LIGHTING: ADDITIONAL ALGORITHMS
(1) render scene from light source; store the z-buffer
(2) render scene from camera view; p is in shadow if \( d_{\text{light}} > d_{\text{blocker}} \)

Issues: resolution of the shadow map image? No soft shadows.

During step (2), consider a fragment \( p \):
- compute \( p' = m_{\text{eye} \rightarrow \text{light}} \cdot p \)
then: \( d_{\text{light}} = -p'_z \), \( d_{\text{blocker}} = \text{zbuffer}[p'_x, p'_y] \)
AMBIENT OCCLUSION

Assume that light is coming from all directions.
For a given point, compute the area of the surrounding hemisphere that is open.
Precompute and store this.

http://www.redway3d.com/
“RADIOSITY” METHOD FOR GLOBAL ILLUMINATION

Form factor $F_{ij}$: Fraction of light leaving surface $i$ and arriving at surface $j$. This depends on the shape, distance, orientation, and relative occlusions of the two surface patches.

Solve a set of simultaneous linear equations for the unknown energies.

Assumes all surfaces are Lambertian, i.e., diffuse. Requires subdivision of scene into patches. The $n^2$ form factors are expensive to compute.

Radiosity: A ray of light that hits a surface is reflected by multiple diffuse rays, which can themselves illuminate other surfaces. Surfaces are subdivided to increase accuracy of the solution.

diffuse $\leftrightarrow$ diffuse transfer
LIGHT BAKING

• precompute and “bake” into texture:
  static lighting of diffuse surfaces

[https://blogs.unity3d.com/2017/03/31/]
PHOTON MAPPING

• trace light paths, “photons”, forward into scene, until they strike a diffuse surface.

• store locations and incoming directions of photons on the diffuse surface.

• efficient rendering of “caustics”

[Henrik Wann Jensen]
PATHTRACING

• global illumination
• trace paths from eye into scene
• Monte-Carlo sampling of directions
• multiple diffuse bounces
• average many sample rays per pixel
• “noisy” images with few samples
  • ML with deep nets to remove noise
RAY TRACING VS PATH TRACING

• Global illumination algorithms
• Rays emitted FROM camera

• Ray Tracing
  • Single ray per pixel
  • Supports indirect lighting only from specular surfaces
    • No color bleeding
  • Shoots shadow rays to compute direct illumination
    • Soft shadows are harder to get

• Path Tracing *(may produce renders indistinguishable from photos)*
  • Many rays per pixel, their color averaged
  • At each interaction, ray direction changes randomly with some distribution
  • No difference between light sources and objects
    • Soft shadows, complex materials, etc.
    • Supports all sorts of indirect lighting