Image Processing and Computer Graphics

Adam Kortylewski
Max Argus
Thomas Brox
Matthias Teschner

UNI FREIBURG

Organization

Computer Graphics	Image Processing
Matthias Teschner	Adam Kortylewski Max Argus Thomas Brox
https://cg.informatik.uni-freiburg.de/teaching.htm	https://lmb.informatik.uni-freiburg.de/lectures/image_processing/

Computer Graphics
Modeling – Rendering – Simulation
Introduction

Matthias Teschner

UNI FREIBURG

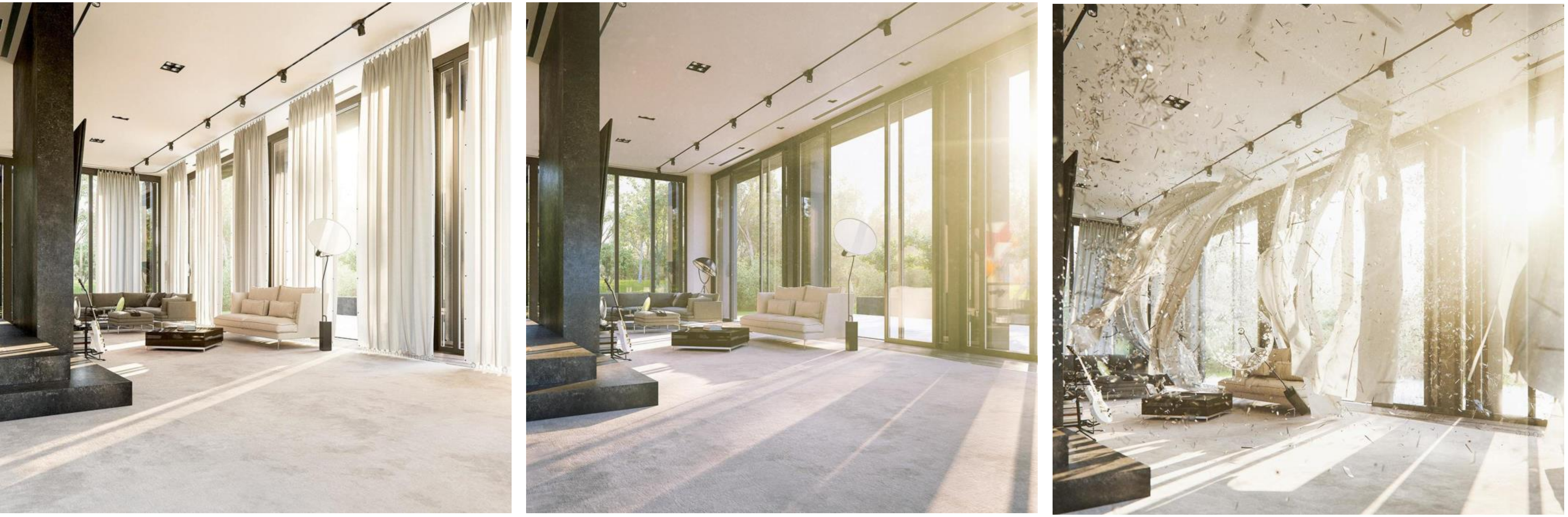
Computer Graphics

Modeling – Rendering – Simulation

© Warner Bros.
Scanline VFX
V-Ray

