Computer Graphics
Modeling - Rendering – Simulation
Introduction

Matthias Teschner
Computer Graphics

– Modeling – Rendering - Simulation

© Warner Bros.
Scanline VFX
V-Ray
Computer Graphics

– Modeling – Rendering - Simulation

© Double Aye
V-Ray
Computer Graphics

- Light is energy or photons transported along lines
- **Modeling**
  - Light sources generate light, sensor measures / absorbs light, geometry, materials and participating media affect light
- **Rendering**
  - Computation of light transport
- **Simulation**
  - Computation of the dynamics of rigid bodies, deformable objects and fluids
Computer Graphics

Modeling → Rendering → Simulation → Computer Graphics

CGI Making of Share a Coke VFX Breakdown by ARMA.
MAKING OF
“SHARE A COKE”

Music by: Chocolate Puma & Firebeatz
I Can’t Understand (Original Mix)
Application Areas

- Movies / visual effects
- Commercials
- Engineering
- Architecture
- Medical imaging
- Scientific visualization
- Games
- Virtual reality / augmented reality
Outline

– Organization
– Our research
– Image generation
– Summary
Contact

– Matthias Teschner
  Computer Graphics
  University of Freiburg

– Georges-Koehler-Allee 052 / 01-005
teschner@informatik.uni-freiburg.de
https://cg.informatik.uni-freiburg.de/
Graphics Courses

- Key course
  - Pattern recognition and computer graphics
    (modeling, rendering, simulation)

- Specialization courses
  - Advanced computer graphics (global illumination)
  - Simulation in computer graphics
    (deformable and rigid solids, fluids)

- Master project, lab course, Master thesis
  - Simulation track, rendering track
# Seminars / Projects / Theses in Graphics

<table>
<thead>
<tr>
<th>Semester</th>
<th>Simulation Track</th>
<th>Rendering Track</th>
</tr>
</thead>
<tbody>
<tr>
<td>Winter</td>
<td>Key Course</td>
<td>Key Course</td>
</tr>
<tr>
<td></td>
<td>Simulation Course</td>
<td></td>
</tr>
<tr>
<td>Summer</td>
<td>Lab Course - Simple fluid solver</td>
<td>Rendering Course</td>
</tr>
<tr>
<td></td>
<td>Simulation Seminar</td>
<td>Lab Course - Simple raytracer</td>
</tr>
<tr>
<td>Winter</td>
<td>Master Project - PPE fluid solver</td>
<td>Master Project - Monte Carlo raytracer</td>
</tr>
<tr>
<td></td>
<td>Rendering Seminar</td>
<td>Rendering Seminar</td>
</tr>
<tr>
<td>Summer</td>
<td>Master Thesis - Research-oriented topic</td>
<td>Master Thesis - Research-oriented topic</td>
</tr>
</tbody>
</table>
Material – Exam

– Slide sets
– Written exam
– Slides, exercises, solutions and test exam on http://cg.informatik.uni-freiburg.de/teaching.htm
Selected Readings

Exercises

- Four exercises
- Optional
- Introduction to OpenGL >3.0
  - Programming interface for rendering
- Eight parts (two per exercise)
  - Related to rasterization, homogeneous notation, projection, Phong shading (check course curriculum)
- Support
  - Sara Todorovikj <sara.todorovik@gmail.com>
Prerequisites

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

– Organization
– Our research
– Image generation
– Summary
Overview

– Simulation of rigid bodies, deformable objects and fluids
Valley

up to 38M fluid particles interacting with more than 650 rigid bricks, highly viscous mud and an elastic tree
Lagrangian Simulation

Fluid / Elastic object / Rigid object

Set of parcels

\( \mathbf{x}_i^t \)
\( \mathbf{v}_i^t \)

Positions and velocities of parcels \( i \) over time \( t \)
Fluid and Solid Parcels
Parcel Movement for Fluids

Task

\[ \mathbf{v}_i^t \rightarrow \mathbf{v}_{i}^{t+\Delta t} \]
\[ \mathbf{x}_i^t \rightarrow \mathbf{x}_{i}^{t+\Delta t} \]

Governing equations

\[ \frac{\mathrm{d}\mathbf{v}_i^t}{\mathrm{d}t} = \mathbf{a}_i = -\frac{1}{\rho_i^t} \nabla p_i^t + \nu \nabla^2 \mathbf{v}_i^t + \mathbf{g} \]
\[ \frac{\mathrm{d}\rho_i^t}{\mathrm{d}t} = -\rho_i^t \nabla \cdot \mathbf{v}_i^t = 0 \]

