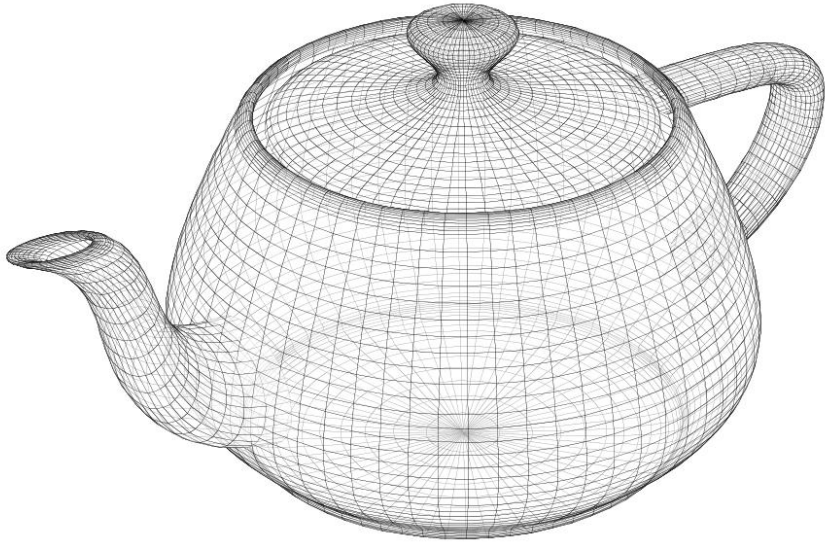


Mirror Reflection

Production Computer Graphics

Eric Shaffer

ILLINOIS

Objectives

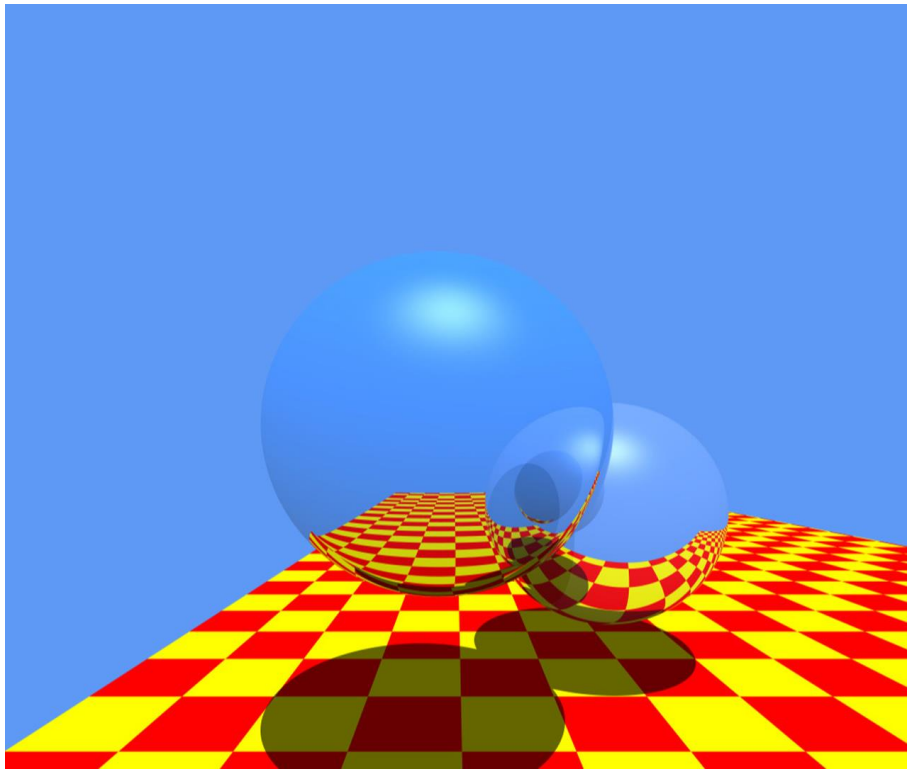
- Learn the core ideas behind the optics of mirrors
- Understand how mirror reflection can be modeled in ray tracing

