Advanced Computer Graphics Introduction

Matthias Teschner

UNI FREIBURG

Computer Graphics

Modeling - Rendering - Simulation

Outline

- Organization
- Concepts
- Applications
- History

Graphics Courses

- Key course
  - Image processing and computer graphics (modeling, rendering, simulation)
- Specialization courses
  - Advanced computer graphics (global illumination)
  - Simulation in computer graphics (solids and fluids)
- B.Sc. / M.Sc. project, lab course, B.Sc. / M.Sc. thesis
  - Simulation track, rendering track
  - By appointment per email, semester-aligned

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

Course Goals

- Concepts for image synthesis / global illumination approaches
- Governing equation and solution techniques
  - Radiometric quantities
  - Rendering equation
  - Radiosity
  - Monte Carlo ray tracing
- Requirements:
  - Key course in graphics and image processing

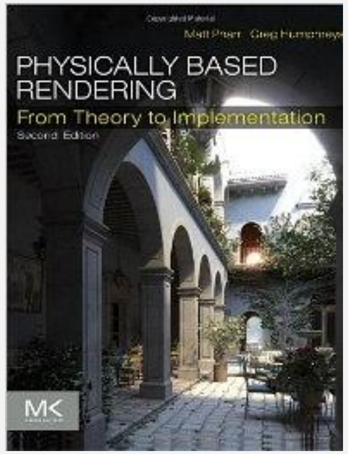
UNI FREIBURG

Material

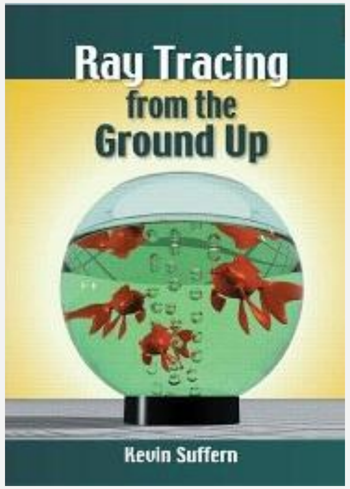
- Slide sets and video recordings on https://cg.informatik.uni-freiburg.de/teaching.htm

Material

– Matt Pharr, Wenzel Jakob, Greg Humphreys
  Physically Based Rendering
  Morgan Kaufmann
  http://www.pbr-book.org/

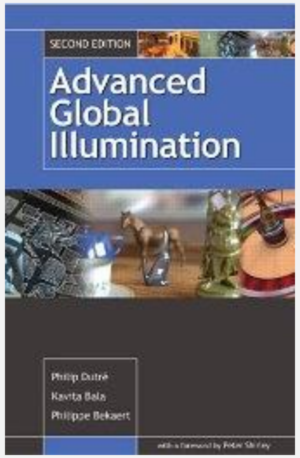


– Kevin Suffern
  Ray Tracing from the Ground Up
  A K Peters

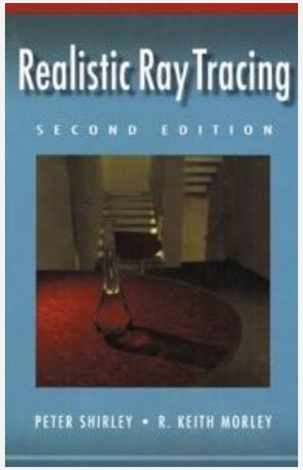


Material

– Philip Dutre, Kavita Bala, Philippe Bekaert
Advanced Global Illumination
A K Peters



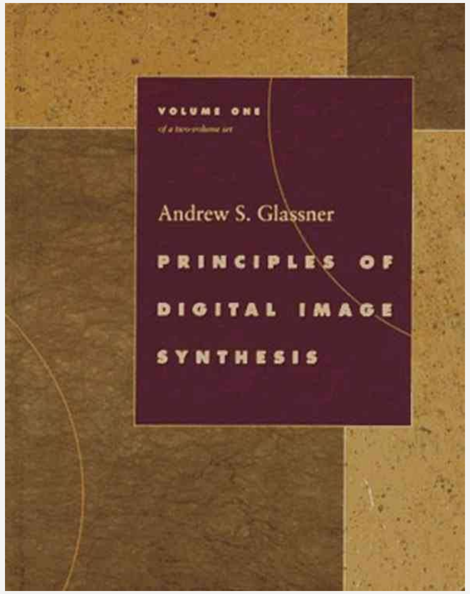
– Peter Shirley, R. Keith Morley
Realistic Ray Tracing
A K Peters



Material

– Andrew Glassner
  Principles of Digital Image Synthesis

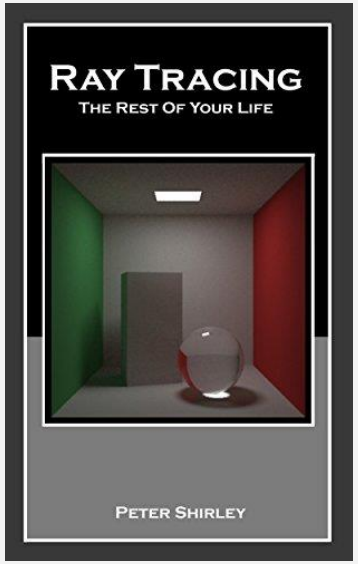
  Available online from
  http://www.realtimerendering.com



Material

– Peter Shirley
  Raytracing in one Weekend
  Raytracing – The Next Week
  Raytracing – The Rest of Your Life

  Available online from
  https://raytracing.github.io/



Exercises

- Development of ray tracing components
- Check web page for information and example frameworks
- Voluntary

UNI FREIBURG

Exam

- Written
- Based on slide sets and recordings
- Relevant material will be summarized
- Text exam on our web page

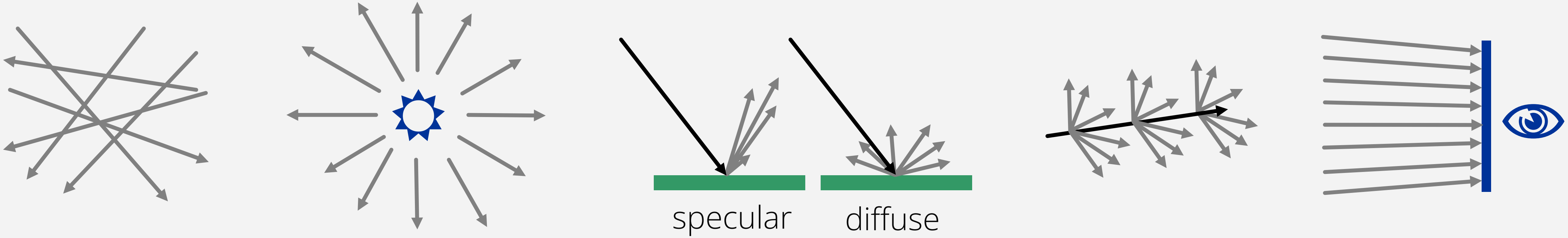
Outline

- Organization
- Concepts
- Applications
- History
- Summary

Light

- Modeled as energy parcels / photons that travel
  - Along geometric rays (radiance L)
  - 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

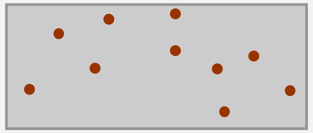
Incoming light is scattered and absorbed at surfaces

Participating media scatters and absorbs light

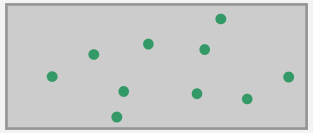
Sensors absorb light

Color

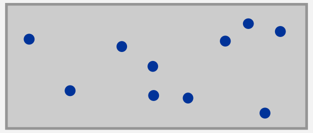
- Photons are characterized by a wavelength within the visible spectrum
- Distribution of wavelengths ⇨ spectrum ⇨ color



