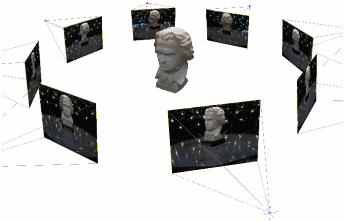
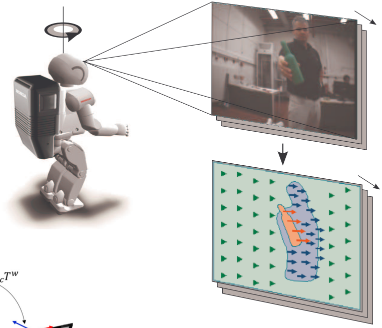
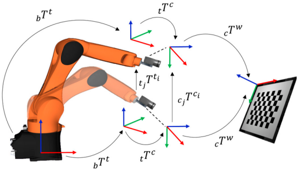
3D Machine Vision

Introduction







Technical University of Applied Sciences Würzburg-Schweinfurt

Building Rome in a Day

Large Scale Structure and Motion Reconstruction

- University of Washington & Microsoft Research (2009)
- huge number of different views from different uncalibrated cameras
- 150.000 images, San Marco square from 14,000 images, reconstructing over 4.5 million 3D points
- `https://grail.cs.washington.edu/rome/`

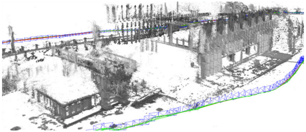
Challenges

- different image resolutions
- large variation of baselines
- parallel computation

LSD-SLAM: Large-Scale Direct Monocular SLAM

Large-Scale Self Localization and Mapping

- Technical University of Munich (2014)
- video stream of one calibrated monocular camera
- trajectories of about 500m length
- `https://cvg.cit.tum.de/research/vslam/lsdslam`



Challenges

- fast movements, large camera rotations, large distance changes
- motion blur, rolling-shutter-effects
- real-time processing

Technical University of Applied Sciences Würzburg-Schweinfurt

Automotive Visual Odometry

Visual Odometry for driver assistance systems

- Technical University of Darmstadt (2017)
- video stream of a calibrated stereo camera
- trajectory of about 2km length
- `own videos!`

Challenges

- large speed range,
  a lot of moving objects
- absolute positioning,
  error accumulation (drift)
- real-time processing

3D Machine Vision

Visual Localization

Visual Pose Estimation for Robots

- Technical University of Darmstadt (2021)
- one image of a calibrated camera
- design of visual fiducial markers
- `own videos!`

Challenges

- very precise pose estimation
- dependency of precision of pose on the pose itself, the number of points and their distribution in space
- real-time processing

3D Machine Vision

Technical University
of Applied Sciences
Würzburg-Schweinfurt

Medical Engineering – Dental Reconstruction

3D reconstruction based on 2D images

- Dentsply Sirona (Bensheim)
- hand-guided image acquisition
- acoustic navigation
- fast data processing