Computer Graphics

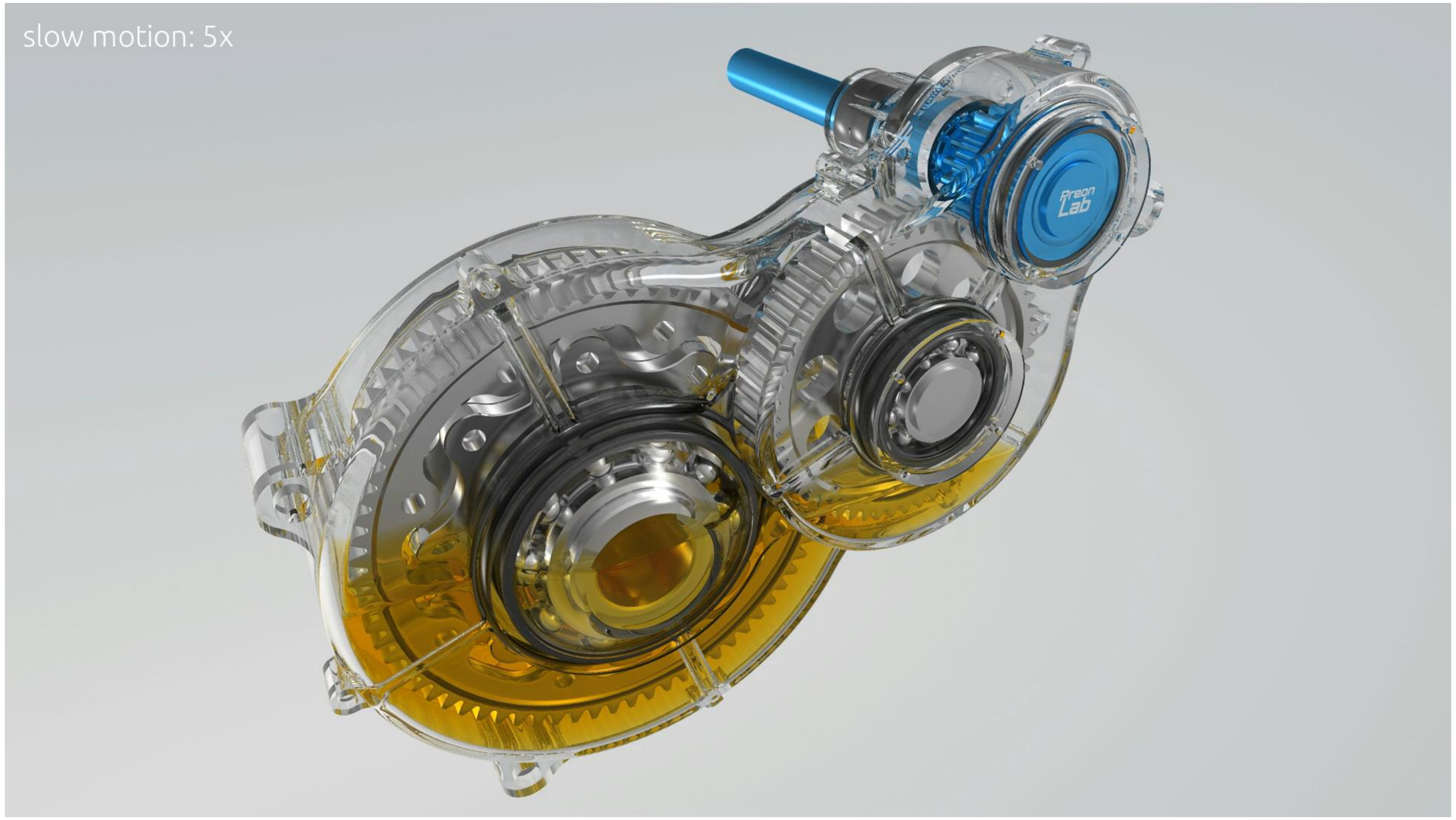
Modeling – Rendering – Simulation



© Double Aye
V-Ray

Computer Graphics

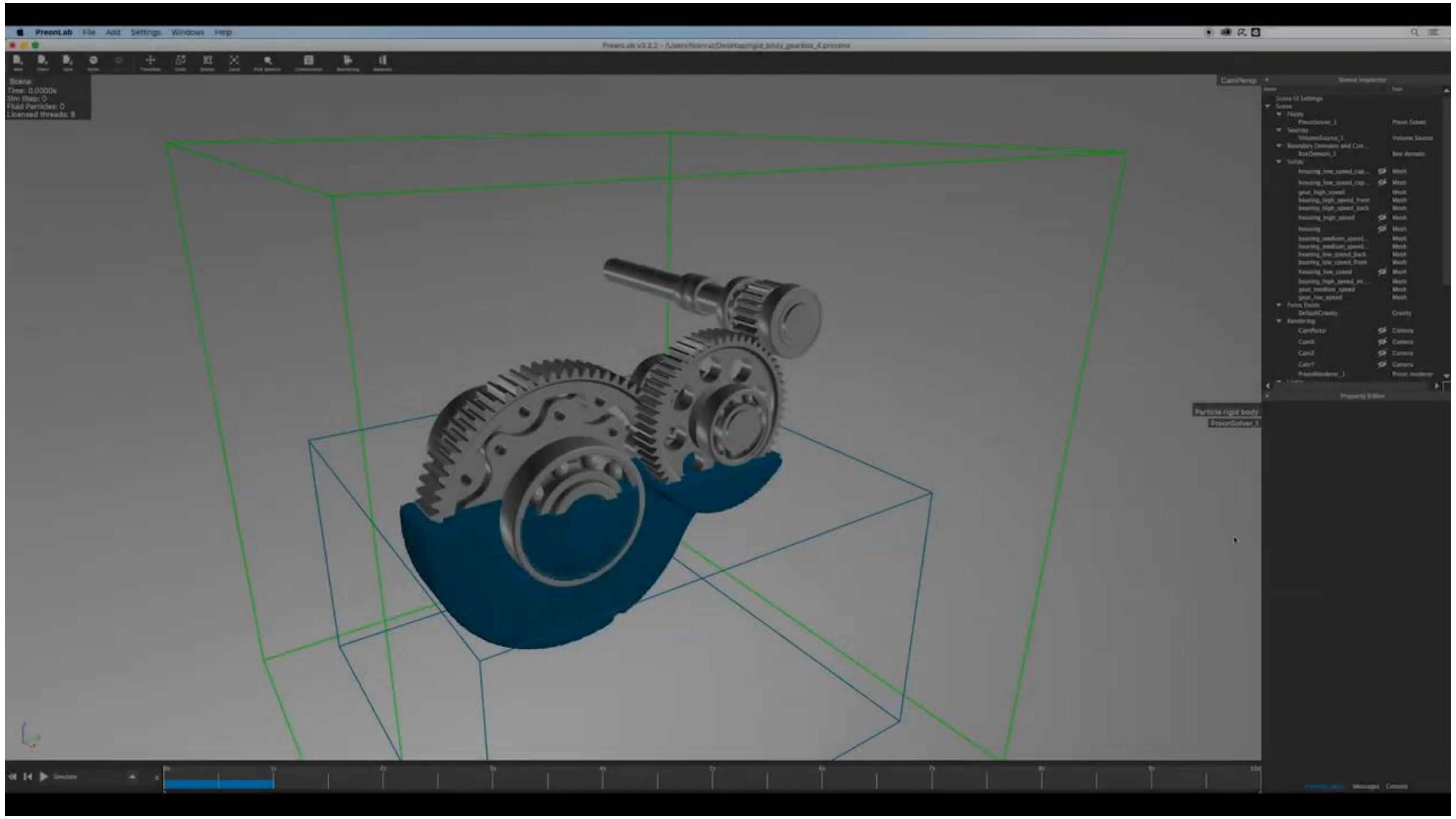
Modeling – Rendering – Simulation



FIFTY2 Technology

Computer Graphics

Modeling – Rendering – Simulation



FIFTY2 Technology

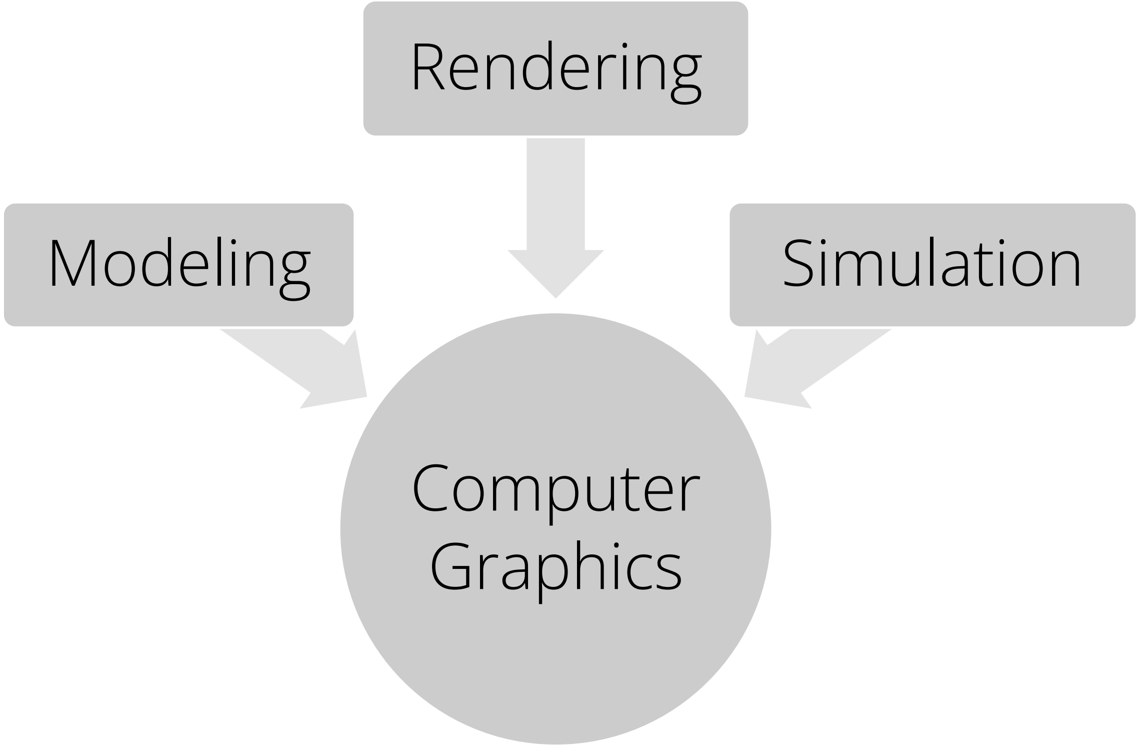
Application Areas

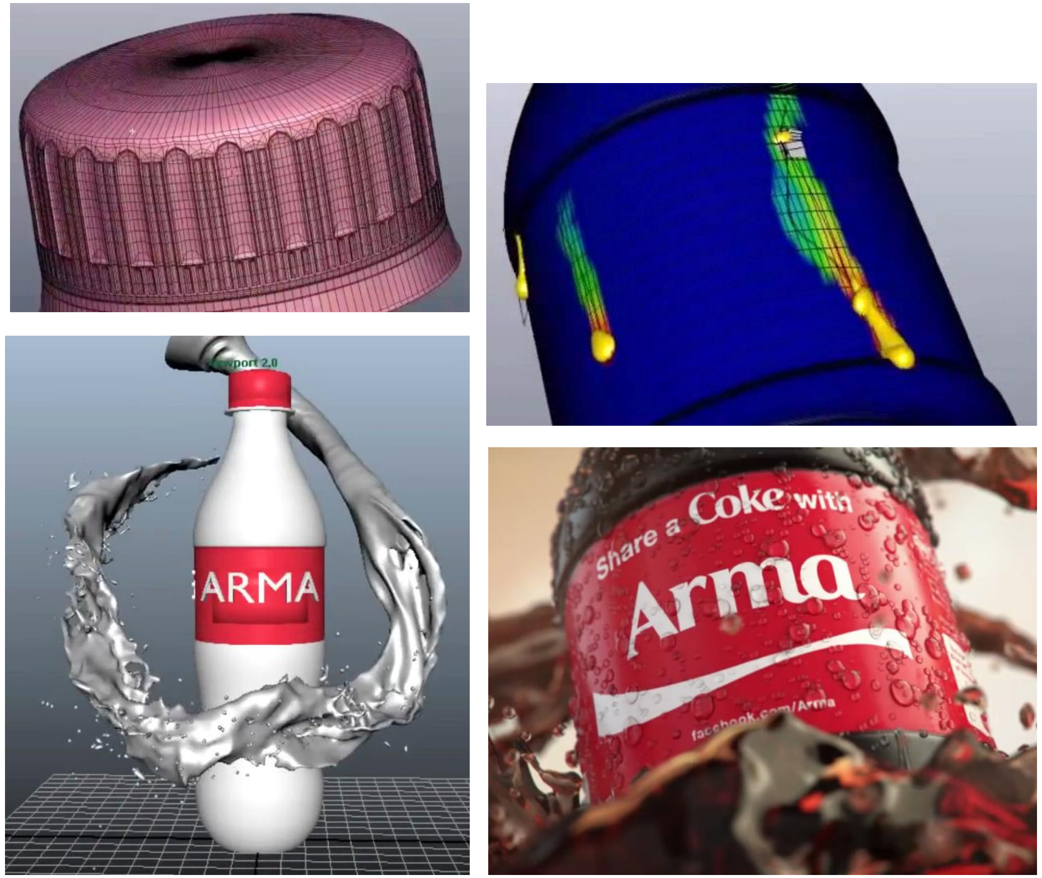
- Visual effects (movies, commercials)
- Architecture
- Engineering
- Medical imaging
- Scientific visualization
- Games
- Virtual reality / augmented reality