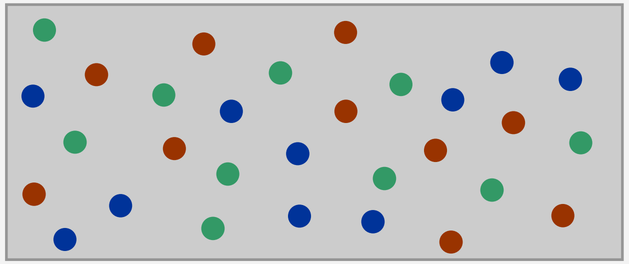
$\Phi_\lambda(\lambda_1)$



$\Phi_\lambda(\lambda_2)$



$\Phi_\lambda(\lambda_3)$



$$\Phi = \int_{\text{VisibleSpectrum}} \Phi_\lambda(\lambda) d\lambda$$
$$\approx \sum_i \Phi_\lambda(\lambda_i) \Delta\lambda_i$$
$$\approx \Phi_{\text{red}} \Delta\lambda + \Phi_{\text{green}} \Delta\lambda + \Phi_{\text{blue}} \Delta\lambda$$

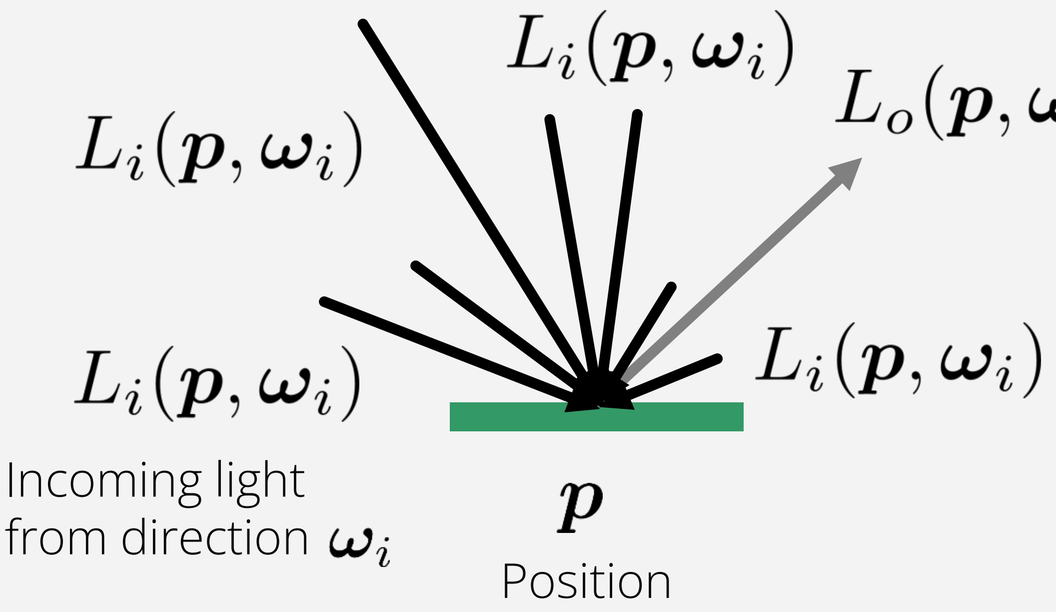
$\Phi_\lambda(\lambda)$: number of photons per time with a wavelength in a range $\Delta\lambda_i$ around λ_i.

Governing Equations

- Light transport is governed by surfaces and by participating media
- Interactions of light with surfaces and volumes are described by governing equations
  - Rendering equation
  - Volume rendering equation

Light at Surfaces – Rendering Equation

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

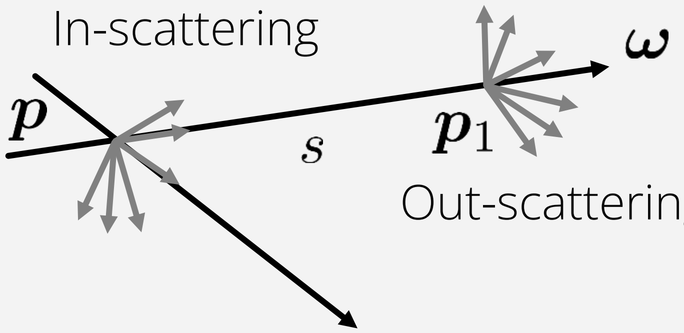


$$L_o(\boldsymbol{p}, \boldsymbol{\omega}_o) = \int_\Omega f_r(\boldsymbol{p}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_o) L_i(\boldsymbol{p}, \boldsymbol{\omega}_i) \cos\theta_i \mathrm{d}\omega_i$$