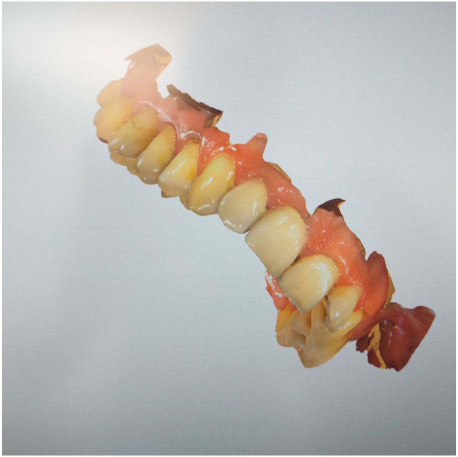
challenges

- high-precision measurement
- photorealistic rendering
- online reconstruction

Images created and provided by dentist Dr. Marco Scherg, Lohr am Main

3D Machine Vision

Medical Engineering – Dental Reconstruction



Example of a surface reconstruction with photorealistic rendering

3D Machine Vision

Medical Engineering – Dental Reconstruction

See video provided by dentist Dr. Marco Scherg, Lohr am Main

3D Machine Vision

Content of Lecture

3D Camera Systems

1. 3D Imaging Techniques
2. 3D Data Structures
3. 3D Data Processing

Basics of Feature Matching

1. Correspondence Problem
2. Features & Correlation
3. Optical Flow
4. Scale Space
5. Descriptors

3D Reconstruction

1. Epipolar Geometry
2. Stereo Vision
3. Stereo Camera Calibration
4. Multiple-View Structure & Motion

3D Pose Estimation

1. Hand-Eye-Calibration
2. PnP Problem
3. 3D Camera Motion

3D Machine Vision

Introduction

General Information

- Organisatorial Stuff
- Literature

Introduction to 3D Machine Vision

- Categories of 3D Vision
- Excursion: Computer Graphics

General Information

- E-Learning

  `https://elearning.thws.de/course/view.php?id=27140`

- Lecturer: Volker Willert

  Campus II: 9.1.09

  E-Mail: `volker.willert@thws.de`

  Telefone: 09721-940-8598

- Appointments: by arrangement

Course of the event

Combined lecture & exercise

Expiration

- start: today Wednesday, 03/20/2024
- end: Wednesday, 07/10/2023
- duration: 16 weeks (excluding holidays)
- number: 32 units, 1.5 hours each
- times/rooms: Wednesday 14:15-15:45, Campus II: 9.E.25
- times/rooms: Wednesday 16:00-17:30, Campus II: 9.E.25

Exam

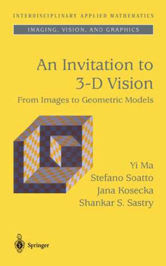
Exam relevant material

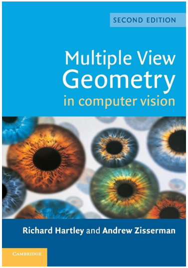
- slide contents are relevant to the exam
- contents of the exercises are relevant to the exam
- blackboard notes are relevant to the exam
- type of examination: written exam
- Date of examination: to be announced!
- Exam duration: 90 minutes
- Auxiliary material: 4 pages hand-written formulary

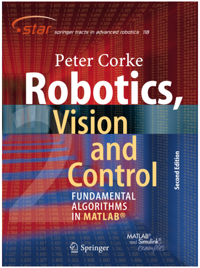
Relevant Literature

Technical University
of Applied Sciences
Würzburg-Schweinfurt

Consolidation of the lecture contents







1. Yi Ma, Stefano Soatto, Jana Kosecka und Shankar S. Sastry, *An Invitation to 3-D Vision - From Images to Geometric Models*, Springer, 2003.
2. Bernd Jähne, *Digitale Bildverarbeitung*, Sechste Auflage, Springer, 2005.
3. Richard Hartley & Andrew Zisserman, *Multiple View Geometry in Computer Vision*, Second Edition, Camebridge University Press, 2003.
4. Peter Corke, *Robotics, Vision and Control*, First Edition, Springers, 2011.

3D Machine Vision

Terms & Application Fields

3D Reconstruction of the environment

- Multi-View Reconstruction
- Structure from Motion
- Visual SLAM (Self Localization and Mapping)

3D Reconstruction of Pose & Motion

- Visual Localization
- Visual Odometry (dead reckoning)
- Feature/Object Tracking
- Visual Servoing (visual control)

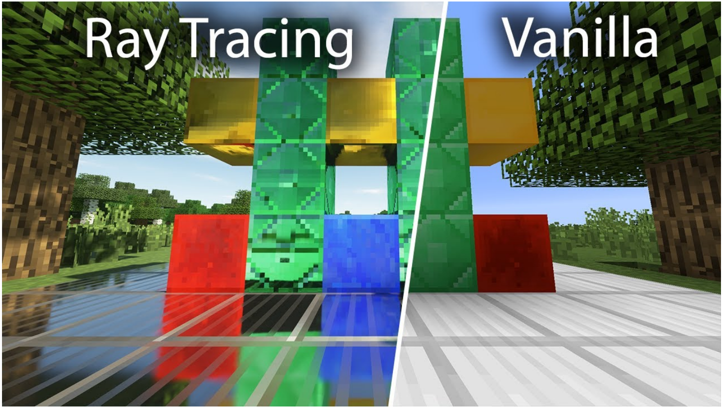
3D Machine Vision

Categories of 3D Vision

- 3D Machine Vision
  - Reconstruction 3D Sensor
    - 3D Pose
    - 3D Motion (Twist)
  - Reconstruction 3D Environment
    - Depth Images
    - Volume Images

