Advanced Computer Graphics

Introduction to Ray Tracing

Matthias Teschner

Computer Science Department
University of Freiburg
Outline

- organization
- introduction
- concepts
- basic components
- syllabus
Course Goals

- ray tracing techniques
- photorealistic rendering
- global illumination techniques

requirements:
- key course in graphics and image processing
- C / C++ / C#
- basics in linear algebra
Contact

- Matthias Teschner
  052 / 01-005
teschner@informatik.uni-freiburg.de
Course Information

- key course
  - pattern recognition and computer graphics (rasterization-based rendering)

- specialization courses
  - advanced computer graphics (ray tracing)
  - simulation in computer graphics (e.g., fluids)

- master project, lab course, Master thesis
  - two tracks: simulation, rendering
## 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>Rasterization Course</td>
<td>Rasterization Course</td>
</tr>
<tr>
<td></td>
<td>Simulation Course</td>
<td></td>
</tr>
<tr>
<td>Summer</td>
<td>Lab Course</td>
<td>Raytracing Course</td>
</tr>
<tr>
<td></td>
<td>- simple fluid solver</td>
<td>Lab Course</td>
</tr>
<tr>
<td></td>
<td>Simulation Seminar</td>
<td>- simple raytracer</td>
</tr>
<tr>
<td>Winter</td>
<td>Master Project</td>
<td>Master Project</td>
</tr>
<tr>
<td></td>
<td>- PPE fluid solver</td>
<td>- Monte Carlo raytracer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Rendering Seminar</td>
</tr>
<tr>
<td>Summer</td>
<td>Master Thesis</td>
<td>Master Thesis</td>
</tr>
<tr>
<td></td>
<td>- research-oriented topic</td>
<td>- research-oriented topic</td>
</tr>
</tbody>
</table>
Course Information

- **key course**
  - pattern recognition and computer graphics (rasterization-based rendering)

- **specialization courses**
  - advanced computer graphics (ray tracing)
  - simulation in computer graphics (animation)

- **master project, lab course, Master thesis**
  - tracks: particle fluids, raytracing
# Projects / Theses in Graphics

<table>
<thead>
<tr>
<th>Semester</th>
<th>Fluid Track</th>
<th>Raytracing Track</th>
</tr>
</thead>
<tbody>
<tr>
<td>Winter</td>
<td>Rasterization Course Simulation Course</td>
<td>Rasterization Course</td>
</tr>
<tr>
<td>Summer</td>
<td>Lab Course - simple fluid solver</td>
<td>Raytracing Course</td>
</tr>
<tr>
<td>Winter</td>
<td>Master Project - PPE fluid solver</td>
<td>Master Project - Monte Carlo raytracer</td>
</tr>
<tr>
<td>Summer</td>
<td>Master Thesis - research-oriented topic</td>
<td>Master Thesis - research-oriented topic</td>
</tr>
</tbody>
</table>

- more information on https://cg.informatik.uni-freiburg.de/student.htm
Material

- slide sets on
  http://cg.informatik.uni-freiburg.de/teaching.htm
Material

- Matt Pharr, Greg Humphreys
  Physically Based Rendering
  Morgan Kaufmann
  http://www.pbrt.org

- Kevin Sufferin
  Ray Tracing from the Ground Up
  A K Peters
  http://www.raytracegroundup.com
Material

- Philip Dutre, Kavita Bala, Philippe Bekaert
  Advanced Global Illumination
  A K Peters
  http://www.advancedglobalillumination.com

- Peter Shirley, R. Keith Morley
  Realistic Ray Tracing
  A K Peters
Tutorials / Exercises

- every second Monday, starting on May 15
  - check web page for changes
- practical exercises
  - development of ray tracing components
  - check web page for information and example frameworks
Outline

- organization
- introduction
- concepts
- basic components
- syllabus
Ray Tracing - Concept

- tracing rays of light through a scene to compute the radiance that is perceived by a sensor
- tracing a path from a camera through a pixel position of a virtual image plane to compute the color of an object that is visible along the path

