Ray Tracing Reflections and Refractions

COSC342

Lecture 18
9 May 2017

Reflection and Refraction

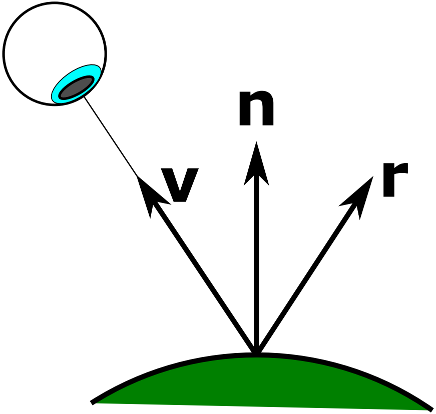
- Reflection
  - Extremes of shiny surfaces
  - Can see objects in reflection
  - Easy to do with ray tracing
- Refraction
  - Transparent objects
  - Light bends at the surface
  - Details get very complicated

Reflection

- Light reflected in surface
- Cast a new ray for reflection
- Reflected direction is

$$\mathbf{r} = 2\mathbf{n}(\mathbf{v} \cdot \mathbf{n}) - \mathbf{v}$$

- $\mathbf{r}$ is the reflected ray
- $\mathbf{v}$ the view direction
- $\mathbf{n}$ the surface normal
- (We assume unit vectors)
- Why this formula?



Reflection Formula

Split rays into two parts:

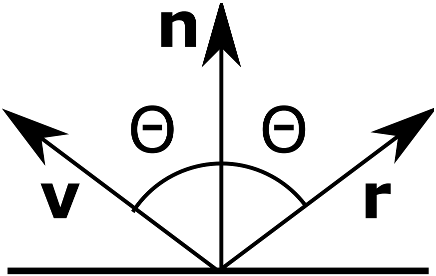
- $\mathbf{u}_{||}$ parallel to $\mathbf{n}$
- $\mathbf{u}_{\perp}$ perpendicular to $\mathbf{n}$
- $\mathbf{u} = \mathbf{u}_{||} + \mathbf{u}_{\perp}$

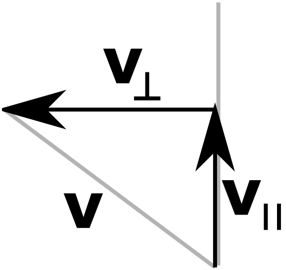
For the view ray

- $\mathbf{v}_{||} = \cos\theta\mathbf{n} = (\mathbf{v}\cdot\mathbf{n})\mathbf{n}$
- $\mathbf{v}_{\perp} = \mathbf{v} - \mathbf{v}_{||} = \mathbf{v} - (\mathbf{v}\cdot\mathbf{n})\mathbf{n}$

For the reflected ray:

- $\mathbf{r}_{||} = \mathbf{v}_{||} = (\mathbf{v}\cdot\mathbf{n})\mathbf{n}$
- $\mathbf{r}_{\perp} = -\mathbf{v}_{\perp} = (\mathbf{v}\cdot\mathbf{n})\mathbf{n} - \mathbf{v}$
- $\mathbf{r} = \mathbf{r}_{||} + \mathbf{r}_{\perp} = 2(\mathbf{v}\cdot\mathbf{n})\mathbf{n} - \mathbf{v}$





Ray Tracing Reflection

Ray tracing the reflection

- Cast a new ray from hit point
- **r** is the direction
- What colour do you see?

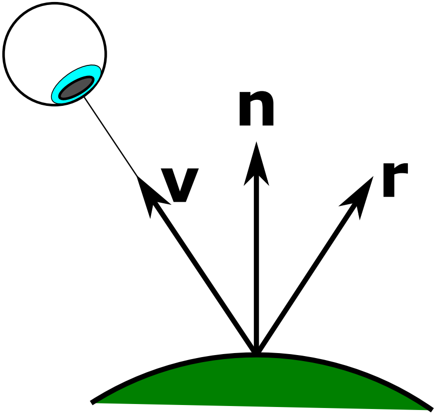
For a perfect mirror

- Use the reflected ray

For semi-reflective surfaces

- Reflective co-efficient k_r

$$k_r(\text{reflection}) + (1 - k_r)(\text{surface})$$



Implementing Reflection

Recursive implementation is fairly simple

```
colour render(ray, scene, fraction, depth)
   hit = intersect(ray, scene)
   if (depth == 0 || hit.kr * fraction < TINY)
      thisColour = hit.colour
   else
      thisColour = (1 - hit.kr) * hit.colour +
            hit.kr * render(reflection, scene,
                            hit.kr*fraction, depth-1)
   return fraction * thisColour
```

First call has `fraction = 1` and some maximum 'ray depth'

Implementing Reflection

Can avoid the recursion

```
colour render(ray, scene, maxDepth)
   depth = 0
   fraction = 1
   colour = background
   while (depth < maxDepth && fraction > TINY)
      hit = intersect(ray, scene)
      colour = (1-fraction)*colour + fraction*hit.colour
      fraction *= hit.kr
      ray = reflect(ray)
      depth++
```

