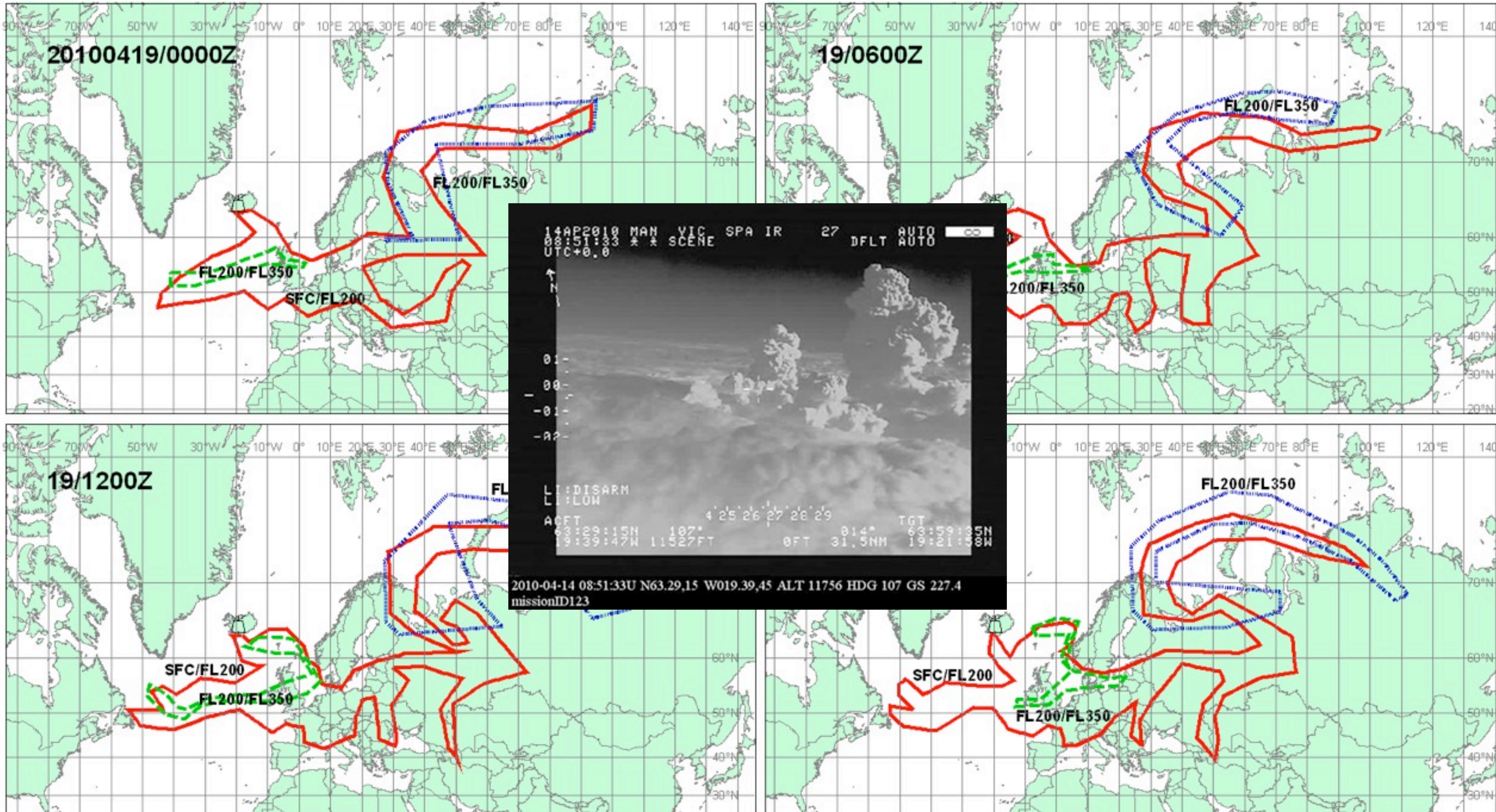
Computer Graphics 1

Chapter 1 (April 22nd, 2010, 2-5pm):

Introduction, coordinate systems, Lin Alg recap

Why I almost didn‘t make it...



sources: http://www.metoffice.gov.uk/aviation/vaac/vaacuk_vag.html, http://sgstb.msn.com/i/66/8A859D72A637F6E25C2AC6DDEB6364.jpg

About this class: Organization

- Bachelor Medieninformatik, 4th semester
- Bachelor Informatik, optional (?!?)
- Diplom Medieninformatik, optional

- Lecture: Andreas Butz, (Sebastian Boring)
  - Thursday, 2-5pm, Theresienstraße, Room B004
  - Run as 3V+2Ü this year, might change to 2V+3Ü in the future
  - change from 2009: no image processing, just 3DCG, but deeper
  - might end a little early sometimes already this year
  - Q: Is 3x 3/4h with pauses OK? Start 14:15h (= 14 Uhr c.t.)