Flying out of the box

Ray tracing in computer games – Minecraft



Quelle: `https://i.ytimg.com/vi/WLmYfFV2n60/maxresdefault.jpg`

Flying out of the box

Driving Simulator – Real Video vs. Car Maker Rendering

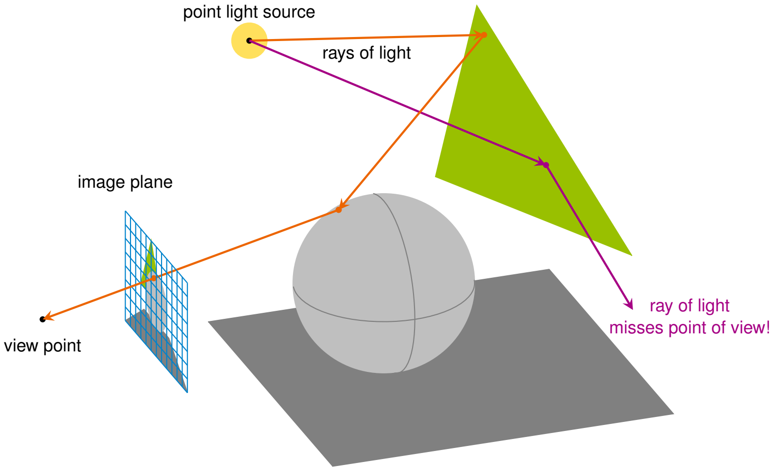
- What is missing in terms of realistic lighting?

- no shadow cast
- no light reflections from other objects
- no reflections, no transparency

Solution: Ray Tracing

Ray Tracing

Idea – tracking light rays

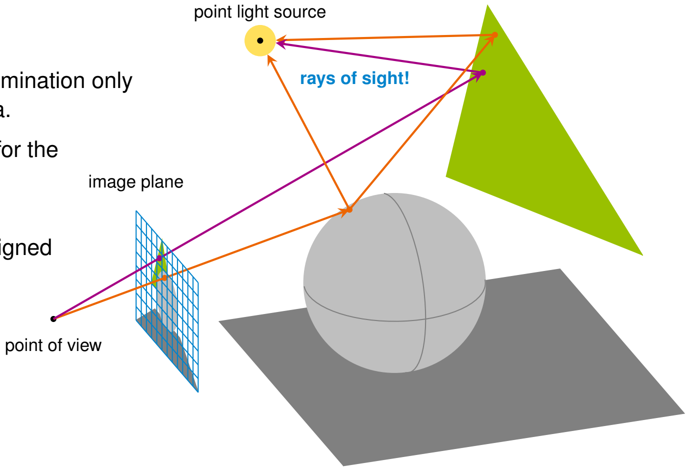


Backward Ray Tracing

Implementation – Tracking sight rays

Advantages

- Calculation of illumination only in the visible area.
- Calculation only for the specified image resolution.
- Each pixel is assigned a color value