Computer Graphics

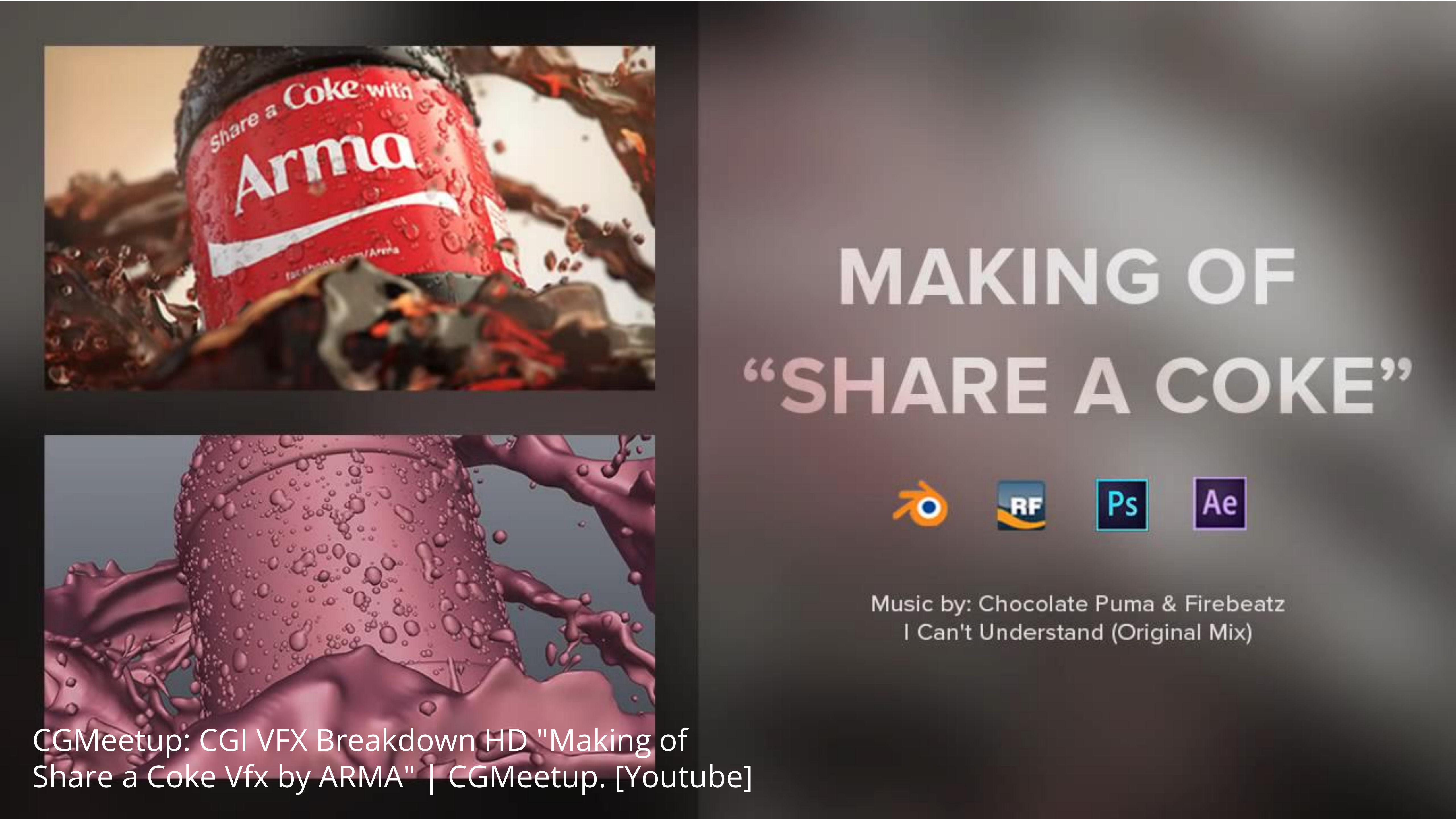
- Light
  - Energy or photons transported along lines
  - Generated by light sources, measured / absorbed by sensors, interacts at surfaces and with participating media
- Modeling
  - Geometry, materials, participating media, illumination
- Rendering
  - Computation of light transport
- Simulation
  - Dynamic rigid bodies, deformable objects and fluids

Computer Graphics





CGMeetup: CGI VFX Breakdown HD "Making of Share a Coke Vfx by ARMA" | CGMeetup. [Youtube]



CGMeetup: CGI VFX Breakdown HD "Making of Share a Coke Vfx by ARMA" | CGMeetup. [Youtube]

Outline

- Organization
- Our research
- Image generation
- Course topics

Graphics Courses

- Key course
  - Image processing and computer graphics
    (modeling, rendering, simulation)
- Specialization courses
  - Advanced computer graphics (global illumination)
  - Simulation in computer graphics
    (deformable and rigid solids, fluids)
- Lab course, Master project, Master thesis
  - Simulation track, rendering track

Seminars / Projects / Theses in Graphics

Semester	Simulation Track	Rendering Track
Winter	Simulation Course	
Summer	Key Course Lab Course - Simple fluid solver Simulation Seminar	Key Course Lab Course - Simple Ray Tracer Rendering Seminar
Winter	Master Project - PPE fluid solver Rendering Seminar	Rendering Course Master Project - Monte Carlo RT Simulation Seminar
Summer	Master Thesis Research-oriented topic	Master Thesis Research-oriented topic

Material – Exam

- Slide sets and recordings
- Slides, recordings, exercises, solutions and test exam on https://cg.informatik.uni-freiburg.de/teaching.htm
- Written exam

Selected Readings

- Thomas Akenine Moeller et al.: *Real-time rendering*. Taylor & Francis, 2018.
- Matt Pharr, Wenzel Jakob, Greg Humphreys.
  Physically based rendering: From theory to implementation. Morgan Kaufmann, 2016.
  Free online version: http://www.pbr-book.org/
- Andrew S. Glassner.
  Principles of digital image synthesis. Morgan Kaufmann, 1995.
  Free download on https://www.realtimerendering.com/
- Steve Marschner, Peter Shirley.
  Fundamentals of computer graphics. CRC Press, 2015.
- Alan Watt. *3D computer graphics*. Addison-Wesley, 1999.
- James D. Foley, Andries van Dam, Steven K. Feiner.
  Computer graphics: principles and practice. Pearson Education, 2014.
- Andrew S. Glassner. *An introduction to ray tracing*. Elsevier, 1989.

Exercises

- Introduction to OpenGL >3.0
  - Programming interface for rendering
- Four exercises
- Two tasks / topics per exercise
  - Related to rasterization, homogeneous notation, projection, Phong shading (check course curriculum)
- Support
  - NN
- Optional

Recommended Prerequisites

- Linear algebra
  - Vector
  - Matrix
- Calculus
  - Differentiation
  - Integration
- Programming language
  - C++, ...