- PDF of the slides: night before class, print out and bring
- Podcast: night after class (if Keynote doesn‘t fail!)

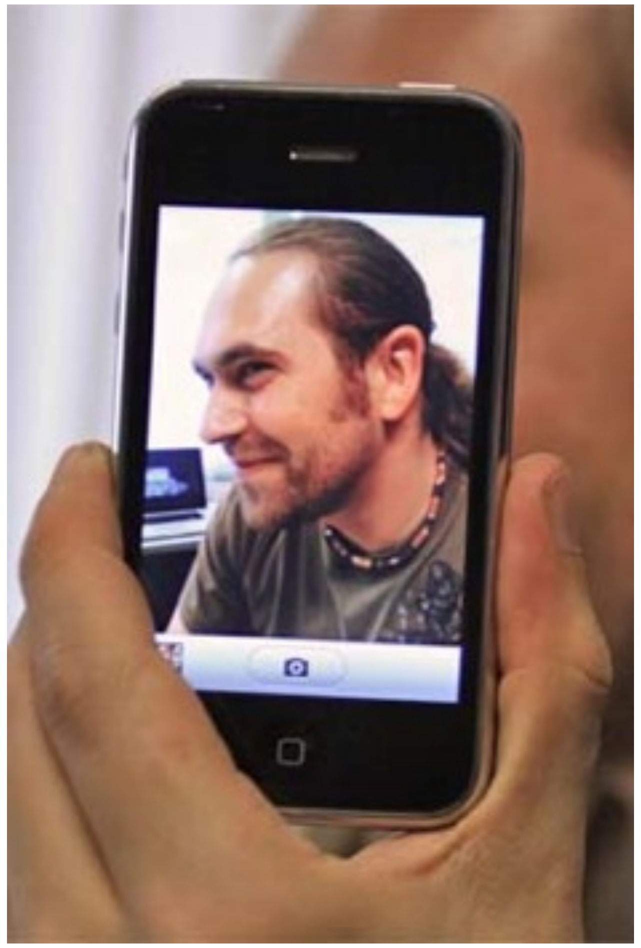


image source: mimuc.de

Übungen "Computergrafik 1"

- Termine:
  - Praktische Anwendung und Ergänzung des Vorlesungsstoffs
  - Erste Übungsstunden in der ersten Semesterwoche (ab Freitag, 23.04.)
  - Alle Übungen: Theresienstr. 39 Raum B134
- Erwerb der Leistungspunkte (6 ECTS) und Benotung:
  - Klausur zu Semesterende (bisherige Planung: 26.07.2010)
- Übungsblätter:
  - Abgabe freiwillig, keine Bedingung für die Teilnahme an der Klausur
  - gute Vorbereitung auf die Klausur
  - Übungsblätter können zusätzlich benutzt werden, um Bonuspunkte für die Klausur zu sammeln (jedes Blatt wird korrigiert, die relative Anzahl der erreichten Punkte wird umgerechnet auf 0 – 15 % der in der Klausur erreichbaren Punkte)
  - Abgabe alleine (die ersten 3 Blätter) oder in festen 4er Gruppen (Einteilung durch die Übungsleiter)
- Plagiarismus:
  - Alle Übungsblätter müssen selbständig gelöst werden
  - Bei eindeutig abgeschriebenen Lösungen (manuelle und automatisierte Prüfung) werden alle Abgaben der Übung (sowohl für den Abschreiber als auch für die Vorlage) mit 0 Punkten gewertet

