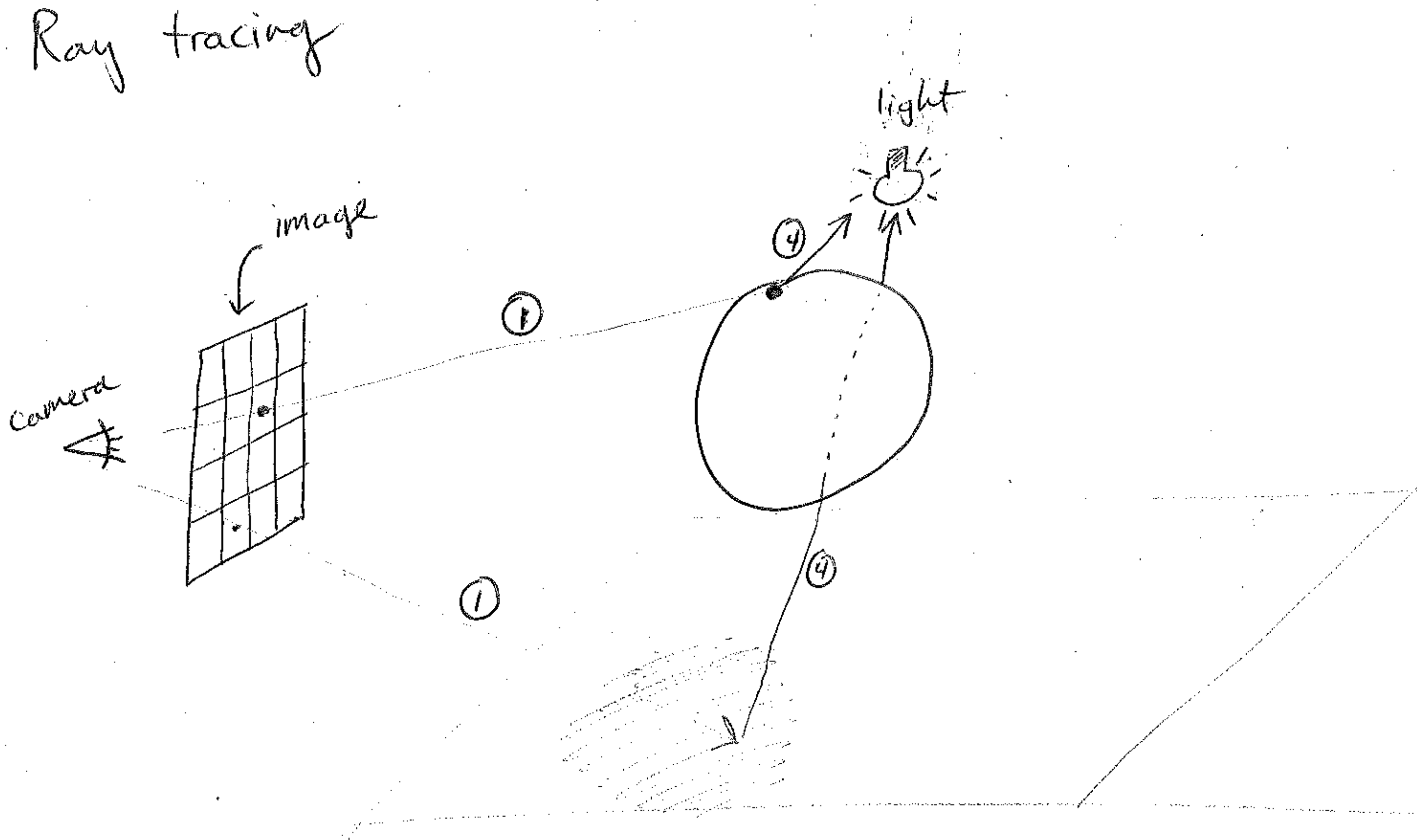
Ray tracing



(1) for each pixel, cast a ray
(2) identify first object hit by ray
(3) compute shading of hit object at point ray hit it
  (4) for shadows, cast ray to each light
    → if hit object before light, ignore the light's contribution to shading

* if ray hits nothing, use background color
  → environment mapping, use environment map rather than background color