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

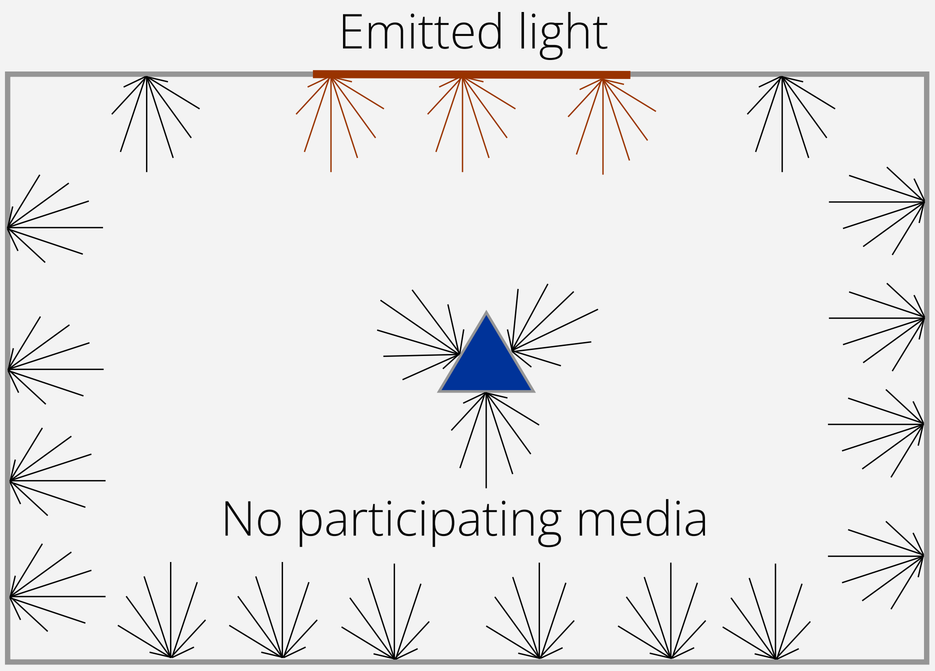
Light in Volumes

- 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

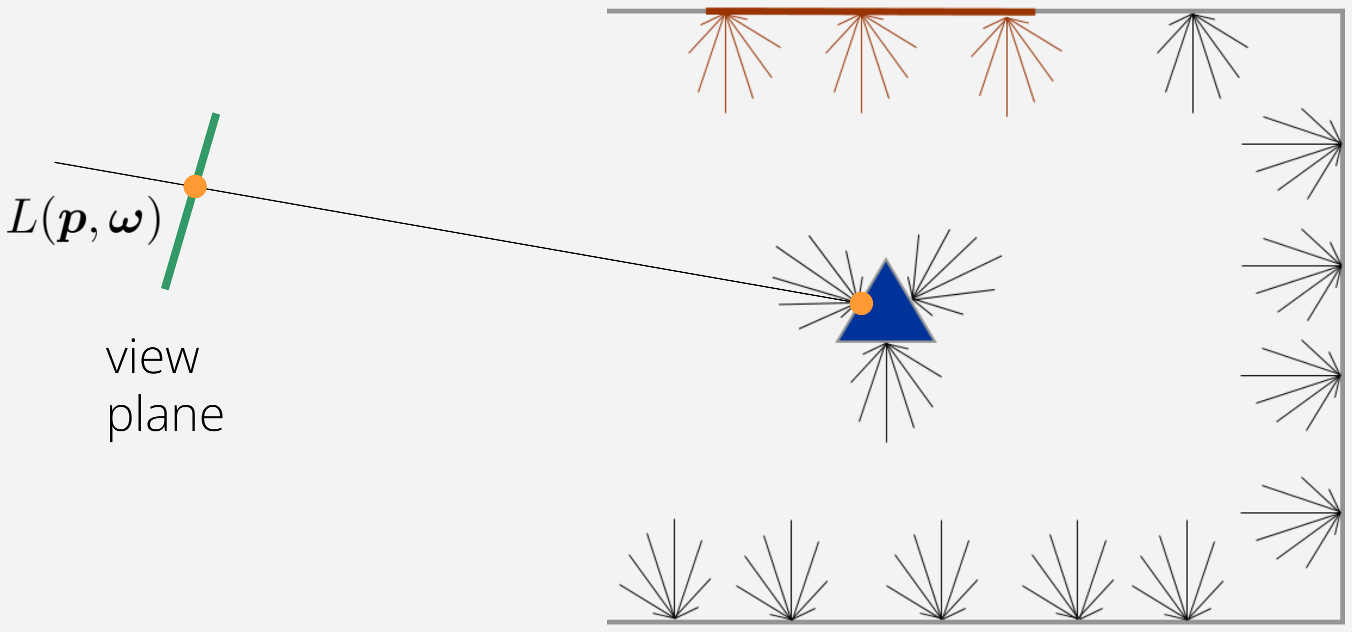
– Governing equations enable the computation of light at all points in space into all direction

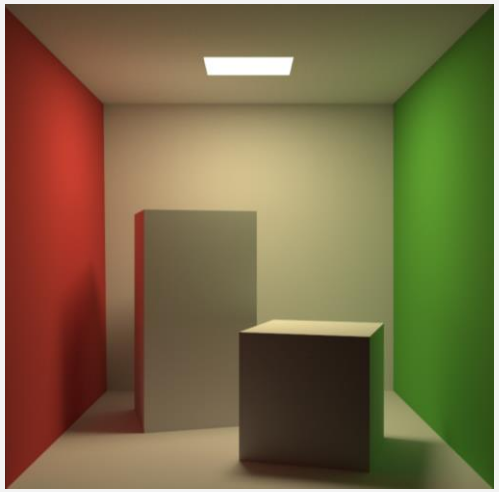


Reflected light due to material properties

Rendering of the Result

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





Example: Cornell box

Rendering Algorithms

– Goal: Solving the entire light transport in a scene as accurate as possible

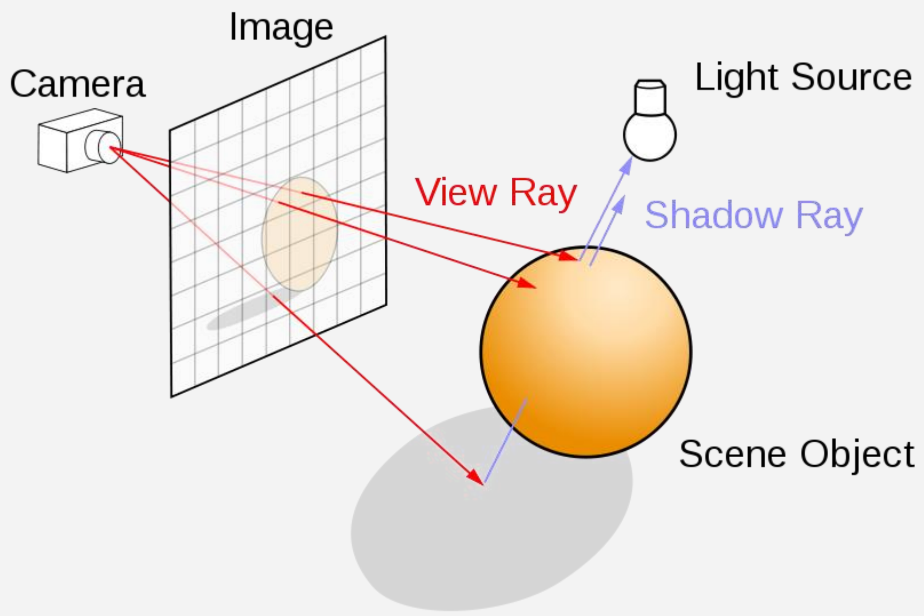
UNI FREIBURG

Radiosity

- Computes reflected light at all surface points into all directions
- Typical 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