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

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

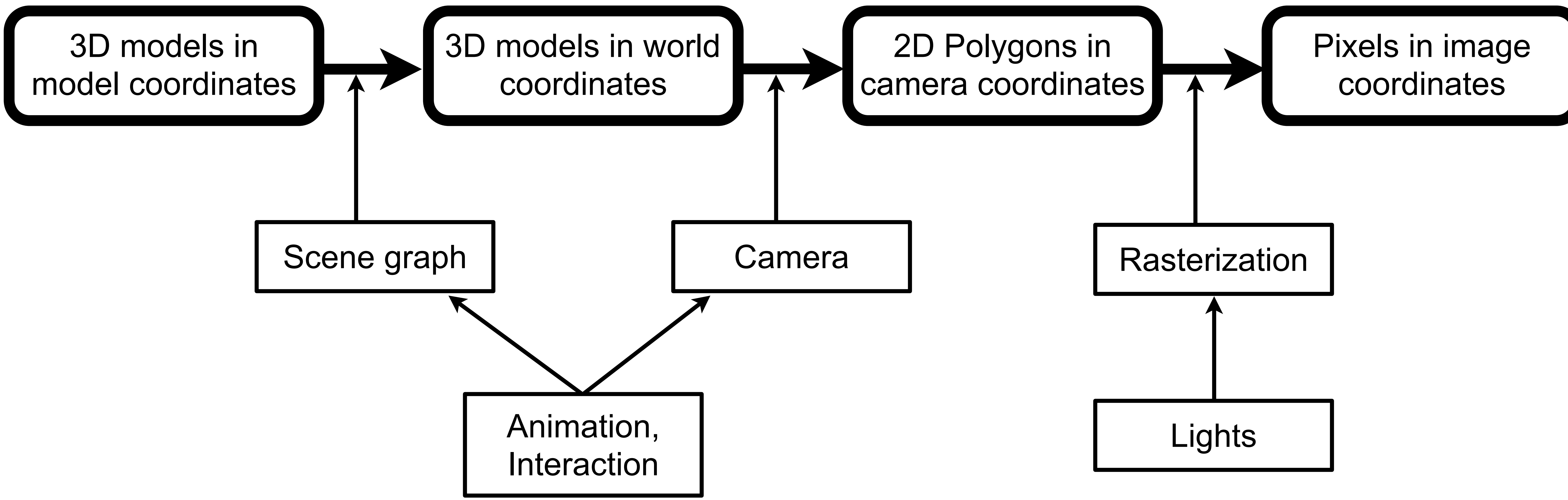
source: http://static.technorati.com/10/01/20/3467/Avatar-movie-Wallpapers.jpg

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

