Global Illumination Methods

Practical Course

5 December 2018

Till Niese, Jochen Görtler

Universität Konstanz

graphics.uni.kn

Work Package II

Tasks

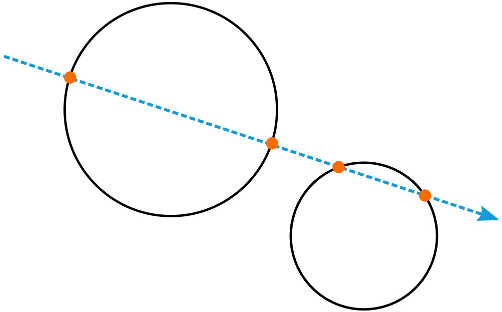
1. Global depth-sorting
2. Diffuse shading
3. Procedural texturing
4. Octree implementation (suggested, but optional)

Date

This assignment is due **December, 19th**. Please bring your Laptop to class.
If you have any questions regarding the assignment, just write us an email.

Task 1

Intersection test



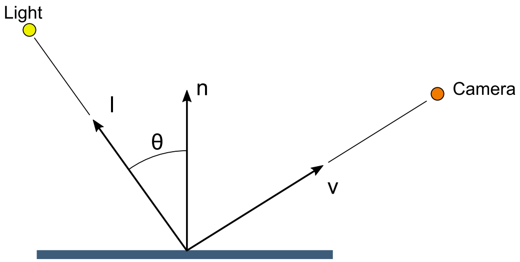
Global depth-sorting

Task 2

Diffuse shading

- Place a light source in the scene.
- Calculate the surface normal at the hit point.
- Diffuse shading (without specular highlight) using lambertian shading.

$$L_d = k_d I \max(0, n \cdot l)$$



Task 3

Procedural texturing

Create a checkerboard texture and apply it to a plane and sphere.

Task 4

Octree

To improve rendering performance for a large number of objects and triangles.

```
/// Store an entity in the correct position of the octree.
void push_back(Entity* object);

/// Returns list of entities that have
/// the possibility to be intersected by the ray.
std::vector<Entity*> intersect(const Ray& ray) const;

/// Subdivides the current node into 8 children.
void Node::partition();
```

Cone - Ray intersection

An infinite cone can be described using the equation: $x^2 + z^2 - y^2 = 0$
The equation for a cone with the apex at p_c aligned along the line $p_c + v_c t$ is:

$$cos^2\alpha(p - p_c - (v_c \cdot (p - p_c))v_c)^2 - sin^2\alpha(v_c \cdot (p - p_c))^2 = 0$$

To find the intersection point substitute point p on the cone with the equation for the ray: $p = p_r + v_r t$

$$cos^2\alpha(p_r + v_r t - p_c - (v_c \cdot (p_r + v_r t - p_c))v_c)^2 - sin^2\alpha(v_c \cdot (p_r + v_r t - p_c))^2 = 0$$

Cone - Ray intersection

$$cos^2\alpha(p_r + v_r t - p_c - (v_c \cdot (p_r + v_r t - p_c))v_c)^2 - sin^2\alpha(v_c \cdot (p_r + v_r t - p_c))^2 = 0$$

To simplify the equation replace $p_r - p_c$ with Δp

$$cos^2\alpha(v_r t + \Delta p - (v_c \cdot (v_r t + \Delta p))v_c)^2 - sin^2\alpha(v_c \cdot (v_r t + \Delta p))^2 = 0$$

Cone - Ray intersection

$$cos^2\alpha(v_r t + \Delta p - (v_c \cdot (v_r t + \Delta p))v_c)^2 - sin^2\alpha(v_c \cdot (v_r t + \Delta p))^2 = 0$$

The coefficients A, B, C of the quadratic equation, to solve t:

$A = cos^2\alpha(v_r - (v_r \cdot v_c) \cdot v_c)^2 - sin^2\alpha(v_r \cdot v_c)^2$

$B = 2cos^2\alpha((v_r - (v_r \cdot v_c) \cdot v_c) \cdot (\Delta p - (\Delta p \cdot v_c) \cdot v_c)) - 2sin^2\alpha(v_r \cdot v_c)(\Delta p \cdot v_c)$

$C = cos^2\alpha(\Delta p - (\Delta p \cdot v_c) \cdot v_c)^2 - sin^2\alpha(\Delta p \cdot v_c)^2$

Cone - Ray intersection

Use the *quadratic formula* to solve it.

```
bool quadratic(double a, double b, double c, double* t0, double* t1) {
        double discriminant = b * b - 4 * a * c;

        if (discriminant < 0) {
                return false;
        } else {
                discriminant = std::sqrt(discriminant);
                *t0 = ((-1 * b) + discriminant) / (2 * a);
                *t1 = ((-1 * b) - discriminant) / (2 * a);
                return true;
        }
}
```

For t_0 and t_1 you need to test if $t >= 0$ and if the intersection point on the infinite cone is within the boundaries of the cone:

$v_c \cdot ((p_r - p_c) + v_r t) > 0$ and $v_c \cdot ((p_r - p_c) + v_r t) < 0$

For the base of the cone you would do a simple ray disc intersection