Cloud Gate in Millennium Park - City of Chicago

ILLINOIS

Reflection

- Mirror reflection is an effect of indirect illumination
  - Up till now we have focused on direct illumination
- Recursive ray-tracing technique first developed by Whitted (1980)
  - you will still sometimes see the term “Whitted-style ray-tracing” used



ILLINOIS

Illumination Model

Indirect illumination can arrive from any direction

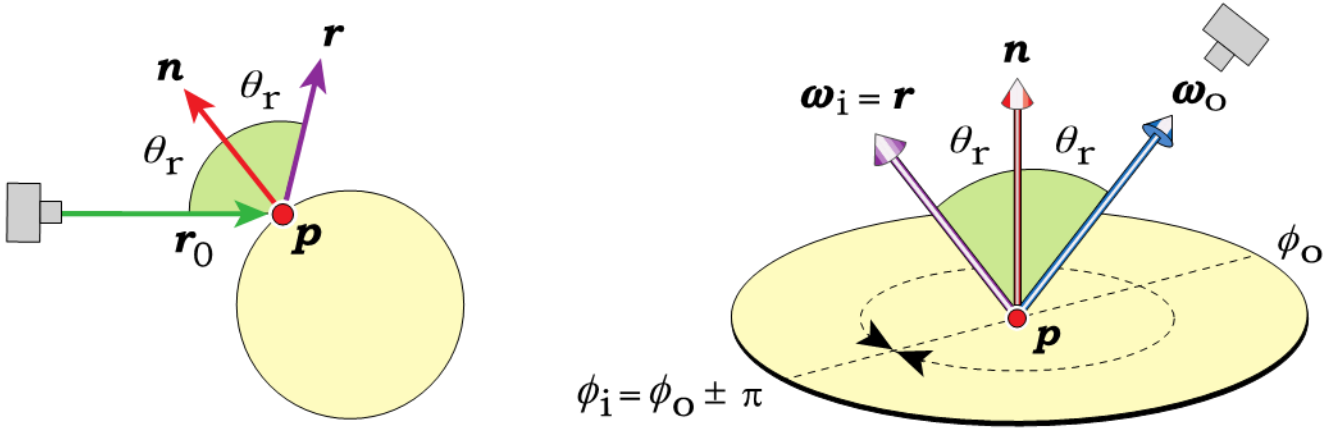
$$L_r(p, w_o) = L_{direct}(p, w_o) + L_{indirect}(p, w_o)$$

$$L_{indirect}(p, w_o) = \int_{2p^+} f_r(p, w_i, w_o) L_i(r_c(p, w_i), -w_i) \cos q_i \, dw_i$$

- L_r is reflected light
- L_i is incident radiance obtained by shooting ray r_c into hemisphere above p
- For now, we'll only consider light obtained in the mirror reflection direction

