Computer Graphics

Introduction

Prof. Dr. Ulrich Schwanecke

RheinMain University of Applied Sciences

Computer Vision & Mixed Reality Group

How to use the HTML slides

- Use the keys **left**/**right** for navigating through the slides.
- Click icon ☰ (top left) to open the navigation menu.
- Press **f**/**ESC** to enter/leave fullscreen mode.
- **Double-click** an item (e.g. an image) to zoom in/out.
- If the bottom boundary flashes on slide change, something was written on the virtual whiteboard.
  - **Scroll down** to see it.

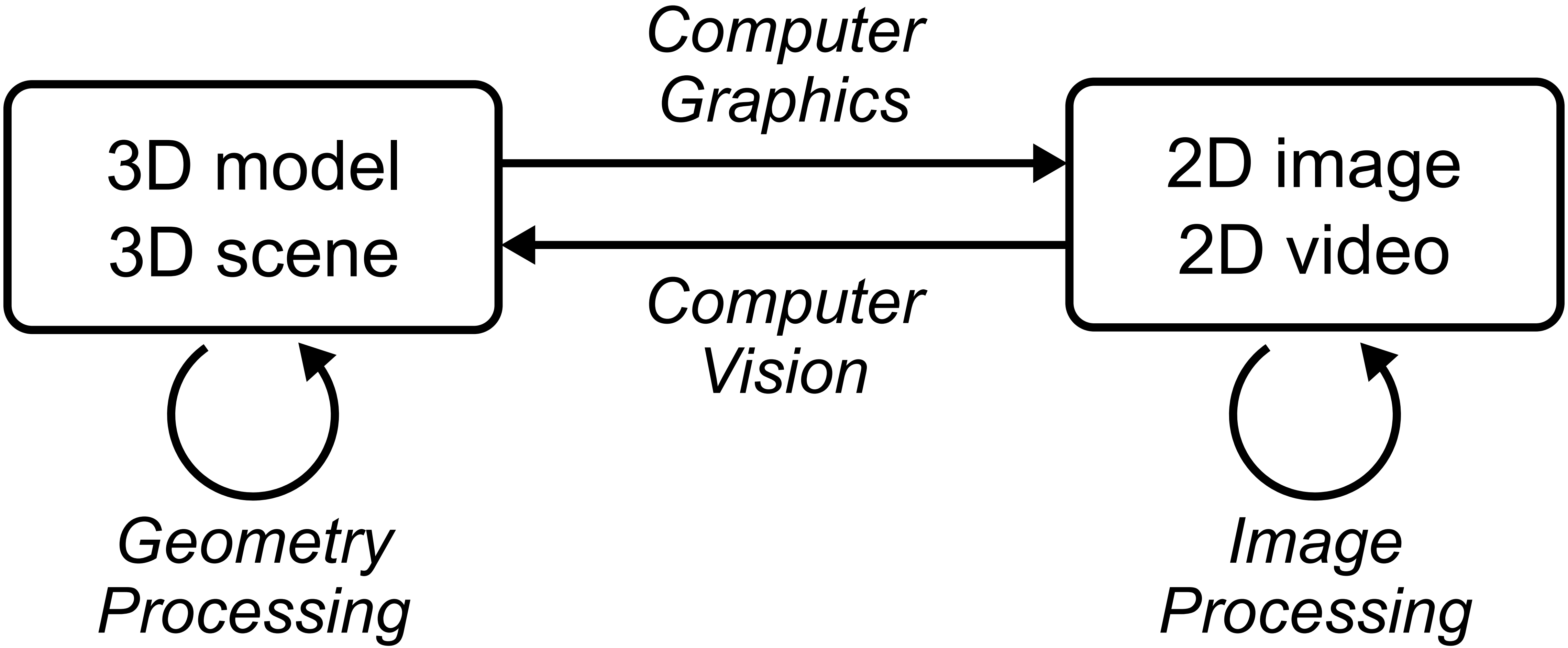
Acknowledgments

Many thanks to Mario Botsch!