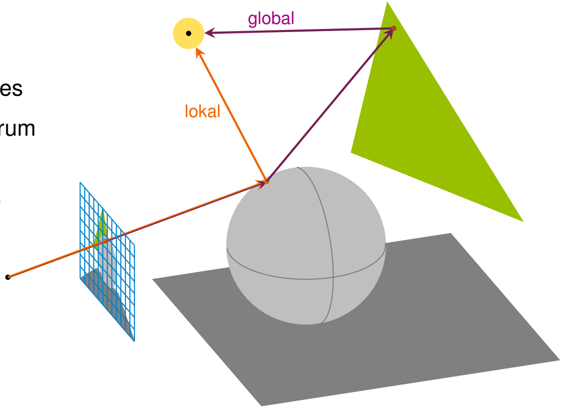


Ray Tracing

Modelling – restrictive assumptions

Modeling the propagation of light

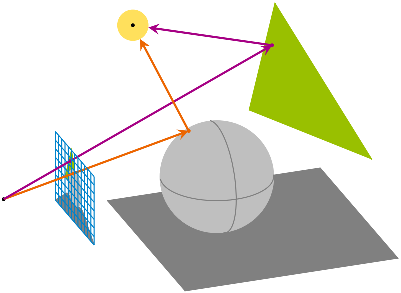
- observer = pinhole camera
- light sources = point light sources
- light = light rays, discrete spectrum
- local illumination
  = specular and diffuse surfaces
- global illumination
  = ideally smooth surfaces



Ray Tracing

Basic Algorithm – Ray Tracing ≡ Ray Casting

```
for all pixel do
  compute ray of sight
  compute intersections with objects
  if ray of sight hits object then
    set pixel to surface color
  else
    set pixel to background color
  end if
end for
```



Ray Tracing

Modelling – Rays of Sight

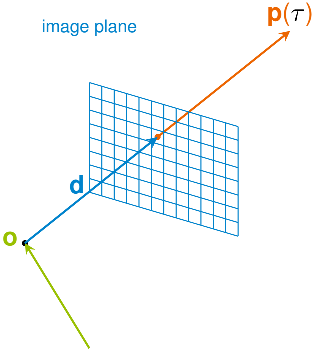
Parametric ray equation

$\mathbf{p}(\tau) = \mathbf{o} + \tau \cdot \mathbf{d}$

Parameter $\tau \geq 1$

3D point of view $\mathbf{o}$

3D direction vector $\mathbf{d}$



Ray Tracing

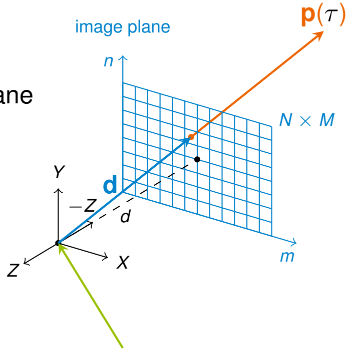
Computation – Direction Vector

Direction vector

$\mathbf{d} = \left[d_x, d_y, d_z\right]^\top$

Distance: point of view – image plane

d



Ray Tracing

Computation – Direction Vector

Ratios

$$\frac{(r-l)}{(d_x-l)} = \frac{M}{(m+0.5)}$$

analogous

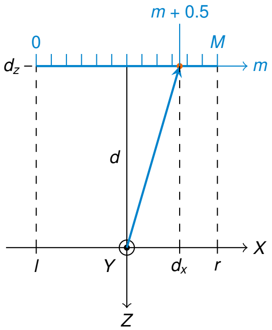
$$\frac{(t-b)}{(d_y-b)} = \frac{N}{(n+0.5)}$$

