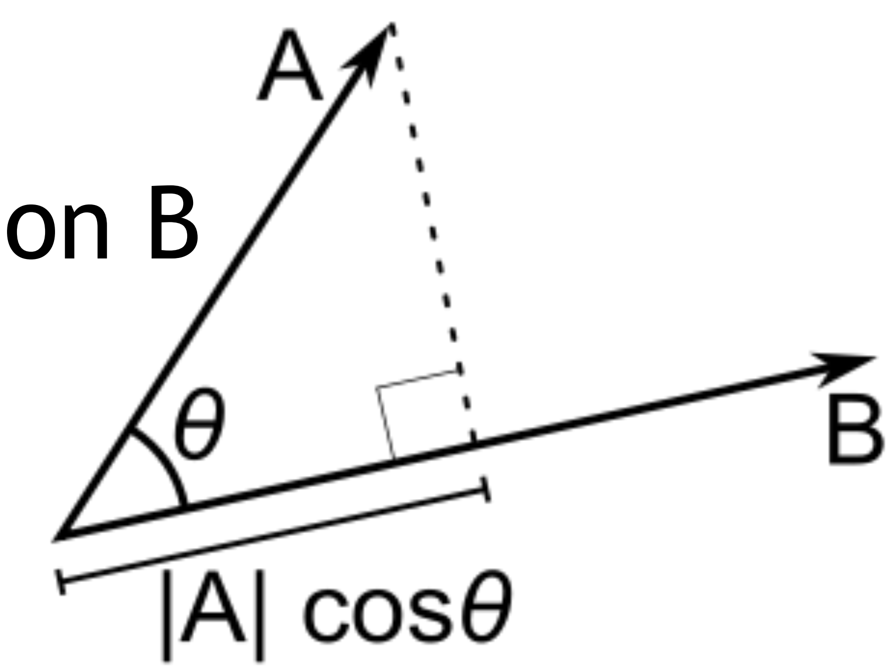


Image sources: Wikipedia

Cross Product (Vector Product)

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

$$\boldsymbol{a} \times \boldsymbol{b} = \begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} \times \begin{pmatrix} b_x \\ b_y \\ b_z \end{pmatrix} = \begin{pmatrix} a_y b_z - a_z b_y \\ a_z b_x - a_x b_z \\ a_x b_y - a_y b_x \end{pmatrix}$$