ILLINOIS

Perfect Mirror Reflection

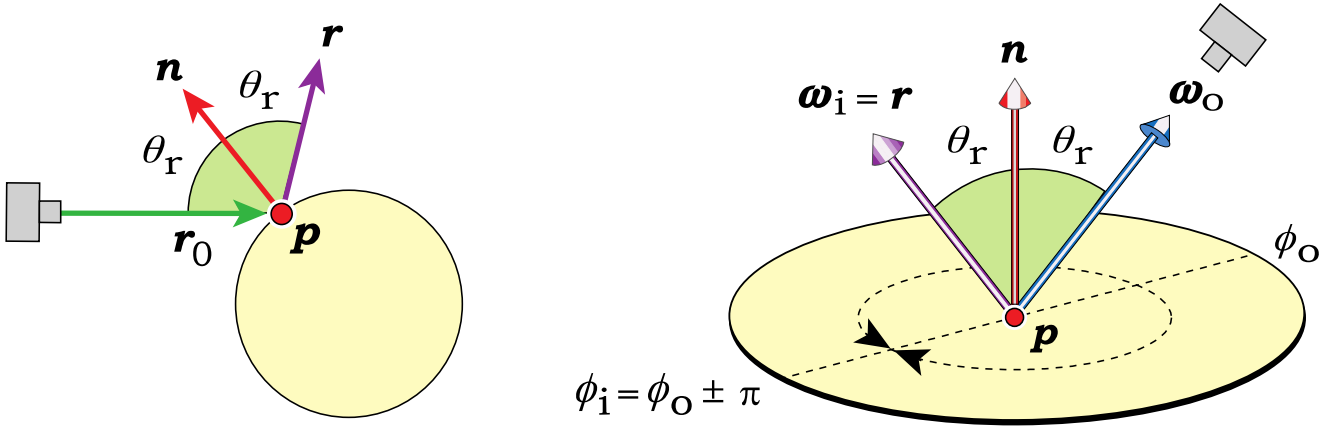


$$r = -w_o + 2(\mathbf{n} \times w_o)\mathbf{n}$$

$$q_i = q_o = q_r$$

ILLINOIS

Perfect Mirror Reflection

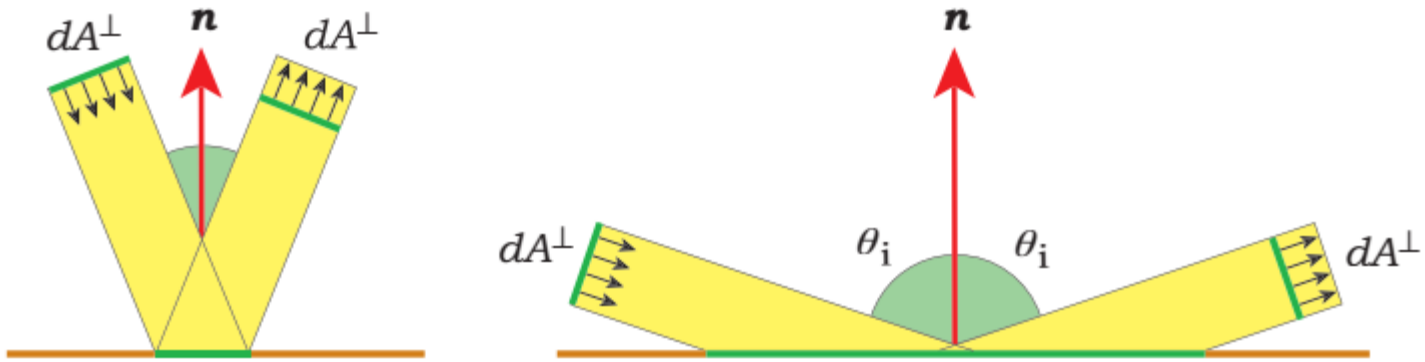


For mirror reflection, incident radiance comes from a single direction

$$L_{indirect}(p, w_o) = f_r(p, w_i, w_o) L_i(p, w_i)$$

ILLINOIS

Perfect Mirror Reflection



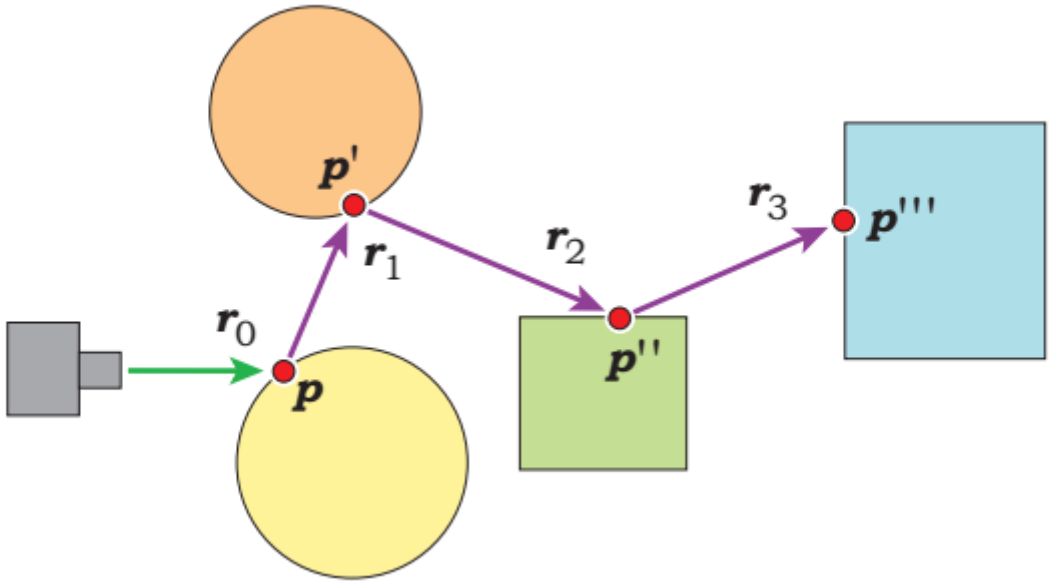
- Note that there is no cosine term for mirror reflection
- There is no scattering
- So the cross-sectional area is unchanged by reflection