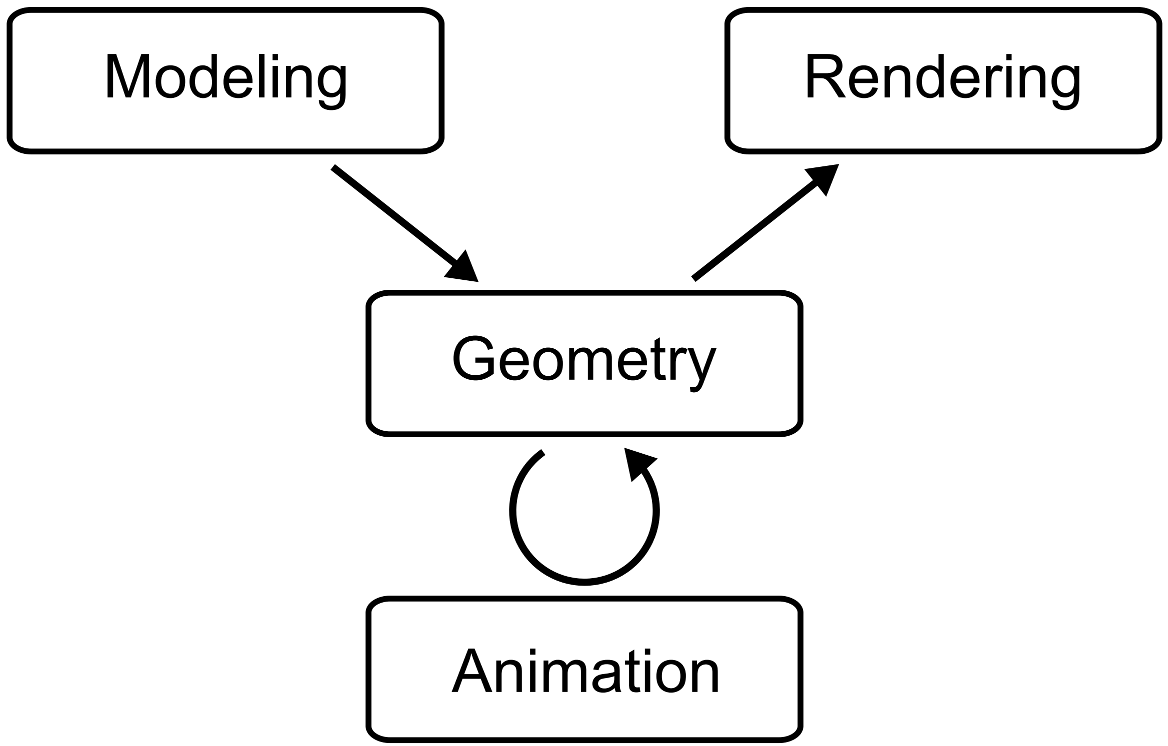
Prof. Mario Botsch,
TU Dortmund

About Computer Graphics

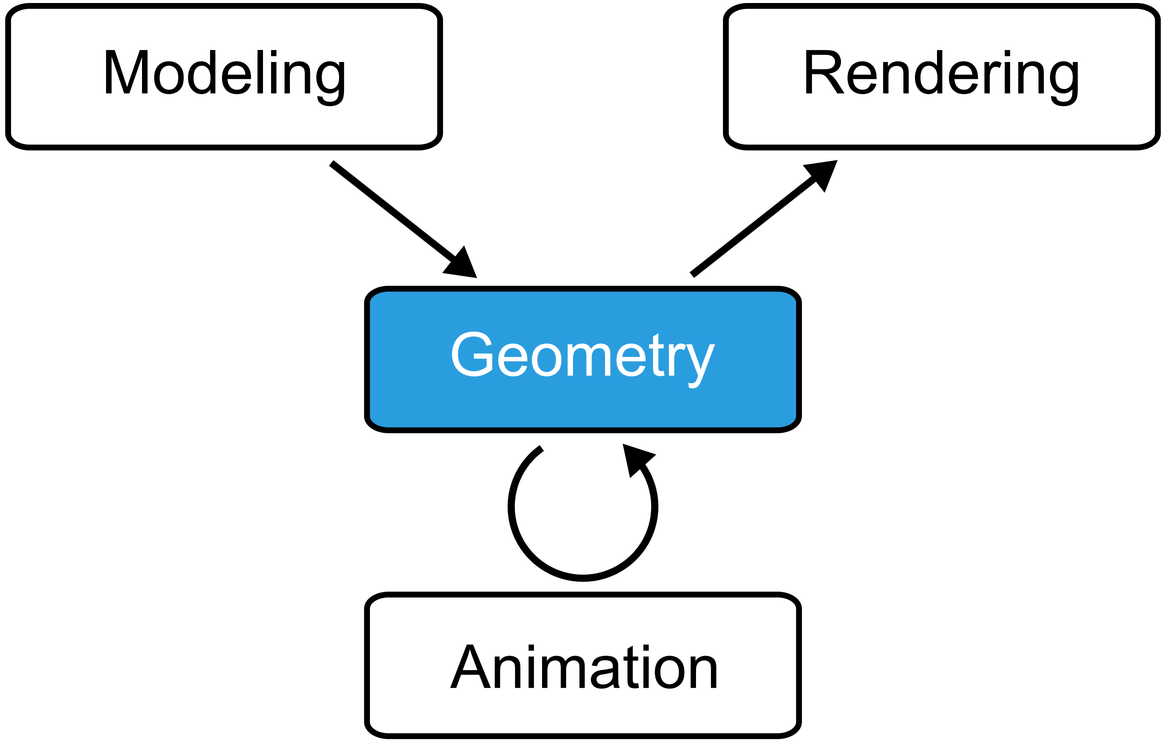
Computer Graphics vs. Computer Vision


Computer
Graphics
3D model
3D scene
2D image
2D video
Computer
Vision
Geometry
Processing
Image
Processing

Computer Graphics & Geometry Processing

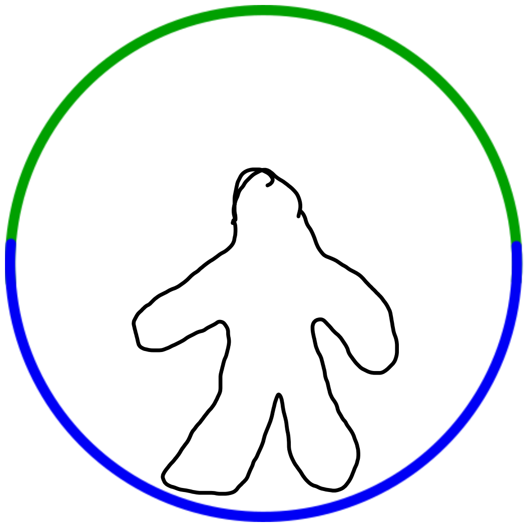


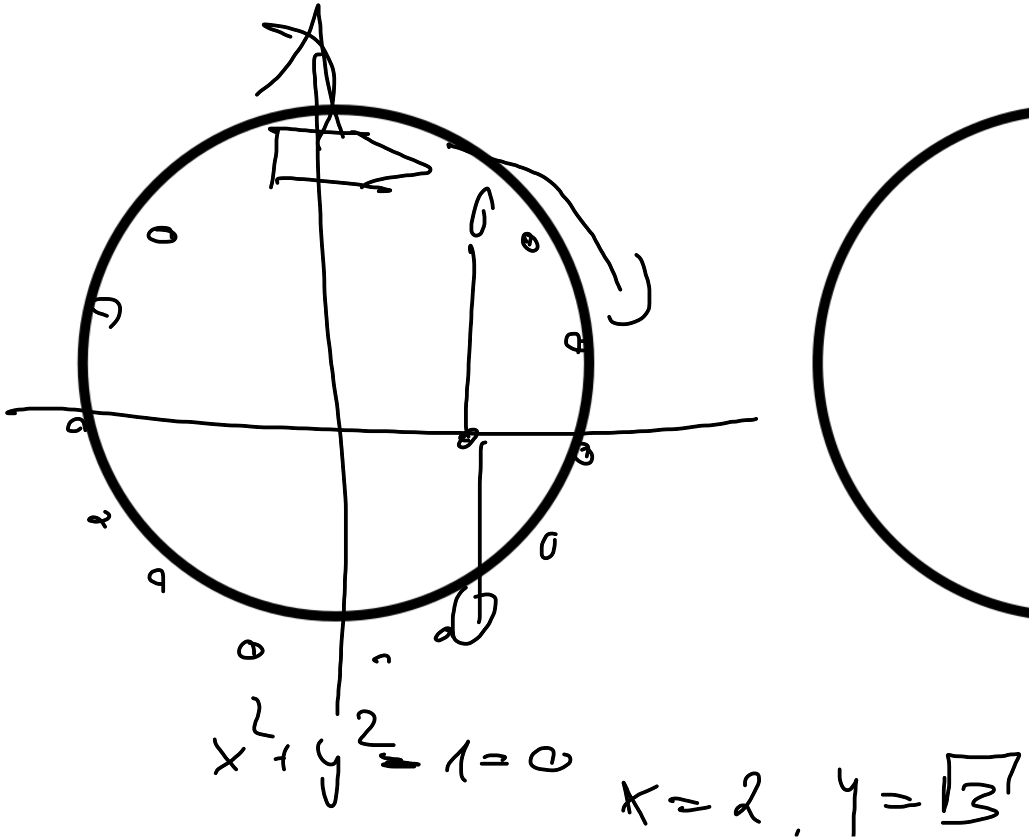
Computer Graphics & Geometry Processing

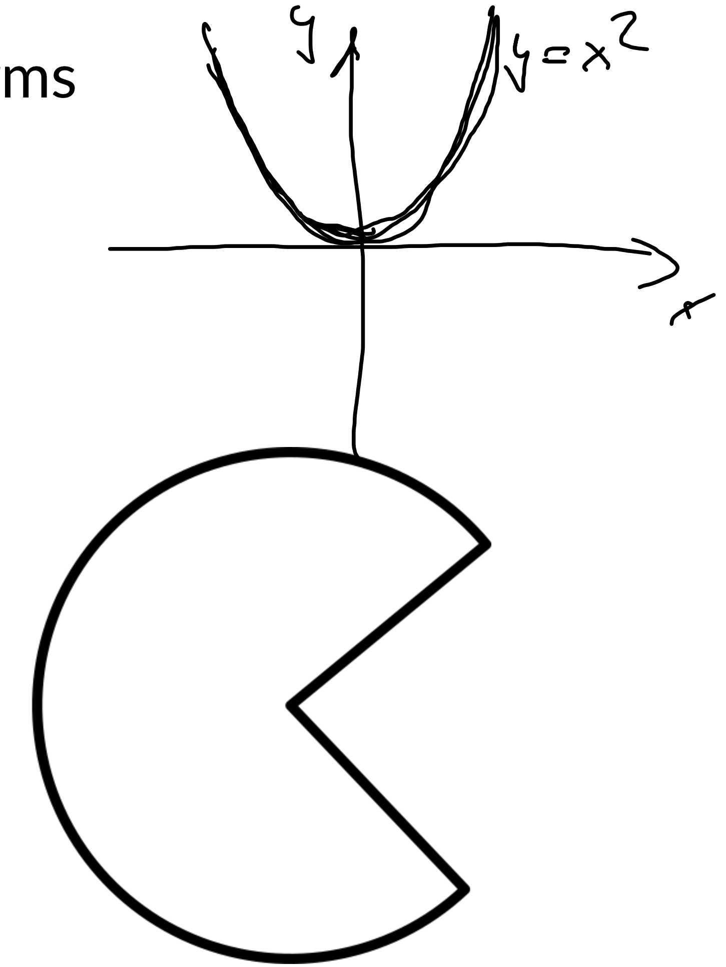


Representation of Objects

- Objects can be represented mathematically in different forms
  - Explicit: $y = \pm\sqrt{r^2 - x^2}$
  - Implicit: $x^2 + y^2 - r^2 = 0$
  - Parametric: $(r \cdot \cos(t), r \cdot \sin(t))^\top, t \in [0, 2\pi)$