Coordinates of direction vector

$$d_x = \frac{1}{M}(r-l)(m+0.5)+l$$

$$d_y = \frac{1}{N}(t-b)(n+0.5)+b$$

$$d_z = -d$$



If world frame = eye frame

$\longrightarrow \mathbf{o} = [0,0,0]^\top$

Ray Tracing

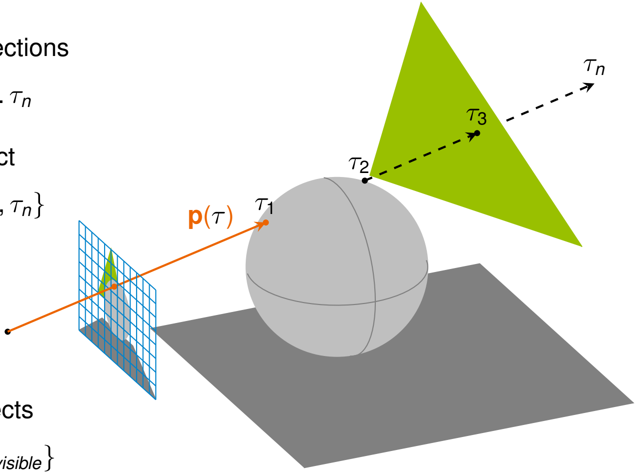
Computation – Intersections & Overlaps

Parameters of the intersections

$1 \leq \tau_1 < \tau_2 < \tau_3 < \ldots \tau_n$

Condition for visible object

$\tau_{visible} = \min\{\tau_1, \tau_2, \tau_3, \ldots, \tau_n\}$



Condition for hidden objects

$\{\tau_{n,hidden}\} = \{\tau_n \,|\, \tau_n > \tau_{visible}\}$

Ray Tracing

Computation – Intersection with Triangle

Parametric plane equation

$$\mathbf{f}(\beta, \gamma) = \mathbf{a} + \beta \cdot (\mathbf{b} - \mathbf{a}) + \gamma \cdot (\mathbf{c} - \mathbf{a})$$

Parametric ray equation

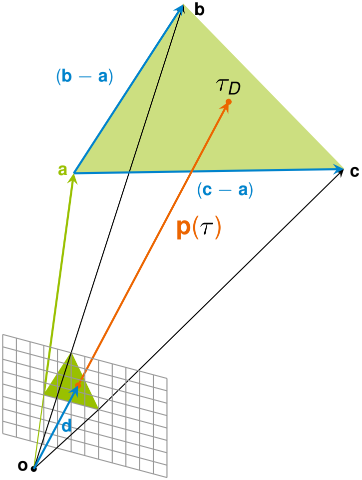
$$\mathbf{p}(\tau) = \mathbf{o} + \tau \cdot \mathbf{d}$$

Condition for intersection $\mathbf{p}(\tau_D)$

$$\mathbf{p}(\tau) \stackrel{!}{=} \mathbf{f}(\beta, \gamma)$$

equals a linear equation system with three unknowns τ, β, γ

$$\mathbf{o} + \tau \cdot \mathbf{d} = \mathbf{a} + \beta \cdot (\mathbf{b} - \mathbf{a}) + \gamma \cdot (\mathbf{c} - \mathbf{a})$$



Ray Tracing

Computation – Intersection with Triangle

Rearranging the linear equations leads to

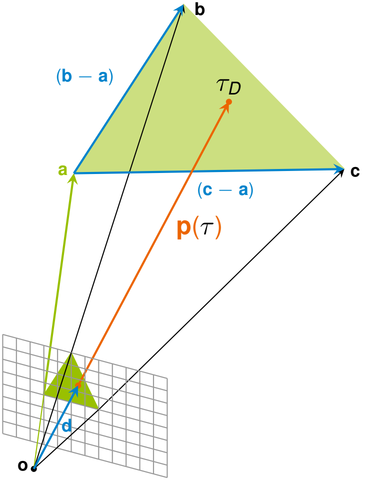
$$\tau \cdot \mathbf{d} + \beta \cdot (\mathbf{a} - \mathbf{b}) + \gamma \cdot (\mathbf{a} - \mathbf{c}) = \mathbf{a} - \mathbf{o}$$

and can be rewritten in matrix form

$$\begin{bmatrix} \mathbf{d} & (\mathbf{a} - \mathbf{b}) & (\mathbf{a} - \mathbf{c}) \end{bmatrix} \begin{bmatrix} \tau \\ \beta \\ \gamma \end{bmatrix} = \begin{bmatrix} \mathbf{a} - \mathbf{o} \end{bmatrix}$$

or more compactly

$$\mathbf{G}\,\mathbf{t} = \mathbf{g}$$



Ray Tracing

Computation – Intersection with Triangle

The solution to the matrix equation

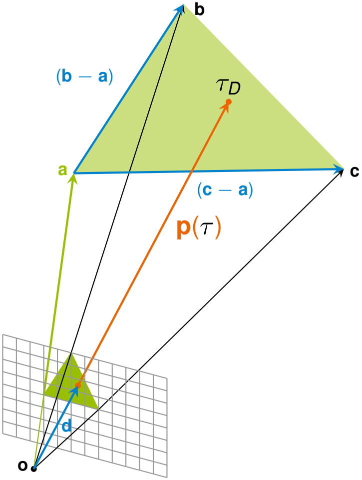
$\mathbf{G}\,\mathbf{t} = \mathbf{g}$

is given via the inverse matrix $\mathbf{G}^{-1}$

$$\begin{bmatrix} \tau \\ \beta \\ \gamma \end{bmatrix} = \mathbf{t} = \mathbf{G}^{-1}\mathbf{g}$$