$$L_{indirect}(p, w_o) = f_r(p, w_i, w_o) L_i(p, w_i)$$

$$L_{\text{indirect}}(\boldsymbol{p}, \boldsymbol{\omega}_{\text{o}}) = k_{\text{r}} \boldsymbol{c}_{\text{r}} L_{\text{i}}(\boldsymbol{p}, \boldsymbol{\omega}_{\text{i}})$$

ILLINOIS

Multiple Reflections and Recursion

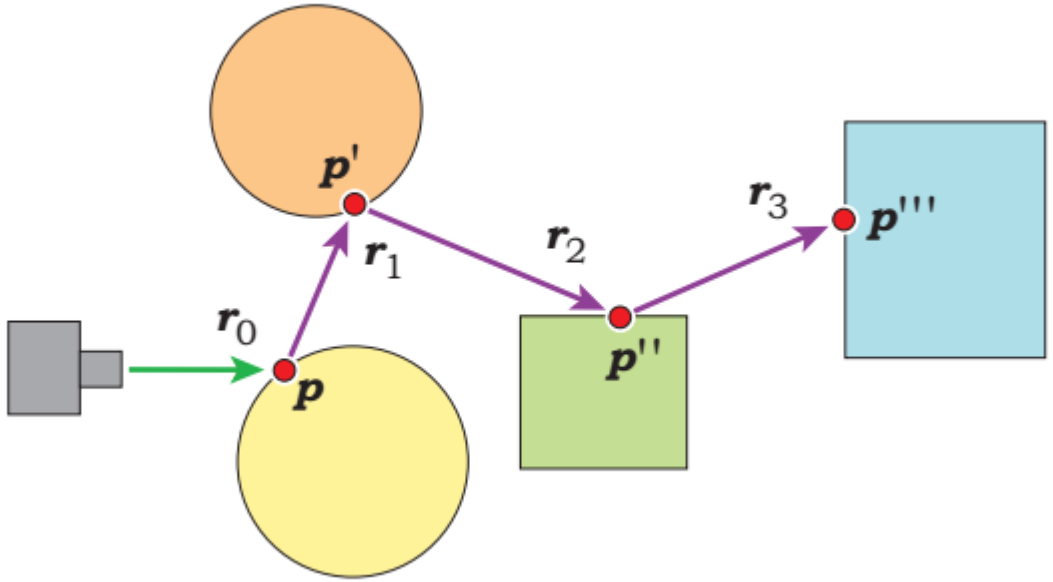


Your reflection ray can

- Hit nothing (returns background color)
- Hit a light (returns emitted light)
- Hit a non-reflective object at point p’
  - Need to compute and return direct illumination at p’
- Hit a reflective object at p’
  - Compute direct illumination at p’ plus indirect, recurse into the scene…