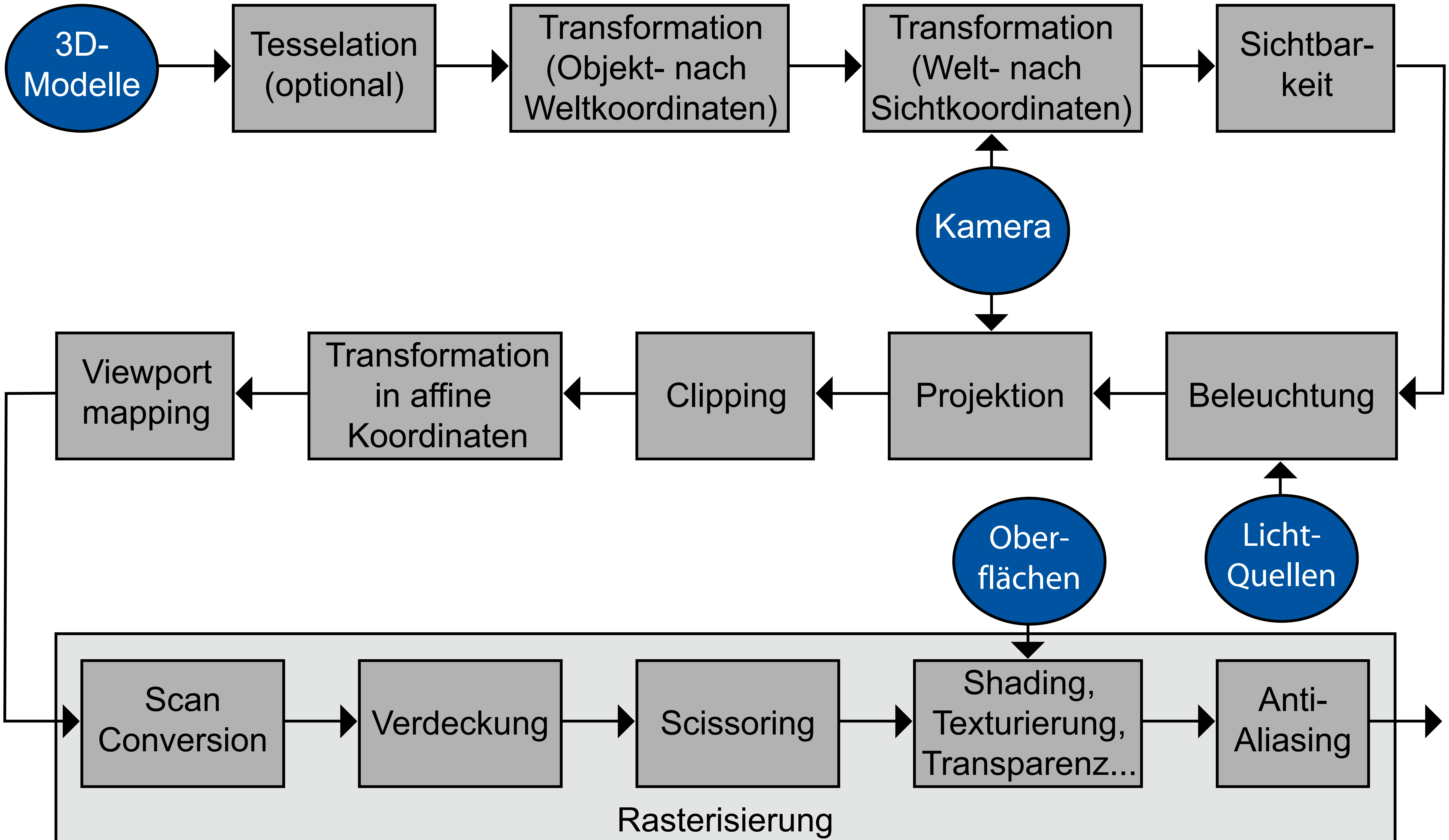
- General problem: current screens are 2D
  - for true 3D perception, we need 2 images for the 2 eyes (stereo)
  - this is technically still difficult (need glasses)
  - research area of volumetric or (auto)stereoscopic displays