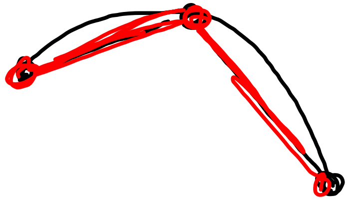


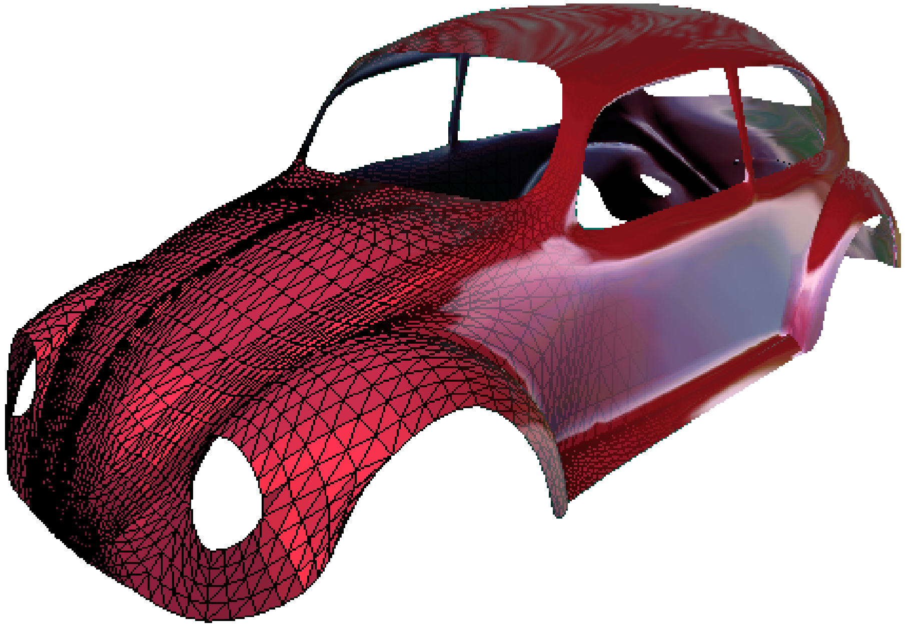




Triangle Meshes

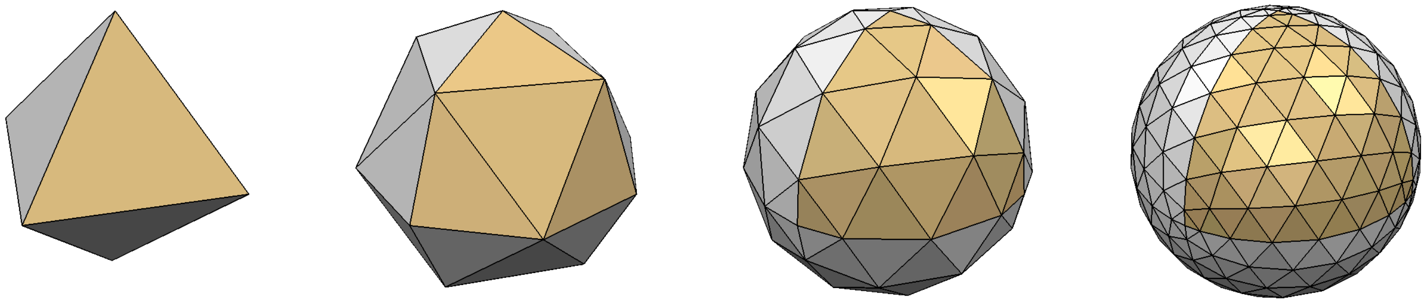
- Triangle meshes can represent arbitrary surfaces





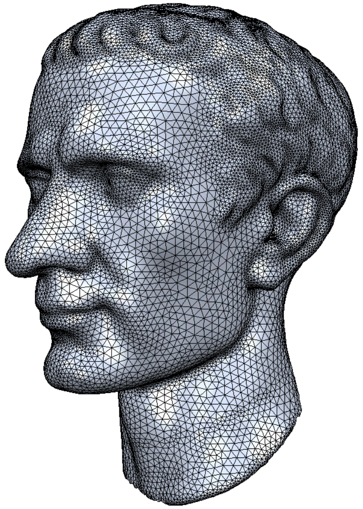
Triangle Meshes

- Triangle meshes can represent arbitrary surfaces
- Approximation error inversely proportional to #triangles

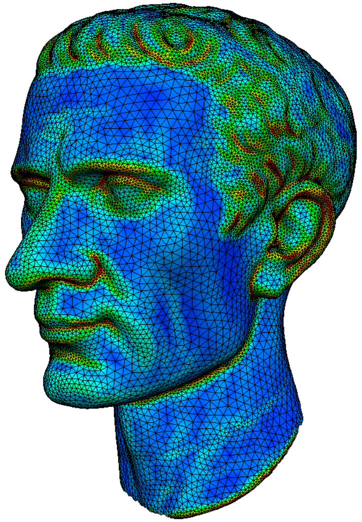


Triangle Meshes

- Triangle meshes can represent arbitrary surfaces
- Approximation error inversely proportional to #triangles
- Adaptive tessellation can adapt to surface curvature