ILLINOIS

Multiple Reflections

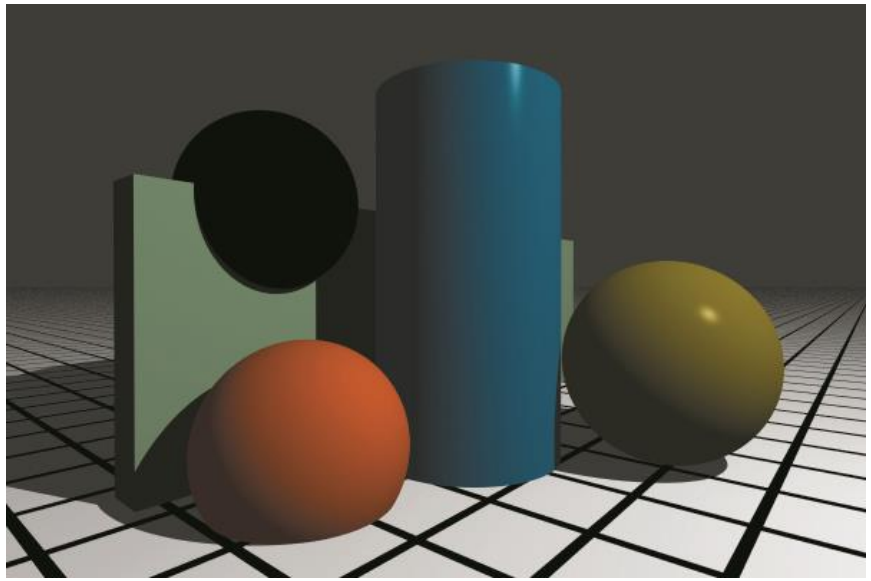


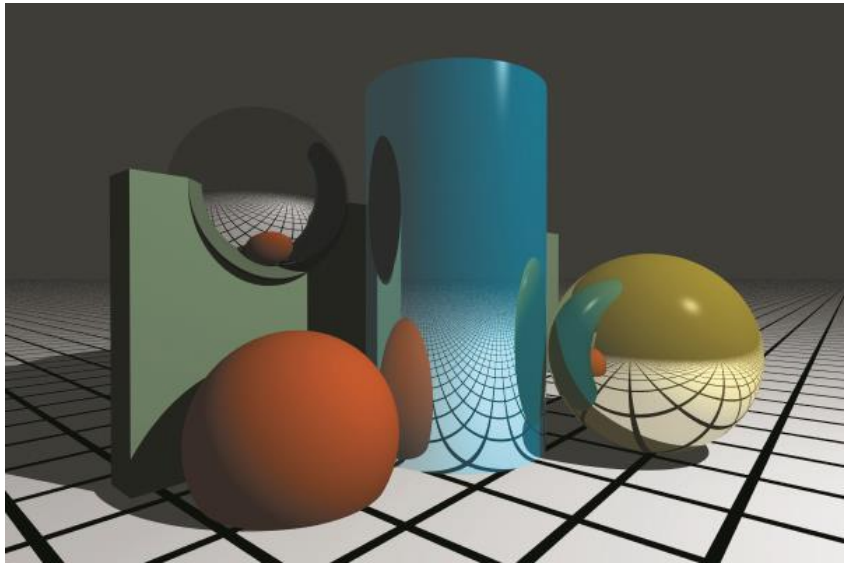
Hit a reflective object at p’

- Compute direct illumination at p’
  - For area lights, this means Monte Carlo integration
- Compute indirect illumination at p’ recursively
  - Need to use a recursion depth limit to stop at some point
- Add the two together and return

