6. Ray Tracing

Reading

Required:

- T. Whitted. An improved illumination model for shaded display. *Communications of the ACM* 23(6), 343-349, 1980.

Recommended:

- Foley, et al, Sections 15.10 and 16.12.
- A. Glassner. An Introduction to Ray Tracing. Academic Press, 1989.

What is light?

- Descartes (ca. 1630)
  - Light is a pressure phenomenon in the "plenum"
- Hooke (1665)
  - Light is a rapid vibration – first wave theory
- Newton (1666)
  - Refraction experiment revealed rectilinear propagation
  - Light is a particle (corpuscular theory)
- Young (1801)
  - Two slit experiment
  - Light is a wave
- Maxwell (ca. 1860)
  - Light is an electromagnetic disturbance
- Einstein (1905)
  - Light comes in quanta – photons

Modern theory: wave-particle duality.

Geometric optics

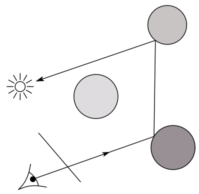
We will take the view of "geometric optics."

- Light is a flow of photons with wavelengths. We'll call these flows "light rays."
- Light rays travel in straight lines in free space.
- Light rays do not interfere with each other as they cross.
- Light rays obey the laws of reflection and refraction.
- Light rays travel form the light sources to the eye, but the physics is invariant under path reversal (reciprocity).

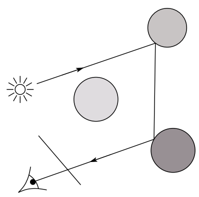
Eye vs. light ray tracing

Where does light begin?

At the eye: eye ray tracing (a.k.a., backward ray tracing)

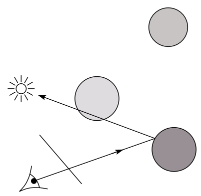


At the light: light ray tracing (a.k.a., photon tracing)

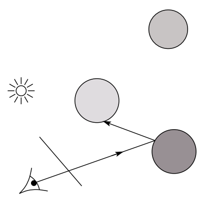


Hybrid methods

- Local illumination
  - Cast one eye ray, then shade according to light

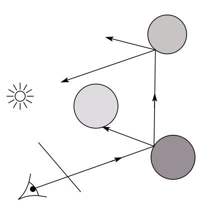


- Appel (1968)
  - Cast one eye ray + one ray to light

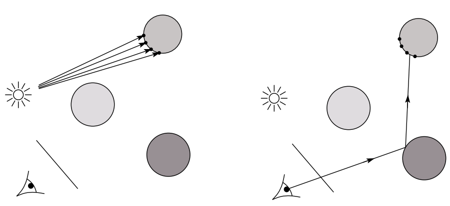


Hybrid methods, cont'd

- Whitted (1980)
  - Eye ray tracing + rays to light
  - Recursive ray tracing

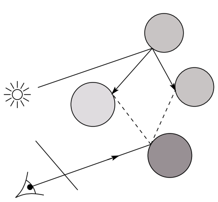


- Heckbert (1990)
  - Eye ray tracing + light ray tracing + light storage on surface

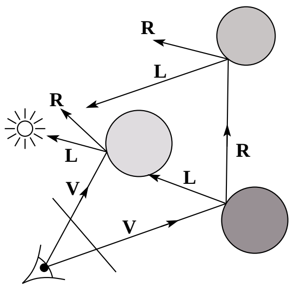


Hybrid methods, cont'd

- Veach (1995)
  - Eye ray tracing + light ray tracing + path connection



Whitted ray-tracing algorithm

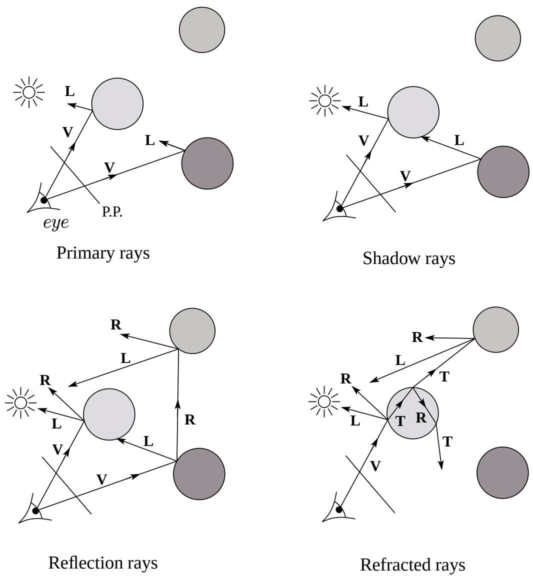


Algorithm:

