University of British Columbia

Ray-Tracing

CPSC 314

Michiel van de Panne

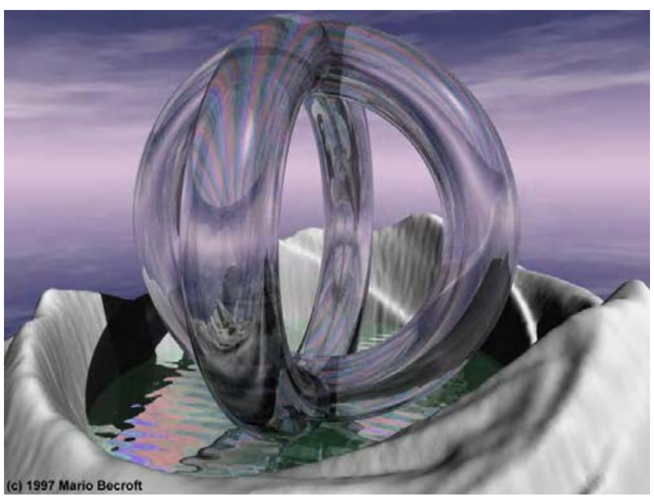
Ray-Tracing

University of British Columbia

CAD Raytraced Image of Audi R8C

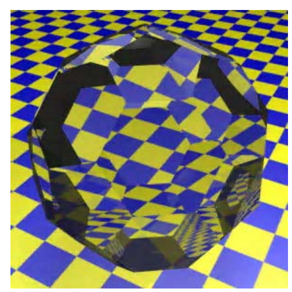
Raytracing

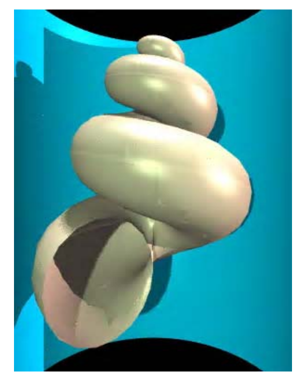
University of British Columbia



Raytracing

University of British Columbia





Overview

University of British Columbia

So far

- projective rendering (hardware)
- radiosity

Ray-Tracing

- simple algorithm for software rendering
- extremely flexible
- well suited to transparent and specular objects
- global illumination (*)
- partly physics-based: geometric optics

Ray-Tracing

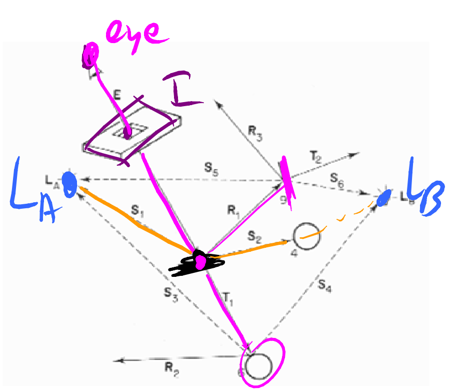
University of British Columbia

```
raytrace( ray ) {
   find closest intersection
   cast shadow ray, calculate colour_local
R  colour_reflect = raytrace( reflected_ray )
T  colour_refract = raytrace( refracted_ray )
   colour = k1*colour_local +
            k2*colour_reflect +
            k3*colour_refract
   return( colour )
}
```

- "raycasting" : only cast first ray from eye

Ray-Tracing

University of British Columbia



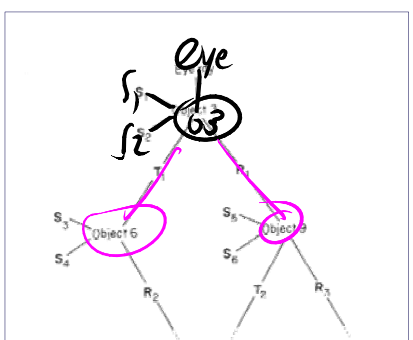
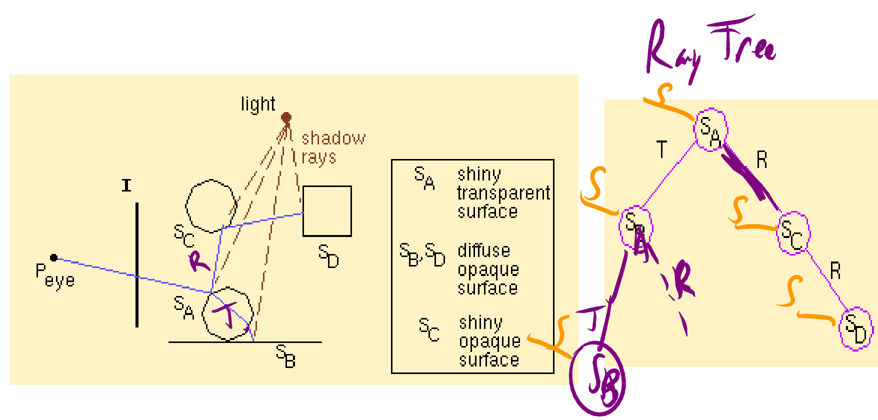


Fig. 12. The ray tree in schematic form.

Figure from Andrew S. Glassner, "An Overview of Ray Tracing" in An Introduction to Ray Tracing, Andrew Glassner, ed., Academic Press Limited, 1989.

