An Improved Illumination Model for Shaded Display & Distributed Ray Tracing

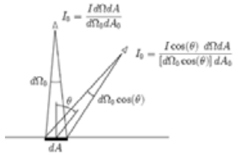
Published 1980 and 1984, respectively

Overview of Improved Illumination

- Previous shading models used local aggregate data rather than global data
- New shading model uses global data to calculate intensities
- Can be extended to assist in ray tracing

Previous work, in increasing order of complexity

- Lambert's cosine law


$I_0 = \frac{I\, d\Omega dA}{d\Omega_0 dA_0}$
$I_0 = \frac{I\cos(\theta)\ d\Omega dA}{[d\Omega_0 \cos(\theta)]\, dA_0}$
$d\Omega_0$
θ
$d\Omega_0 \cos(\theta)$
dA

- Phong model $I = I_a + k_d \sum_{j=1}^{j=ls} (\bar{N}\cdot\bar{L}_j) + k_s \sum_{j=1}^{j=ls} (\bar{N}\cdot\bar{L}_j')^n,$
- Blinn and Newell
- Kay (refraction model)

Improved Model

- Use classical optics to calculate reflection and diffusion

$$I = I_a + k_d \sum_{j=1}^{j=ls} (\bar{N}\cdot\bar{L}_j) + k_s S + k_t T,$$

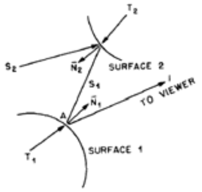
S = the intensity of light incident from the $\bar{R}$ direction
k_t = the transmission coefficient,
T = the intensity of light from the $\bar{P}$ direction.

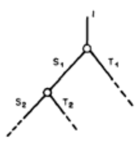
- Ideally, ks and kt would be functions of Fresnel reflection law
- Here they are used as coefficients. If ks is smaller and kt larger, surface is glossy
- Random perturbations added to simulate roughened surface

Improved Model

- Simulate reflections from multiple surface by building a tree, recursively follow all branches, applying surface shading

s





- Can be used to find which areas are in

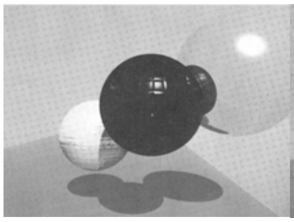
Visible Surface Processor

- Used for ray tracing, sends rays from viewer rather than from light source
- When a ray hits an object, new rays are created by diffusion towards light source
- Cannot clip background objects – might be caught in a reflection (use bounding box)

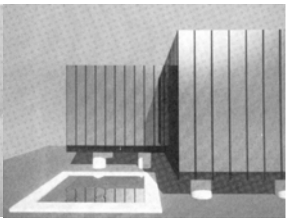
Visible Surface Processor

- Use spherical bounding boxes in a hierarchy
- Low-pass filter regions in danger of aliasing
- Pixel described by four point square
- Get intensity by either interpolation, or, for large differences, subdivision into more squares

Results



44 minutes on VAX-11/780
(Runs at 1 MIPS, 1977 model)



Time not given

Future Work

- Diffuse reflection from distributed light sources
- Better handling of specular reflections
- Overall, rather inefficient

Overview of Distributed Ray Tracing

- Ray tracing is limited to sharp images
- Distributing rays is an easy way to get fuzzy images
- Effects such as motion blur become possible

Previous Work

- Fuzzy samples would have previously required a great deal of oversampling for each ray
- Ray tracing was limited to sharp images and shadows

New Model

- Distribute rays rather than add more
- Makes heavy use of antialiasing; this makes it possible to sample motion and shading
- Shading with rays distributed according to formula (with some simplifications):

$$I(\phi_r,\theta_r) = \int_{\phi_i}\int_{\theta_i} L(\phi_i,\theta_i)R(\phi_i,\theta_i,\phi_r,\theta_r)\,d\phi_i\,d\theta_i$$

New Model

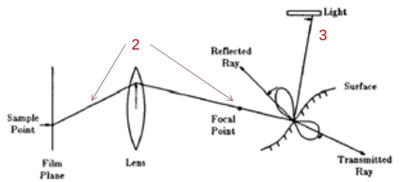
- Gloss (blurred reflections) created by distributing new rays caused by reflections
- Translucency much in the same way, but with transmittance
- Penumbras (caused by partially obscured light sources) by distributing rays traced from surface to light source

New Model

- For depth of field (objects out of focus), distribute initial rays from a single point to being across the “lens”
- For motion blur, distribute the rays being traced across discrete time steps as an object moves through the scene
- Use antialiasing to prevent strobing of motion blurred objects

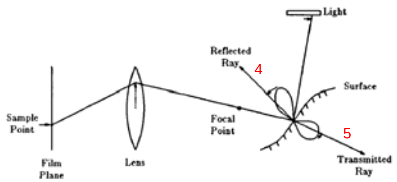
Algorithm

- 1. Choose time for ray, move objects in scene
- 2. Make ray from lens to screen, and from ray to focal point of lens, find what is visible
- 3 Trace ray from point on light source to vi

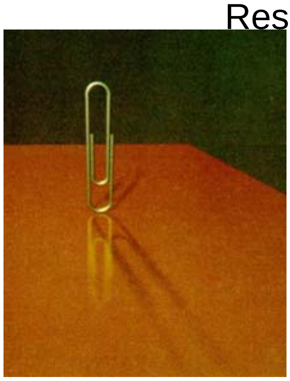


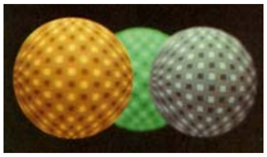
Algorithm

- 4. For reflection, distribute around mirror reflection, trace ray from that point to visible point. # rays ~ amount of light from that direction
- 5. Same for transmitted light



Results

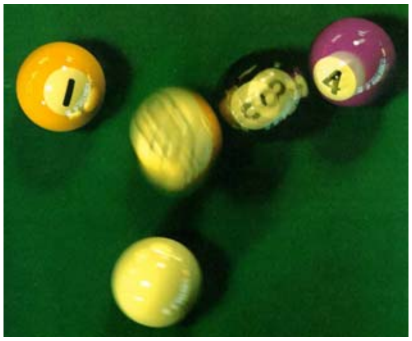




Depth of Field (35mm, f2.8, ray tracing)

Reflection, Shading, and Penumbra (scanline)

Results



Motion Blur (ray traced)

Questions Posed w.r.t. Improved Illumination

- How are S, T in Eq. 2 determined? Does a ray need to intersect a light source to transmit?
- Does use of bounding sphere create problems for higher resolutions/smaller objects?
- Why draw rays from viewer/objects to light?
- Why do we still use Phong if Blinn is better? Is something wrong with refraction in Fig. 7?

Questions Posed w.r.t Distributed Ray Tracing

- What are diffraction effects in DOF? Is there a better method for using it for ray tracing?
- Do real-time applications currently use this kind of DOF algorithm?
- Is treating anti-aliasing as a black box the best we can do? Or can we adaptively change sample rate?