ILLINOIS

Examples



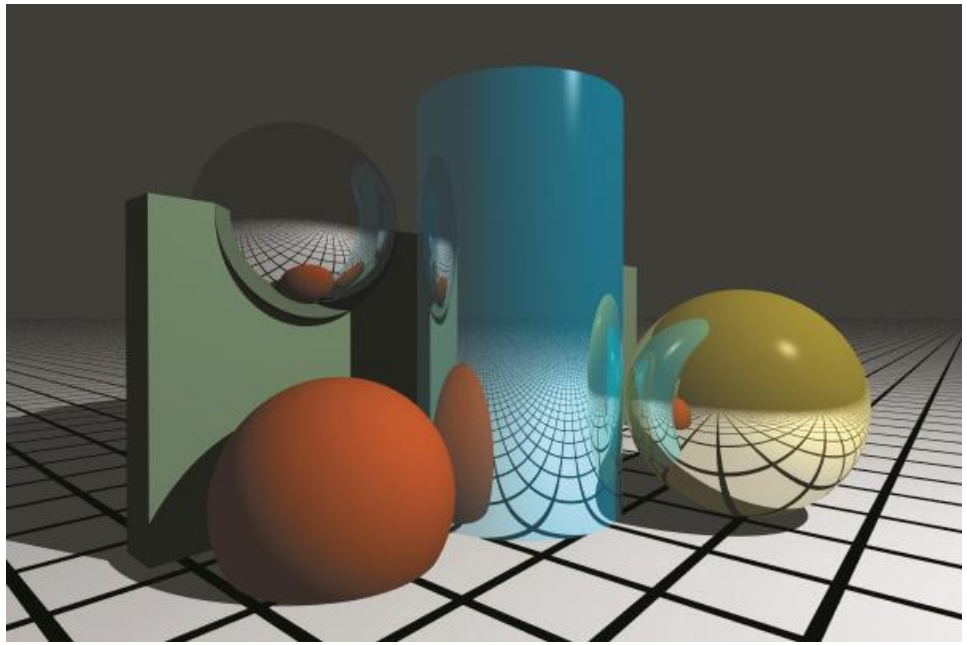


black sphere reflects no direct illumination

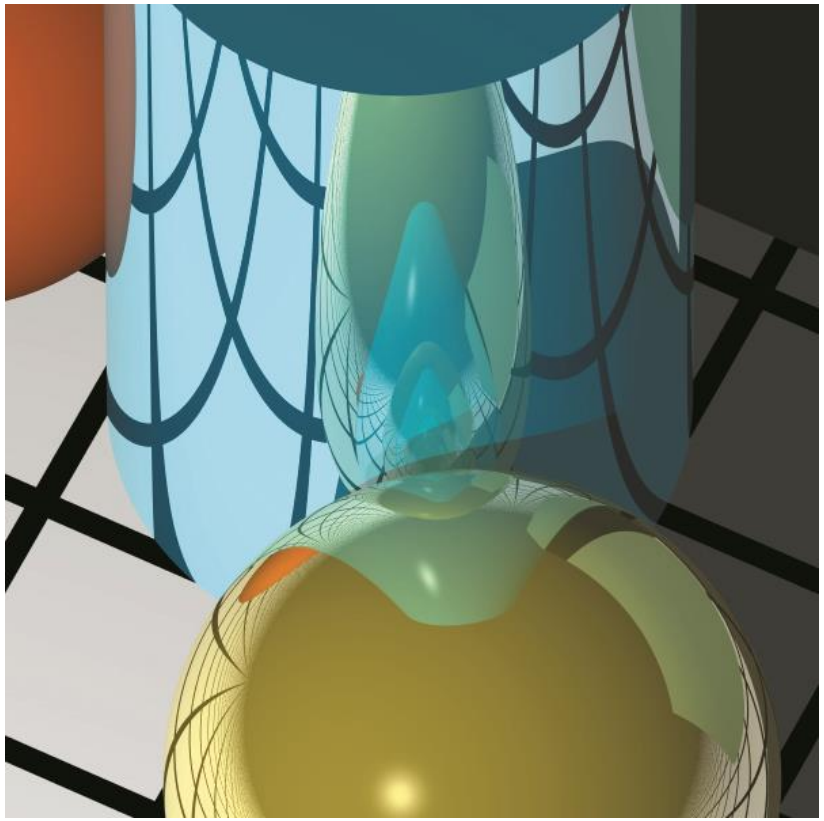
shows difference between recursion depth 0 and recursion depth 1

