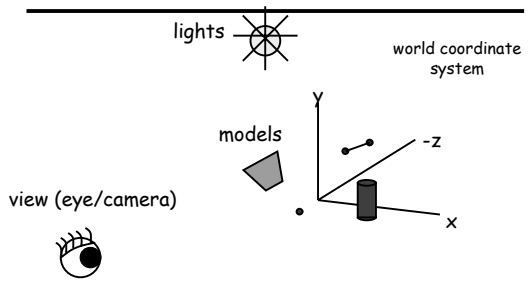
cs155 – z sweedyk

viewing

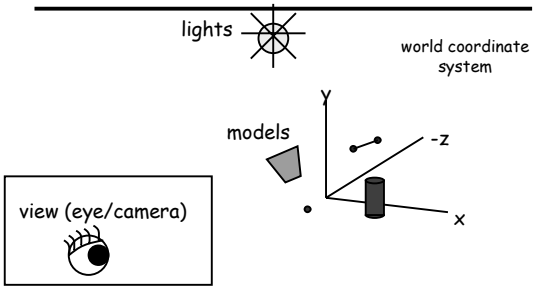
3d scene



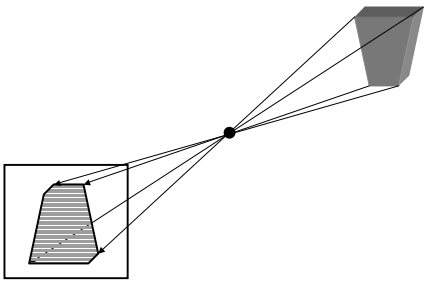
overview

- 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

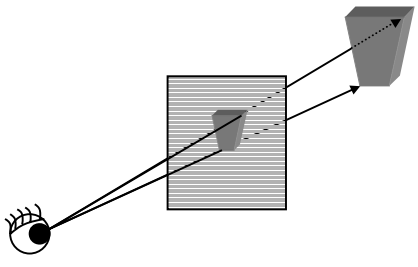
3d scene



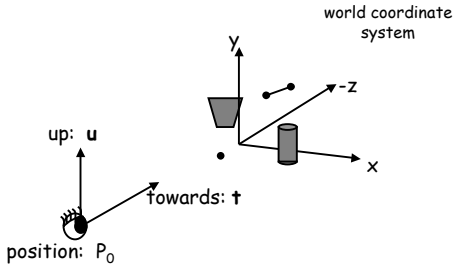
pinhole camera model



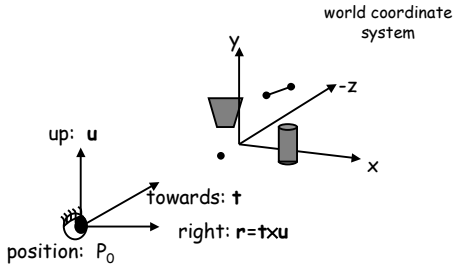
projection



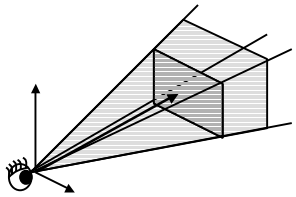
how is eye situated?



how is eye situated?

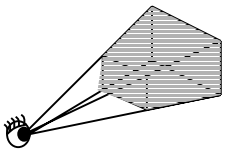


how much of the world is seen?



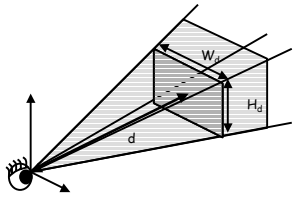
the *view volume* is a pyramid that is axis-aligned with the toward, up, and right vectors

how much of the world is seen?



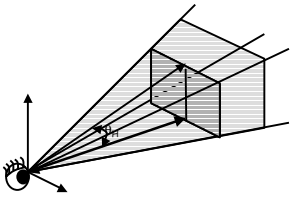
or a truncated pyramid, called a frustum

view volume



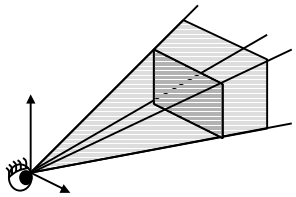
we can specify the view volume by its height and width at some distance d from the viewpoint

view volume



or, we can specify the the view volume by its height angle θ_h and width angle θ_w

view volume

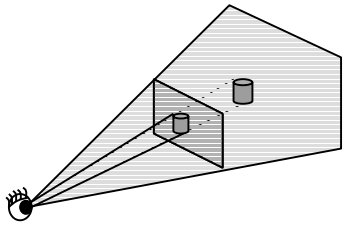


or …

specification recap

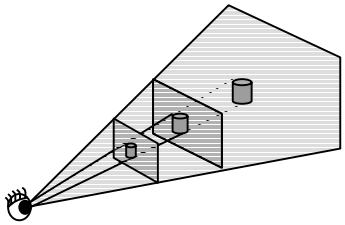
- viewpoint
  - position
  - toward, up, right
- view volume
  - height and width angles

projection



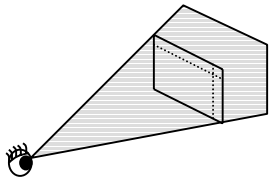
objects in the view volume are projected onto the *view plane*

where is the view plane?