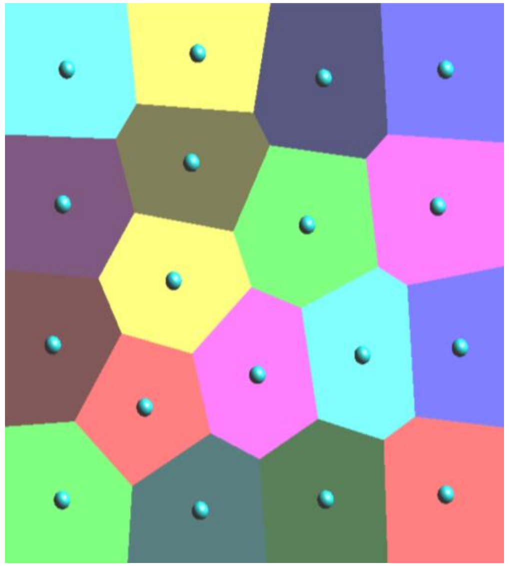
Outline

- Organization
- Our research
- Image generation
- Course topics

Gissler et al., ACM Transactions on Graphics, 2019

Lagrangian Simulation

Fluid / Elastic object / Rigid object



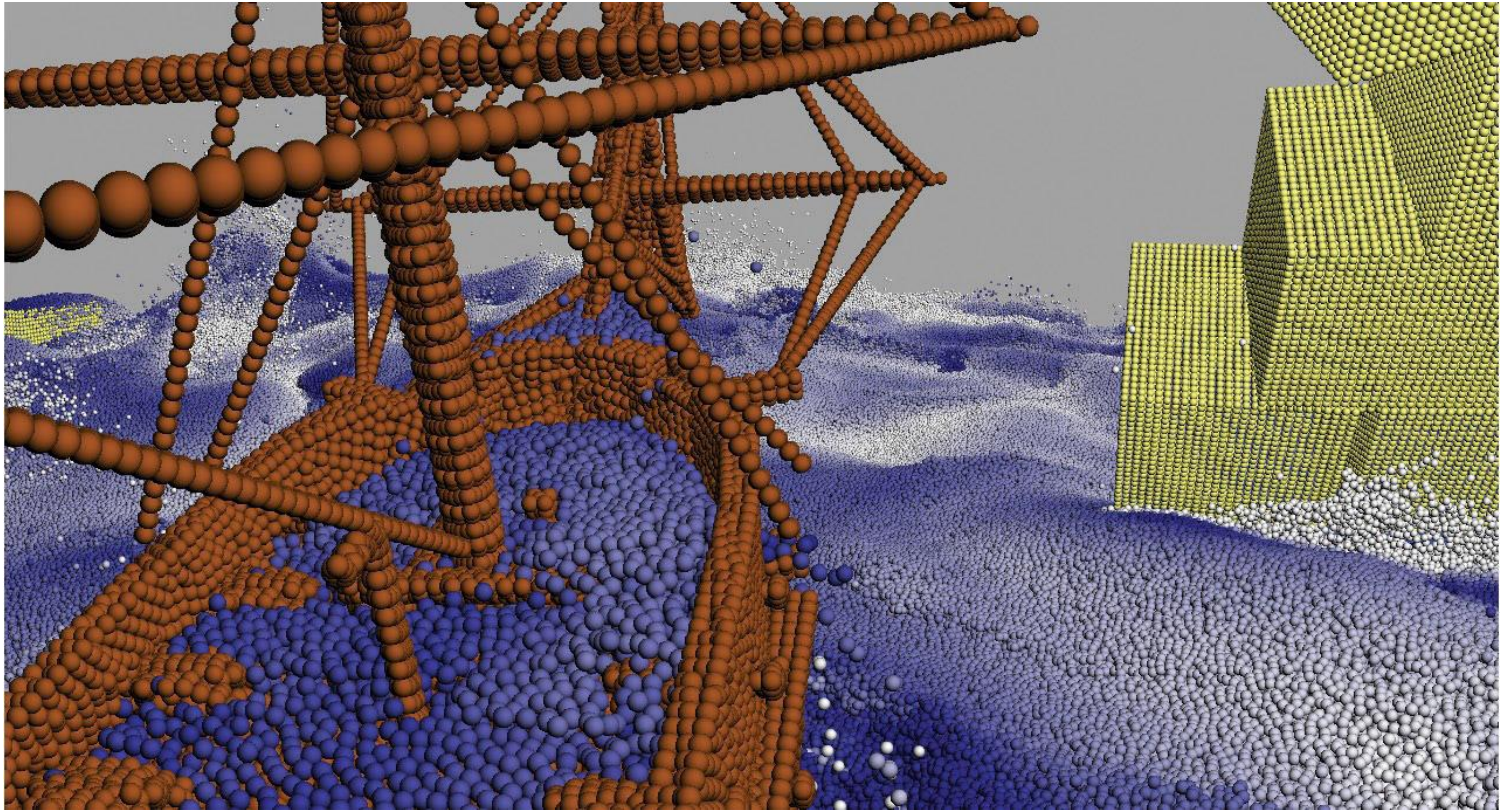
Set of parcels

$$\boldsymbol{x}_i^t$$

$$\boldsymbol{v}_i^t$$

Positions and velocities of parcels *i* over time *t*

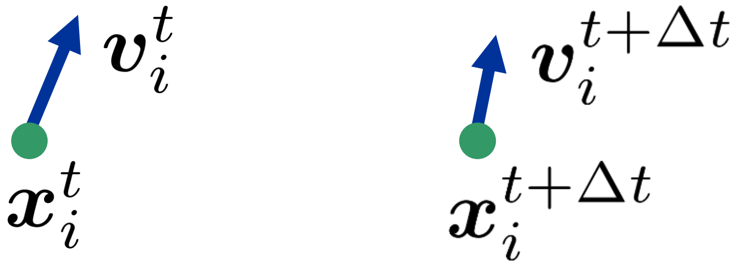
Fluid and Solid Parcels



Akinci et al., *ACM SIGGRAPH*, 2012.

Parcel Movement for Fluids

Task



Governing equations

$$\frac{\mathrm{d}\boldsymbol{v}_i^t}{\mathrm{d}t} = \boldsymbol{a}_i^t = -\frac{1}{\rho_i^t}\nabla p_i^t + \nu \nabla^2 \boldsymbol{v}_i^t + \boldsymbol{g} \qquad \frac{\mathrm{d}\boldsymbol{x}_i^t}{\mathrm{d}t} = \boldsymbol{v}_i^t$$

$$\frac{\mathrm{d}\rho_i^t}{\mathrm{d}t} = -\rho_i^t \nabla \cdot \boldsymbol{v}_i^t = 0$$

Numerics

$$\nabla p_i^t \approx \sum_j \frac{m_j}{\rho_j^t} p_j^t \nabla W_{ij}^t \qquad \nabla^2 \boldsymbol{v}_i^t \approx \sum_j \ldots$$

$$\boldsymbol{v}_i^{t+\Delta t} = \ldots \qquad \boldsymbol{x}_i^{t+\Delta t} = \ldots$$

Typical Steps of a Fluid Solver

- Neighbors j of i
- Predicted velocity

$$\boldsymbol{v}_i^* = \boldsymbol{v}_i^t + \Delta t \left(\nu \nabla^2 \boldsymbol{v}_i^t + \boldsymbol{g} \right)$$

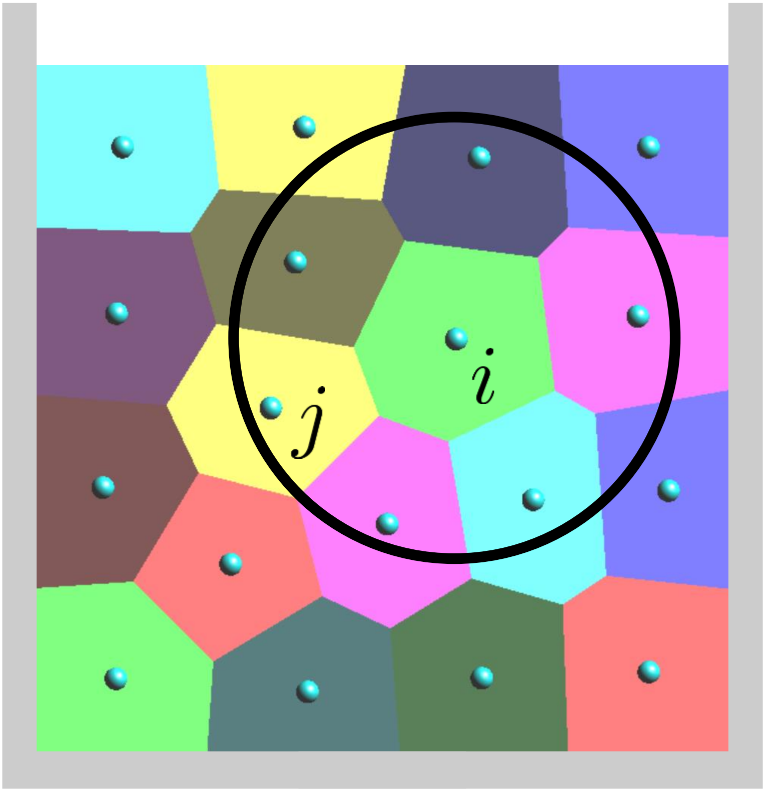
- Pressure

$$\nabla \cdot \boldsymbol{v}_i^* + \nabla \cdot (-\Delta t \frac{1}{\rho_i^t} \nabla p_i^t) = 0$$