[Wikipedia: Ray Tracing]
Ray Tracing - Motivation

- light is modeled as geometric rays
  - travels in straight lines (e.g., no diffraction / bending)
  - travels at infinite speed (steady state of light is computed)
  - is emitted by light sources
  - is absorbed or scattered / reflected at surfaces
- radiance
  - characterizes strength and direction of radiation / light
  - is measured by sensors
  - is computed in computer-generated images
  - is preserved along lines in space
  - does not change with distance
Ray Tracing - Capabilities

- reflection
- refraction
- soft shadows
- caustics
- diffuse interreflections
- specular interreflections
- depth of field
- motion blur

[sean.seanie, www.flickr.com]
rendered with POVray 3.7
Photorealistic Rendering - History

- rasterization
  - 1965: rasterized lines (Bresenham)
  - 1967: rasterized flat-shaded polygons (Wylie)
  - 1971: Gouraud shading
  - 1973: Phong illumination model
  - 1974: texture mapping (Blinn)
  - 1974: depth buffer (Catmull)
  - 1975: Phong shading
  - 1977: shadow volumes (Crow)
  - 1978: shadow maps (Williams)
Photorealistic Rendering - History

- ray tracing
  - 1968: viewing and shadow rays, non-recursive (Appel)
- recursive ray tracing
  - 1980: ideal reflection, refraction (Whitted)
- rendering equation
  - 1986: general description of light distribution in a scene (Kajiya)
    - $L_o(p, \omega_o) = L_e(p, \omega_o) + \int_{2\pi} f_r(p, \omega_i, \omega_o) L_i(p, \omega_i) \cos \theta_i \, d\omega_i$
    - arbitrary global illumination effects can be considered
- distribution ray tracing
  - 1986: Monte-Carlo evaluation of integrals (Cook)
    - approximately solves the rendering equation
Ray Tracing vs. Rasterization

- **rasterization**
  - given a set of viewing rays and a primitive, efficiently compute the subset of rays hitting the primitive
  - loop over all primitives
  - no explicit representation of rays

- **ray tracing**
  - given a viewing ray and a set of primitives, efficiently compute the subset of primitives hit by the ray
  - loop over all viewing rays
  - explicit representation of rays
Ray Tracing vs. Rasterization

- **rasterization**
  - simple and well-established algorithms
  - popular in interactive applications
  - efficient parallel processing of primitives and fragments
  - independent processing of primitives and fragments
does not account for global illumination effects, e.g. shadows and interreflections

- **ray tracing**
  - natural incorporation of numerous visual effects
  - no special algorithms for, e.g.,
    - shadows (additional geometry or additional rendering passes)
    - transparency (depth sorting)
  - trade-off between quality and performance
Ray Tracing - Challenges

- efficient ray shooting
  - ray shooting algorithms build spatial data structures to accelerate ray shooting queries
  - dynamic scenes are more challenging compared to static scenes

- optimal number of rays
  - per pixel
    - for antialiasing
  - at ray-object intersections
    - for interreflections
    - soft shadows
    - approximate evaluation of the rendering equation

- optimal recursion depth
Ray Tracing - Applications

- visual effects in movies and commercials
  - major software packages have built-in ray tracers, e.g. Maya, 3ds Max (Autodesk), Houdini (Side Effects Software)
- visualization of architectural design
  - consideration of realistic indoor and outdoor illumination
- automotive design
- flight and car simulators
- computer games
Ray Tracing - Software

- mental ray (NVIDIA ARC)
- Maxwell Render (Next Limit Technologies)
- Brazil (SplutterFish)
- Arnold (Solid Angle)
- POV-Ray
- Blender
- pbtrt
Ray Tracing - Applications

- all images rendered with mental ray

Spiderman 3 (Columbia Pictures)

Bioshock 2 (Game trailer by Blur studio)
Ray Tracing - Applications

- all images rendered with mental ray

Mies van der Rohe Farnsworth House
(Artist Alessandro Prodan)

Delta Tracing

[www.mentalimages.com]
Ray Tracing - Applications

- all images rendered with mental ray

zerone cgi GmbH and Daimler AG

[www.mentalimages.com]
Ray Tracing - Applications

- video is rendered with mental ray (cooperation with Pixar Animation Studios)
Ray Tracing - Applications