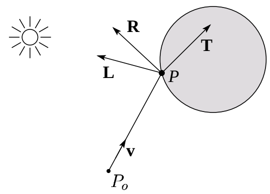
1. For each pixel, trace a "primary ray" to the first visible surface.
2. For each intersection, trace "secondary rays":
   - "Shadow rays" in directions $\mathbf{L}_i$ to light sources.
   - "Reflected ray" in direction $\mathbf{R}$.
   - "Refracted ray" in direction $\mathbf{T}$.

Whitted ray-tracing algorithm, cont'd

Let's look at this in stages:



Shading



Let $I(P_o, \mathbf{v})$ be the intensity seen from point P_o along direction $\mathbf{v}$:

$$I(P_o, \mathbf{v}) = I_{\text{direct}} + I_{\text{reflected}} + I_{\text{transmitted}}$$

where

- I_{direct} is computed from the Phong model (next lecture)
- $I_{\text{reflected}} = k_{\text{reflected}} I(P, \mathbf{R})$
- $I_{\text{transmitted}} = k_{\text{transmitted}} I(P, \mathbf{T})$

Shading, cont'd

In the next lecture, we will discuss the Phong shading equation.

For now, we will just say that surfaces can:

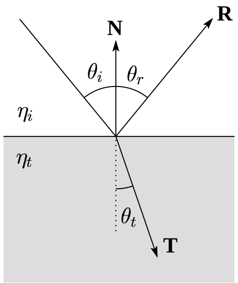
- reflect light ideally specularly (mirror-like) – weighted by coefficient k_s
- transmit light ideally specularly (glass-like) – weighted by coefficient k_t
- reflect light diffusely – weighted by coefficient k_d

So, in the previous equation, we will replace $k_{\text{reflected}}$ with k_s, and $k_{\text{transmitted}}$ with k_t.

Further, the diffuse lighting term will be shown to be equal to $k_d \mathbf{L} \cdot \mathbf{N}$, clamped to positive values.

All shading calculations are done separately on each color channel.

Reflection and transmission



Law of reflection:

$$\theta_i = \theta_r$$

Snell's law of refraction:

$$\eta_i \sin\theta_i = \eta_t \sin\theta_t$$

where η_i, η_t are "indices of refraction."

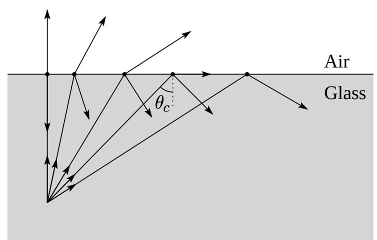
Total Internal Reflection

The equation for the angle of refraction can be computed from Snell's law:

What happens when $\eta_i > \eta_t$?

When θ_t is exactly 90^o, we say that θ_i has achieved the "critical angle," θ_c.

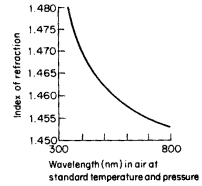
For $\theta_i > \theta_c$, no rays are transmitted, and only reflection occurs, a phenonenon known as "total internal reflection" or TIR.



Refraction and wavelength dependence

The index of refraction varies according to the properties of the material.

Medium	Index of refraction
Vaccum	1
Air	1.0003
Water	1.33
Fused quartz	1.46
Glass, crown	1.52
Glass, dense flint	1.66
Diamond	2.42



Index of refraction variation for fused quartz

The index of refraction varies with wavelength – "dispersion."

Usually, this effect is ignored in computer graphics.

Fresnel coefficient

The amount of light that is reflected is determined by the Fresnel coefficient. It varies with:

- Angle of incidence
- Wavelength

Dielectrics: light reflected and transmitted must sum to incident light.

Example: Glass is transparent viewed head on, but acts like a mirror at grazing angles.

