Advanced Ray Tracing

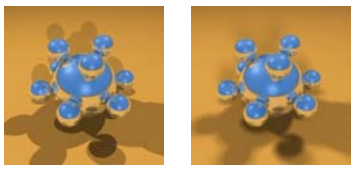
Distributed Ray Tracing

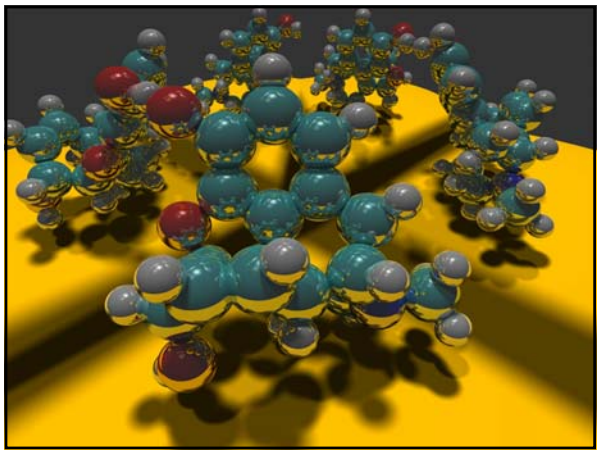
- *Distributed ray tracing* is an elegant technique that tackles many problems at once
  - Stochastic ray tracing: distribute rays stochastically across pixel
  - Distributed ray tracing: distribute rays stochastically across everything
- Distributed ray tracing is basically a Monte Carlo estimation technique

Distributed Ray Tracing

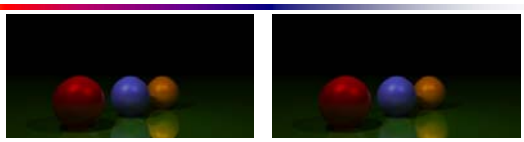
- Distribute rays stochastically across:
  - *Pixel* for antialiasing
  - *Light source* for soft shadows
  - *Reflection function* for soft (glossy) reflections
  - *Time* for motion blur
  - *Lens* for depth of field
- Cook: 16 rays suffice for all of these

- Soft Shadows

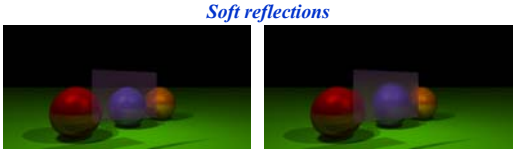




Soft reflections/refractions

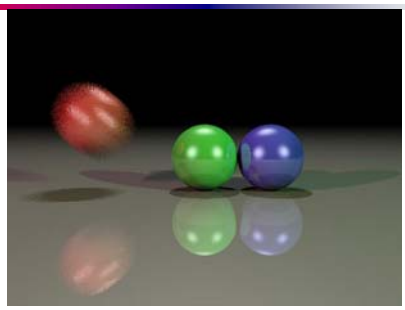


Soft reflections



Soft refractions

Motion Blur



Motion Blur

Depth of Field

Backwards Ray Tracing

- Traditional ray tracing traces rays from the eye, through the pixel, off of objects, to the light source
- *Backwards ray tracing* traces rays from the light source, into the scene, into the eye
- *Why might this be better?*

Backwards Ray Tracing

- Backwards ray tracing can capture:
  - Indirect illumination
  - Color bleeding
  - Caustics

Backwards Ray Tracing

- Usually implies two passes:
  - Rays are cast from light into scene
  - Rays are cast from the eye into scene, picking up illumination showered on the scene from the first pass

Backwards Ray Tracing

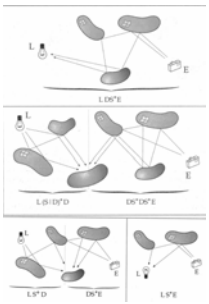
- Q: *How might these two passes "meet in the middle?"*

Backwards Ray Tracing

- Arvo: *illumination maps* tile surfaces with regular grids, like texture maps
  - Shoot rays outward from lights
  - Every ray hit deposits some of its energy into surface's illumination map
    - Ignore first generation hits that directly illuminate surface (*Why?*)
  - Eye rays look up indirect illumination using bilinear interpolation

Different Paths = Different Methods

- Consider all light paths: L(D|S)*E
- These can be broken in different sets, and each set solved separately
  - LDS*E: Eye ray tracing – direct illumination
  - L(S|D)*D: Light path tracing
  - DS*DS*E: Extended eye ray tracing
  - LS+D: *Caustic* paths
  - DS*E: Eye ray tracing – indirect illumination
  - Many other possibilities



Advanced Ray Tracing Wrapup

- Backwards ray tracing accounts for indirect illumination by considering more general paths from light to eye
- Distributed ray tracing uses a Monte Carlo sampling approach to solve many ray-tracing aliasing problems