adaptive meshing

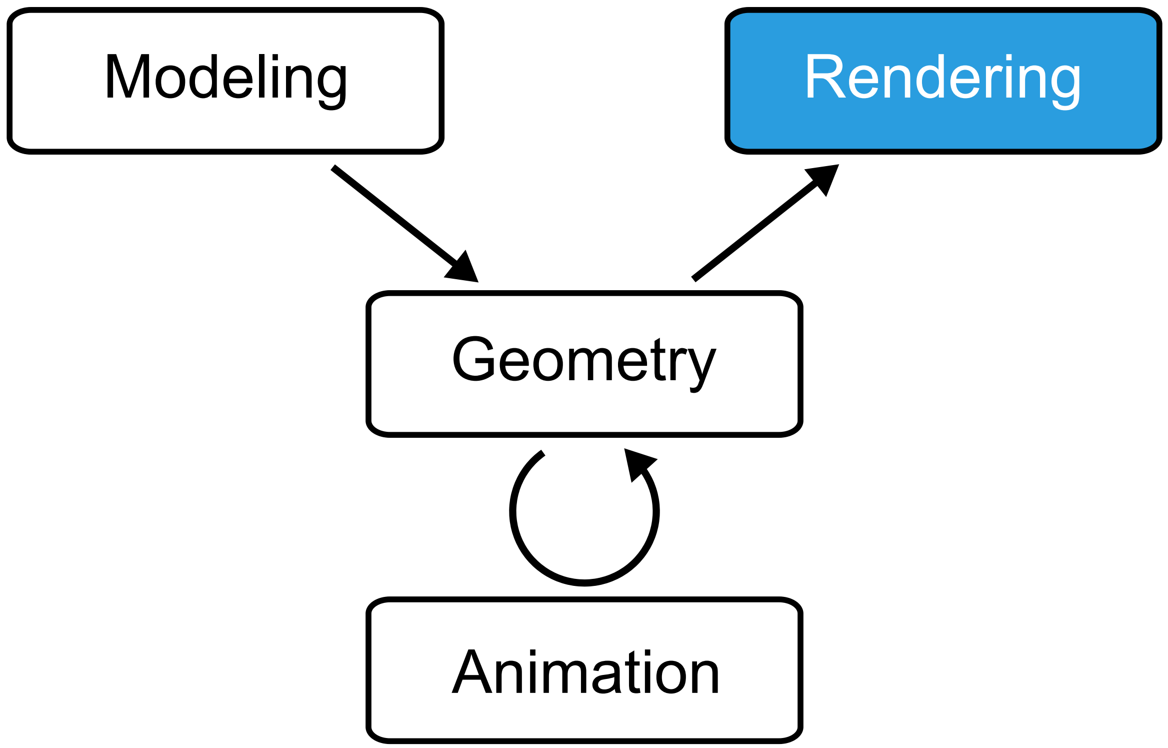


curvature visualization

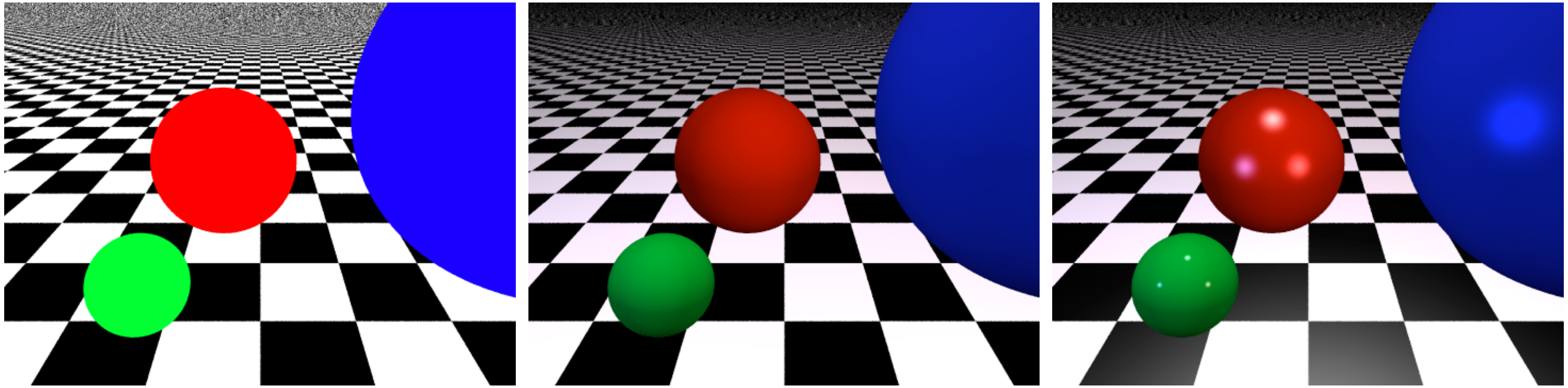
Triangle Meshes

- Triangle meshes can represent arbitrary surfaces
- Approximation error inversely proportional to #triangles
- Adaptive tessellation can adapt to surface curvature
- Simple primitives can efficiently be processed by CPU/GPU