Refraction

Light ‘bends’ between materials

- Material has a *refractive index*
- Ratio of refractive indices determines the bending

Snell’s law:

$$\eta_1 \sin \theta_1 = \eta_2 \sin \theta_2$$

$$\theta_2 = \sin^{-1}\left(\frac{\eta_1}{\eta_2} \sin \theta_1\right)$$

Some common materials:

- Vacuum: $\eta = 1$
- Air: $\eta \approx 1$
- Water: $\eta \approx 1.333$
- Window glass: $\eta \approx 1.52$

Ray Tracing Refraction

Much the same as reflection

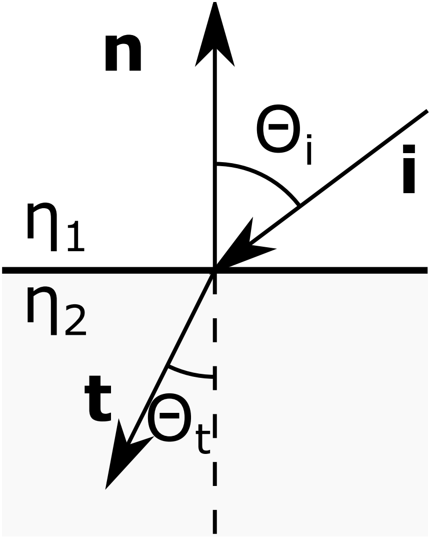
- Materials have k_t

  $$k_t(\text{refraction}) + (1 - k_t)(\text{surface})$$

- The refracted ray is

  $$\mathbf{t} = \frac{\eta_1}{\eta_2}(\mathbf{i} + (\mathbf{n}\cdot\mathbf{i})\mathbf{n}) - \sqrt{1 - \frac{\eta_1^2}{\eta_2^2}(1 - (\mathbf{i}\cdot\mathbf{n})^2)}\mathbf{n}$$

- Why this formula?



Finding the Refracted Ray

Incoming ray:

- $\mathbf{i}_{||} = -(\mathbf{n} \cdot \mathbf{i})\mathbf{n}$
- $\mathbf{i}_{\perp} = (\mathbf{i} + (\mathbf{n} \cdot \mathbf{i})\mathbf{n})$

Refracted ray:

- $\mathbf{t}_{\perp}$ is parallel to $\mathbf{i}_{\perp}$
- Lengths are sines of angles, so

$$\mathbf{t}_{\perp} = \frac{\eta_1}{\eta_2}\mathbf{i}_{\perp} = \frac{\eta_1}{\eta_2}(\mathbf{i} + (\mathbf{n} \cdot \mathbf{i})\mathbf{n})$$

- We want unit vectors so

$$1 = \|\mathbf{t}\|^2 = \|\mathbf{t}_{\perp}\|^2 + \|\mathbf{t}_{||}\|^2$$

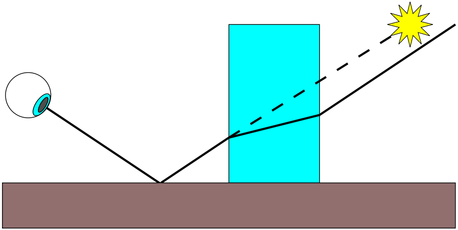
- $\mathbf{t}_{||}$ is parallel to $\mathbf{n}$, so

$$\begin{aligned}
\mathbf{t}_{||} &= -\sqrt{1 - \|\mathbf{t}_{\perp}\|^2}\mathbf{n} \\
&= -\sqrt{1 - \sin^2\theta_t}\mathbf{n} \\
&= -\sqrt{1 - \frac{\eta_1^2}{\eta_2^2}\sin^2\theta_i}\mathbf{n} \\
&= -\sqrt{1 - \frac{\eta_1^2}{\eta_2^2}(1 - \cos^2\theta_i)}\mathbf{n} \\
&= -\sqrt{1 - \frac{\eta_1^2}{\eta_2^2}(1 - (\mathbf{i} \cdot \mathbf{n})^2)}\mathbf{n}
\end{aligned}$$

Problems with Refraction

Refraction makes some parts of ray tracing harder:

- How do you determine lighting though a refracting object
- What direction do you cast shadow rays?
- Recursive algorithm is not efficient in some languages



Total Internal Refraction

- Angle of refraction:
  $\theta_2 = \sin^{-1}\left(\frac{\eta_1}{\eta_2}\sin\theta_1\right)$
- What if $\frac{\eta_1}{\eta_2}\sin\theta_1 > 1$?
- Total internal refraction
- Light reflects off the surface

Ray Tracing Summary

Types of ray:

- Primary Ray
  – from camera to scene
- Shadow Rays
  – from hit to light
- Reflection Rays
  – bouncing off surface
- Refraction Rays
  – bending through surface