Ray Tracing
Geometry
The Camera Model

- Based on a simple pin-hole camera model
  - Simplest lens model
  - Pure geometric optics – based on similar triangles
  - Perfect image if hole infinitely small
  - Inverted image
Basic Ray Tracing Algorithm

for every pixel  
    cast a ray from the eye through pixel
for every object in the scene
    find intersections with the ray
    keep it if closest
compute color at the intersection point
Construct a Ray

• 3D parametric line
  \[ r(t) = \text{eye} + t (\text{p-eye}) \]
  \[ r(t): \text{ray equation} \]
  \[ \text{eye}: \text{eye (camera) position} \]
  \[ \text{p}: \text{pixel position} \]
  \[ t: \text{ray parameter} \]

Question: How to calculate the pixel position P?
What are given?

- Camera (eye) position
- View direction or center of interest
- Camera orientation (which way is up?)
  - specified by an “up” vector
- Field of view + aspect ratio
- Distance to the image plane
- Pixel resolutions in x and y
We need to have a ‘view coordinate system, i.e., compute the $u$, $v$, $w$ vectors

\[ u \perp v \perp w \]

$w \perp$ image plane

Eye + $w$ goes through the image plane center
Camera Setup

“up” vector

v
w

direction
u

w: known
u = w \times “up”
v = u \times w

“up” may not be perpendicular to w
x: cross product

CSE 681
Pixel Calculation

Coordinate (in u,v,n space) of upper left corner of screen

Assume virtual screen is one unit away (D=1) in w direction

\[
\text{Eye + w} - (\text{xres}/2)\times \text{PixelWidth}\times u + (\text{yres}/2)\times \text{PixelHeight}\times v
\]
Coordinate (in u,v,n space) of upper left corner of screen

Assume virtual screen is one unit away (D=1) in w direction

\[ \text{Eye} + w - \frac{xres}{2} \times \text{PixelWidth} \times u + \frac{yres}{2} \times \text{PixelHeight} \times v \]

CSE 681
Camera Setup

\[
\tan\left(\frac{\theta}{2}\right) = \frac{\text{yres} \times \text{pixelHeight}}{2D}
\]

\[
\text{pixelHeight} = 2 \times \tan\left(\frac{\theta_y}{2}\right) \times \frac{D}{\text{yres}}
\]

\[
\text{pixelWidth} = 2 \times \tan\left(\frac{\theta_x}{2}\right) \times \frac{D}{\text{xres}}
\]
Screen Placement

How do images differ if the resolution doesn’t change?

Assume virtual screen is one unit away (D=1) in w direction
Pixel Calculation

\[
\tan(\frac{\theta}{2}) = \frac{yres \times \text{pixelHeight}}{2}
\]

\[
\text{pixelHeight} = 2 \times \frac{\tan(\frac{\theta_y}{2})}{yres}
\]

\[
\text{pixelWidth} = 2 \times \frac{\tan(\frac{\theta_x}{2})}{xres}
\]

Pixel AspectRatio = pixelWidth/pixelHeight

Coordinate (in xyz space) of upper left corner of screen = ?
Pixel Calculation

Coordinate (in xyz space) of upper left corner of screen = ?

\[ \tan(\theta/2) = \frac{yres \cdot \text{pixelHeight}}{2} \]

\[ \text{pixelHeight} = \frac{2 \cdot \tan(\theta/2)}{yres} \]

\[ \text{pixelWidth} = \frac{2 \cdot \tan(\theta/2)}{xres} \]

\[ \text{Eye} + w - \left( \frac{xres}{2} \right) \cdot \text{PixelWidth} \cdot u + \left( \frac{yres}{2} \right) \cdot \text{PixelHeight} \cdot v \]
Pixel Calculation

Coordinate (in xyz space) of upper left pixel center = ?