Computer Graphics & Geometry Processing

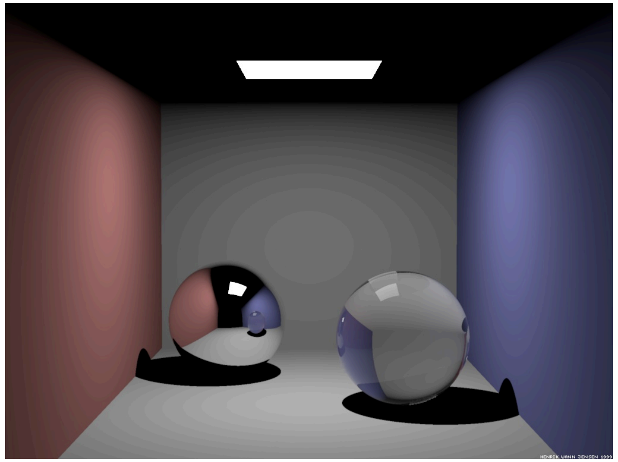


Realistic Lighting Computations

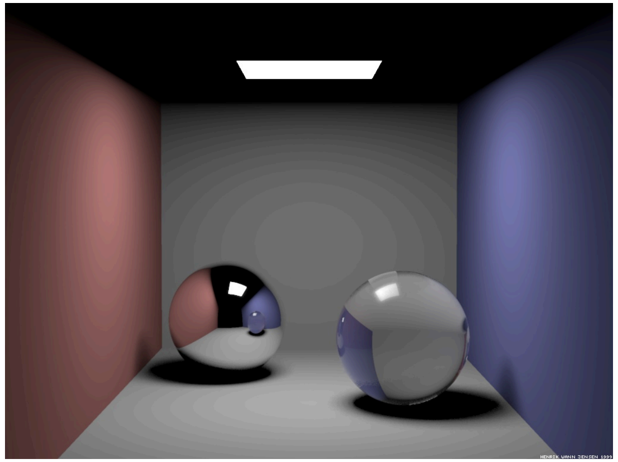


color only *+diffuse* *+specular*

Realistic Lighting Computations



shadows + reflection + refraction



+soft shadows

Realistic Rendering for Movie Industry

Lord of the Rings

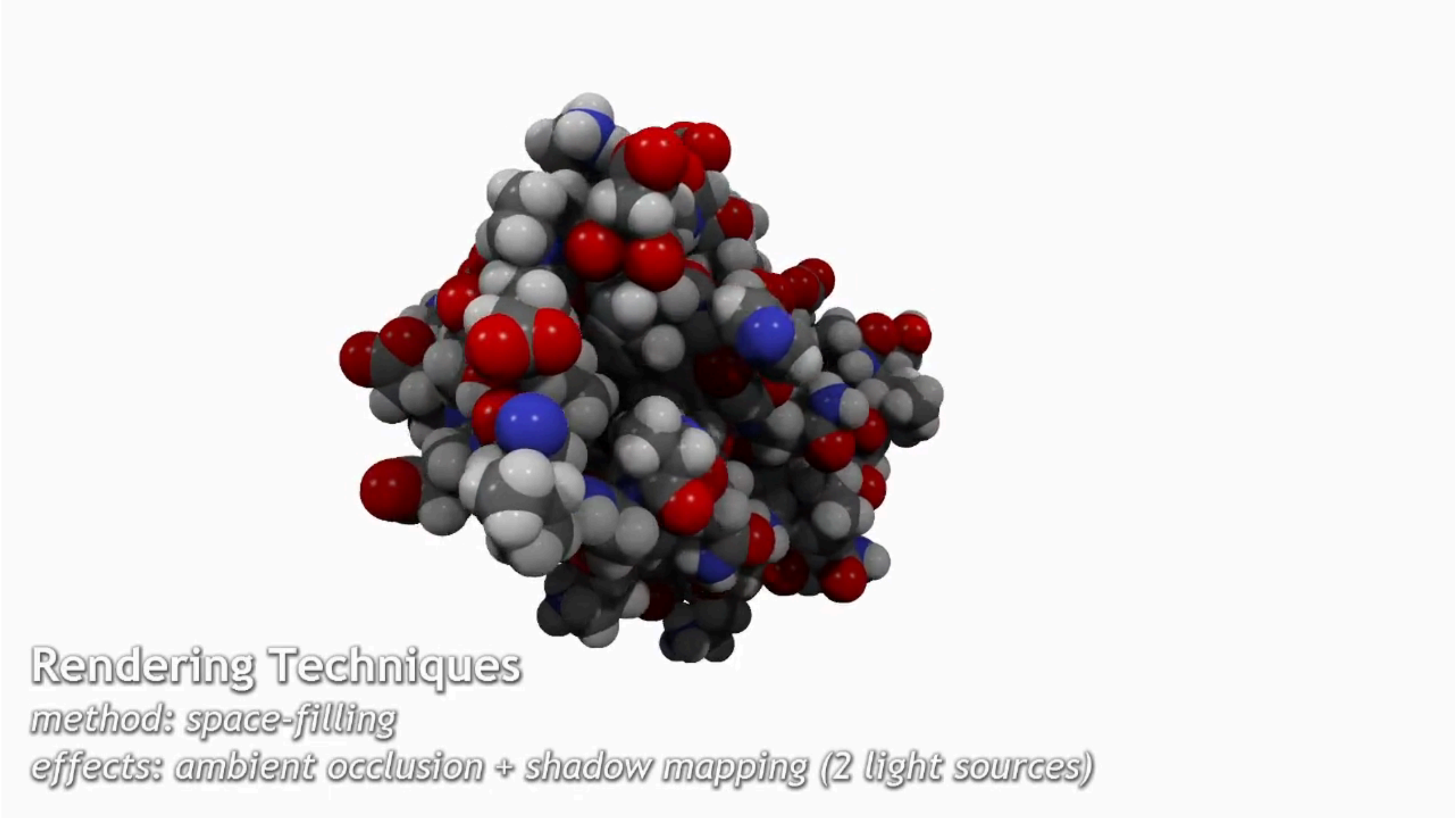
Alita Battle Angel

Realtime Rendering for Computer Games

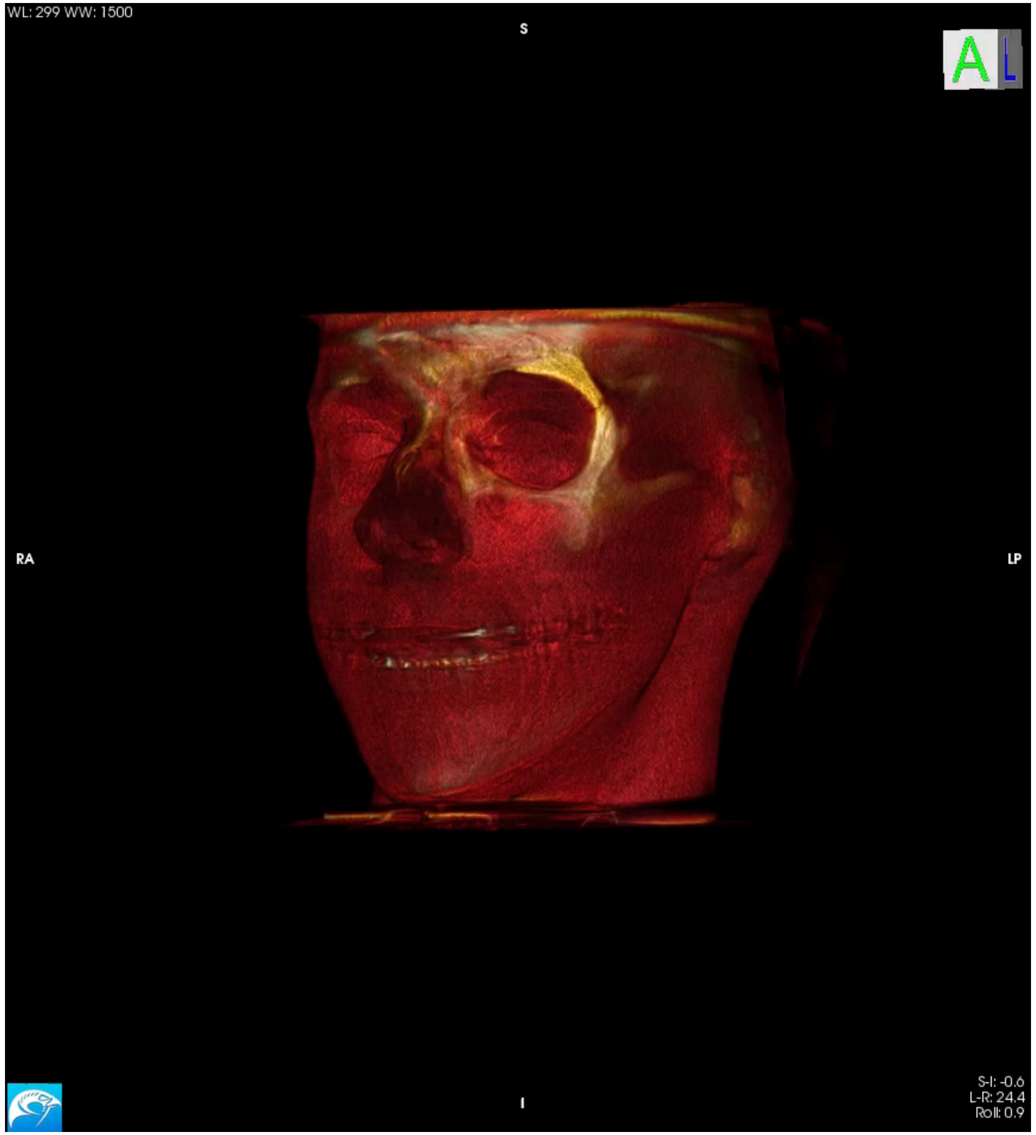
