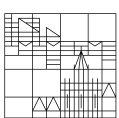
Global Illumination Methods

Practical Course

9 December 2019

Till Niese

Universität Konstanz



graphics.uni.kn

Work Package III

Tasks

1. Reflection
2. Refraction
3. Rough surfaces (optional)
4. Shadows
5. Clouds and Fog (optional)

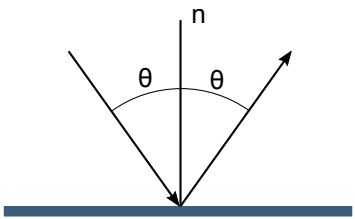
Date

This assignment is due **January, 6th**. Please bring your Laptop to class.
If you have any questions regarding the assignment, just write us an email.

Task 1

Reflection

- Extend the Material to contain the amount of reflection
- You can use an environment map to generate the illusion of a surrounding room (reflection mapping, optional)
- Emit a new ray at the hit-point
- Maximum number ray creations



Task 1: Reflection Mapping

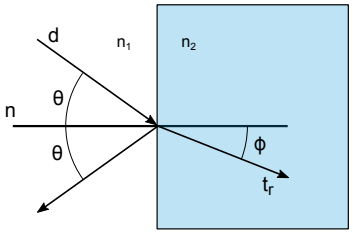
Terminator 2 (1991)

Task 2

Refraction

- Extend the Material to contain transparency and refraction information (m_i)
- Emit a new ray at the hit-point
- A ray traveling from one medium to another bends according to Snell's law
  $m_1 \sin\theta = m_2 \sin\phi$
- Total reflection:

$$\theta_{crit} = \arcsin\left(\frac{m_2}{m_1}\sin\phi\right) = \arcsin\left(\frac{m_2}{m_1}\right)$$



Task 2

Refraction

The direction of refraction t_r is [Shirley and Marschner, 2009]:

$$\boldsymbol{t}_r = \frac{m_1(\boldsymbol{d} - \boldsymbol{n}(\boldsymbol{d} \cdot \boldsymbol{n}))}{m_2} - \boldsymbol{n}\sqrt{1 - \frac{m_1^2(1 - (\boldsymbol{d} \cdot \boldsymbol{n})^2)}{m_2^2}}$$

Typical refractive indices are:

- Air: 1.00
- Water: 1.33
- Window glass: 1.51, optical glass: 1.49 - 1.92
- Diamond: 2.42

Task 3

Rough surfaces

- Rough surfaces do not have perfect reflection or refraction
- Emit multiple rays at the hit-point
- Add a random deflection to each of the new rays
- Average the color of each of the rays

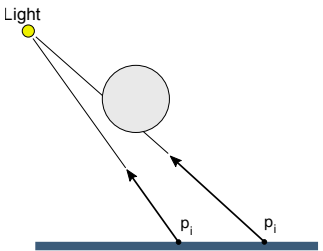
Task 4

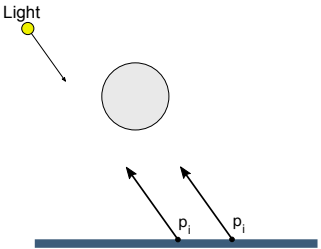
Shadows

- Perform shadow test by implementing a visibility function $V(p \leftrightarrow L_j)$

Problems

- Self intersection because for numerical imprecision
- can be solved by adding an offset to p in direction of the surface normal



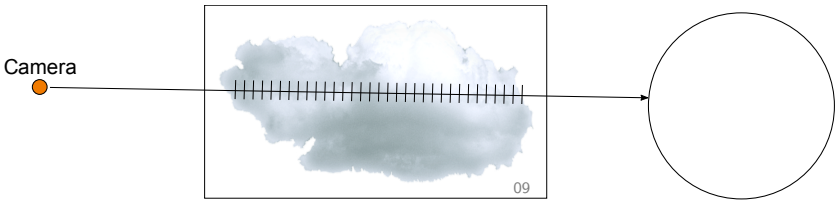


Task 5

Clouds and Fog

For clouds and fog the color and transparency has to be integrated over the volume of the object. Clouds and fog can be represented by either a volume texture or by noise functions.

- first check if the ray hits an object
- check if the ray intersects with one or more cloud object.
- integrate over the cloud object from the start point of the ray to the hit-point of the object or abort if fully opaque



Optional: Parallelization with OpenMP

CMakeLists.txt

```
find_package(OpenMP)
if (OPENMP_FOUND)
  set (CMAKE_C_FLAGS
       "${CMAKE_C_FLAGS} ${OpenMP_C_FLAGS}")
  set (CMAKE_CXX_FLAGS
       "${CMAKE_CXX_FLAGS} ${OpenMP_CXX_FLAGS}")
endif()
```

Optional: Parallelization with OpenMP

Rendering Loop

```
#pragma omp parallel for
for (int y = 0; y < h; ++y) {
  for (int x = 0; x < w; ++x) {
    if(_running) {
      // ...

      #pragma omp critical
      _image->setPixel(x, y, color);
    }
  }
}
```