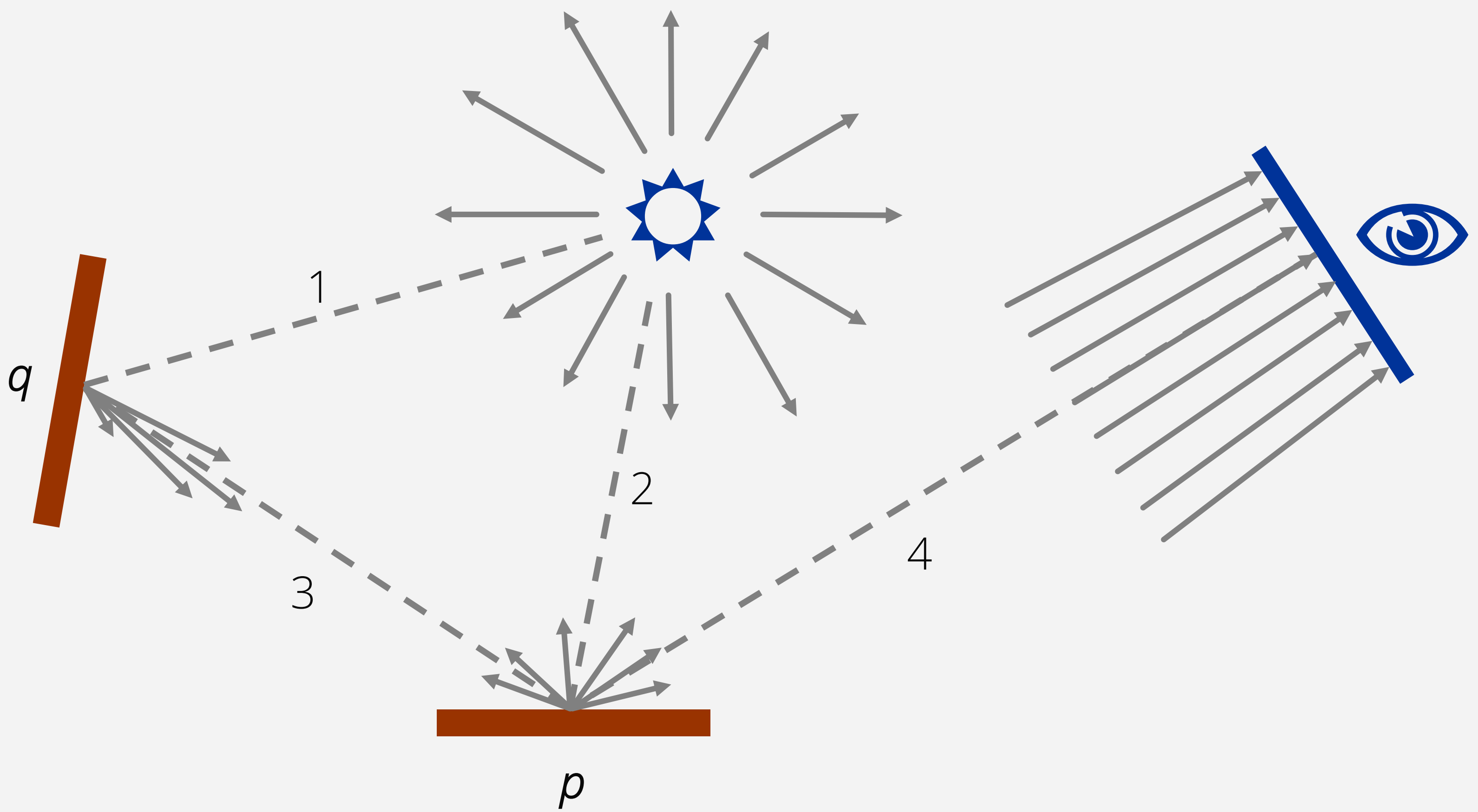
Ray Tracing

- Computation of light transport along selected line segments / rays
- Minimal setup
  - Consideration of rays from the scene towards the sensor (viewing rays)
  - Consideration of rays from the light source towards visible scene elements (shadow rays)



[Wikipedia: Ray Tracing]

Ray Tracing - Challenge



Path 4
Computation of outgoing radiance from surface towards camera is the **main goal of a ray tracer**

Path 1, 2, 3 …
Incoming / outgoing radiance at all other paths is required to compute radiance at path 4

Path 3
Two surfaces illuminate each other. Outgoing radiance from *q* towards *p* depends on outgoing radiance from *p* towards *q* which depends on …

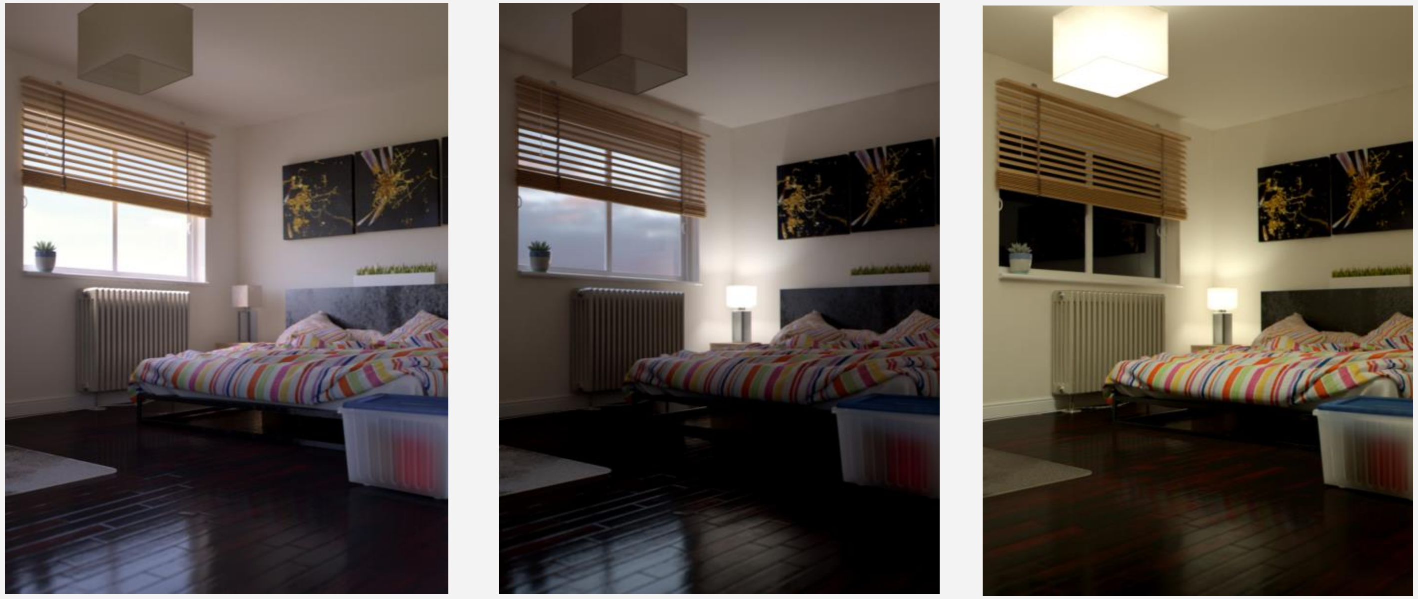
Towards Realistic Images

- Accurate modeling of the light interaction with surfaces and participating media
- Parameterizing realistic light sources and materials
- Computing the light transport for as many rays as possible
- In case of limited resources, choose relevant rays with larger radiances

UNI FREIBURG

Towards Realistic Images

- Capturing all direct and indirect illumination from all directions
  - Less realistic images consider few and simple light sources