* important primitive operations
  → intersect (ray, object)
  → shade object

Phong shading needs

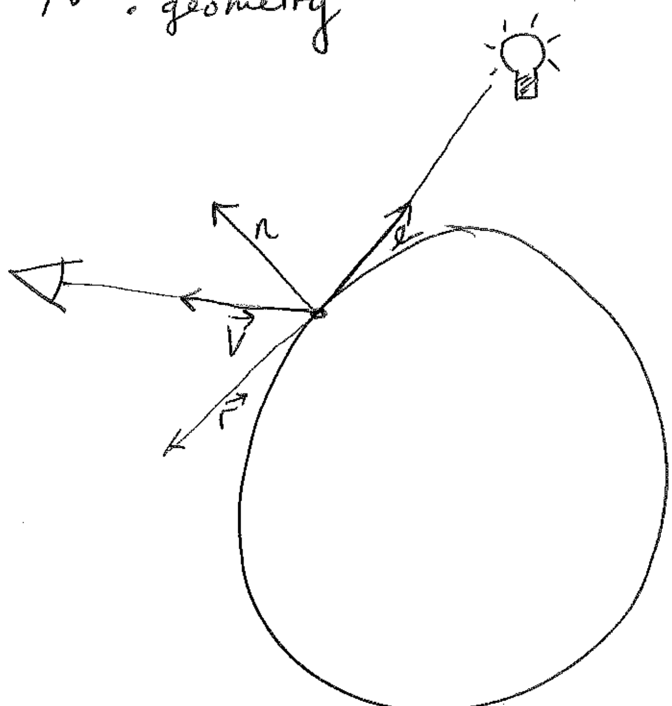
$$I = L_a R_a + L R_d (\vec{n} \cdot \vec{l}) + L R_s (\vec{r} \cdot \vec{v})^e$$

inputs : L_a , L → from lights

R_a, R_d, R_s, e → from shader (material properties)

↳ objects have shaders to describe their appearance

$\vec{l}, \vec{n}, \vec{r}, \vec{v}$: geometry



$\vec{n}$: object can compute this given intersection location

$\vec{V}$: from camera and intersection locations

$\vec{l}$: from light and intersection locations

$\vec{r}$: from $\vec{n}$ and $\vec{l}$

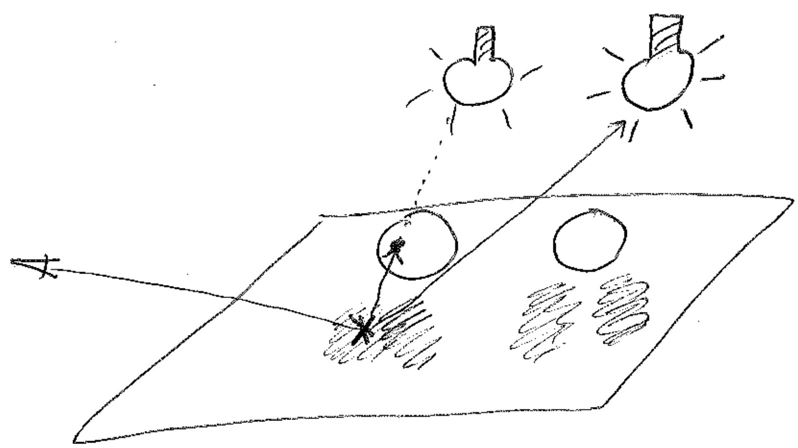
Lights

note that L, L_a, $\vec{l}$ are <u>per light</u>

compute for each light, sum shading

Shadows

is light visible? cast ray to light and see (reuse existing routine)



Other types of shaders

Flat shader → fixed color (ignore lights)

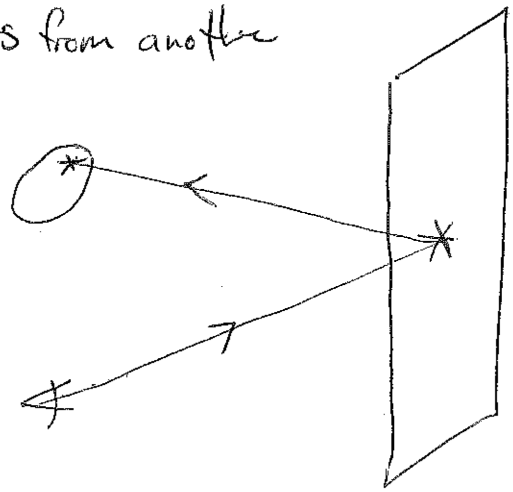
good for debugging but little else

Reflective Shader

Shiny surface

~~light reflects from object~~

light from one object reflects from another
and reaches you
→ indirect light



* cast a new ray from intersection point in reflection direction

* reflectivity β $\quad 0 \le \beta \le 1$
* surface shader
    ↳ surface appearance <u>ignoring</u> reflection $\Rightarrow I_0$