doesn't matter

image coordinates



what matters is the sampling density;

image size is specified by height and width in pixels

specification recap

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

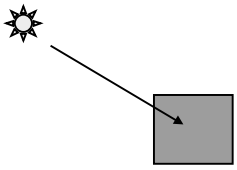
overview

- 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

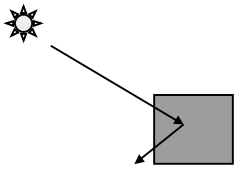
approximate the interaction of light and surface materials in a way that is
- consistent with our perception
- computationally efficient

light/material interaction



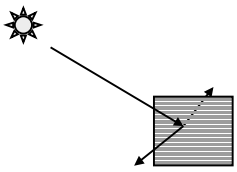
- light
  - incident angle
  - incident intensity

light/material interaction



- light
  - incident angle
  - incident intensity
- material
  - angle of reflection
  - reflected intensity

light/material interaction

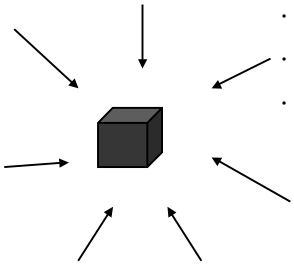


- light
  - incident angle
  - incident intensity
- material
  - angle of reflection
  - reflected intensity
  - angle of refraction
  - transmitted intensity

light sources (in cs155)

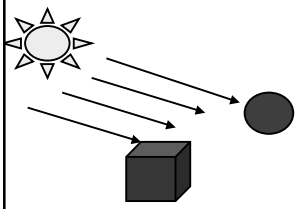
- ambient light
- directional light
- point light
- spot light

ambient light



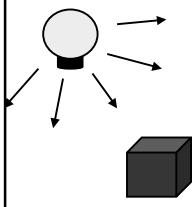
- "source-less" light
- incident angle: falls uniformly from every angle
- incident intensity: constant everywhere
- specification
  - red, green, and blue intensity

directional light



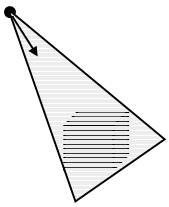
- light positioned at "infinity"
- incident angle: constant everywhere
- incident intensity: constant everywhere subject to occlusion
- specification
  - incident angle (direction)
  - red, green, and blue intensity

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



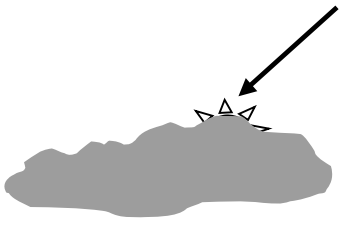
- local light
- incident angle: depends on position of light and surface point
- incident intensity: drops off with distance and angle
- specifications
  - location in world coordinates
  - direction
  - red, green, and blue intensity
  - how the light drops off with distance
  - how light drops off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

material properties

how does the surface material
- reflect light
- transmit light

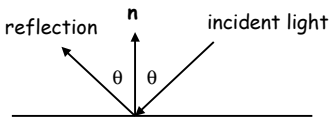
diffuse reflection

rough/matte surface: light reflects uniformly in all directions



specular reflection

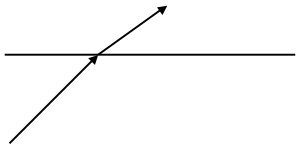
smooth (mirror) surfaces: light reflects in one (primary) direction



transparency

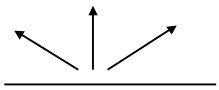
does surface transmit light?

is transmitted light refracted?



hybrid light/material

- surface emits light



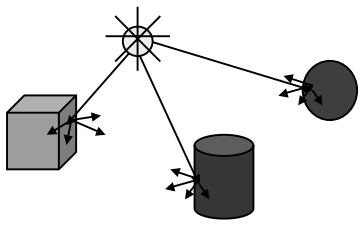
material property specification

1. red, green, and blue emission
2. diffuse reflectivity coefficent 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 (!)