- Velocity and position

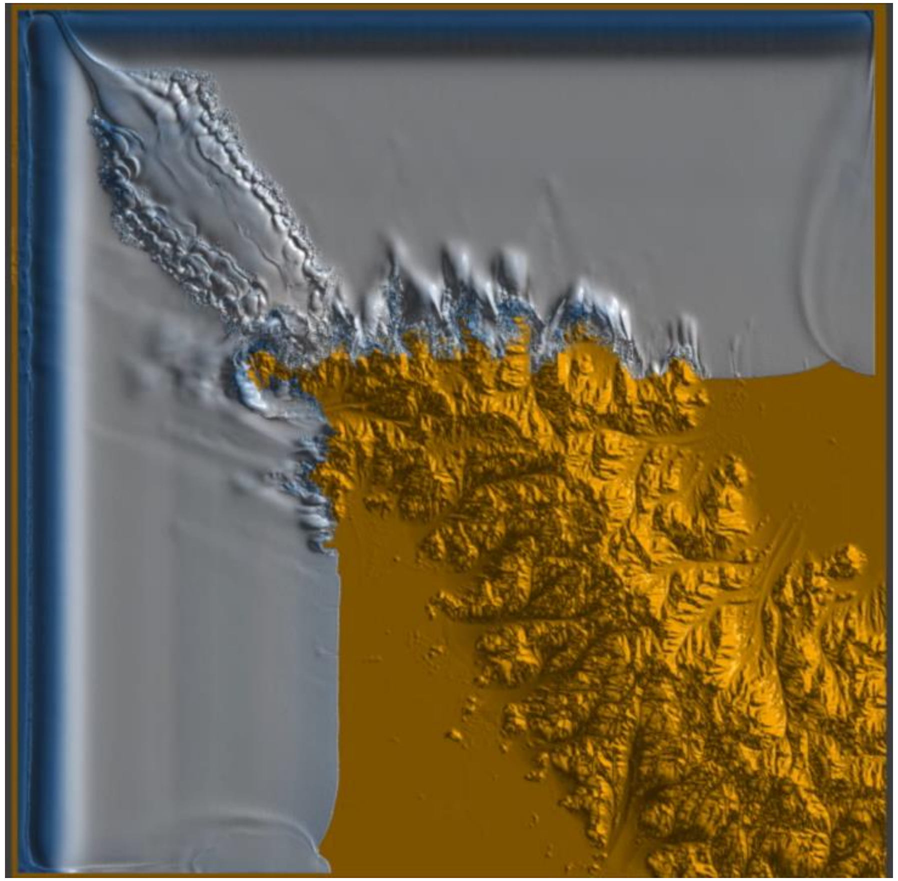
$$\boldsymbol{v}_i^{t+\Delta t} = \boldsymbol{v}_i^* - \Delta t \frac{1}{\rho_i^t} \nabla p_i^t$$

$$\boldsymbol{x}_i^{t+\Delta t} = \boldsymbol{x}_i^t + \Delta t \boldsymbol{v}_i^{t+\Delta t}$$



Neighbor Search

- Huge numbers of neighbors have to be estimated
- Uniform grid
  - Sorted list
  - Compact hashing
  - 1 million samples: 20 ms
  - 1 billion samples: 30 s
- Minimized secondary data structures



Ihmsen et al.,
Computer Graphics Forum, 2011.

Pressure Computation

- Solving a pressure Poisson equation
  - Matrix-free
  - OpenMP, MPI
  - Up to 1 billion samples on desktop PCs

Ihmsen et al., *IEEE Transactions on Visualization and Graphics*, 2014.

$$\nabla \cdot \boldsymbol{v}_i^* + \nabla \cdot \left(-\Delta t \frac{1}{\rho_i^t} \nabla p_i^t\right) = 0$$

$$\Downarrow$$

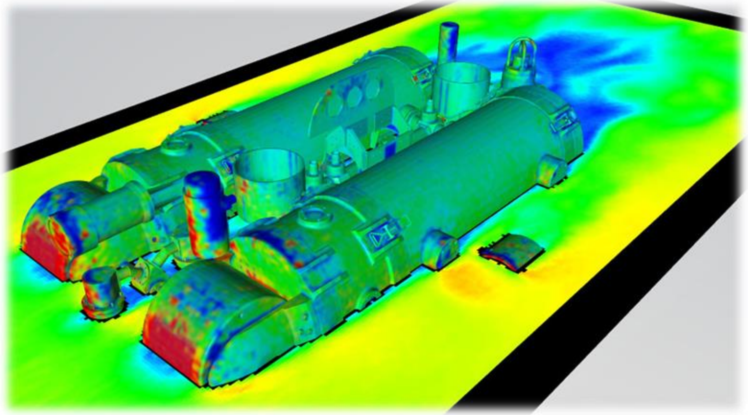
$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix} \begin{pmatrix} p_1^t \\ p_2^t \\ \vdots \\ p_n^t \end{pmatrix} = \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{pmatrix}$$

