Ray Tracing

CS 351-50

Turner Whitted

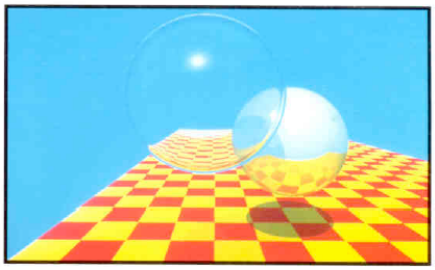
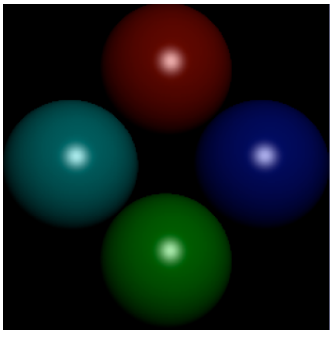
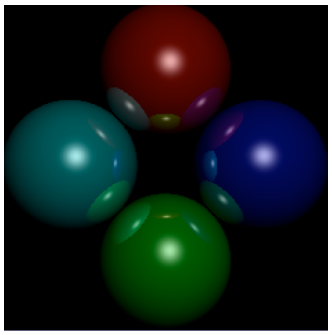


Plate III.10 Spheres and checkerboard. An early image produced with recursive ray tracing (Section 16.12). (Courtesy of Turner Whitted, Bell Laboratories.)

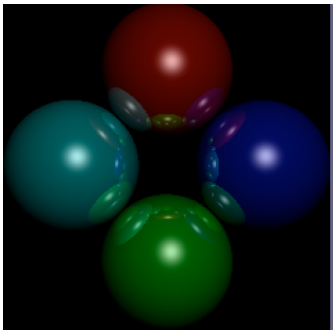
No Bounces



One Bounce

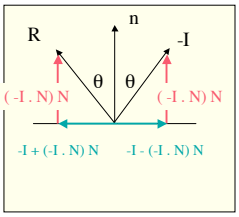


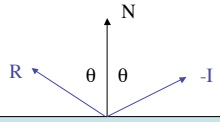
Two Bounces



Reflection Ray

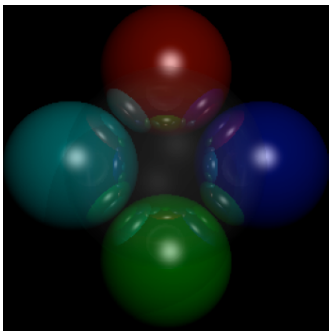
I = incident ray = -d
N = normal vector
R = reflected ray





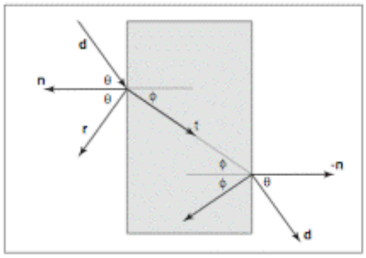
R = (-I . N) N - I + (-I . N) N
= -I - 2 (I . N) N

Adding an object: transparency

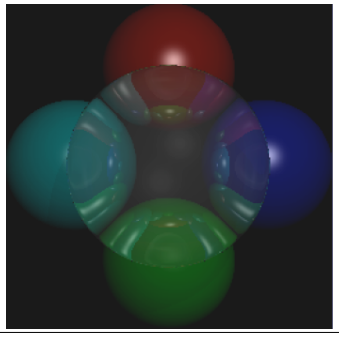


Refraction

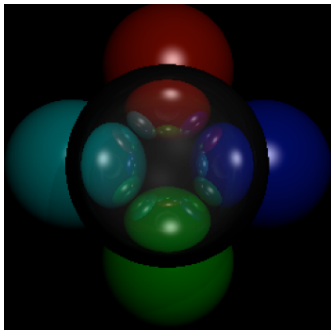
$n \sin \theta = n_t \sin \phi$



Index of Refraction > 1



Index of Refraction < 1

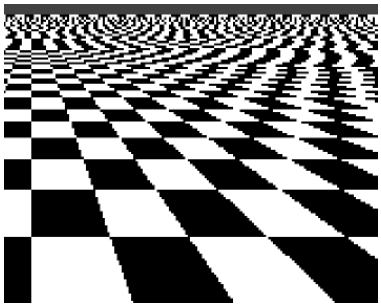


total internal reflection

Algorithm

```
Trace (ray){
    For each object in scene
        intersect (ray, object)
        If no intersections
            return background_color
    Color  = c_a * c_r   (Ambient * reflected diffuse color of surface)
    For each light
        for each object in scene
            intersect(shadowRay, object)
            Accumulate local illumination
    Trace (ReflectionRay)
    Trace (TransmissionRay)
    Accumulate global illumination
    Return illumination
    }
```

Sampling



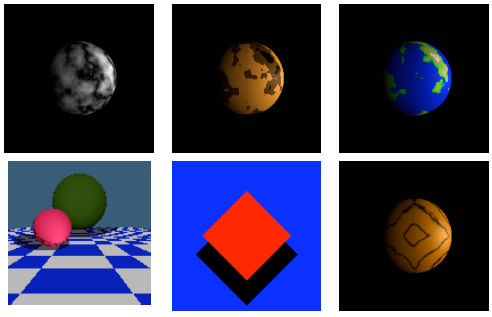
Ray Tracing Demos

http://www.siggraph.org/education/materials/HyperGraph/raytrace/rt_java/raytrace.html

http://www.cs.berkeley.edu/~efros/java/tracer/tracer.html

Sampling Demo

Texture & Materials



Links

- http://www.acm.org/tog/GraphicsGems/