Image Processing and Computer Graphics

Thomas Brox
Matthias Teschner
# Organization

<table>
<thead>
<tr>
<th>Computer Graphics</th>
<th>Image Processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Matthias Teschner</td>
<td>Thomas Brox</td>
</tr>
<tr>
<td><a href="https://cg.informatik.uni-freiburg.de/teaching.htm">Link</a></td>
<td><a href="https://lmb.informatik.uni-freiburg.de/lectures/image_processing/">Link</a></td>
</tr>
</tbody>
</table>
Computer Graphics
Modeling – Rendering – Simulation
Introduction

Matthias Teschner
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

Modeling

Rendering

Simulation

CGMeetup: CGI VFX Breakdown HD "Making of Share a Coke Vfx by ARMA" | CGMeetup. [Youtube]
MAKING OF
“SHARE A COKE”

Music by: Chocolate Puma & Firebeatz
I Can't Understand (Original Mix)

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

<table>
<thead>
<tr>
<th>Semester</th>
<th>Simulation Track</th>
<th>Rendering Track</th>
</tr>
</thead>
<tbody>
<tr>
<td>Winter</td>
<td>Simulation Course</td>
<td></td>
</tr>
<tr>
<td>Summer</td>
<td>Key Course</td>
<td>Key Course</td>
</tr>
<tr>
<td></td>
<td>Lab Course - Simple fluid solver</td>
<td>Lab Course - Simple Ray Tracer</td>
</tr>
<tr>
<td></td>
<td>Simulation Seminar</td>
<td>Rendering Seminar</td>
</tr>
<tr>
<td>Winter</td>
<td>Master Project - PPE fluid solver</td>
<td>Rendering Course</td>
</tr>
<tr>
<td></td>
<td>Rendering Seminar</td>
<td>Master Project - Monte Carlo RT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Simulation 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 and recordings
– Slides, recordings, exercises, solutions and test exam on https://cg.informatik.uni-freiburg.de/teaching.htm
– Written exam
Selected Readings

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
  - Student assistant Philipp Sandweg <screenfun13@web.de>
- Optional
Recommended Prerequisites

– Linear algebra
  – Vector
  – Matrix

– Calculus
  – Differentiation
  – Integration

– Programming language
  – C, ...
Outline

– Organization
– Our research
– Image generation
– Course topics
Lagrangian Simulation

Fluid / Elastic object / Rigid object

Set of parcels

$x_i^t$

$u_i^t$

Positions and velocities of parcels $i$ over time $t$
Fluid and Solid Parcels

Akinci et al., *ACM SIGGRAPH*, 2012.
Parcel Movement for Fluids

**Task**

\[ \begin{align*}
\mathbf{v}_i^t &\rightarrow x_i^t & \quad \mathbf{v}_i^{t+\Delta t} &\rightarrow x_i^{t+\Delta t}
\end{align*} \]

**Governing equations**

\[ \begin{align*}
\frac{d\mathbf{v}_i^t}{dt} &= a_i^t = -\frac{1}{\rho_i^t} \nabla p_i^t + \nu \nabla^2 \mathbf{v}_i^t + \mathbf{g} \\
\frac{d\rho_i^t}{dt} &= -\rho_i^t \nabla \cdot \mathbf{v}_i^t = 0
\end{align*} \]

**Numerics**

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

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

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

Pressure Computation

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

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

\[
\begin{bmatrix}
  a_{11} & a_{12} & \cdots & a_{1n} \\
  a_{21} & a_{22} & \cdots & a_{2n} \\
  \vdots & \vdots & \ddots & \vdots \\
  a_{n1} & a_{n2} & \cdots & a_{nn}
\end{bmatrix}
\begin{bmatrix}
  p_{1}^t \\
  p_{2}^t \\
  \vdots \\
  p_{n}^t
\end{bmatrix}
= \begin{bmatrix}
  s_{1} \\
  s_{2} \\
  \vdots \\
  s_{n}
\end{bmatrix}
\]

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
– 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 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 $\leftrightarrow$ spectrum $\leftrightarrow$ 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.

\[ L(p, \omega_i) \]

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

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
  \[
  L(p_1, \omega) = L(p, \omega) + s \frac{dL}{ds}
  \]

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

- Emission
  \[
  \frac{dL}{ds} = L_e(p, \omega)
  \]

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

- In-scattering
  \[
  \frac{dL}{ds} = L_j(p, \omega)
  \]
Light Transport

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

- 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

\[ L(p, \omega) \]
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.

Transform

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

Matrix in homogeneous notation
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 \( \text{mat} \) \( \Rightarrow \) reflected light towards sensor point \( q \)

- 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