$$\boldsymbol{a}\times\boldsymbol{b} = -(\boldsymbol{b}\times\boldsymbol{a})$$

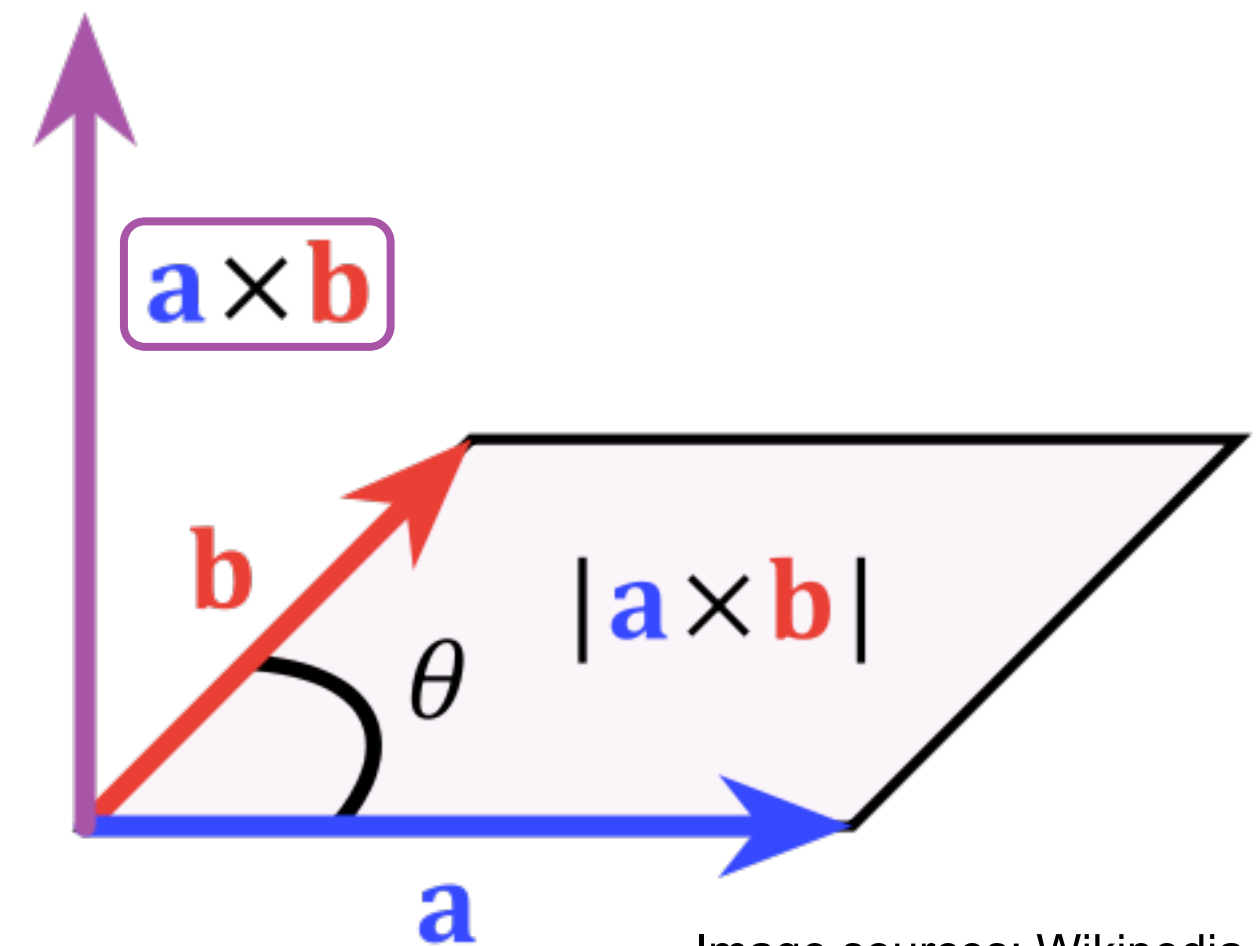
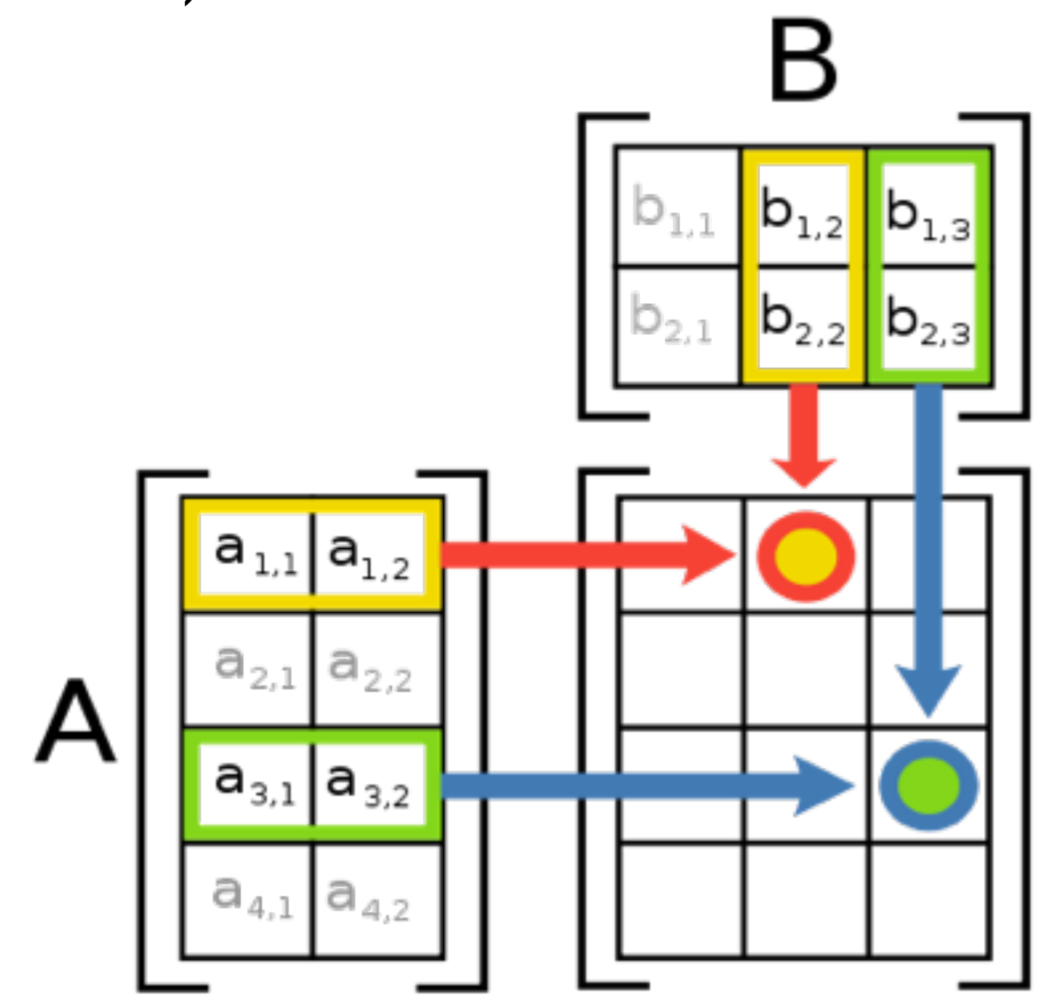


Image sources: Wikipedia

Matrices

- A *matrix* is an (m×n) arrangement of real numbers (m rows, n 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×p)-matrix A multiplied by (p×n)-matrix B gives (m×n)-matrix C

$$C_{i,j} = \sum_{k=1}^{p} A_{i,k} \cdot B_{k,j} \qquad \begin{matrix} 1 \leq i \leq m \\ 1 \leq j \leq n \end{matrix}$$



Multiplying a Matrix and a Vector

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

$$w_j = \sum_{k=1}^{p} A_{i,k} \cdot v_k$$

- If this all sounded difficult or long-forgotten:
  - Dig out your old school books
  - Re-read your Linear Algebra scripts
  - Attending the tutorials and doing the assignments will help!
- There will be more math in the rest of the lecture
- *There will be math in the exam!*

http://jasinski.ukw.edu.pl/images/133b.jpg