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

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



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



Lecture Content

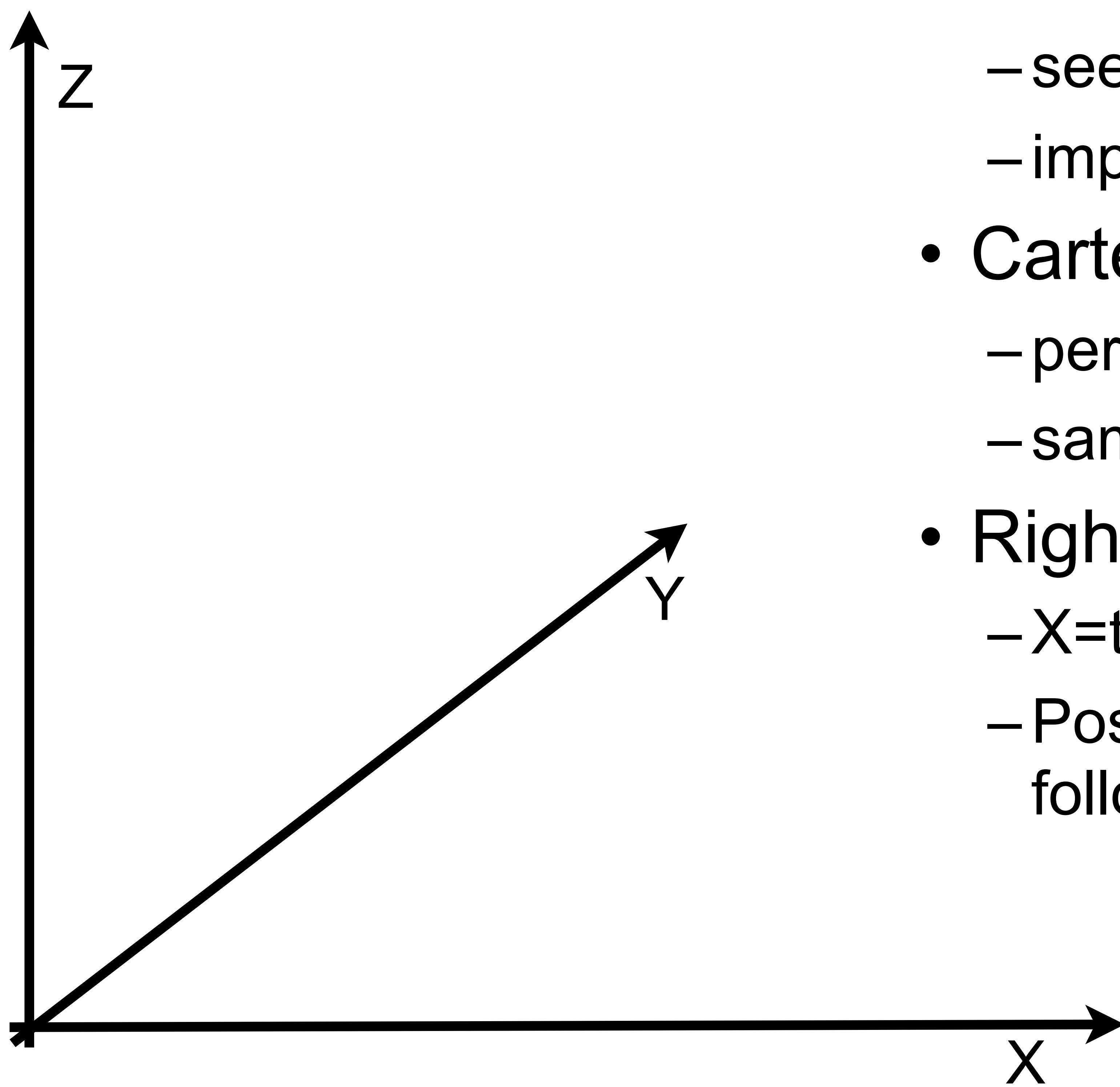
- 11 lecture appointments

- VL1 Intro, Lin Alg
- VL2 3D Modelling
- VL3 Bezier curves and Patches
- VL4 Scene graphs
- VL5 Camera, culling, Z-Buffer
- VL6 Light, Phong Model, Shadows
- VL7 Surfaces, Materials, Maps
- VL8 Shading, Rendering
- VL9 Animation
- VL10 Interaction
- VL11 Time Buffer

- 13 exercise appointments

- Ü1 Lin Alg
- Ü2 C + QT
- Ü3 C++ + Debugging
- Ü4 OpenGL
- Ü5 Meshes
- Ü6 More light
- Ü7 Materials, Textures
- Ü8 Non Photorealistic Rendering
- Ü9 Animation, Paths
- Ü10 Picking