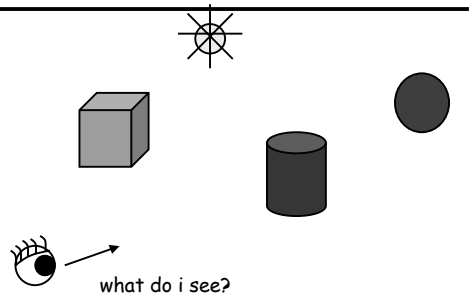
overview

- 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

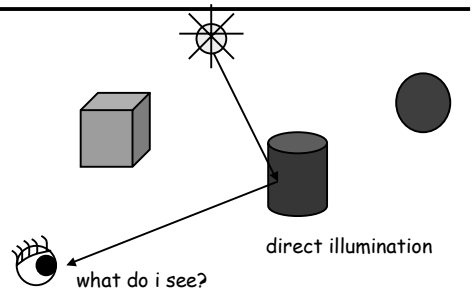
global illumination



local illumination



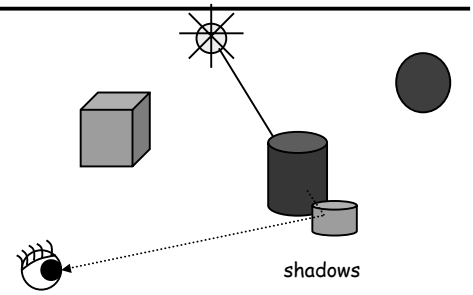
local illumination



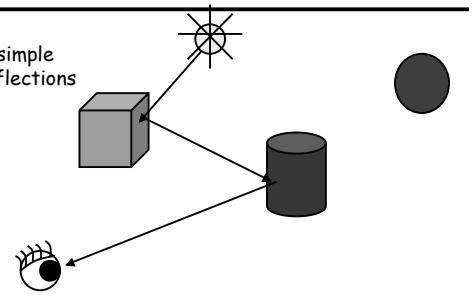
local illumination models

ignore
- shadows
- reflections
- refraction

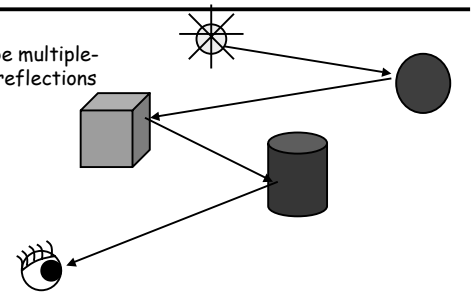
semi-local illumination



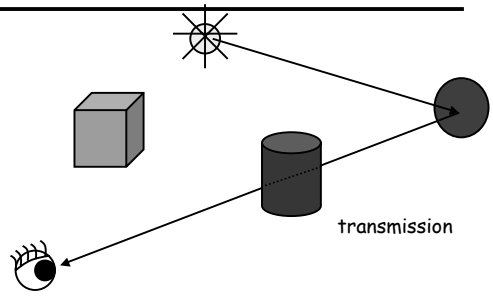
semi-local illumination



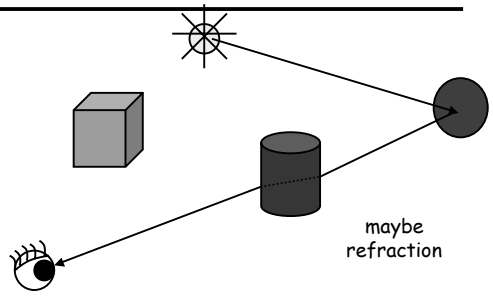
semi-local illumination



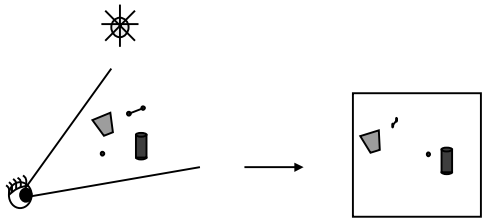
semi-local illumination



semi-local illumination



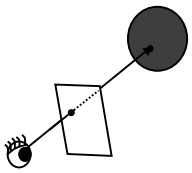
rendering



rendering

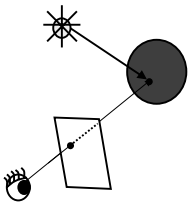
- image based algorithms
  - ray casting/tracing
- object-based algorithms
  - vertex pipeline

ray casting



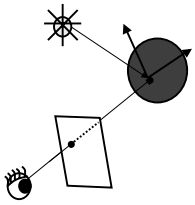
- cast ray through pixel into scene
- find closest intersection (if any)
- compute luminance at intersection

ray casting



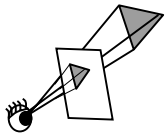
- cast ray through pixel into scene
- find closest intersection (if any)
- compute luminance at intersection
  - direct illumination

ray tracing



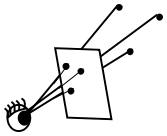
- cast ray through pixel into scene
- find closest intersection (if any)
- compute luminance at intersection
  - direct illumination
  - reflections
  - transmission

object-based



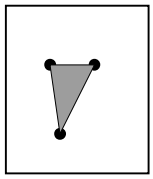
- project each object
- hidden surface removal

vertex pipeline



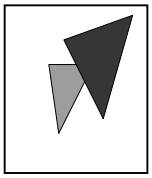
- project vertices each polygon

vertex pipeline



- project vertices of each polygon
- turn on "inside" pixels

vertex pipeline



- project vertices of each polygon
- turn on inside pixels
- use hidden surface removal to resolve conflicts