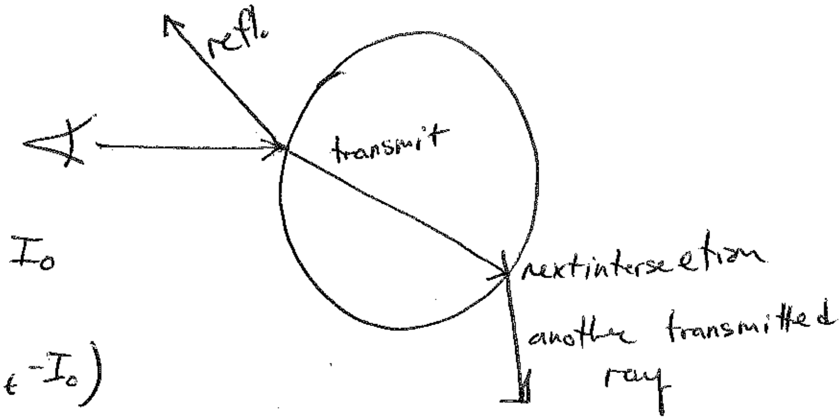
* color from reflected ray $\Rightarrow I_r$

* effective color $\Rightarrow I_0 + \beta(I_r - I_0)$

  note: $\beta = 0 \Rightarrow I = I_0$
  $\beta = 1 \Rightarrow I = I_r$

Transmission

* cast two rays
  → reflected ray ⇒ I_r
  → transmitted ray I_t



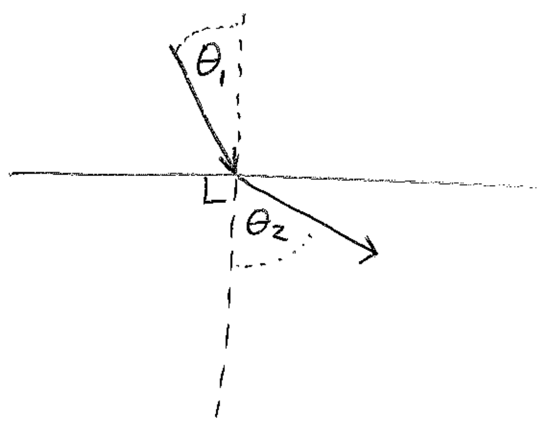
* combine with object shade I_o

$$I = I_o + \beta(I_r - I_o) + \gamma(I_t - I_o)$$

$$\beta + \gamma \le 1$$

* direction of transmitted ray given by snell's law

$$\frac{\sin\theta_1}{\sin\theta_2} = \frac{n_1}{n_2}$$



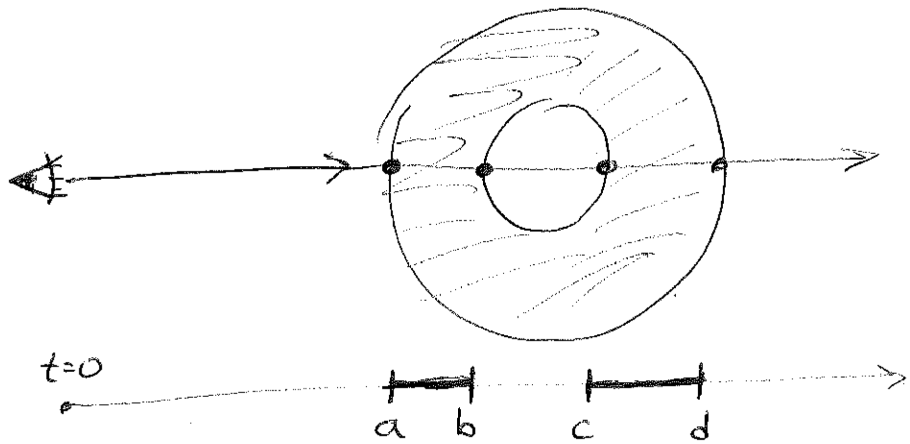
n_1, n_2 index of refraction

air: $n \approx 1.00$
glass: $n \approx 1.46$
water: $n \approx 1.33$
vacuum: $n = 1$
diamond: $n \approx 2.42$