Applications

Pixar Animation Studios, Emeryville

FIFTY2 Technology, Freiburg

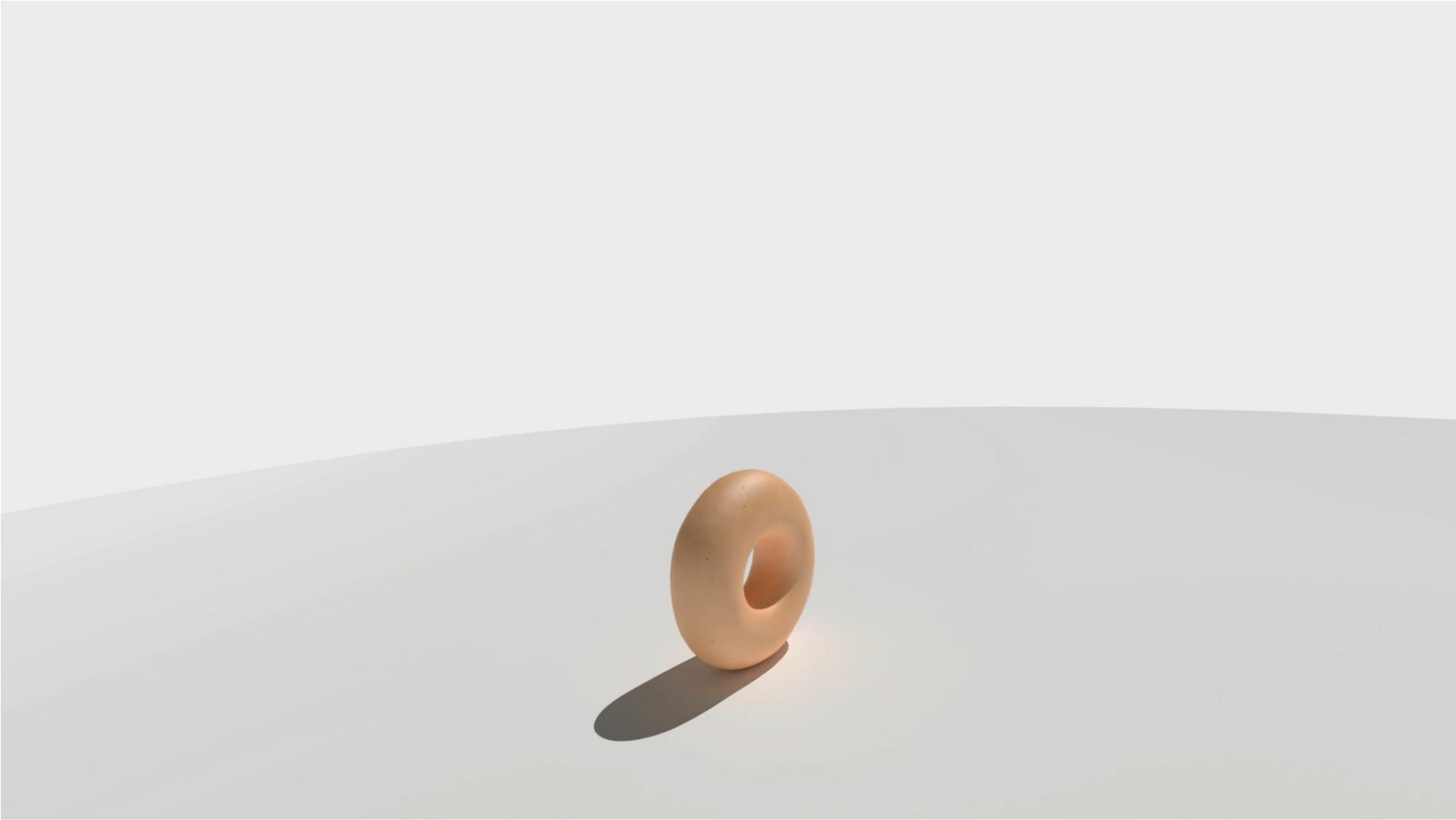
Studio Claudia Comte, Grancy / Berlin



Robotics Innovation Center DFKI, Bremen

UNI FREIBURG

Fluids Meet Art

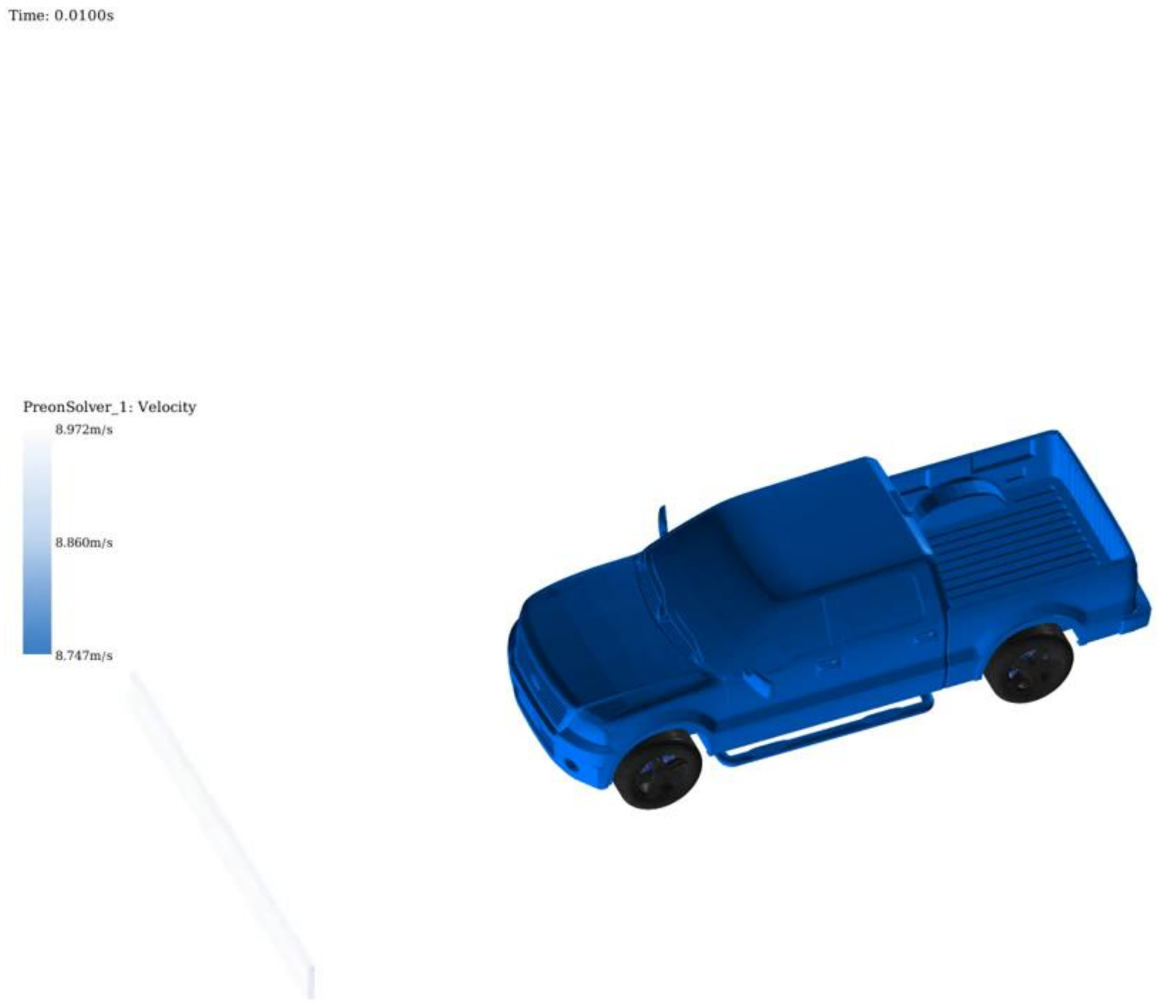


Studio Claudia Comte
Grancy / Berlin

Andreas Peer
University of Freiburg

PreonLab
FIFTY2 Technology

Fluids in Engineering



PreonLab
FIFTY2 Technology

FORD F-150

Water wading

Outline

- Organization
- Our research
- Image generation
- Course topics

Setup Aspects

- Light
- Scene
  - Light sources, sensor / eye / camera
  - Geometry, materials / reflection properties
  - Participating media, e.g. haze, fog
- Dynamics
  - Simulation of fluids, elastic and rigid solids

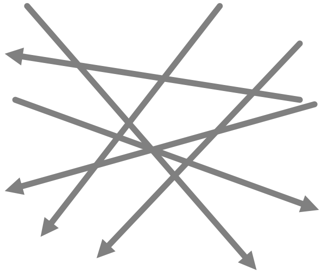
Rendering Aspects

- What is visible by the sensor?
  - Rasterization
  - Ray Tracing
- Which color / intensity does it have?
  - Local evaluation of governing equations (Phong illumination model)
  - Global evaluation of governing equations for light interaction at surfaces (rendering equation) and in participating media (volume rendering equation)

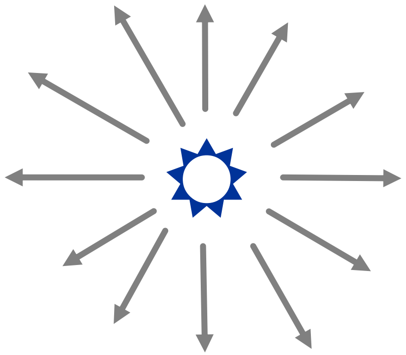
Light

- Modeled as energy parcels / photons that travel
  - Along geometric rays
  - At infinite speed
- Emitted by light sources
- Scattered / absorbed at surfaces
- Scattered / absorbed by participating media
- Absorbed / measured by sensors

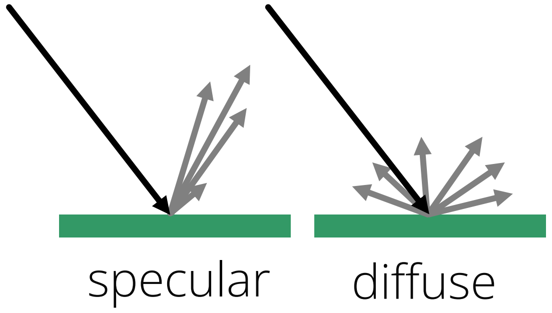
Light



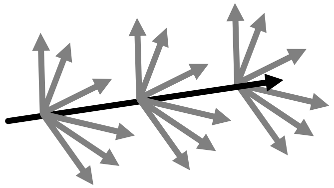
Light travels along rays



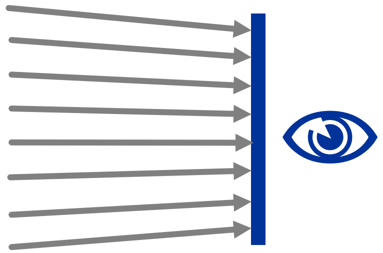
Light is generated at light sources



Incoming light is scattered and absorbed at surfaces



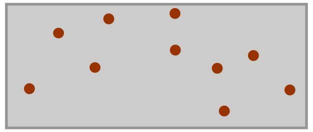
Participating media scatters and absorbs light



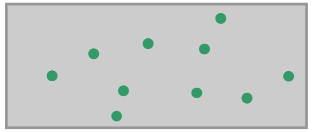
Sensors absorb light

Color

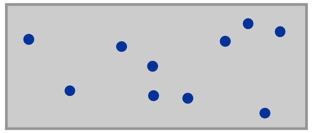
- Photons are characterized by a wavelength within the visible spectrum
- Distribution of wavelengths ⇨ spectrum ⇨ color