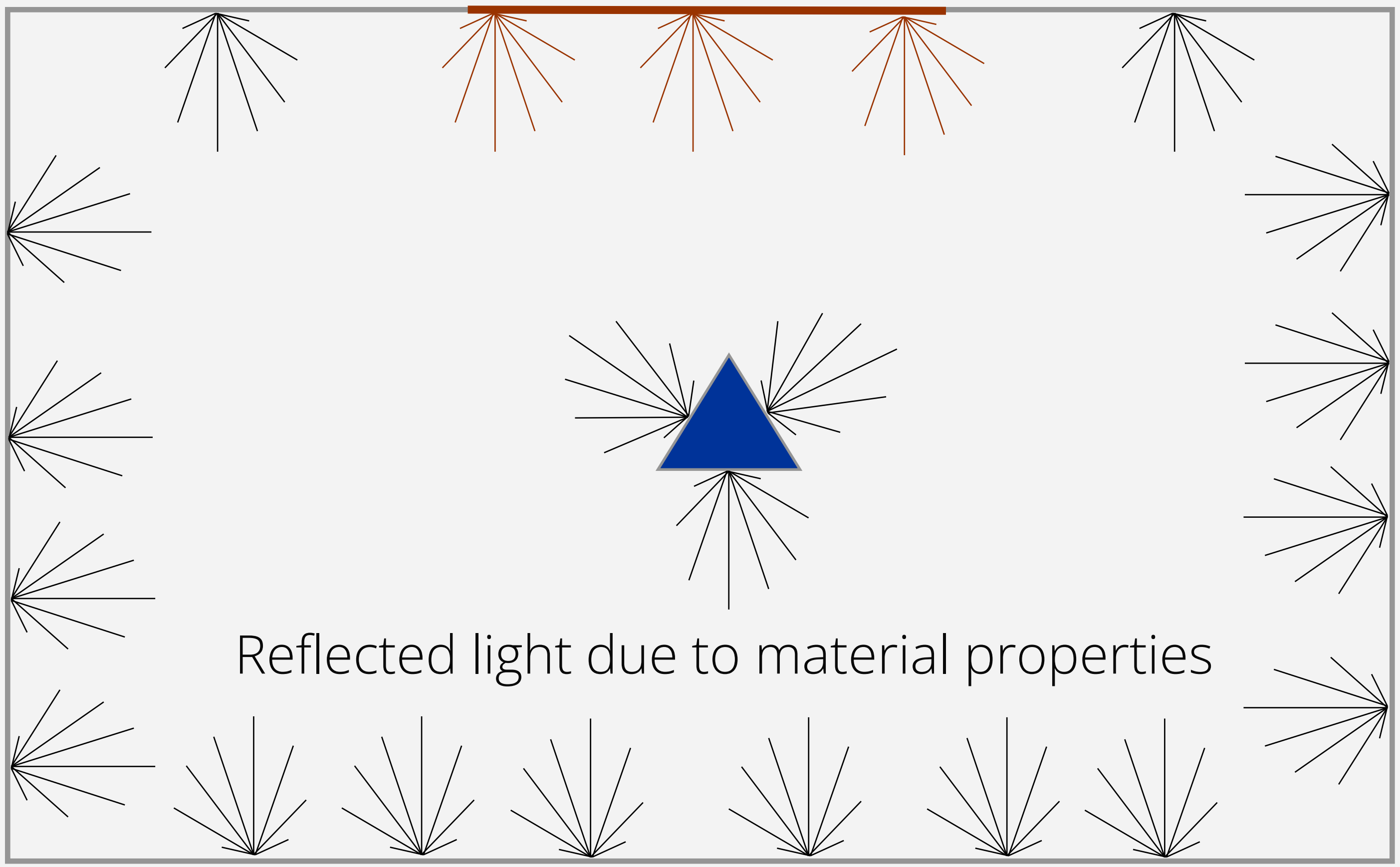
https://imgur.com/gallery/MXbNt

Towards Realistic Images

- Realistic reflection properties of materials
  - Surfaces are not perfectly diffuse or specular

Next Limit / Maxwell Render
http://support.nextlimit.com/display/maxwelldocs/IOR+files

Towards Realistic Images



Outline

- Organization
- Concepts
- Applications
- History
- Summary

Areas

- Movies and commercials
- Architecture
- Automotive
- Flight and car simulators
- Computer games

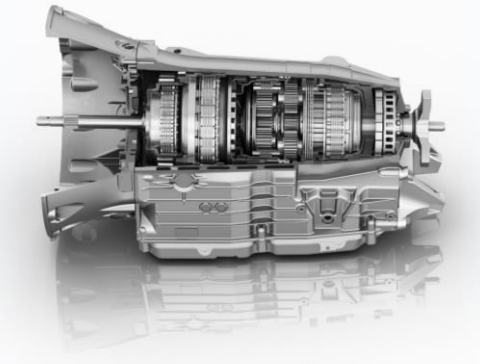
Architecture

Mies van der Rohe Farnsworth House
(Artist Alessandro Prodan)

Delta Tracing

[www.mentalimages.com]

Automotive


zerone cgi GmbH and Daimler AG [www.mentalimages.com]

Commercials

– Spellwork Pictures

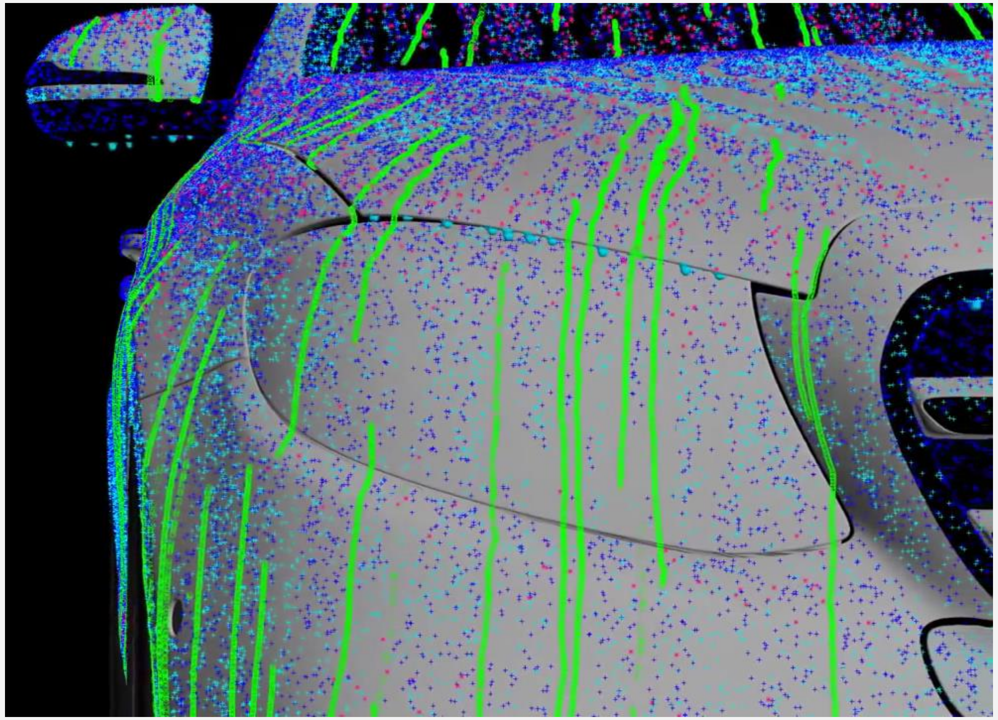
Modeling

Rendering

UNI FREIBURG

Commercials

– Spellwork Pictures



Animation

Animation + Rendering

Commercials

– Spellwork Pictures

Outline

- Organization
- Concepts
- Applications
- History
- Summary

Rasterization

1965: Rasterized lines (Bresenham)

1967: Rasterized flat-shaded polygons (Wylie)

1973: Phong illumination model

1974: Depth buffer (Catmull)

1977: Shadow volumes (Crow)

1978: Shadow maps (Williams)

Ray Tracing

- Ray casting

  1968: viewing and shadow rays, non-recursive (Appel)
- Recursive ray tracing

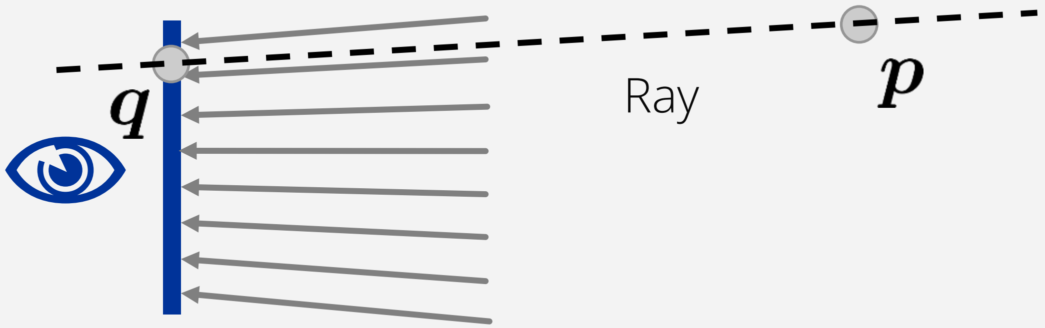
  1980: ideal reflection, refraction (Whitted)
- Rendering equation

  1986: general description of light interaction at surfaces (Kajiya): $L_o(\boldsymbol{p}, \boldsymbol{\omega}_o) = L_e(\boldsymbol{p}, \boldsymbol{\omega}_o) + \int_\Omega f_r(\boldsymbol{p}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_o) L_i(\boldsymbol{p}, \boldsymbol{\omega}_i) \cos\theta_i \mathrm{d}\omega_i$
- Stochastic ray tracing

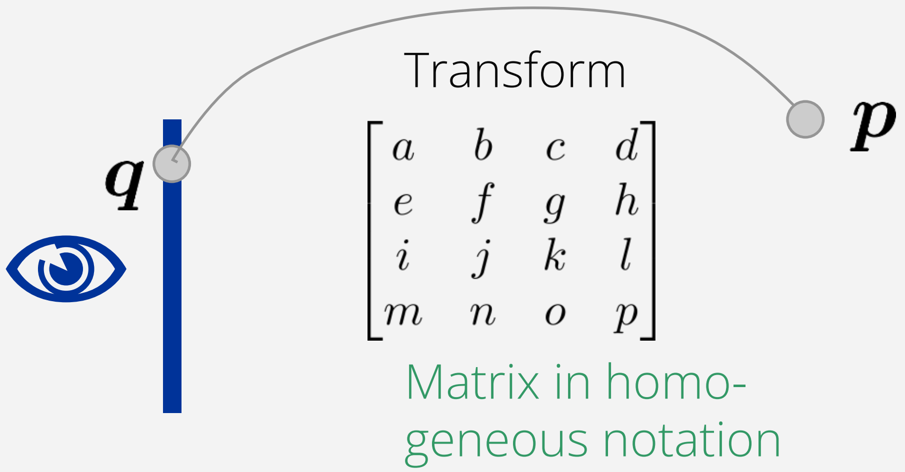
  1986: Monte-Carlo integration for approximately solving the rendering equation (Kajiya), e.g. path tracing

Ray Casting and Rasterization

– Solve the visibility problem



Ray Casting computes ray-scene intersections to estimate *q* from *p*.