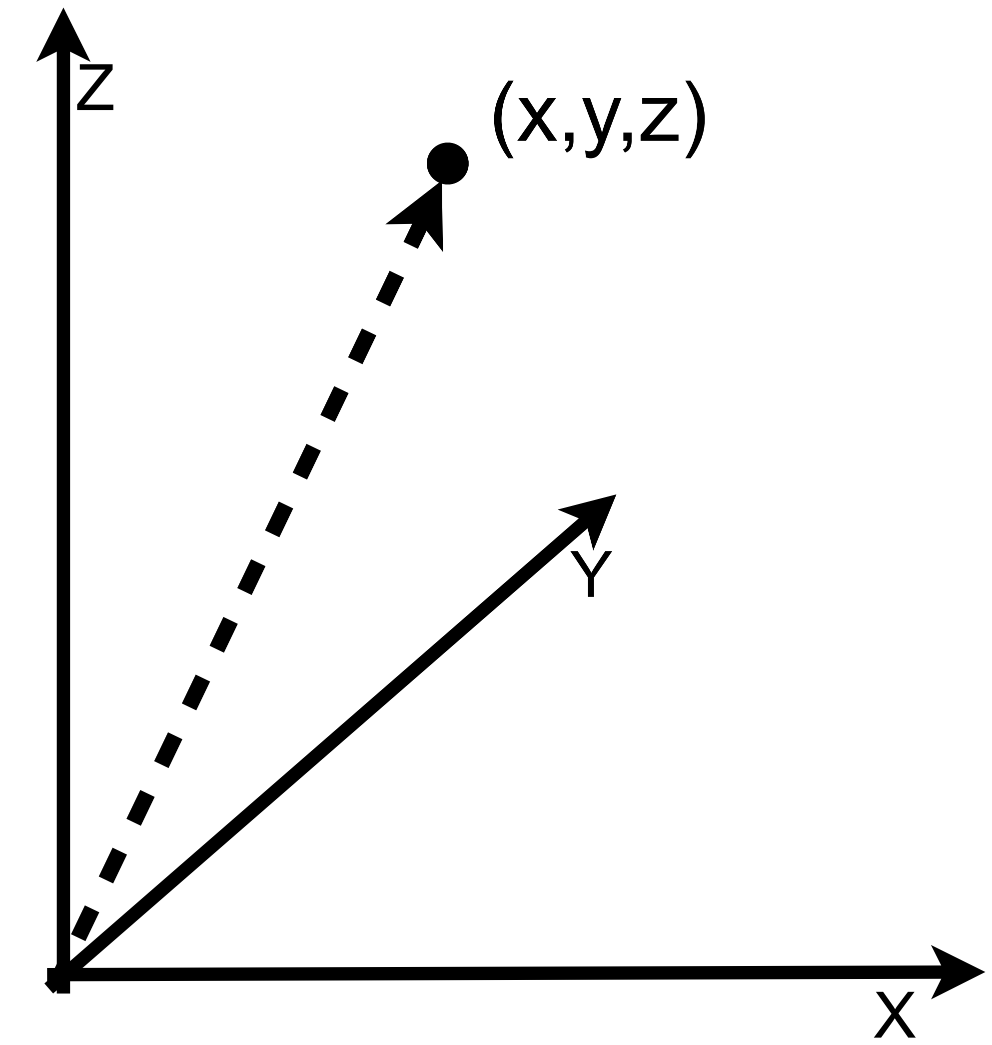
Coordinate systems

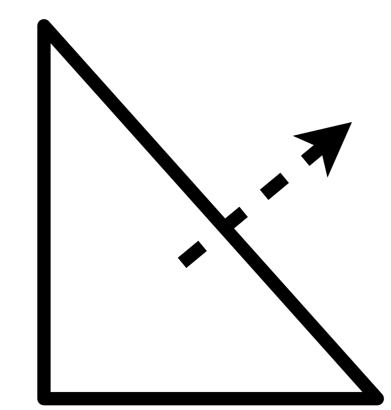


- 3D vector space $\mathbb{R}^3$ in limited precision
  - computation errors accumulate!
  - see numerical mathematics
  - implementations optimized, but be aware!
- Cartesian coordinate system
  - perpendicular axes
  - same linear scales on all axes
- Right-handed coordinate system
  - X=thumb, Y=index, Z=middle finger of the RH
  - Positive rotation follows fingers if positive axis follows thumb

Points, Vectors, Lines, Polygons, Normals

- Every point has a unique description (x,y,z)
- Vector (x,y,z) from origin (0,0,0) to point (x,y,z)
  - in CG we don‘t care whether row or column vectors!
- Line from point (x_1,y_1,z_1) to (x_2,y_2,z_2)
- Polygon = sequence of lines in which
  - the end of one line is the start of the next
- Closed polygon: last point = first point
- Planar polygon: all points within one plane
- Normal vector of a plane is perpendicular to it
  - undefined for non-planar polygons
- Triangle = Polygon with 3 points
  - always planar: normal is always defined 8-)
- Point normal = normal of the polygon at the point





Affine Transformations

- straight lines will remain straight lines
- distances and angles might change

- basic transformations are translation, rotation, scaling and shearing

- all combinations of these are affine transformations again
- combination is associative, but not commutative
- inverse transformation normally exists
  - counterexamples??

- neutral transformation is the identity transformation

Translation

$$\begin{pmatrix} x_{neu} \\ y_{neu} \\ z_{neu} \end{pmatrix} = \begin{pmatrix} x_{alt} \\ y_{alt} \\ z_{alt} \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} = \begin{pmatrix} x_{alt} + t_x \\ y_{alt} + t_y \\ z_{alt} + t_z \end{pmatrix}$$

- add a vector t
- Hmm, not much to say here ;-)
- Inverse operation?
- Neutral operation?

Scaling

$$\begin{pmatrix} x_{neu} \\ y_{neu} \\ z_{neu} \end{pmatrix} = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{pmatrix} \begin{pmatrix} x_{alt} \\ y_{alt} \\ z_{alt} \end{pmatrix} = \begin{pmatrix} s_x x_{alt} \\ s_y y_{alt} \\ s_z z_{alt} \end{pmatrix}$$

- Uniform scaling: $s_x = s_y = s_z$
- Non-uniform scaling: not equal
- Mirroring: $s_x * s_y * s_z < 0$
  - example: $s_x = s_y = 1, s_z = -1$
  - what happens to the handedness of a coordinate system?
  - what happens to all surface normals?