- video is rendered with mental ray
Ray Tracing - Applications

- cooperation with Fifty2 Technology GmbH

PreonLab: Drive Through
Ray Tracing - Applications

- cooperation with Fifty2 Technology GmbH
Outline

- organization
- introduction
- concepts
- basic components
- syllabus
Ray Tracing

- ray generation
- ray traversal
- intersection
- shading
- frame buffer

Viewing rays return a radiance value.
Shadow rays return an occlusion value which is also a radiance.
Ray Tracing

Recursive Ray Tracing

- ray generation
- ray traversal
- intersection
- shading
- frame buffer

Viewing rays return a radiance value.
Shadow rays return an occlusion / radiance value.
Reflection and refraction rays return a radiance value.
Recursive Ray Tracing

Distribution Ray Tracing
(Stochastic Ray Tracing)

- generates more than one (randomly perturbed)
  - viewing ray per pixel
  - reflection / refraction ray at a surface point
  - shadow ray at a surface point
- examples
  - distributing shadow rays over an area light source for soft shadows
  - distributing reflection rays over a solid angle about the exact reflection direction to blur the reflection
  - perturbing ray origins per pixel to enable depth-of-field effects
  - distributing rays per pixel over time to get motion blur effects
  - distributing rays over the hemisphere of a surface point to capture the incident radiance at this point (Monte Carlo integration for solving the rendering equation)
Distribution Ray Tracing

Outline

- organization
- introduction
- concepts
- basic components
- syllabus
Components

- camera
  - generates viewing rays
- light distribution
  - location and radiant intensity of light sources
- ray-object intersection
  - with additional information, e.g. normal
- visibility
  - of light sources
- surface scattering model
  - describes how light interacts with a surface
- recursion
  - important for reflections on shiny surfaces
- ray propagation
  - variation of radiance in, e.g., fog or smoke
Camera

- a camera simulator generates viewing rays
- pinhole camera with a virtual image plane (near plane) in front of the pinhole
- pinhole is referred to as the eye
- for a position on the image, a camera simulator generates rays along which light is known to contribute to that position, e.g.
  - a ray from the eye through the image position
  - a ray that considers one or multiple lenses
**Light Distribution**

- determining the amount of light energy arriving at the differential area around the intersection point
- therefore, geometric and radiometric distribution of light has to be known
  - for emitted light from point light sources
  - for emitted light from area light sources
  - for reflected light for object surfaces
Ray-Object Intersection

- determine whether a ray intersects an object
- determine the first intersection (closest to the ray origin)
- determine further geometric information at the intersection, e.g.
  - surface normal
  - partial derivatives of position and normal with respect to the local surface parameterization
- efficient implementations heavily rely on spatial data structures
Visibility

- determine whether a light source is visible from a surface point to be shaded
- shadow rays are casted from the object to the light source
- if the distance to the first ray-object intersection along this ray is shorter than the distance to the light source, the surface point is in shadow
Surface Scattering

- computes the radiance scattered back along a viewing ray
- from previous components, we have
  - ray-object intersection and further geometric information
  - information on incident lighting
- we further know appearance properties, e.g.
  - a local illumination model
  - a Bidirectional Reflectance Distribution Function BRDF
    (how much light is reflected from an incoming direction to an outgoing direction)
Recursion

- recursively invoke the ray-tracing components if appropriate
- if, e.g., a viewing ray hits a mirror
  - the viewing ray can be reflected at the mirror
  - the ray-tracing routine is applied to the reflected ray
  - the resulting radiance is considered as additional illumination of the mirror
- to approximately solve the rendering equation,
  - various rays are generated that sample the hemisphere above the surface (Monte Carlo integration)
Ray Propagation

- can consider participating media, e.g., smoke, fog, dust
- in vacuum, radiance along a ray does not change
- in presence of participating media, light can be attenuated or extinguished by scattering it in different directions
- participating media can be characterized by its transmittance
Outline

- organization
- introduction
- concepts
- basic components
- syllabus
Course Topics

- aspects that affect efficiency and quality of the rendering
  - transformations
  - primitives
  - ray traversal / ray shooting
  - sampling / antialiasing
  - radiometric quantities
  - rendering equation
  - Monte Carlo integration