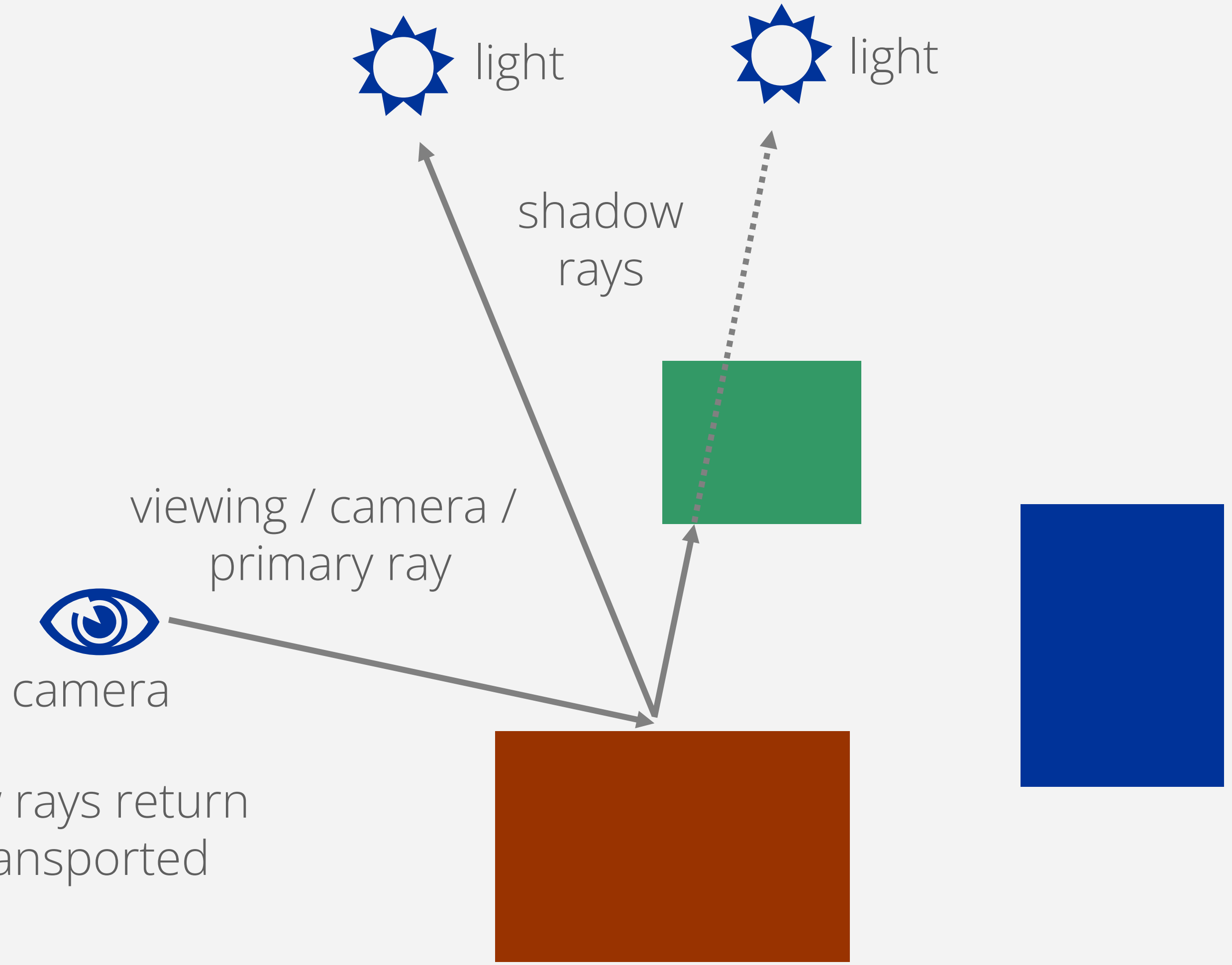
Rasterizers apply transformations to *p* in order to estimate *q*. *p* is projected onto the sensor plane.

Ray Tracing and Rasterization

- Ray tracing
  - Can potentially compute entire light transport in a scene
  - Natural incorporation of numerous visual effects with unified concepts
  - Trade-off between quality and performance
- Rasterization
  - Focus on light transport along viewing rays
  - Specialized realizations of global illumination effects
  - Well-established, parallelizable algorithms
  - Popular in interactive applications

Some Ray Tracing History

- Ray generation
- Ray traversal
- Intersection
- Shading
- Frame buffer

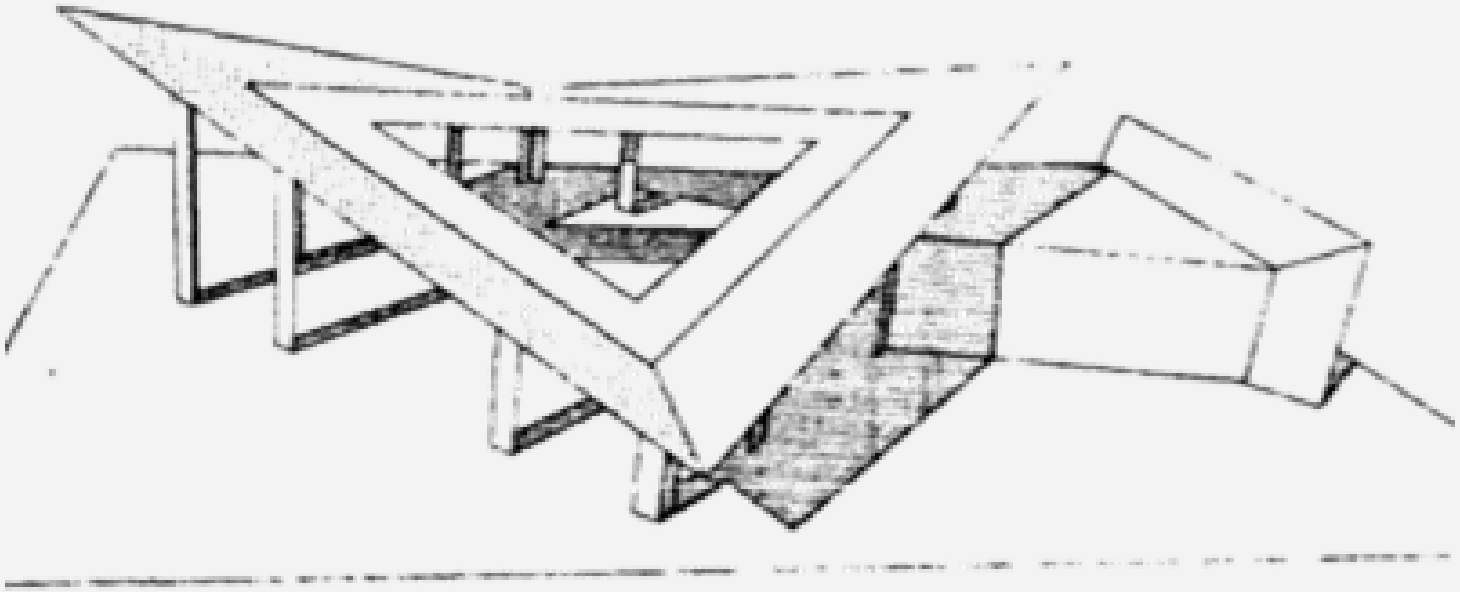


Viewing and shadow rays return a radiance value / transported light along the ray.