Far Cry

Tomb Raider

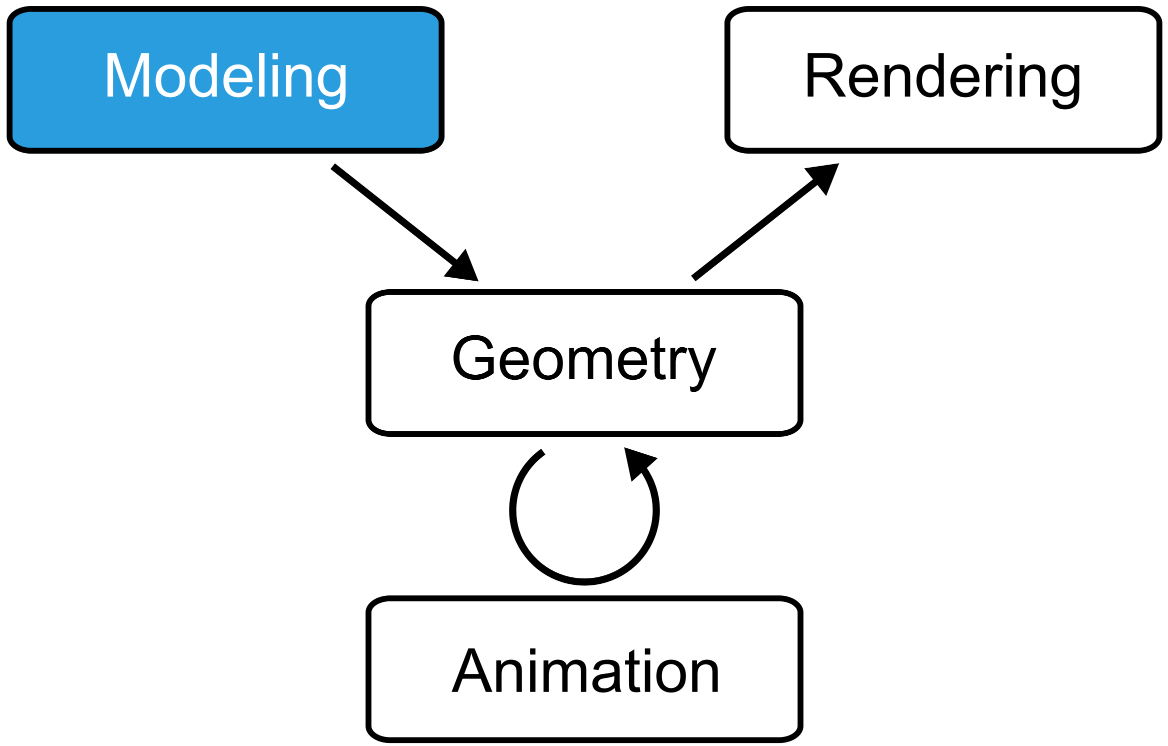
Scientific Visualization



Visualization in Medical Imaging (Volume Rendering)


WL: 299 WW: 1500
S
RA
LP
I
S-I: -0.6
L-R: 24.4
Roll: 0.9

Computer Graphics & Geometry Processing

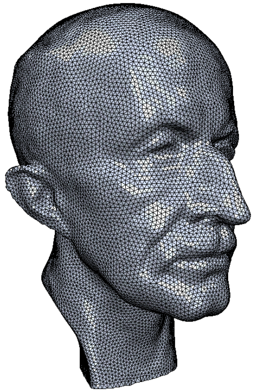


(Laser) Scanning

Get a nice model...

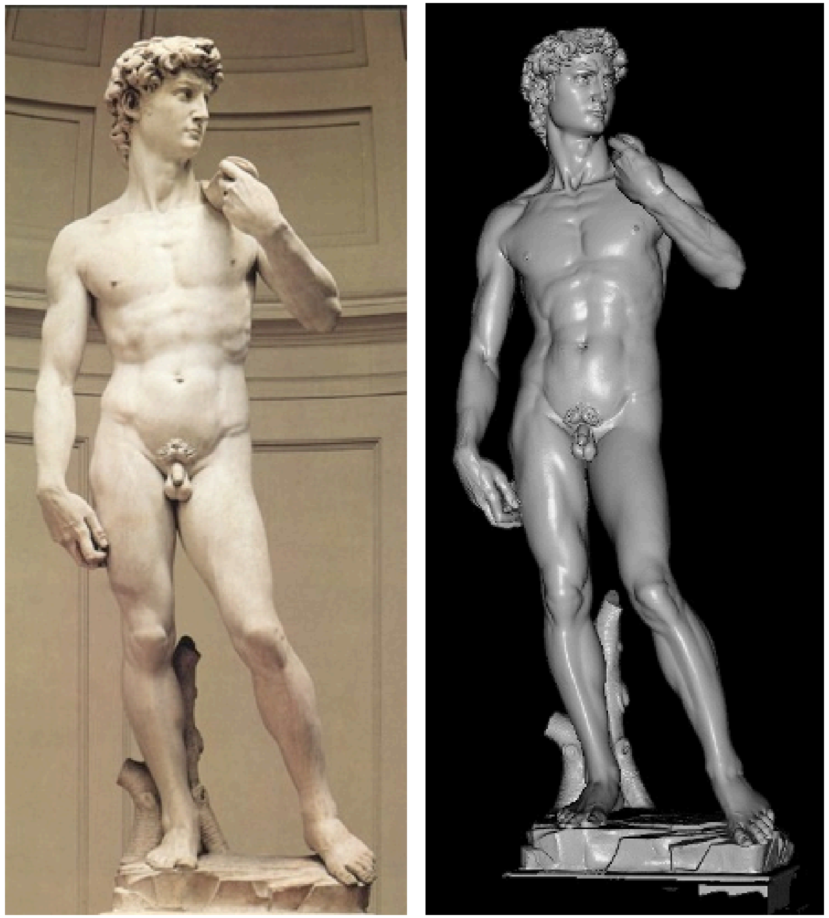
...buy a laser scanner

...perform some scans



...and reconstruct a virtual model!

