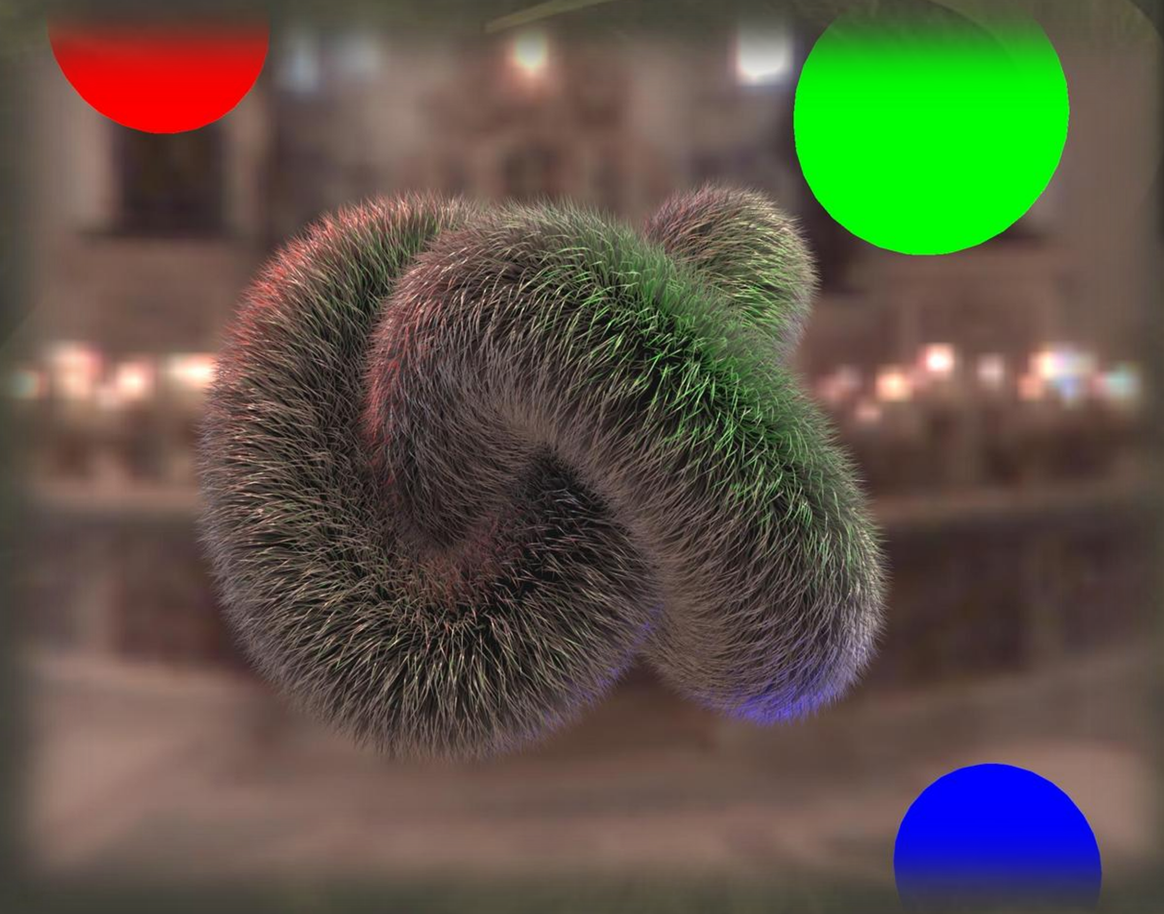


SIGGRAPH2010
The People Behind the Pixels

Fast Furry Ray Gathering

Ivan Neulander
Rhythm & Hues

Objective

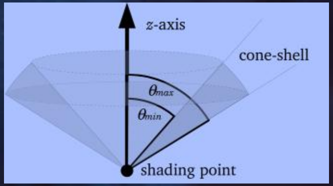
- Render fur convincingly:
  - Lighting interacts with geometry in the scene
  - Objects and environment maps serve as light sources
- Rendering must be fast and art-directable