Numerics

\[ \nabla p_i^t \approx \sum_j \frac{m_j}{\rho_j} p_j^t \nabla W_{ij}^t \]
\[ \nabla^2 \mathbf{v}_i^t \approx \sum_j \ldots \]

\[ \mathbf{v}_i^{t+\Delta t} = \ldots \]
\[ \mathbf{x}_i^{t+\Delta t} = \ldots \]
Typical Steps of a Fluid Solver

- Neighbors $j$ of $i$
- Predicted velocity
  \[ \mathbf{v}_i^* = \mathbf{v}_i^t + \Delta t \left( \nu \nabla^2 \mathbf{v}_i^t + \mathbf{g} \right) \]
- Pressure
  \[ \nabla \cdot \mathbf{v}_i^* + \nabla \cdot \left( -\Delta t \frac{1}{\rho_i} \nabla p_i^t \right) = 0 \]
- Velocity and position
  \[ \mathbf{v}_i^{t+\Delta t} = \mathbf{v}_i^* - \Delta t \frac{1}{\rho_i} \nabla p_i^t \]
  \[ \mathbf{x}_i^{t+\Delta t} = \mathbf{x}_i^t + \Delta t \mathbf{v}_i^{t+\Delta t} \]
Neighbor Search

- $\sim 10^{14}$ neighbors have to be identified per simulation
- Uniform grid
  - Sorted list
  - Compact hashing
  - 1 million samples: 20 ms
  - 500 million samples: 10 s
- Minimized secondary data structures

Pressure Computation

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

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

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

Applications

Pixar Animation Studios, Emeryville

FIFTY2 Technology, Freiburg

Studio Claudia Comte, Grancy / Berlin

Robotics Innovation Center DFKI, Bremen
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
– Summary
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
Aspects

- Rendering
  - What is visible by the sensor?
    - Rasterization
    - Raytracing
  - Which color / intensity does it have?
    - Local shading models / Phong illumination model
    - 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 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 wavelength $\rightarrow$ spectrum $\rightarrow$ color

$\Phi_\lambda(\lambda_1)$  
$\Phi_\lambda(\lambda_2)$

$\Phi_\lambda(\lambda_3)$  
$\Phi_\lambda(\lambda):$ number of photons per time with a wavelength in a range $\Delta\lambda_i$ around $\lambda_i.$

$$\Phi = \int_{\text{Visible Spectrum}} \Phi_\lambda(\lambda) d\lambda$$

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

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

\[ L(p, \omega_o) = \int_{\Omega} \text{mat}(p, \omega_i, \omega_o) L(p, \omega_i) \cos \theta_i \, d\omega_i \]

Outgoing light into direction \( \omega_o \) is a sum of incident light from all directions weighted with material properties \( \text{mat}(p, \omega_i, \omega_o) \).
Light in Volumes

– Governing equations for light changes along rays through participating media, e.g. haze or fog

– Setting

\[ L(p, \omega) \xrightarrow{ds} L(p, \omega) + dL \]

– Absorption

\[ \frac{dL}{ds} = -\kappa L(p, \omega) \]

– Emission

\[ \frac{dL}{ds} = q(p, \omega) \]

– Out-scattering

\[ \frac{dL}{ds} = -\sigma L(p, \omega) \]

– In-scattering

\[ \frac{dL}{ds} = j(p, \omega) \]
Light Transport

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

Emitted light

No participating media

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

- Raytracing, 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 raytracing
Raytracing 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.

Matrix in homogeneous notation

\[
\begin{pmatrix}
a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \\
\end{pmatrix}
\]
Shading

- Solve $L(p, \omega_o) = \int_{\Omega} \text{mat}(p, \omega_i, \omega_o) L(p, \omega_i) \cos \theta_i \, d\omega_i$ at a surface point $p$ with, e.g., Monte-Carlo raytracing
  - Accumulate all illumination onto $p$ weighted with material properties $\text{mat} \iff$ reflected light towards sensor point $q$
- Phong illumination model
  - Simplified equation
  - Considers light, sensor and normal direction and material properties
Challenges for Realistic Images

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

– Organization
– Our research
– Image generation
– Summary
Course Curriculum

1. Introduction
   – Modeling, rendering, simulation
   – Concepts, challenges, applications
2. Visibility with raytracing
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. Sampling
9. Particle fluids
10. Outlook
    – Radiosity, Monte Carlo raytracing, simulation
Guest Lecture

- Dr. Markus Ihmsen, Co-founder of FIFTY2 Technology
- Modeling, Rendering and Simulation in Industry
- Dec 10, 10-12, 082 00 006
Modeling – Rendering - Simulation

© Spellwork Pictures

Modeling

Rendering

University of Freiburg – Computer Science Department – 50
Modeling – Rendering - Simulation

Animation

Rendering

© Spellwork Pictures