An intersection point exists if the following conditions are met

$$\left.\begin{array}{l} \beta > 0, \quad \gamma > 0, \\ \beta + \gamma < 1, \end{array}\right\} \rightarrow \tau = \tau_D$$



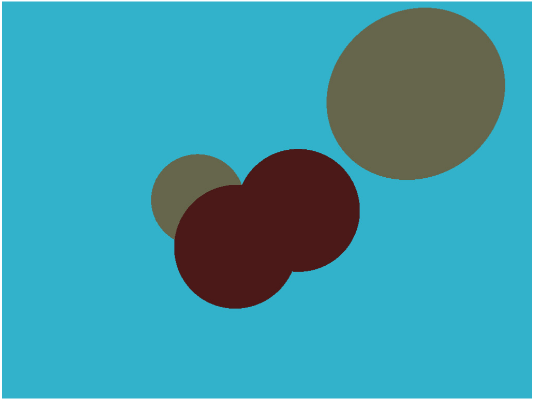
Ray Tracing

Example – Ray Tracing $\equiv$ Ray Casting

✓ occlusions
× local illumination
× shadows

global illumination

× reflective surfaces
× transparent objects

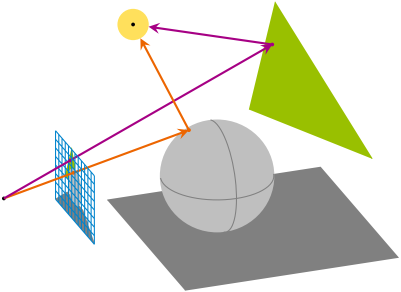


Quelle: `https://github.com/ssloy/tinyraytracer/`

Ray Tracing

Basic algorithm – Ray Tracing + local illumination

```
for all pixel do
  compute ray of sight
  compute intersections with objects
  if ray of sight hits an object then
    compute local illumination
    (e.g. Phong-Shading)
  else
    set pixel to background color
  end if
end for
```



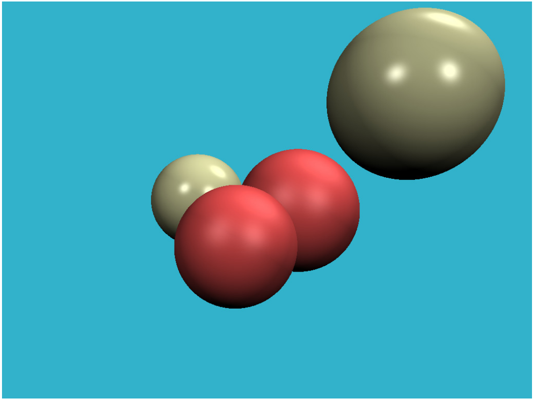
Ray Tracing

Example – local illumination

- ✓ occlusions
- ✓ local illumination
- × shadows

global illumination

- × reflective surfaces
- × transparent objects



Quelle: `https://github.com/ssloy/tinyraytracer/`

Ray Tracing

Basic Algorithm + Shadows

```
for all pixel do
  compute ray of sight
  compute intersections with objects
  if ray of sight hits an object then
    send shadow ray
    if shadow ray hits object then
      set pixel to black
    else
      compute surface normal
      compute local illumination
      set pixel to computed color
    end if
  end if
end for
```

