overview

- models
  - simple primitives (for now)
  - modeling transforms
  - hierarchical coordinates
- view system (eye/camera)
  - view volume
  - projection
  - image coordinates
- lights and material properties
- illumination models
  - local
  - global

pinhole camera model

projection
how is eye situated?

world coordinate system

position: \( P_0 \)
towards: \( t \)
up: \( u \)

right: \( r = tu \)

how much of the world is seen?

the view volume is a pyramid that is axis-aligned with the toward, up, and right vectors

or a truncated pyramid, called a frustum

view volume

we can specify the view volume by its height and width at some distance \( d \) from the viewpoint

or, we can specify the view volume by its height angle \( \theta_h \) and width angle \( \theta_w \)
view volume

specification recap
- viewpoint
  - position
  - toward, up, right
- view volume
  - height and width angles

projection

where is the view plane?

doesn't matter

image coordinates

specification recap
- viewpoint
  - position
  - toward, up, right
- view volume
  - height and width angles
- image size
  - height and width in pixels
overview

- models
  - simple primitives (for now)
  - modeling transforms
  - hierarchical coordinates
- view system (eye/camera)
  - view volume
  - projection
  - image coordinates
- lights and material properties
  - illumination models
    - local
    - global

objective

approximate the interaction of light and surface materials in a way that is
- consistent with our perception
- computationally efficient

light/material interaction

- light
  - incident angle
  - incident intensity

- material
  - angle of reflection
  - reflected intensity
  - angle of refraction
  - transmitted intensity

light/material interaction

- light
  - incident angle
  - incident intensity

- material
  - angle of reflection
  - reflected intensity

light/material interaction

- light
  - incident angle
  - incident intensity

- material
  - angle of reflection
  - reflected intensity

light/material interaction

- light
  - incident angle
  - incident intensity

- material
  - angle of reflection
  - reflected intensity
  - angle of refraction
  - transmitted intensity

light sources (in cs155)

- ambient light
- directional light
- point light
- spot light
ambient light

- "source-less" light
- incident angle: falls uniformly from every angle
- incident intensity: constant everywhere
  - specification
    - red, green, and blue intensity

directional light

- light positioned at "infinity"
- incident angle: constant everywhere
- incident intensity: constant everywhere subject to occlusion
  - specification
    - incident angle (direction)
    - red, green, and blue intensity

point light

- local light
- incident angle: depends on position of light and surface point
- incident intensity: drops off with distance
  - specification
    - location in world coordinates
    - red, green, and blue intensity
    - how the light drops off with distance

spot light

- local light
- incident angle: depends on position of light and surface point
- incident intensity: drops off with distance and angle
  - specification
    - location in world coordinates
    - direction
    - red, green, and blue intensity
    - how the light drops off with distance
    - how light drops off with angle from center

material properties

how does the surface material
- reflect light
- transmit light

diffuse reflection

rough/matte surface: light reflects uniformly in all directions
specular reflection
smooth (mirror) surfaces: light reflects in one (primary) direction

transparency
does surface transmit light? is transmitted light refracted?

hybrid light/material
- surface emits light

material property specification
1. red, green, and blue emission
2. diffuse reflectivity coefficient for red, green, and blue light
3. specular reflectivity coefficient for red, green, and blue light
4. transparency and refractive index
5. cheap tricks
6. ambient response (!)

overview
- models
  - simple primitives (for now)
  - modeling transforms
  - hierarchical coordinates
- view system (eye/camera)
  - view volume
  - projection
  - image coordinates
- lights and material properties
- illumination models
  - local
  - global

global illumination
local illumination

what do i see?

local illumination

direct illumination

what do i see?

local illumination models

ignore
- shadows
- reflections
- refraction

semi-local illumination

shadows

semi-local illumination

maybe multiple-hop reflections

semi-local illumination

simple reflections
**Semi-local Illumination**

- Transmission
- Maybe refraction

**Rendering**

- Image-based algorithms
  - Ray casting/tracing
- Object-based algorithms
  - Vertex pipeline

**Ray Casting**

- Cast ray through pixel into scene
- Find closest intersection (if any)
- Compute luminance at intersection
- Direct illumination
Ray Tracing:
- Cast ray through pixel into scene
- Find closest intersection (if any)
- Compute luminance at intersection
  - Direct illumination
  - Reflections
  - Transmission

Object-Based:
- Project each object
- Hidden surface removal

Vertex Pipeline:
- Project vertices of each polygon
- Turn on "inside" pixels
- Use hidden surface removal to resolve conflicts