ILLINOIS

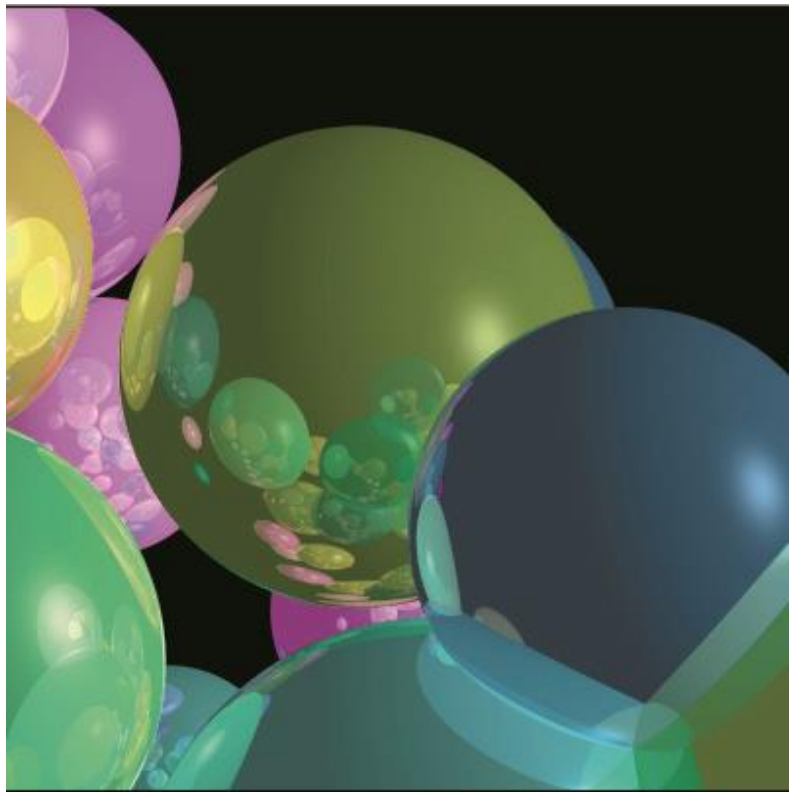
Examples



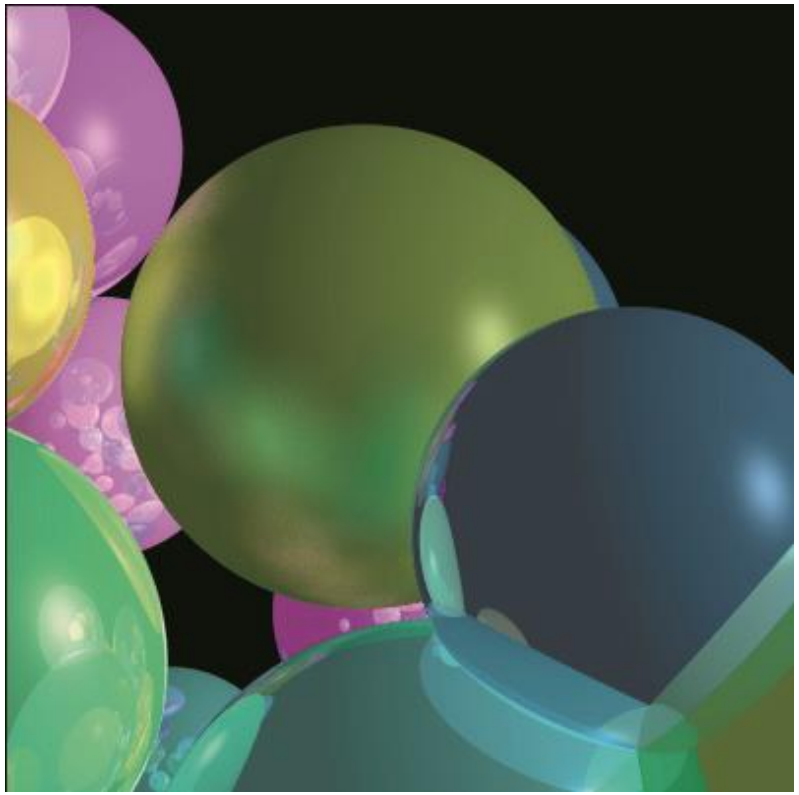
recursion depth = 10



Specular Inconsistency



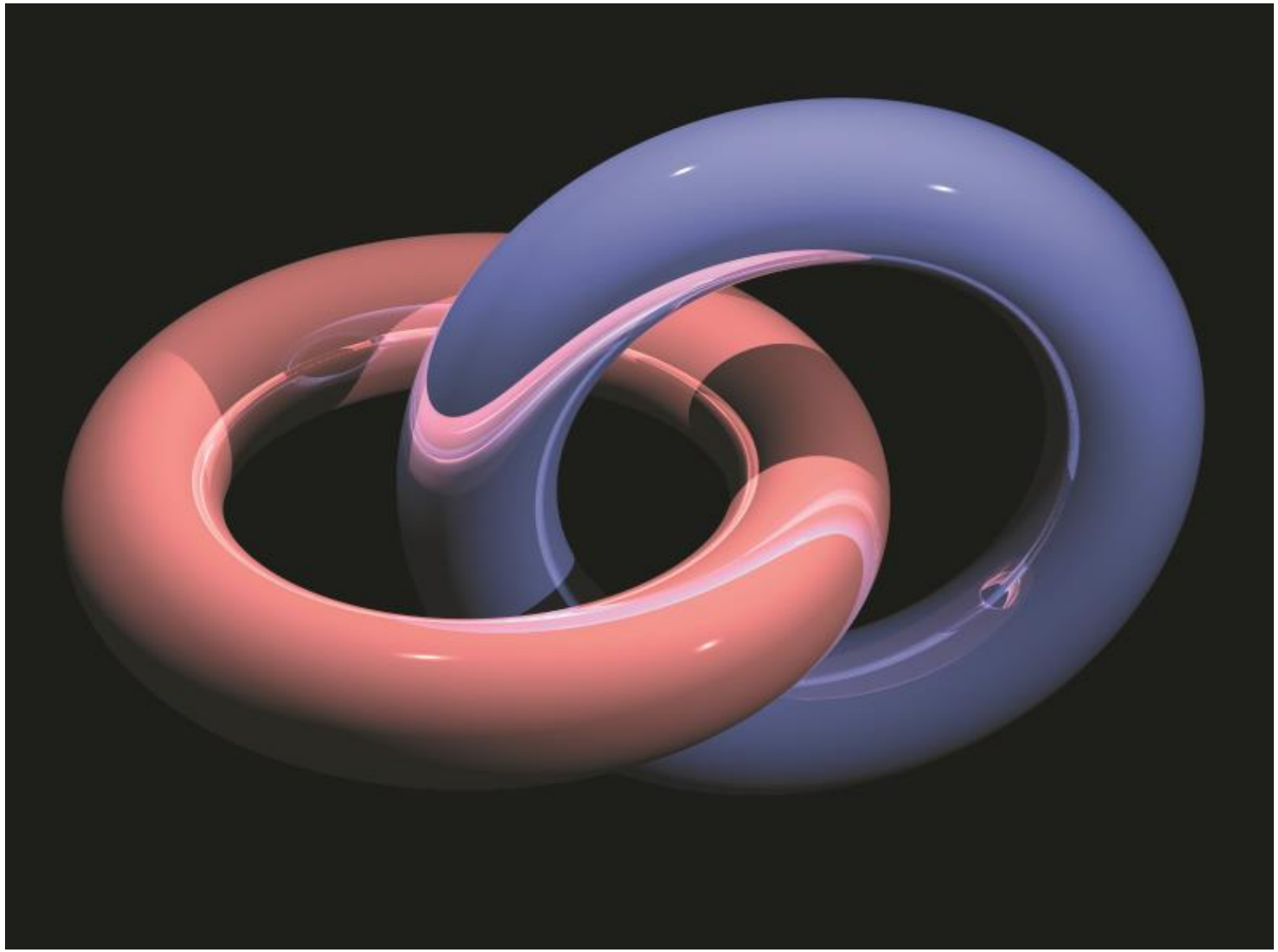
A mirror should not have perfect reflection and specular highlights (which implies scattering)



A correct version with glossy specular reflection

ILLINOIS

Self-Reflection



ILLINOIS

Reflective Meshes

Flat shading versus smooth change the appearance

ILLINOIS

Reflective Meshes

Low poly

High poly

Flat shading versus smooth change the appearance

ILLINOIS