- This scaling is always about the origin.
- How about when we want to scale about an arbitrary point?
  –
  –
  –

Rotation about X

$$\begin{pmatrix} x_{neu} \\ y_{neu} \\ z_{neu} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{pmatrix} \begin{pmatrix} x_{alt} \\ y_{alt} \\ z_{alt} \end{pmatrix} = \begin{pmatrix} x_{alt} \\ \cos\alpha y_{alt} - \sin\alpha z_{alt} \\ \sin\alpha y_{alt} + \cos\alpha z_{alt} \end{pmatrix}$$

- x stays constant
- y and z mix
- special cases: 90, 180, 270 degrees, what happens >360 degrees?
- How can we rotate about arbitrary axes?
- Lemma: Any rotation can be composed from 3 basic rotations
  - no proof here (see math lectures)
- Rotation about an axis which doesn‘t go through the origin??
  -
- Rotations can also be described by Quaternions (maybe later ;-)
- ...or how else?? Advantages, Disadvatages?
  -
  -
  -

Elementary rotations

- Combine to express arbitrary rotation
- This is not always intuitive
- Order matters (a lot!)
- Likely source of bugs!

$$\begin{pmatrix} x_{neu} \\ y_{neu} \\ z_{neu} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{pmatrix} \begin{pmatrix} x_{alt} \\ y_{alt} \\ z_{alt} \end{pmatrix} = \begin{pmatrix} x_{alt} \\ \cos\alpha y_{alt} - \sin\alpha z_{alt} \\ \sin\alpha y_{alt} + \cos\alpha z_{alt} \end{pmatrix}$$

$$\begin{pmatrix} x_{neu} \\ y_{neu} \\ z_{neu} \end{pmatrix} = \begin{pmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{pmatrix} \begin{pmatrix} x_{alt} \\ y_{alt} \\ z_{alt} \end{pmatrix} = \begin{pmatrix} \cos\beta x_{alt} + \sin\beta z_{alt} \\ y_{alt} \\ \cos\beta z_{alt} - \sin\beta x_{alt} \end{pmatrix}$$

$$\begin{pmatrix} x_{neu} \\ y_{neu} \\ z_{neu} \end{pmatrix} = \begin{pmatrix} \cos\chi & -\sin\chi & 0 \\ \sin\chi & \cos\chi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{alt} \\ y_{alt} \\ z_{alt} \end{pmatrix} = \begin{pmatrix} \cos\chi x_{alt} - \sin\chi y_{alt} \\ \sin\chi x_{alt} + \cos\chi y_{alt} \\ z_{alt} \end{pmatrix}$$

Shearing along X

$$\begin{pmatrix} x_{neu} \\ y_{neu} \\ z_{neu} \end{pmatrix} = \begin{pmatrix} 1 & m & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{alt} \\ y_{alt} \\ z_{alt} \end{pmatrix} = \begin{pmatrix} x_{alt} + m y_{alt} \\ y_{alt} \\ z_{alt} \end{pmatrix}$$

- z and y remain
- x_{neu} depends on y now
- areas and volumes remain the same
- angles change
- analog formulas for Y and Z
- shearing along an arbitrary axis?
  –
  –
  –

Combining Multiple Transformations

- Rotation, scaling and shearing are expressed as matrices
  - associative, hence can all be combined into one matrix
  - many of these operations can also be combined into one matrix
- Translation is expressed by adding a vector
  - adding vectors is also associative
  - many translations can be combined into a single vector
- Combination of Translation with other operations?
  - series of matrices and vectors, no way to combine al in one
  - ...except if there was a matrix to express translation ?!?
  –
  –
  –

Projection 3D to 2D

$$\begin{pmatrix} x_{neu} \\ y_{neu} \\ z_{neu} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_{alt} \\ y_{alt} \\ z_{alt} \end{pmatrix} = \begin{pmatrix} x_{alt} \\ y_{alt} \\ 0 \end{pmatrix}$$

- assumption: project onto Z plane
- x and y remain the same, z=0
- hence: not reversible!!!
- This is the orthographic (parallel) projection
- what does this mean visually?
  –
  –
  –
- Any ideas for perspective projection?
  – will come in camera chapter of this class
  –
  –
  –
  –

Homogeneous Coordinates

- Trick to express translation as a matrix
- Add one dimension
  - for vectors, add a 1 as the 4th component
  - for matrices, add 0 and a 1 on the diagonal
- Matrices for rotation, scaling and shearing are the same in the upper left corner
- Translation matrix contains the translation vector in the last column
- Now, all affine transformations can be expressed as matrices and combined
- How about computational cost?
  –
  –
  –

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \Rightarrow \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} m_{1,1} & m_{1,2} & m_{1,3} \\ m_{2,1} & m_{2,2} & m_{2,3} \\ m_{3,1} & m_{3,2} & m_{3,3} \end{pmatrix} \Rightarrow \begin{pmatrix} m_{1,1} & m_{1,2} & m_{1,3} & 0 \\ m_{2,1} & m_{2,2} & m_{2,3} & 0 \\ m_{3,1} & m_{3,2} & m_{3,3} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