Cultural Heritage

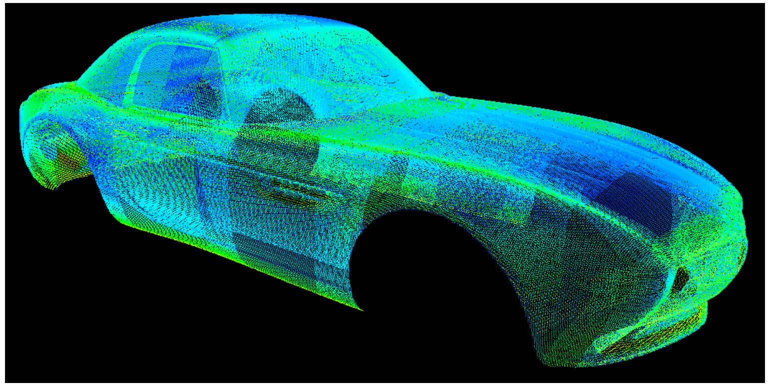


David
1G sample points

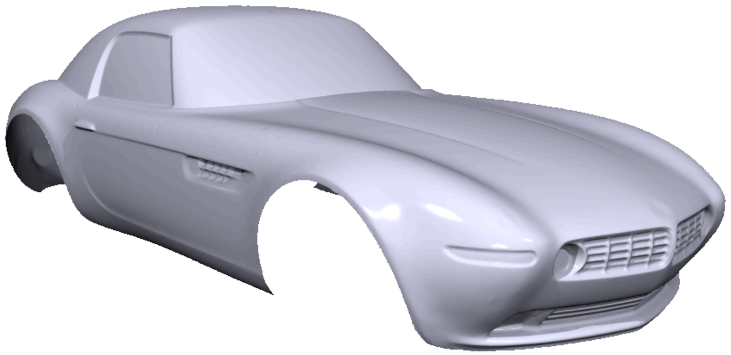
St. Matthew
4G sample points

Automotive Industry

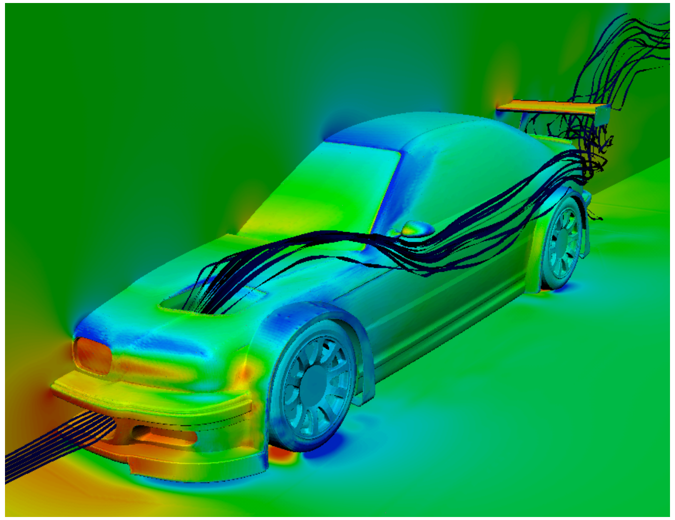
Real model



Scanner point cloud

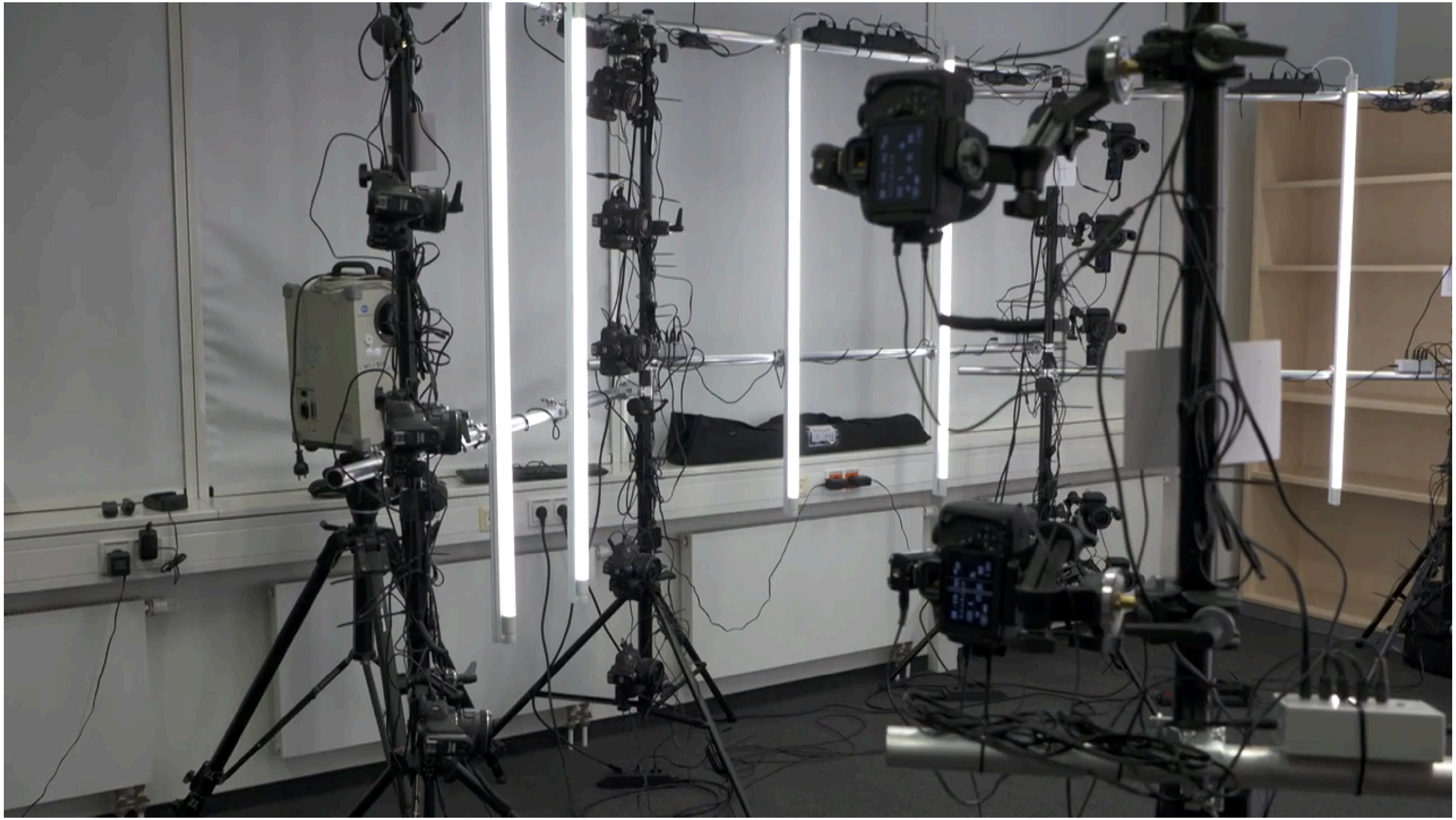


Reconstructed surface



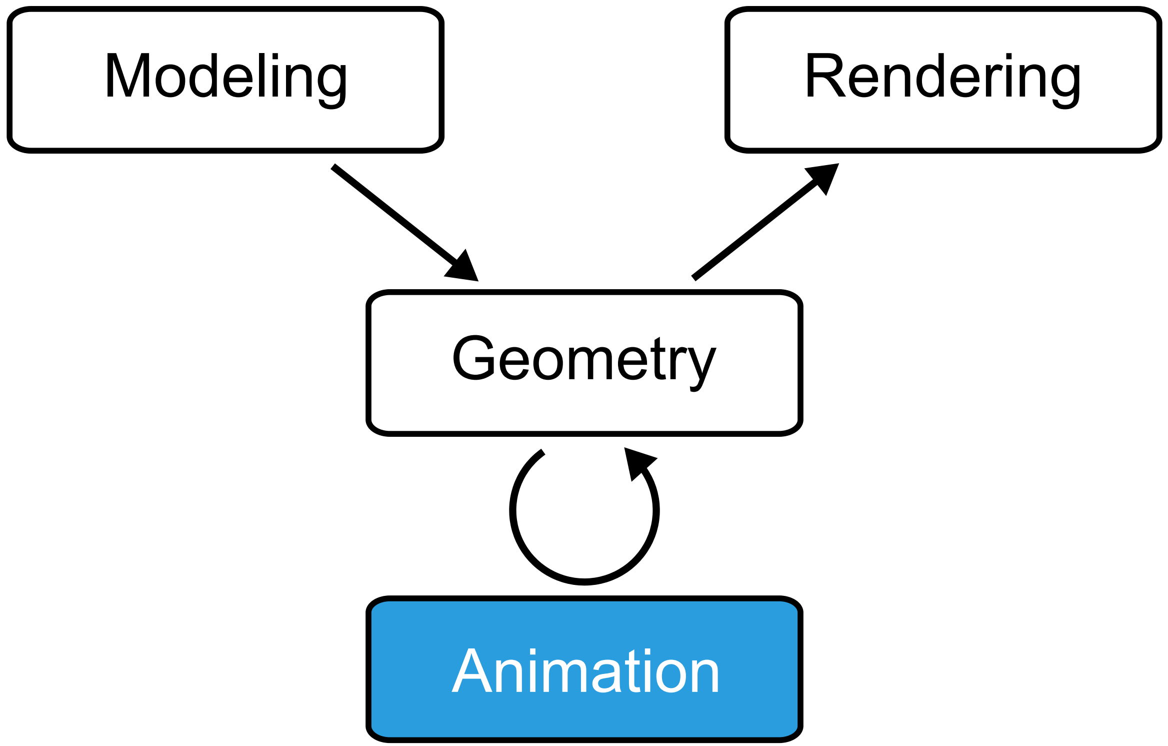
Flow simulation

Avatar Generation



© Mario Botsch

Computer Graphics & Geometry Processing



Motion Capturing & Character Animation

Real-Time Motion Capturing

Face Animation

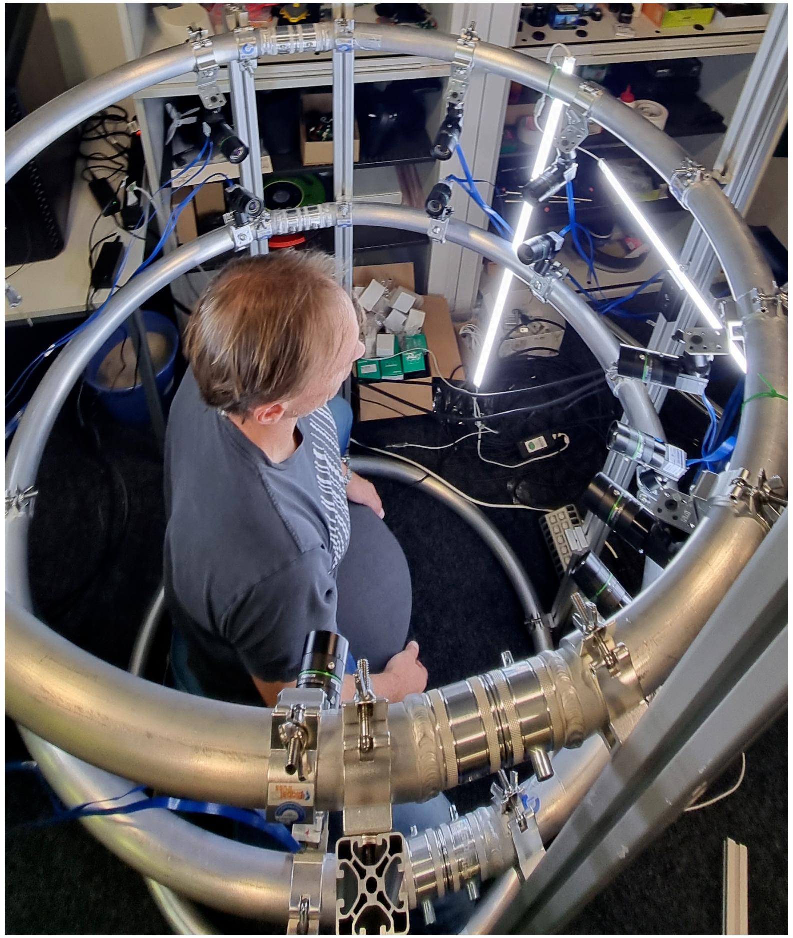
Facial Animation by Linear Delta-Blendshapes

- Facial expressions by blending example poses

$$\mathbf{x}_i = \mathbf{x}_i^{(0)} + \sum_{k=1}^{n} w_k \left(\mathbf{x}_i^{(k)} - \mathbf{x}_i^{(0)} \right)$$

  - Examples (blendshapes) have identical triangulation
  - Linear interpolation per vertex $\mathbf{x}_i$
  - Weights w_k typically are in $[0, 1]$.
    (see what happens at 200% on Youtube)

Face Scanner at HSRM



16 synchronized 12MP, 30fps cameras

Expressions

Face Animation

Cloth Simulation

Collisions

Elastic Deformations

Plasticity & Fracture

Cutting Simulation

About this course

Course Goal and Content

- **Goal**
  - Gain an understanding of the theoretical and practical concepts of computer graphics
- **After attending the course**, you should be able to
  - understand the basic principles of image generation using local and global illumination
  - implement a rendering system based on real-time OpenGL or offline ray-tracing
  - know the pros/cons of the main geometry representations for 3D models or scenes

- **(Planned) Content (12 Lectures)**
  1. Introduction, Organization
  2. Primitives, Transformations
  3. Raytracing
  4. Colors
  5. Lighting
  6. Rendering Equation
  7. Meshes
  8. 3D Transformations
  9. Projections
  10. Rasterization
  11. OpenGL
  12. Textures and Shadows

Organization

- **SWS 2V + 2Ü, 6 ECTS, Total Workload: 180h**
- **Lecture**
  - Thursday D14, 10:00-11:30
- **Exercise Sessions**
  - Thursday D12, Group B: 11:45-13:15, Group A: 14:15-15:45
  - Friday D12, Group C: 14:15-15:45
  - Exercises are mandatory
  - Three evaluated assignments
- **Exam**
  - Content: lectures and exercises
  - Very likely written (date and time will be announced)

Course Materials

- **Books (Graphics in General, Global Illumination and a little bit about Matrices)**
  - A. Watt, *3D Computer Graphics*, Addison-Wesley Longman, 1999
  - J. D. Foley, A. van Dam, S. K. Feiner, *Computer Graphics: Principles and Practice*, Addison Wesley, 2012
  - T. Akenine-Möller, E. Haines, N. Hoffman, *Real-time Rendering*, A K Peters/CRC Press, 2018