UNI FREIBURG

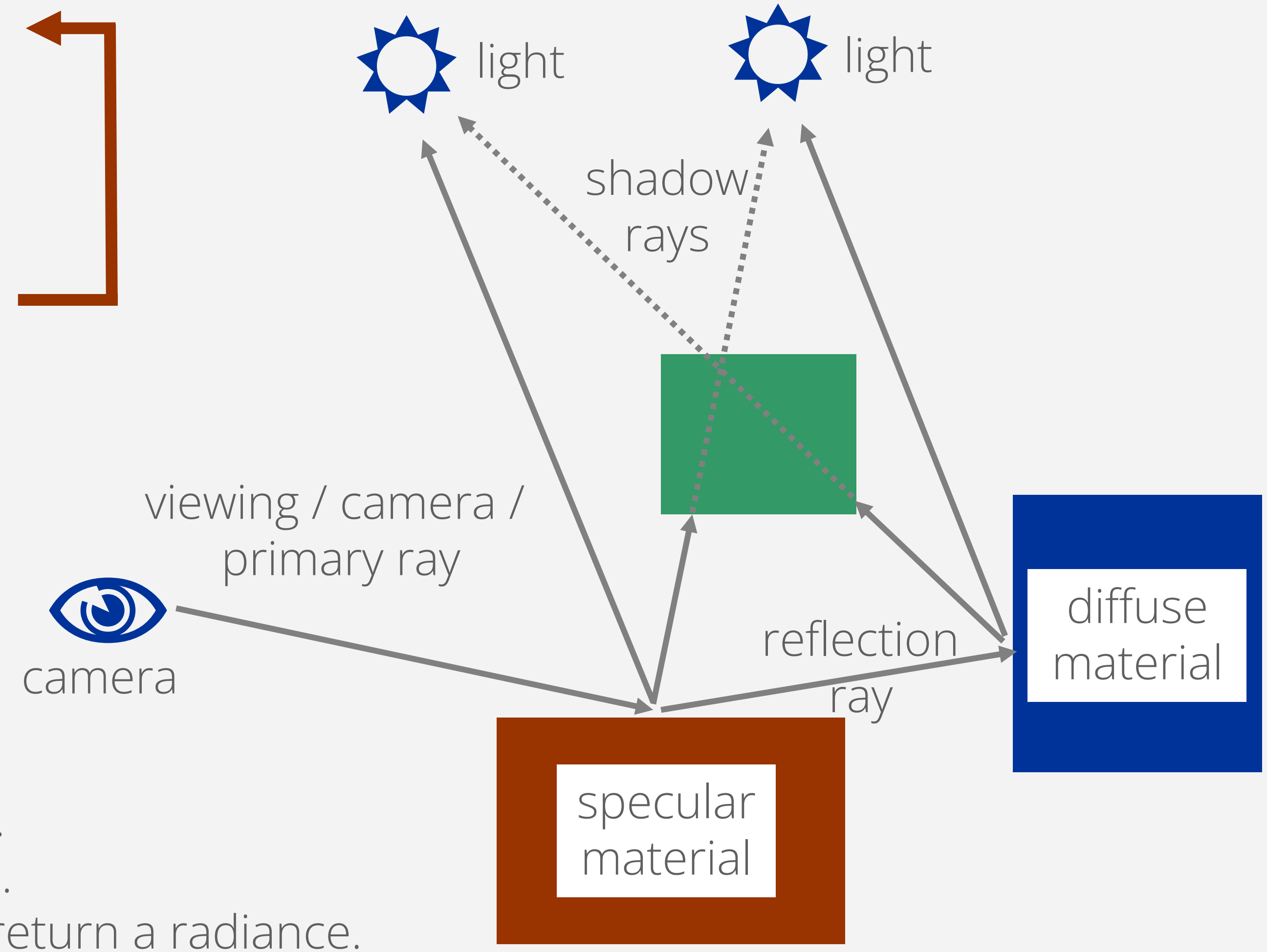
Some Ray Tracing History

– Arthur Appel: Some techniques for shading machine renderings of solids, 1968.



Recursive Ray Tracing

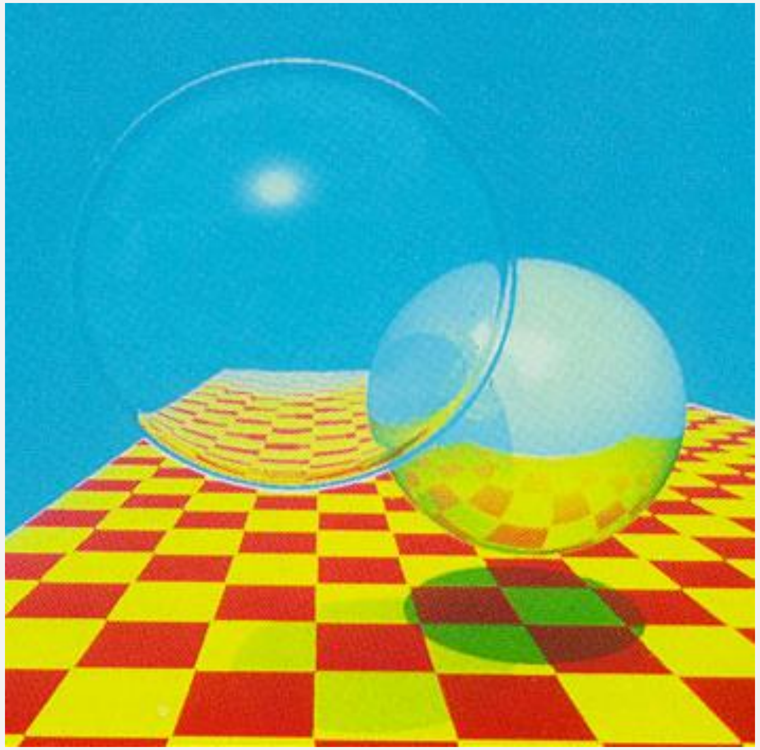
– Ray generation
– Ray traversal
– Intersection
– Shading
– Frame buffer



Viewing rays return a radiance.
Shadow rays return a radiance.
Reflection and refraction rays return a radiance.