$$\sin\theta_2 = \frac{n_1}{n_2}\sin\theta_1$$

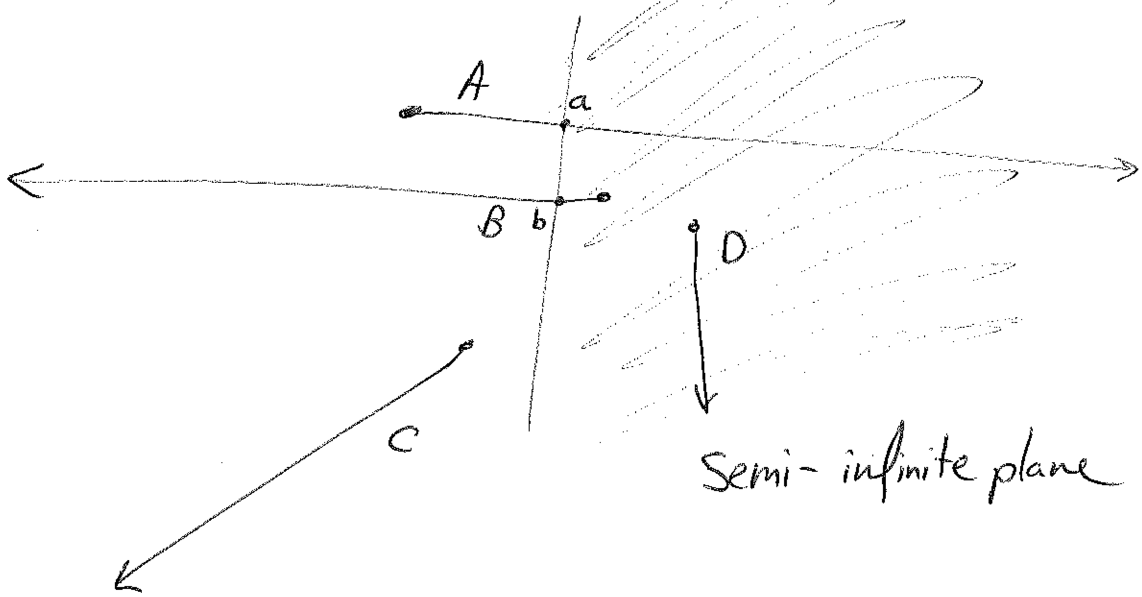
* what if $\sin\theta_2 > 1$?
  ↳ complete internal reflection
  $\gamma = 0$

Intersections



ray intersects object for $t \in [a,b]$ and $t \in [c,d]$

$\hookrightarrow (a,b,c,d)$



$A \to [a, \infty) \quad \to (a)$

$B \to [0, b] \quad \to (0, b)$

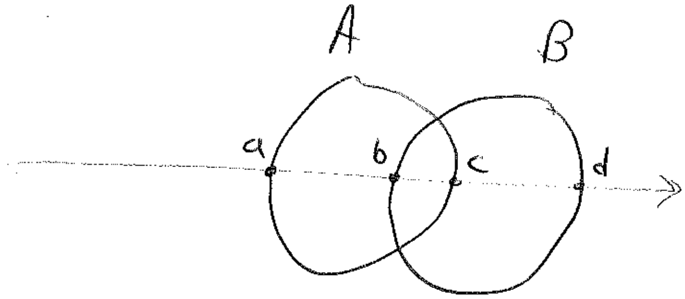
$C \to \emptyset \quad \to (\,)$

$D \to [0, \infty) \quad \to (0)$

Intersections II

why not just store first intersection?

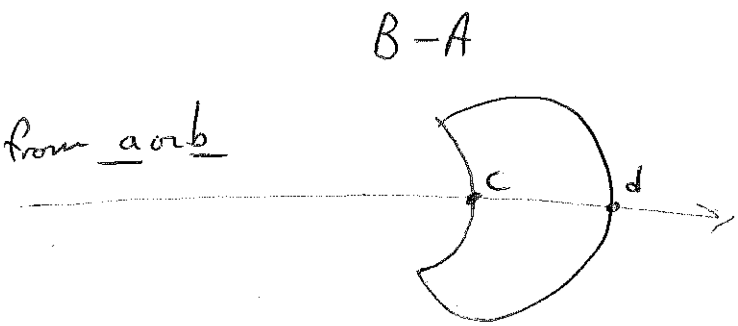
→ booleans



first hit on $A = a$

first hit on $B = b$

first hit on $B - A = c$



c cannot be deduced from a or b