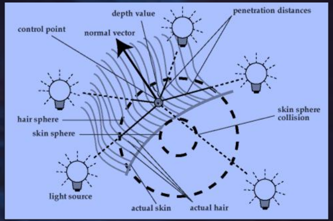
Objective

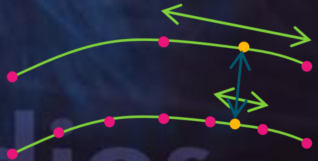
- Our approach: “ray gathering”
  - Sample incident radiance along many directions
  - Weight it by material’s BRDF
  - Determine reflected radiance
- Furry creatures deform ➔ can’t precompute
- How do we do this *efficiently*?

Overview

1. Hair Scattering Model
   - Concisely generates gather rays
2. Volumetric Occlusion Model
   - Eliminates ray-hair intersections
3. Reflection Cache
   - Evaluate fewer shading samples







1) Hair Scattering Model

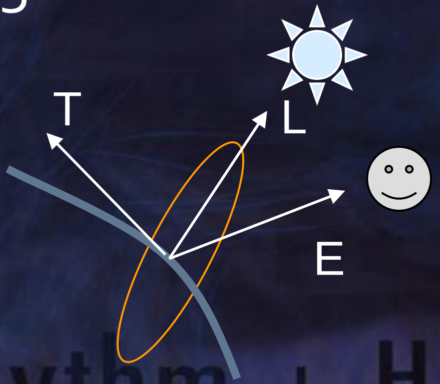
- Kajiya-Kay model [SIGGRAPH ’89]
  - Treats hair strand as an infinitesimally thin cylinder
  - Consists of separate diffuse, specular submodels
  - Describes reflectance in terms of hair tangent

1) Hair Scattering Model: Kajiya-Kay Diffuse Reflectance

Diffuse $= \sin(\mathbf{T},\mathbf{L}) = \mathrm{sqrt}(1 - \mathbf{T} \cdot \mathbf{L}^2)$

$= \pi\ \mathrm{avg}(\max(\mathbf{N} \cdot \mathbf{L}, 0))$ all around **T**

$\propto$ average Lambert reflectance around strand



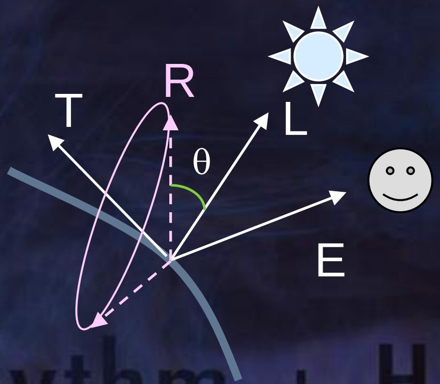
1) Hair Scattering Model: Kajiya-Kay Specular Reflectance

$$\text{Specular} = [\sin(\mathbf{T},\mathbf{L})\,\sin(\mathbf{T},\mathbf{E}) - (\mathbf{T}\cdot\mathbf{L})(\mathbf{T}\cdot\mathbf{E})]^{p}$$

$$= [\mathrm{sqrt}(1 - \mathbf{T}\cdot\mathbf{L}^2)\,\mathrm{sqrt}(1 - \mathbf{T}\cdot\mathbf{E}^2) - (\mathbf{T}\cdot\mathbf{L})(\mathbf{T}\cdot\mathbf{E})]^{p}$$