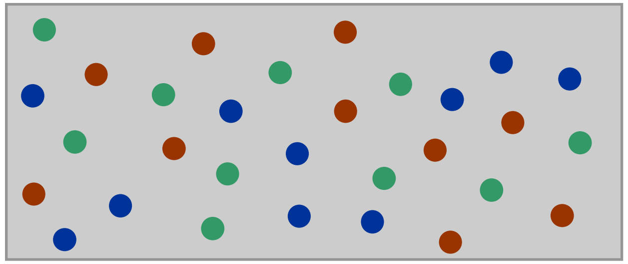
$\Phi_\lambda(\lambda_1)$



$\Phi_\lambda(\lambda_2)$



$\Phi_\lambda(\lambda_3)$



$$\Phi = \int_{\mathrm{VisibleSpectrum}} \Phi_\lambda(\lambda) d\lambda$$

$$\approx \sum_i \Phi_\lambda(\lambda_i)\Delta\lambda_i$$

$$\approx \Phi_{\mathrm{red}}\Delta\lambda + \Phi_{\mathrm{green}}\Delta\lambda + \Phi_{\mathrm{blue}}\Delta\lambda$$

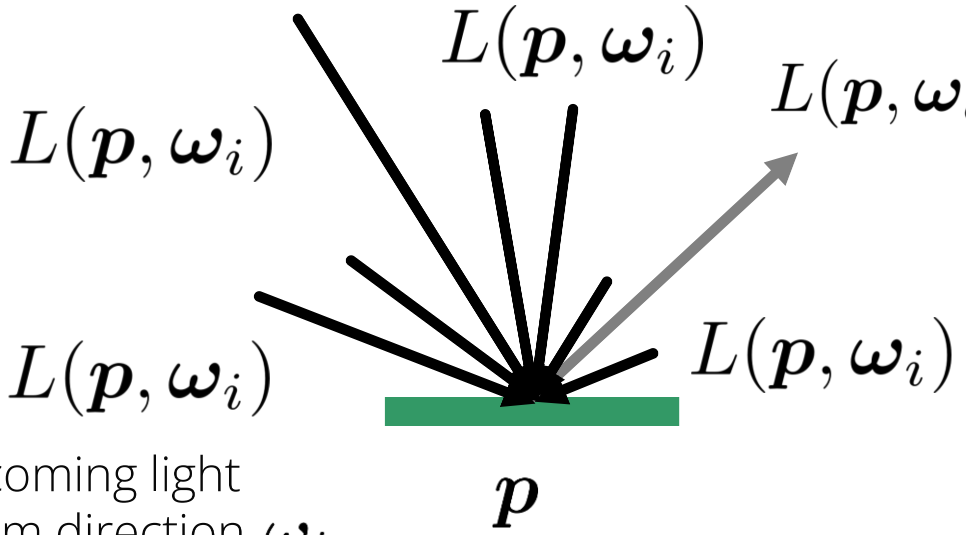
$\Phi_\lambda(\lambda)$: number of photons per time with a wavelength in a range $\Delta\lambda_i$ around λ_i.

Governing Equations

- Light is affected by surfaces and by participating media
- Processes described by governing equations
  - Rendering equation
  - Volume rendering equation

Light at Surfaces

– Governing equation for reflected light at surfaces into a particular direction given incident light from all directions



Incoming light from direction $\boldsymbol{\omega}_i$

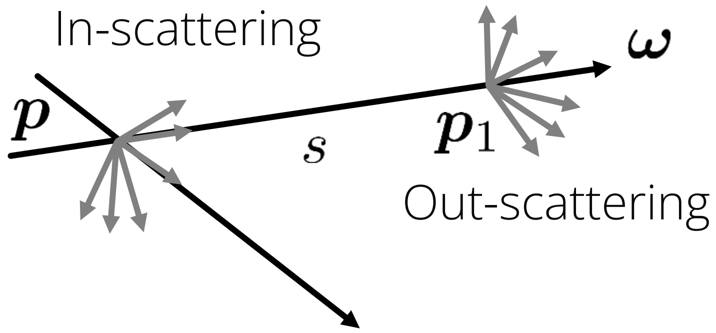
Position

$$L(\boldsymbol{p}, \boldsymbol{\omega}_o) = \int_\Omega \mathrm{mat}(\boldsymbol{p}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_o) L(\boldsymbol{p}, \boldsymbol{\omega}_i) \cos\theta_i \mathrm{d}\omega_i$$

Outgoing light into direction $\boldsymbol{\omega}_o$ is a sum of incident light from all directions weighted with material properties $\mathrm{mat}(\boldsymbol{p}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_o)$

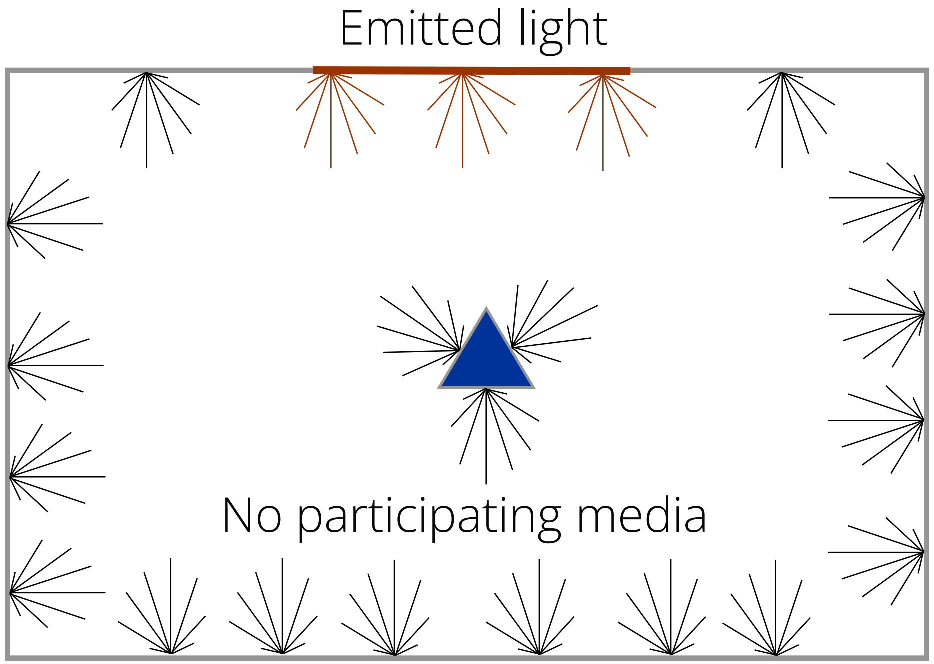
Light in Volumes

- Governing equations for light changes along rays through participating media, e.g. haze or fog
- Setting $L(\boldsymbol{p}_1, \boldsymbol{\omega}) = L(\boldsymbol{p}, \boldsymbol{\omega}) + s\frac{\mathrm{d}L}{\mathrm{d}s}$
- Absorption $\frac{\mathrm{d}L}{\mathrm{d}s} = -\kappa L(\boldsymbol{p}, \boldsymbol{\omega})$
- Emission $\frac{\mathrm{d}L}{\mathrm{d}s} = L_e(\boldsymbol{p}, \boldsymbol{\omega})$
- Out-scattering $\frac{\mathrm{d}L}{\mathrm{d}s} = -\sigma L(\boldsymbol{p}, \boldsymbol{\omega})$
- In-scattering $\frac{\mathrm{d}L}{\mathrm{d}s} = L_j(\boldsymbol{p}, \boldsymbol{\omega})$

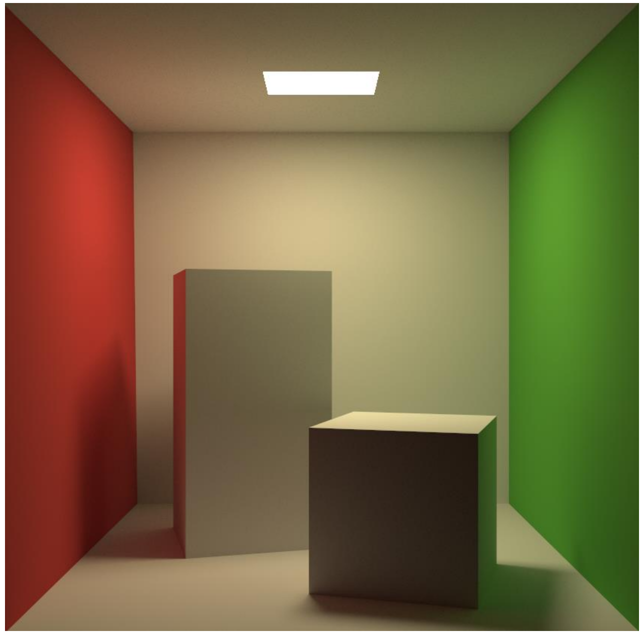


Light Transport

– Governing equations enable the computation of light at all points in space into all direction