    http://www.realtimerendering.com/
  - A. S. Glassner, *An Introduction to Ray Tracing*, Academic Press, 1989.
    http://www.realtimerendering.com/raytracing/An-Introduction-to-Ray-Tracing-The-Morgan-Kaufmann-Series-in-Computer-Graphics-.pdf
  - M. Pharr, W. Jakob, g. Humphreys, *Physically Based Rendering*, 2016.
    http://www.pbr-book.org/

Course Materials

- **Books (OpenGL)**
  - J. Kessenich, G. Sellers, D. Shreiner, *OpenGL Programming Guide*, 9th edition, Addison-Wesley, 2016
    http://www.opengl-redbook.com
  - R. J. Rost, *OpenGL Shading Language*, 3rd edition, Addison-Wesley, 2009
  - R. S. Wright, N. Haemel, G. Sellers, B. Lipchak, *OpenGL SuperBible: Comprehensive Tutorial and Reference*, Addison Wesley, 2015
  - E. Angel, D. Shreiner, *Interactive Computer Graphics with WebGL*, Addison Wesley, 2015
- **Books (Curves and Surfaces)**
  - G. Farin, *Curves and Surfaces for CAGD: A Practical Guide*, 5th edition, Morgan Kaufmann, 2002
- **Books (Matrices)**
  - K. B. Petersen, M. S. Pedersen, *The Matrix Cookbook*
    http://www.cs.toronto.edu/~bonner/courses/2012s/csc338/matrix_cookbook.pdf

Course Materials

- **Tutorials**
  - The Python Tutorial: https://docs.python.org/3/tutorial
  - Numpy Quickstart: https://numpy.org/devdocs/user/quickstart.html
- **Frameworks, IDEs**
  - Visual Studio Code: https://code.visualstudio.com/

Prerequisites

- **Basic math skills**
  - Linear Algebra
    - Vectors: $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$
    - Matrices: $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m \times n}$
    - Operations:
      - $\mathbf{x}^\top \mathbf{y}, \mathbf{x} \times \mathbf{y}$, $\mathbf{A}\mathbf{x}$
      - $\mathbf{A}^\top, \mathbf{A}^{-1}, \text{trace}(\mathbf{A}), \det(\mathbf{A}), \mathbf{A} + \mathbf{B}, \mathbf{A}\mathbf{B}$
    - Norms: $||\mathbf{x}||_1, ||\mathbf{x}||_2, ||\mathbf{x}||_\infty, ||\mathbf{A}||_F$
    - Eigenvalues, Eigenvectors, SVD: $\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$
  - Calculus
    - (Multivariate) functions: $f : \mathbb{R}^n \rightarrow \mathbb{R}$
    - (Partial) derivatives: $\frac{\partial f}{\partial x_i}, i = 1, \ldots, n$, Gradient
    - Integrals: $\int f(x)dx$

- **Basis computer science skills**
  - Variables
  - Functions
  - Loops
  - Classes
  - Algorithms
- **Basic Python coding skills**
  - https://docs.python.org/3/turorial/

Time Management

Activity	Times	Total
Attending (watching) the lecture	2h / week	24h
Self-study of lecture materials	2h / week	24h
Participation in exercise	2h / week	24h
Solving the assignments	6h / week	72h
Preparation for the final exam	36h	36h
Total workload		**180h**