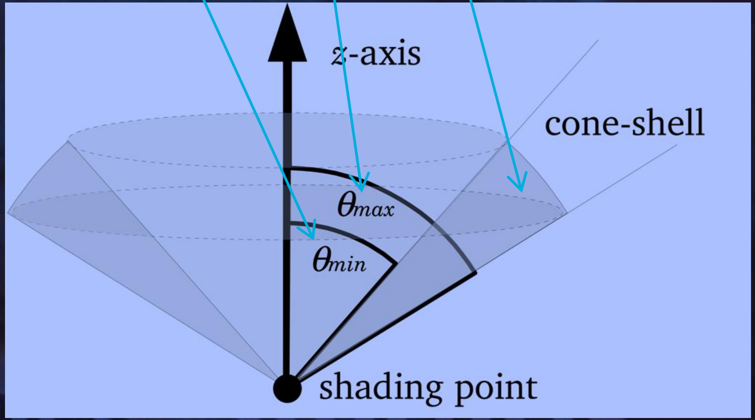
$$= \cos^{p}\theta$$

- *p* is a phong-style exponent



1) Hair Scattering Model: Our *Cone-Shell* Model

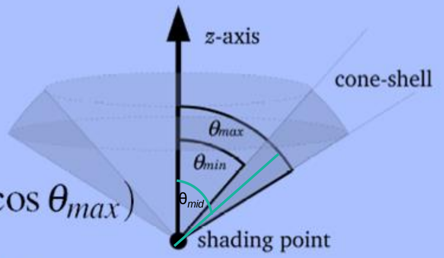
- Defined by a pair of concentric cones about *z*-axis, bounded by: θ_{min}, θ_{max}
  - Solid angle between the cones is the "Cone-Shell"
- Thickness of cone-shell defines blurriness of reflection



1) Hair Scattering Model: Our *Cone-Shell* Model

- U(0,1) pair ξ_1, ξ_2 generates ray in direction (x,y,z) with BRDF *weight*

$$
\begin{aligned}
x &= \sqrt{1-z^2}\cos(2\pi\xi_2) \\
y &= \sqrt{1-z^2}\sin(2\pi\xi_2) \\
z &= \cos\theta_{max} + \xi_1\,(\cos\theta_{min} - \cos\theta_{max}) \\
weight &= z\cos\theta_{mid} + \sqrt{1-z^2}\sin\theta_{mid}
\end{aligned}
$$



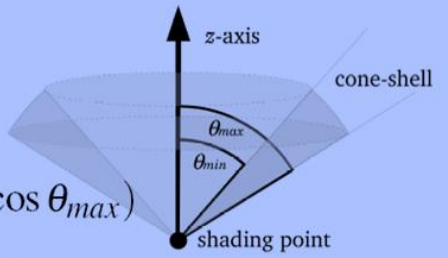
- Intuition for weight:
  - cosine of angle ray makes with middle of cone-shell ($\theta_{mid} = \theta_{(min+max)/2}$)
  - $= \cos(\theta_{mid} - \theta_{ray})$
  - $= \cos\theta_{mid}\cos\theta_{ray} - \sin\theta_{mid}\sin\theta_{ray}$
  - $= z\cos\theta_{mid} - (1 - z^2)^{1/2}\sin\theta_{mid}$

1) Hair Scattering Model: Our *Cone-Shell* Model

- At render time, we express these in terms of
  - E = eye vector
  - T = tangent vector
  - β = blur angle radius

- For diffuse, $\cos\theta_{mid} = 0$

$$
\begin{aligned}
x &= \sqrt{1-z^2}\cos(2\pi\xi_2)\\
y &= \sqrt{1-z^2}\sin(2\pi\xi_2)\\
z &= \cos\theta_{max} + \xi_1\left(\cos\theta_{min} - \cos\theta_{max}\right)\\
weight &= z\cos\theta_{mid} + \sqrt{1-z^2}\sin\theta_{mid}\\
\cos\theta_{mid} &= -\mathbf{E}\cdot\mathbf{T}\\
\sin\theta_{mid} &= \sqrt{1-\cos^2\theta_{mid}}
\end{aligned}
$$