Ray-Tracing

University of British Columbia



Ray-Tracing

University of British Columbia

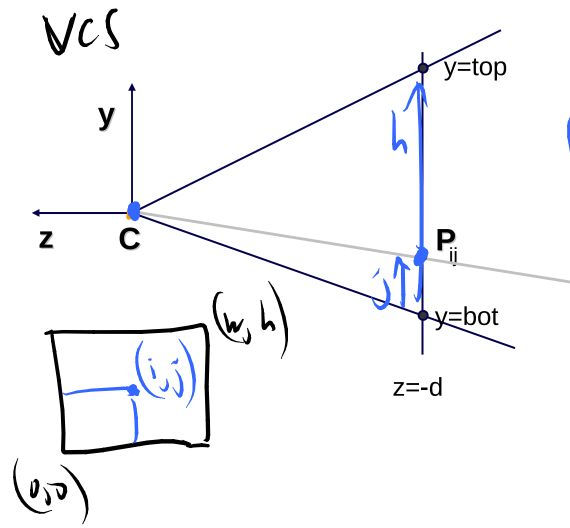
Ray Termination Criteria:

- **ray hits a diffuse surface**
- **ray exits the scene**
- **threshold on contrib. towards final pixel colour**
- **maximum recursion depth**

Ray-Tracing – Generation of Rays

University of British Columbia

Camera Coordinate System



$C(0,0,0)$

$P_{ij}(x_{ij}, y_{ij}, z_{ij})$

$z_{ij} = -d$

$y_{ij} = bot + \frac{j}{h}(top - bot)$

$x_{ij} = left + \frac{i}{w}(right - left)$

Ray-Tracing – Generation of Rays

University of British Columbia

Ray in 3D Space:

$$\mathrm{R}_{i,j}(t) = C + t \cdot (P_{i,j} - C)$$
$$= C + t \cdot v_{i,j}$$

where $t = 0 \ldots \infty$

Task:

- Given an object o, find ray parameter t, such that $\mathbf{R}_{i,j}(t)$ is a point on the object
  - *Such a value for t may not exist*
- Intersection test depends on geometric primitive

Ray Intersections

University of British Columbia

Sphere at origin:

- Implicit function:

$$S(x, y, z): x^2 + y^2 + z^2 = r^2$$

- Ray equation:

$$\mathrm{R}_{i,j}(t) = C + t \cdot \mathbf{v}_{i,j} = \begin{pmatrix} c_x \\ c_y \\ c_z \end{pmatrix} + t \cdot \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix} = \begin{pmatrix} c_x + t \cdot v_x \\ c_y + t \cdot v_y \\ c_z + t \cdot v_z \end{pmatrix}$$

Ray Intersections

University of British Columbia

To determine intersection:

- Insert ray $\mathbf{R}_{i,j}(t)$ into $S(x,y,z)$:

$$(c_x + t \cdot v_x)^2 + (c_y + t \cdot v_y)^2 + (c_z + t \cdot v_z)^2 = r^2$$

- Solve for t (find roots)
  - *Simple quadratic equation*

Ray Intersections

University of British Columbia

Triangles:

$$P = \overset{\alpha}{(1-\beta-\gamma)}\, a + \beta b + \gamma c$$

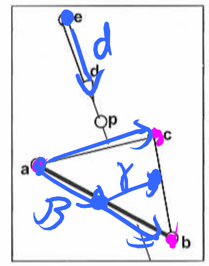


Figure 10.5. The ray hits the plane containing the triangle at point **p**.

ray

point on plane containing triangle

$$\mathbf{e} + t\mathbf{d} = \mathbf{a} + \beta(\mathbf{b} - \mathbf{a}) + \gamma(\mathbf{c} - \mathbf{a}).$$

$$x_e + t x_d = x_a + \beta(x_b - x_a) + \gamma(x_c - x_a),$$
$$y_e + t y_d = y_a + \beta(y_b - y_a) + \gamma(y_c - y_a),$$
$$z_e + t z_d = z_a + \beta(z_b - z_a) + \gamma(z_c - z_a).$$

$$\begin{bmatrix} x_a - x_b & x_a - x_c & x_d \\ y_a - y_b & y_a - y_c & y_d \\ z_a - z_b & z_a - z_c & z_d \end{bmatrix} \begin{bmatrix} \beta \\ \gamma \\ t \end{bmatrix} = \begin{bmatrix} x_a - x_e \\ y_a - y_e \\ z_a - z_e \end{bmatrix}.$$

Ray Tracing

University of British Columbia

Triangle Intersection (cont.)

Cramer's rule gives us

$$\beta = \frac{j(ei - hf) + k(gf - di) + l(dh - eg)}{M},$$

$$\gamma = \frac{i(ak - jb) + h(jc - al) + g(bl - kc)}{M},$$

$$t = -\frac{f(ak - jb) + e(jc - al) + d(bl - kc)}{M},$$

where

$$M = a(ei - hf) + b(gf - di) + c(dh - eg).$$

Ray Tracing

University of British Columbia

Triangle intersection (cont.): check bounds

Check

$0 \le \beta \le 1$

$0 \le \gamma \le 1$

$0 \le 1 - \beta - \gamma \le 1$

$t > 0$

Ray-Tracing – Geometric Transformations

University of British Columbia

Ray Transformation:

- For intersection test, it is only important that ray is in same coordinate system as object representation
- Transform ray into object coordinates $\vec{c}$ $\vec{v}$ or P_i
  - *Transform camera point and ray direction by* *<u>inverse</u> of model/view matrix*
- Shading has to be done in world coordinates (where light sources are given)
  - *Transform object space intersection point to world coordinates*
  - *Thus have to keep both world and object-space ray*