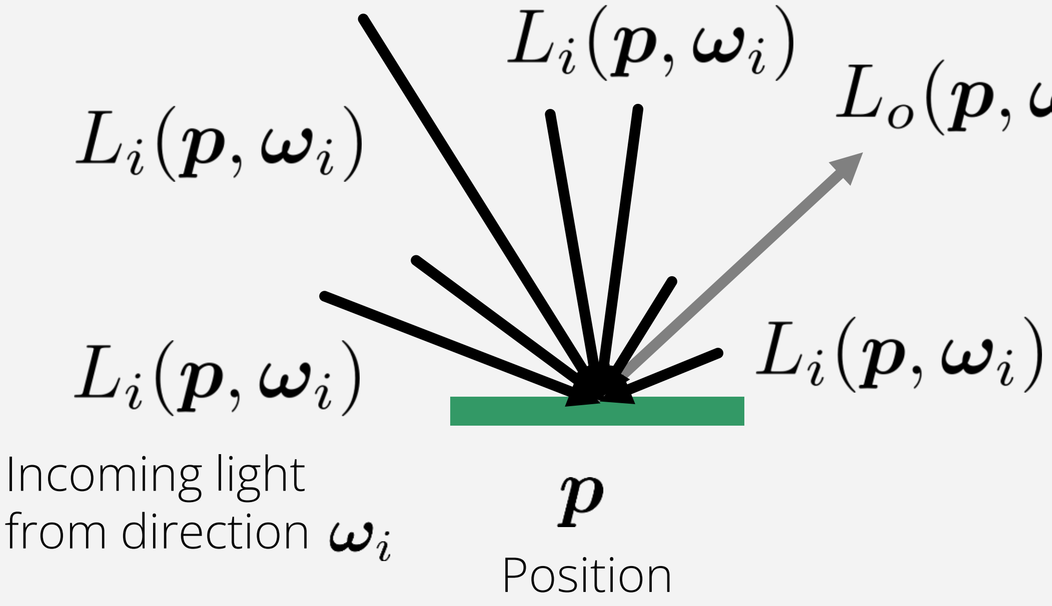
Recursive Ray Tracing

– Turner Whitted: An Improved Illumination Model for Shaded Display, 1980.



Stochastic Ray Tracing

– Consider randomly sampled reflection / refraction rays to approximately solve the rendering equation

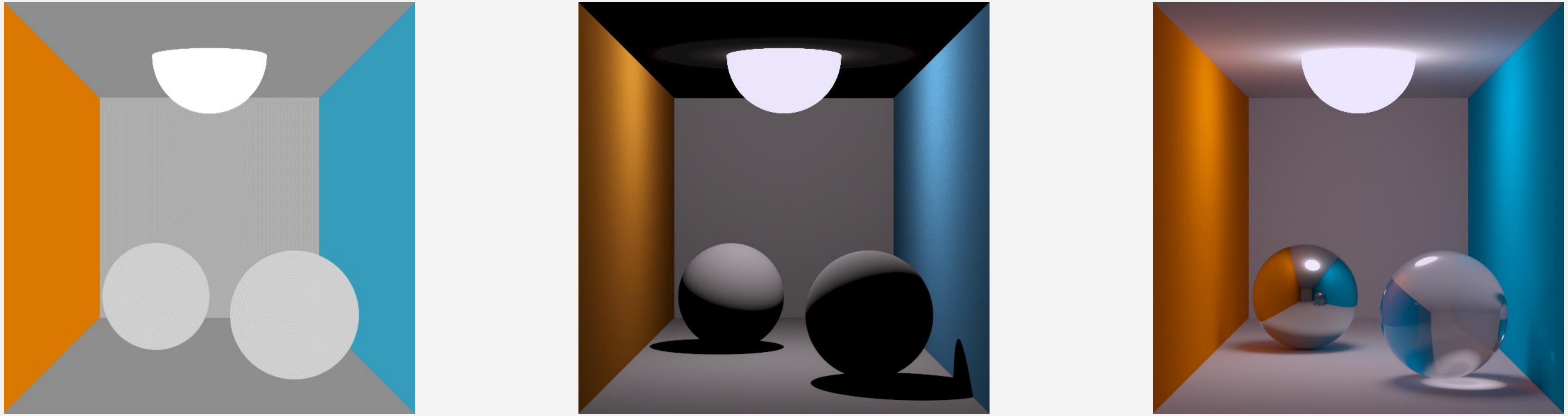


$$L_o(\boldsymbol{p}, \boldsymbol{\omega}_o) = \int_\Omega f_r(\boldsymbol{p}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_o) L_i(\boldsymbol{p}, \boldsymbol{\omega}_i) \cos\theta_i \mathrm{d}\omega_i$$

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

Ray Tracing

– More light transport paths → more visual effects, realism and improved accuracy



Viewing rays

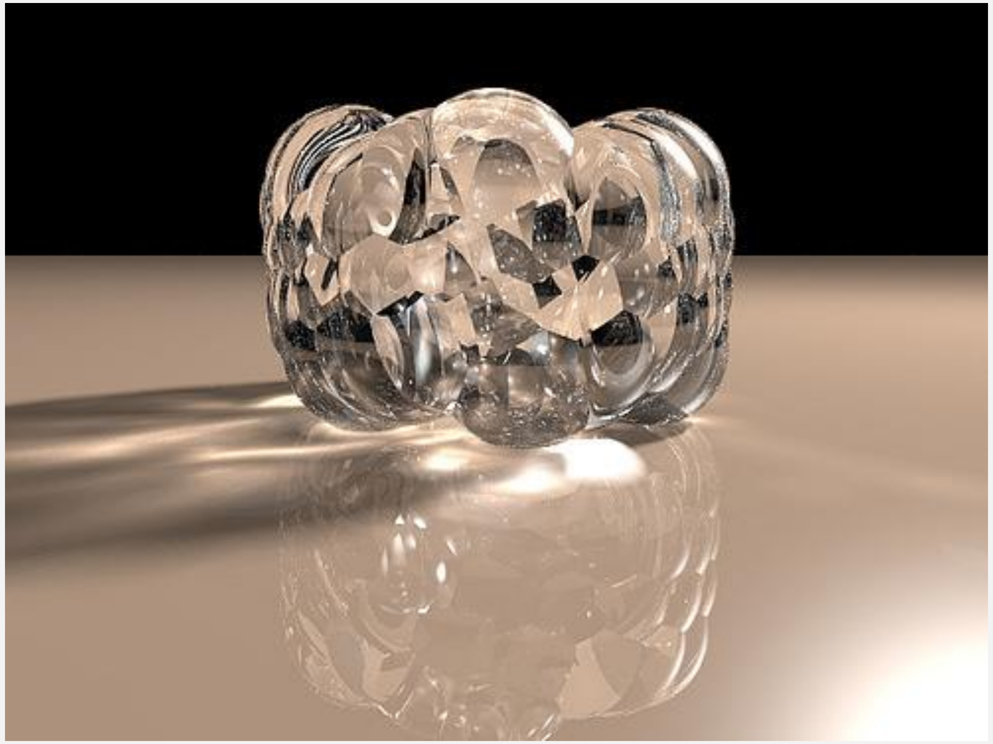
Viewing and shadow rays

Recursive stochastic ray tracing

[Thomas Kabir: Wikipedia Raytracing]

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

Ray Tracing - Challenges

- Ray-primitive intersections
  - Spatial data structures for accelerated ray traversal
  - Dynamic scenes are particularly challenging
- Number of rays (quality vs. costs)
  - More rays typically improve the rendering quality
- Recursion depth (quality vs. costs)
  - Dependent on the recursion depth, effects are captured or not, e.g. transparency or caustics

Outline

- Organization
- Concepts
- Applications
- History
- Summary

Aspects for High-quality Image Synthesis

- Radiometric quantities
- Reflection properties of surfaces / materials
- Rendering equation
- Radiosity
- Monte Carlo ray tracing
- Ray-primitive intersections (see key course)
- Data structures for ray traversal

Announcement

- Monday, 10:15, Advanced Computer Graphics
- Monday, 16:15, Simulation in Computer Graphics
- Wednesday, 10:15, Rendering Seminar
- Wednesday, 12:15, Animation Seminar
- Thursday, 10:15, Proseminar Graphik
- Tuesday next week, 12:15, Simulation Tutorial
- Tuesday next week, 14:15, Rendering Tutorial
- No tutorials tomorrow