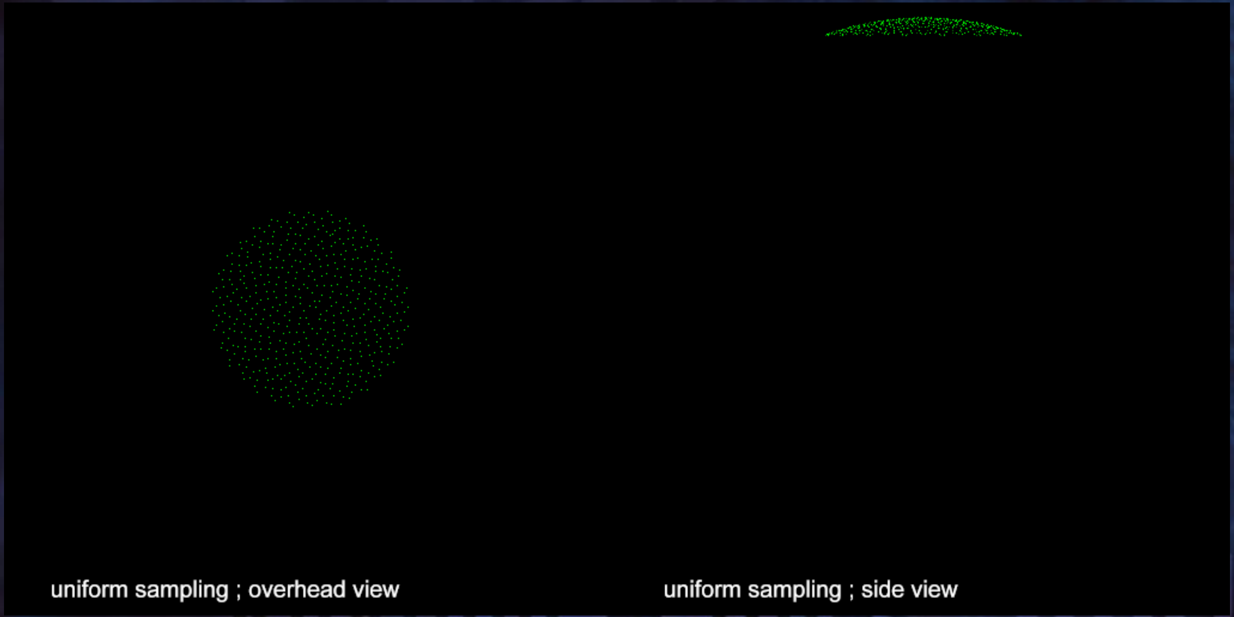


- We taper weight for $\beta < \pi/2$

$$weight := weight + \frac{\cos\beta}{\cos\beta - 1}(1 - weight)$$

1) Hair Scattering Model: Our *Cone-Shell* Model

- Specular Cone-Shell (500 rays, $\beta = 20^\circ$)