Ray Tracing

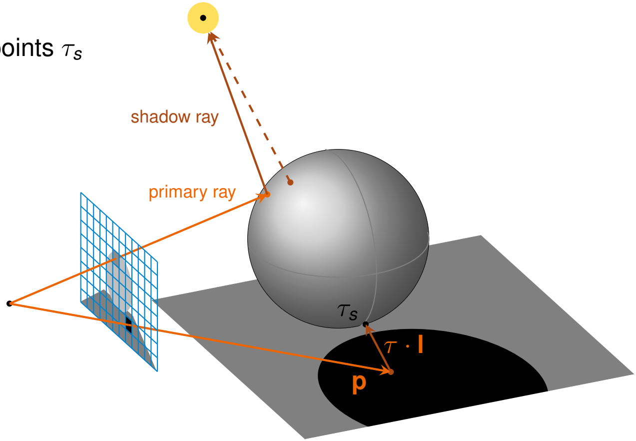
Computation – Shadows

Calculate intersection points τ_s

of each shadow ray

$\mathbf{s}(\tau) = \mathbf{p} + \tau \cdot \mathbf{l}$

with each object



Technical detail
$\tau \geq \epsilon > 0$

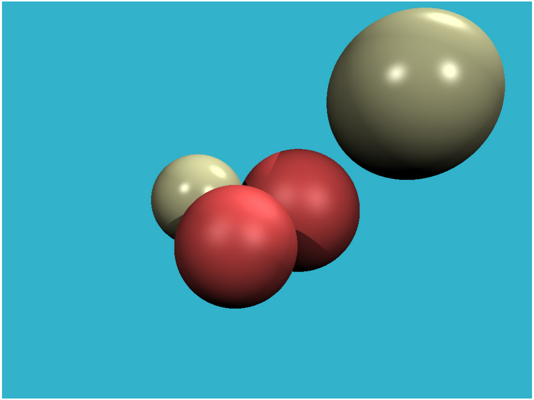
Ray Tracing

Example – Shadows

- ✓ occlusions
- ✓ local illumination
- ✓ shadows

global illumination

- × reflective surfaces
- × transparent objects



Quelle: `https://github.com/ssloy/tinyraytracer/`

Recursive Ray Tracing

Modelling - Direct Reflexions

Law of reflection

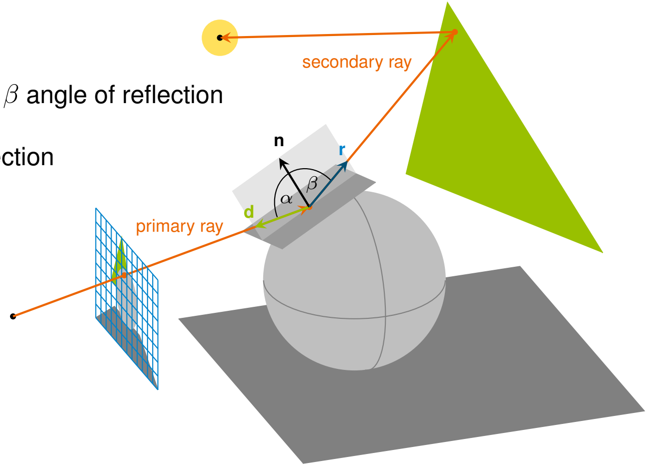
Angle of incidence $\alpha = \beta$ angle of reflection

Direction vector of reflection

$\mathbf{r} = 2 \cdot (\mathbf{d}^\top \mathbf{n}) \cdot \mathbf{n} - \mathbf{d}$

