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

\[ I = n_0 \cdot L \cdot R_a + LR_d (\hat{n} \cdot \hat{L}) + LR_s (\hat{n} \cdot \hat{V})^e \]

inputs: \( L_0, \hat{L} \) → from lights
\( R_a, R_d, R_s, e \) → from shader (material properties)
\( \hat{L} \) → objects have shaders to describe their appearance

\( \hat{n}, \hat{V}, \hat{r}, \hat{v} \) : geometry

\( \hat{n} \): object can compute this given intersection location
\( \hat{V} \): from camera and intersection locations
\( \hat{l} \): from light and intersection locations
\( \hat{r} \): from \( \hat{n} \) and \( \hat{l} \)
Lights
Note that $L, l_a, l$ are per light.
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

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

* cast a new ray from intersection point in reflection direction

* reflectivity \( \beta \), \( 0 \leq \beta \leq 1 \)

* surface shader

\( I \): surface appearance ignoring 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 \( \Rightarrow I_r \)
  → Transmitted ray \( I_t \)

* Combine with object-shade \( I_0 \)

\[
I = I_0 + \beta (I_r - I_0) + \gamma (I_t - I_0)
\]

\[
\beta + \gamma \leq 1
\]

* 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: \( n \approx 1.33 \)
crystal: \( n = 1 \)
diamond: \( n \approx 2.42 \)

\[
\sin \theta_2 = \frac{n_1}{n_2} \sin \theta_1
\]

* What if \( \sin \theta_2 > 1 \) ?
  \( \rightarrow \) Complete internal reflection
  \( \gamma = 0 \)
Intersections

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

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

Semi-infinite plane

$A \rightarrow [a, \infty) \rightarrow (a)$

$B \rightarrow [0, b] \rightarrow (0, b)$

$C \rightarrow \emptyset \rightarrow ()$

$D \rightarrow [0, \infty) \rightarrow (0)$
Why not just store first intersection?

→ booleans

\[ \text{first hit on } A = a \]
\[ \text{first hit on } B = b \]
\[ \text{first hit on } B \cup A = c \]

\( c \) cannot be deduced from \( a \cup b \)