Metals: no light is transmitted (absorbed instead), can have strong wavelength dependence.

Example: brass gives a yellowed reflection.

Ray-tracing pseudocode

function *RayTrace*(P_o, $\mathbf{v}$):
  (P, *Obj*) $\leftarrow$ *RayCast*(P_o, $\mathbf{v}$)
  $I \leftarrow 0$
  for each light source ℓ **do**:
    (P', *LightObj*) $\leftarrow$ *RayCast*(P, *Dir*(P, ℓ))
    if *LightObj* $= \ell$ **then**:
      $I \leftarrow I + \langle\text{diffuse term}\rangle + \langle\text{spec term}\rangle$
    end if
  end for
  $I \leftarrow I + Obj.k_s * RayTrace(P, \mathbf{R})$
  $I \leftarrow I + Obj.k_t * RayTrace(P, \mathbf{T})$
  return I
end function

Terminating recursion

Q: How do you bottom out of recursive ray tracing?

Possibilities:

Ray-surface intersection

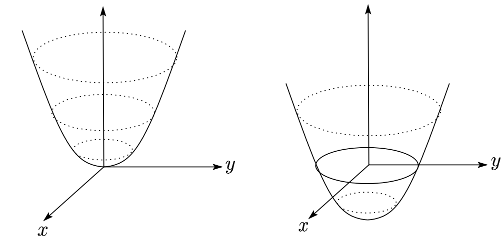
We can intersect rays with two basics types of surfaces: implicit or parametric. Let's first consider these terms for curves in 2D. We can define a parametric curve as:

$$x = x(u)$$
$$y = y(u)$$

Example: A circle with radius r centered at the origin is given by:

$$x = r\cos u$$
$$y = r\sin u$$

By contrast, an "implicit" representation of the circle is:



Ray-surface intersection, cont'd

We can define surfaces in 3D implicitly as:

$$F(x, y, z) = 0$$

or parametrically as:

$$(x(u, v), y(u, v), z(u, v))$$

We define the points along a ray as: $P = P_o + t\mathbf{v}$

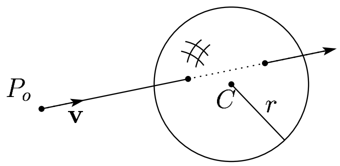
Intersection with implicit surfaces can be reduced to root finding. Given:

$$F(x, y, z) = 0$$
$$x = x_o + tv_x$$
$$y = y_o + tv_y$$
$$z = z_o + tv_z$$

We can solve for t by substitution:

$$F(x_o + tv_x, y_o + tv_y, z_o + tv_z) = \tilde{F}(t) = 0$$

Intersecting rays with spheres

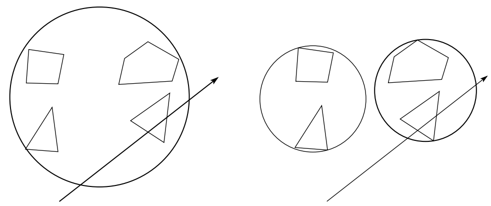


Acceleration: hierarchical bounding volumes

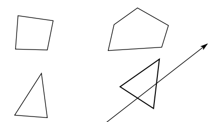
Vanilla ray tracing is really slow!

In practice, some acceleration technique is almost always used.

One approach is to use "hierarchical bounding volumes."



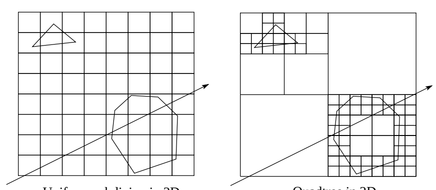
Intersect with largest B.V... ...then intersect with children...



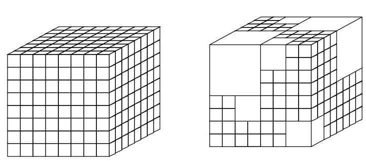
...until you reach the leaf nodes - the primitives.

Acceleration: spatial subdivision

Another approach is "spatial subdivision."



Uniform subdivion in 2D Quadtree in 2D



Uniform subdivion in 3D Octree in 3D

Idea:

- Partition objects spatially.
- Trace ray through voxel array.

Partition can be uniform or adaptive (e.g., octrees).