with

$\|\mathbf{r}\| = \|\mathbf{d}\| = \|\mathbf{n}\| = 1$



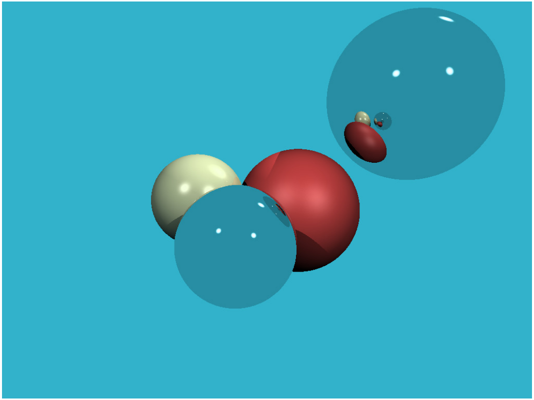
Recursive Ray Tracing

Example – reflective surfaces

- ✓ occlusions
- ✓ local illumination
- ✓ shadows

global illumination

- ✓ reflective surfaces
- × transparent objects



Quelle: `https://github.com/ssloy/tinyraytracer/`

Recursive Ray Tracing

Modelling – Transmission

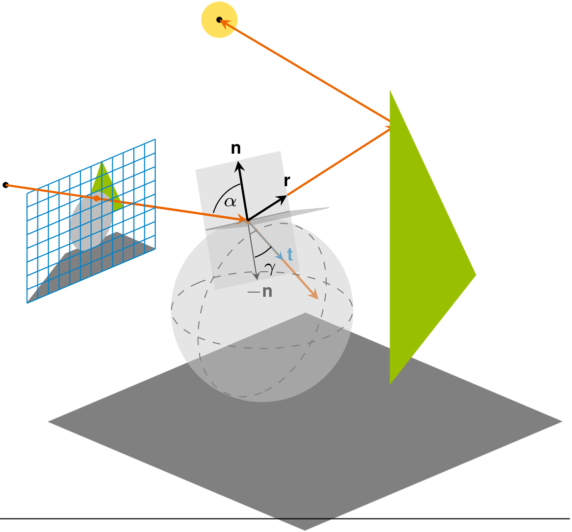
Law of refraction

$$\frac{\sin(\alpha)}{\sin(\gamma)} = \frac{\eta_t}{\eta}$$

Refraction index

e.g. glas $\eta_t > \eta$ air

Direction of transmission **t**



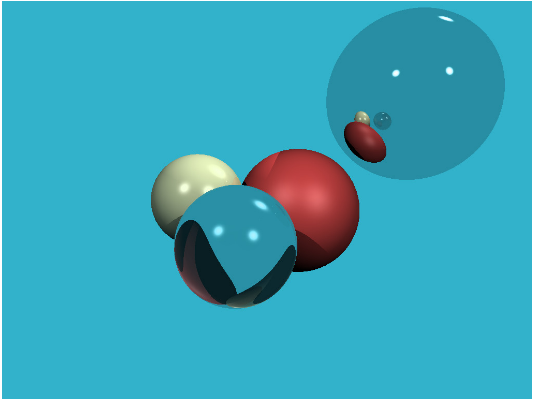
Recursive Ray Tracing

Example – Transmission

- ✓ occlusions
- ✓ local illumination
- ✓ shadows

global illumination

- ✓ reflective surfaces
- ✓ transparent objects



Quelle: `https://github.com/ssloy/tinyraytracer/`

Ray Tracing

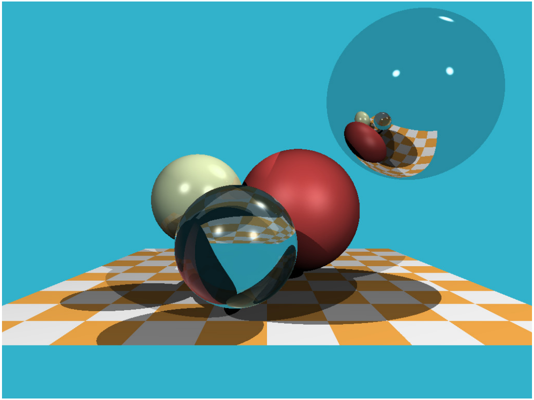
Summary – Pros and Cons

Pros

- ✓ correct object reflections
- ✓ correct shadows
- ✓ includes occlusions

Cons

- × unnatural hard lighting transitions
- × no diffus indirect illumination
- × aliasing-artefacts
- × large computational cost



Quelle: `https://github.com/ssloy/tinyraytracer/`

Ray Tracing vs. 3D Reconstruction

What can we take away for 3D vision?

1. advanced methods of image processing and 3D reconstruction

✓ Automated generation of photorealistic, error-free labeled synthetic datasets for neural network learning
✓ Calculation of ray intersections with object surfaces when the point cloud of the 3D object has been measured or reconstructed $\rightarrow$ e.g. for subpixel accurate reconstruction of 3D points
✓ Segmentation of shadow areas if the point cloud of the 3D object has been measured or reconstructed and the 3D pose of the light source(s) is known

2. Challenges of 3D reconstruction: Almost never brightness constancy!

× shadows
× reflections from other objects
× transparency
× object movements and light source movements