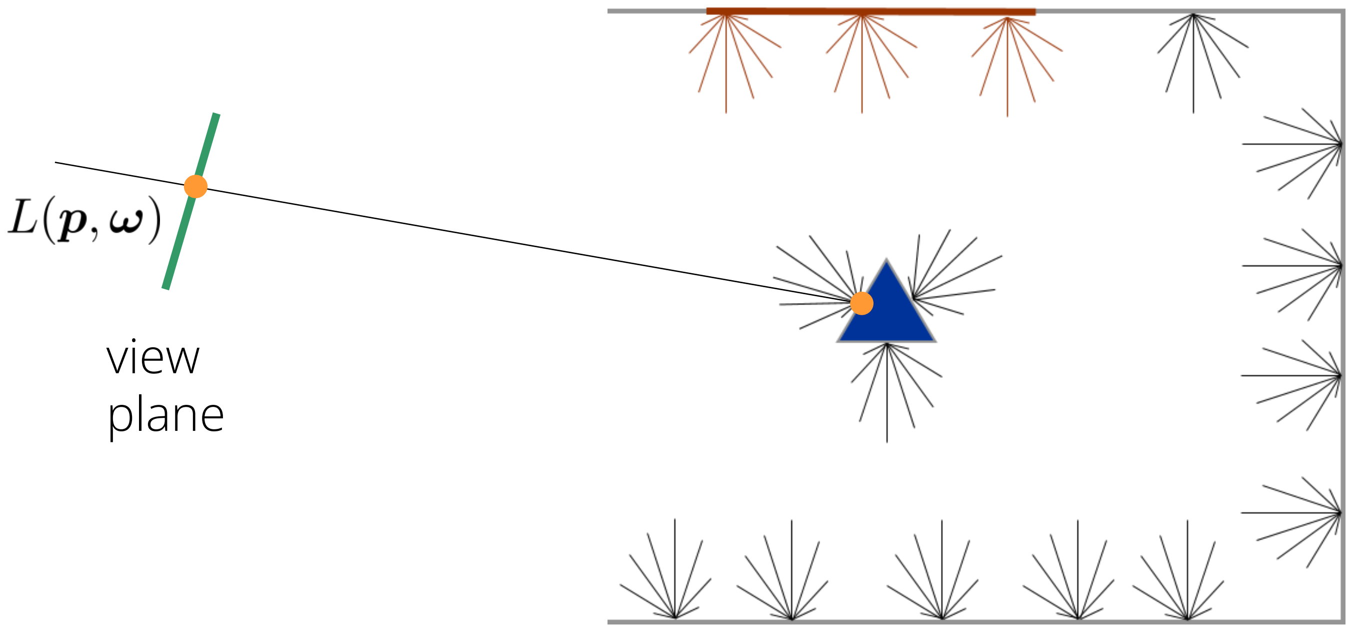
Reflected light due to material properties



Cornell box

Rendering

- At an arbitrarily placed and oriented sensor
  - Cast rays into the scene
  - Lookup light that is transported along these rays



Rendering Algorithms

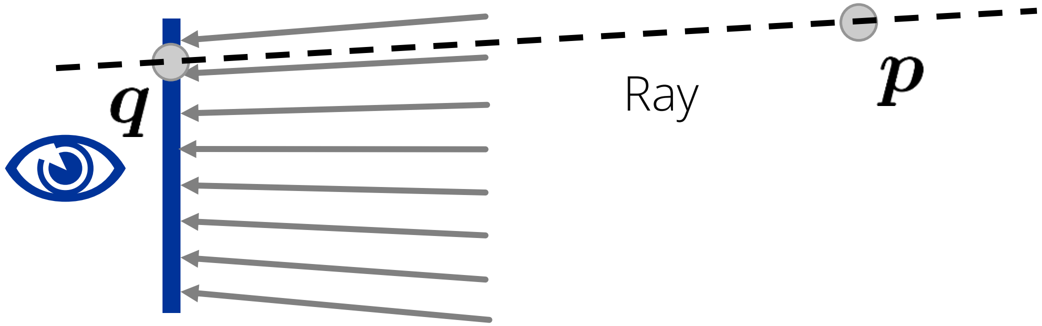
- Approximately solve the light transport in a scene
- Radiosity
  - Computes reflected light at all surface points into all directions
  - Simplifications: No participating media, diffuse surfaces, equal reflected light per finite-size surface patch, e.g. triangle
  - Linear system with unknown reflected light per surface patch

Rendering Algorithms

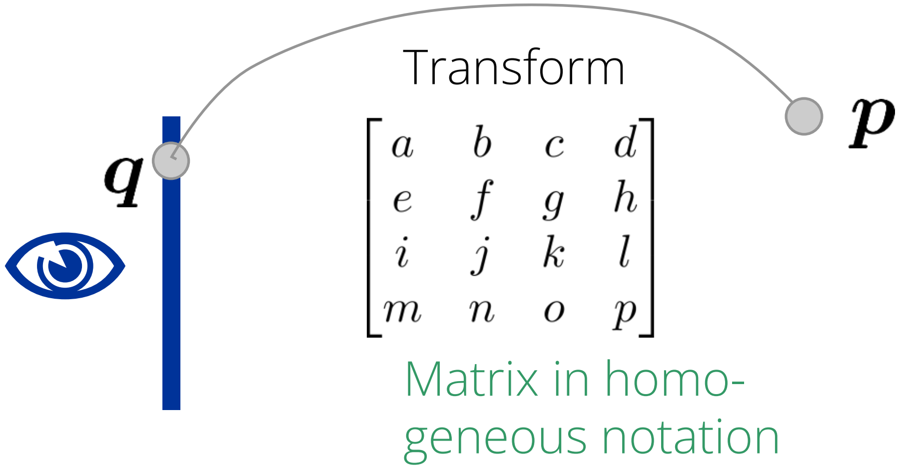
- Ray Tracing, Rasterization
  - Compute visible surfaces
    (What is visible by the sensor?)
  - Have to be combined with shading algorithms
    (Which color does it have?)
    - Phong illumination model
    - Monte-Carlo Ray Tracing

Ray Tracing and Rasterization

– Solve the visibility problem



Ray Tracers compute ray-scene intersections to estimate *q* from *p*.



Rasterizers apply transformations to *p* in order to estimate *q*. *p* is projected onto the sensor plane.

Shading

- Solve $L(\boldsymbol{p}, \boldsymbol{\omega}_o) = \int_\Omega \mathrm{mat}(\boldsymbol{p}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_o) L(\boldsymbol{p}, \boldsymbol{\omega}_i) \cos\theta_i \mathrm{d}\omega_i$ at a surface point $\boldsymbol{p}$ with, e.g., Monte-Carlo raytracing
  - Accumulate all illumination onto $\boldsymbol{p}$ weighted with material properties mat $\Rightarrow$ reflected light towards sensor point $\boldsymbol{q}$
- Phong illumination model
  - Simplified setting
  - Considers light, sensor and normal direction and material properties

Challenges for Realistic Images

- Rendering
  - Computing the entire light transport
  - Understanding simplifications introduced by practical concepts
- Modeling
  - Detailed geometry and material properties
  - Properties of participating media
  - Realistic light sources
- Simulation

Outline

- Organization
- Our research
- Image generation
- Course topics

Course Curriculum

1. Introduction
   - Modeling, rendering, simulation
   - Concepts, challenges, applications
2. Visibility with Ray Tracing
3. Shading
4. Homogeneous coordinates
   - Prerequisite for projection
5. Visibility with projection

Course Curriculum

6. Rasterization
   – Concepts for vertex and fragment processing
7. Curves and surfaces
8. Particle fluids
9. Summary and outlook
   – Test exam
   – Radiosity, Monte Carlo ray tracing, simulation

Summary

- Organization
- Our research
- Image generation
- Course topics