Eye + w - (xres/2)*PixelWidth*u + (yres/2)*PixelHeight *v

+ (pixelWidth/2)*u - (pixelHeight/2)*v

CSE 681
Interate through pixel Centers

\[
\begin{align*}
\text{pixelCenter} &= \\
\text{scanlineStart} &= \text{Eye} + \\
& \quad \text{w - } \\
& \quad (\text{xres}/2)\text{PixelWidth}\text{u} + \\
& \quad (\text{yres}/2)\text{PixelHeight} \text{v} + \\
& \quad (\text{pixelWidth}/2)\text{u} - \\
& \quad (\text{pixelHeight}/2)\text{v}
\end{align*}
\]

\[
\text{pixelCenter} += \text{pixelWidth} \times u
\]

\[
\text{scanlineStart} -= \text{pixelHeight} \times v
\]
Pixel loops

ScenlineStart = [from previous slide]
For each scanline {
    pixelCenter = scanlineStart
    For each pixel across {
        form ray from camera through pixel
        ....
        pixelCenter += pixelWidth*u
    }
    scanlineStart -= pixelHeight*v
}
Process Objects

For each pixel {
    Form ray from eye through pixel
    $\text{distance}_{\text{min}} = \infty$
    For each object {
        If ($\text{distance} = \text{intersect}(\text{ray}, \text{object})$) {
            If ($\text{distance} < \text{distance}_{\text{min}}$) {
                closestObject = object
                $\text{distance}_{\text{min}} = \text{distance}$
            }
        }
    } Color pixel according to intersection information
}
After all objects are tested

If (distance_{\text{min}} > \text{infinityThreshold}) 
  \begin{align*}
  \text{pixelColor} &= \text{background color} \\
  \text{else} \\
  \text{pixelColor} &= \text{color of object at distance}_{\text{min}} \text{ along ray}
  \end{align*}

CSE 681
Ray-Sphere Intersection - geometric

Knowns
C, r
Eye
Ray

t = |C-eye|
d+k = (C-eye) \cdot Ray

\[ t^2 = (k+d)^2 + s^2 \]

\[ r^2 = k^2 + s^2 \]

d = (k+d) - k

CSE 681
Ray-Sphere Intersection - algebraic

\[ x^2 + y^2 + z^2 = r^2 \]

\[ P(t) = \text{eye} + t*\text{Ray} \]

Substitute definition of \( p \) into first equation:

\[ (\text{eye.x+ t }*\text{ray.x})^2 + (\text{eye.y+ t }*\text{ray.y})^2 + (\text{eye.z+ t }*\text{ray.z})^2 = r^2 \]

Expand squared terms and collect terms based on powers of \( u \):

\[ A* t^2 + B* t + C = 0 \]
Ray-Sphere Intersection (cont’d)

For a sphere with its center at $c$

A sphere with center $c = (xc, yc, zc)$ and radius $R$ can be represented as:

$$(x-xc)^2 + (y-yc)^2 + (z-zc)^2 - R^2 = 0$$

For a point $p$ on the sphere, we can write the above in vector form:

$$(p-c).(p-c) - R^2 = 0 \quad \text{(note ‘.’ is a dot product)}$$

Solve $p$ similarly
Quadratic Equation

When solving a quadratic equation

\[ at^2 + bt + c = 0 \]

Discriminant:

\[ d = \sqrt{b^2 - 4ac} \]

And Solution:

\[ t_{\pm} = \frac{-b \pm d}{2a} \]
Ray-Sphere Intersection

\( b^2 - 4ac < 0 \) : No intersection
\( b^2 - 4ac > 0 \) : Two solutions (enter and exit)
\( b^2 - 4ac = 0 \) : One solution (ray grazes the sphere)

\[
d = \sqrt{b^2 - 4ac}
\]
Determine Color

FOR LAB #1

Use z-component of normalized normal vector

Clamp to [0.3..1.0]

objectColor*Nz

What’s the normal at a point on the sphere?