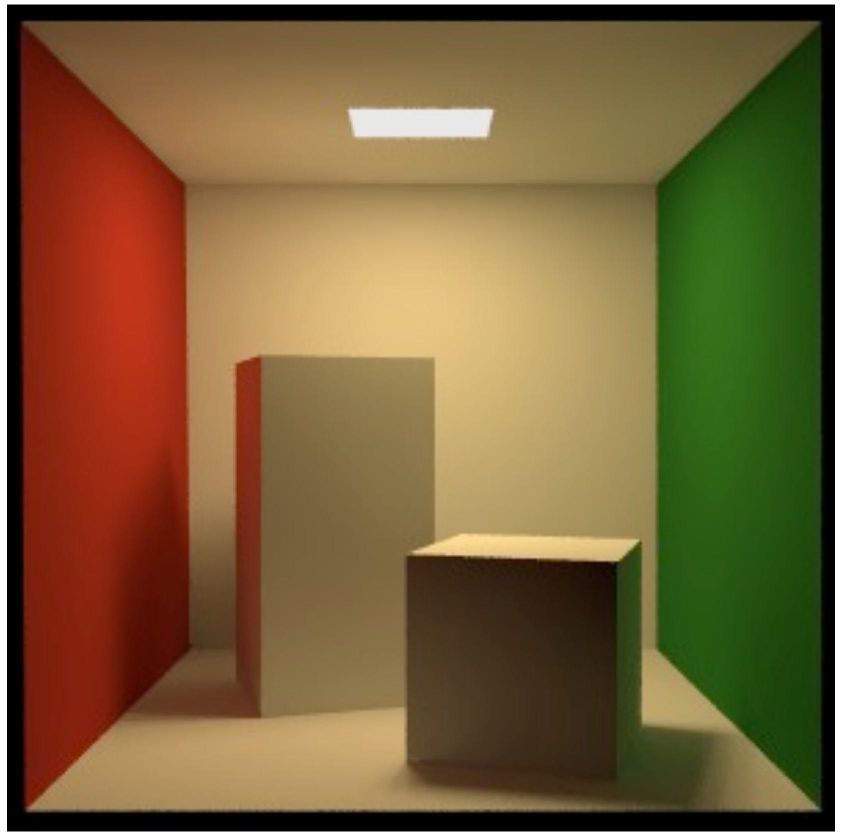
$$\begin{pmatrix} x_{neu} \\ y_{neu} \\ z_{neu} \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{alt} \\ y_{alt} \\ z_{alt} \\ 1 \end{pmatrix} = \begin{pmatrix} x_{alt} + t_x \\ y_{alt} + t_y \\ z_{alt} + t_z \\ 1 \end{pmatrix}$$

About Teapots and Bunnies

- Origin: Martin Newell (1975), University of Utah (hence „Utah teapot“)
- Purpose: show how a complex 3D object can be modeled nicely with only a few Bezier patches
- Is a primitive object in some 3D packages
  - e.g., GLUT
- Original pot was scaled along Z
- Similar stories for Stanford Bunny and Cornell Box (will see later)

http://de.academic.ru/pictures/dewiki/111/original_utah_teapot.jpg

http://en.wikipedia.org/wiki/File:Stanford_Bunny.png



http://upload.wikimedia.org/wikipedia/commons/2/24/Cornell_box.png

http://commons.wikimedia.org/wiki/File:Utah_teapot_simple_2.png

Free Software

- These are only two examples I have recently tried!
  - many other out there, list never complete!

- Blender 3D modelling and rendering software
  - http://www.blender.org/
  - really powerful tool
  - has been used for movies
  - UI can really be confusing

- Google Sketchup
  - http://sketchup.google.com/
  - simplicity is a priority in the UI
  - many models available

Literature Recommendations and links

- Malaka, Butz, Hussmann: Medieninformatik, Pearson Studium 2009
  - Kapitel 8: 3D-Grafik, später noch Kapitel 7: 2D-Grafik
- Bungartz, H. et al.: Einführung in die Computergraphik, 2. Auflage, Vieweg, 2002
- Foley, Van Dam, Feiner: Computer Graphics – Principles and Practice, 2nd edition, Addison-Wesley, 1996
- Watt, A. et al.: Advanced Animation and Rendering Techniques.: Theory and Practice, Addison Wesley, 1992
- WEB3D consortium: Open Standards for Real-Time 3D Communication - http://www.web3d.org/