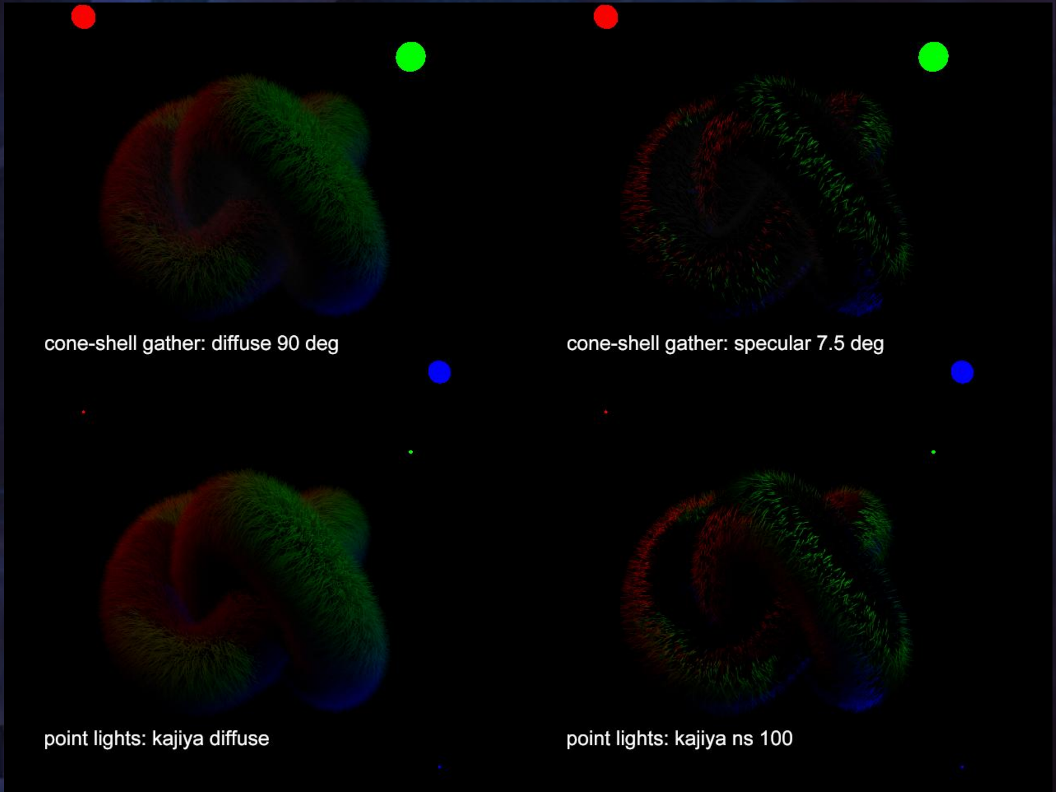
1) Hair Scattering Model: Our *Cone-Shell* Model

- Compare:

  Area-lit cone-shell

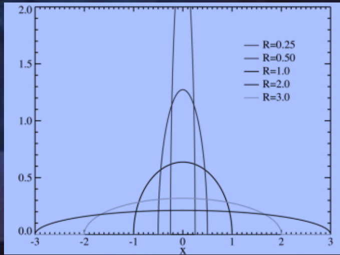
  vs

  Point-lit Kajiya-Kay
  - Diffuse
  - Specular



1) Hair Scattering Model: BRDF Importance Sampling

- IS → Draw samples with probability proportional to BRDF weight
  - Essential for $\cos^n$ lobe, less for cone-shell
- "ideal" pdf is [Wigner Semicircle Distribution]
  - Ideal only if incoming radiance is uniform
  - We chose a *raised triangle distribution*

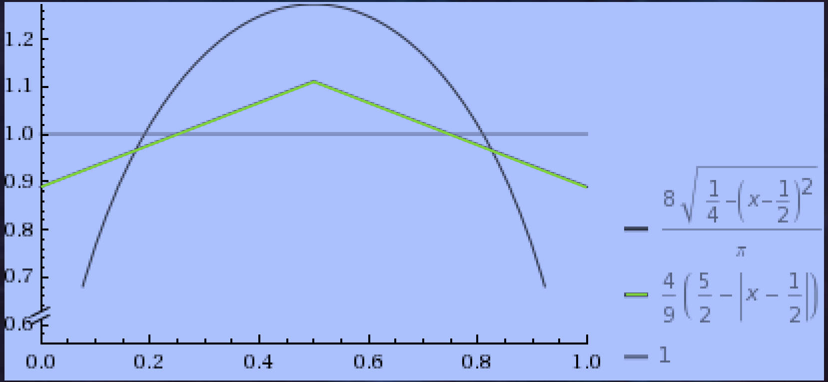


1) Hair Scattering Model: BRDF Importance Sampling

- Raised Triangle Distribution:

$$\frac{4}{9}\left(\frac{5}{2} - \left|x - \frac{1}{2}\right|\right)$$

pdf



- How to generate:

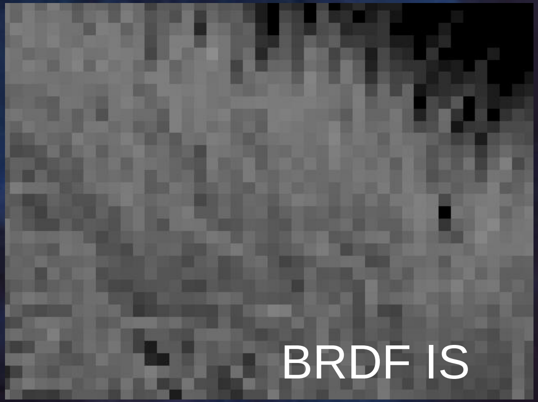
$$\xi_1 = \begin{cases} \sqrt{\frac{1}{2}(8+9\xi)} - 2 & : \quad \xi \le 1/2 \\ 3 - \sqrt{\frac{1}{2}(17-9\xi)} & : \quad \xi > 1/2 \end{cases}$$

1) Hair Scattering Model: BRDF Importance Sampling

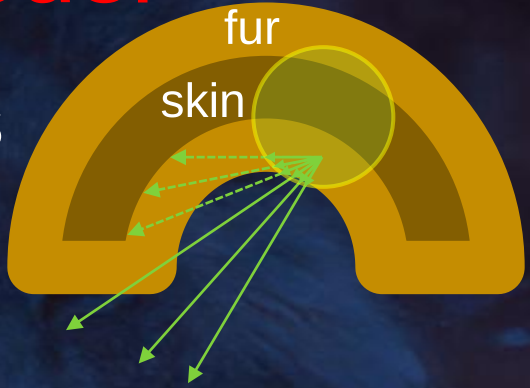
- Diffuse reflection of white sphere (30 rays/pixel)

uniform sampling

importance sampling



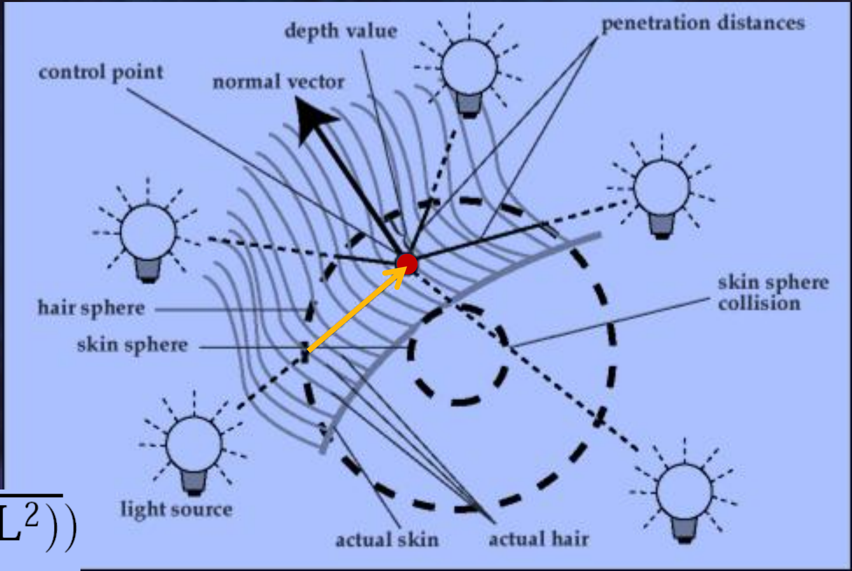
2) Volumetric Occlusion Model



- Goal: omit ray-hair intersections
- Assume hair is short and uniformly covers skin
- Compute each ray’s fractional occlusion using volumetric local occlusion model
- Supplement occlusion estimate with accurate ray-traced occlusion to skin geometry

2) Volumetric Occlusion Model: Details

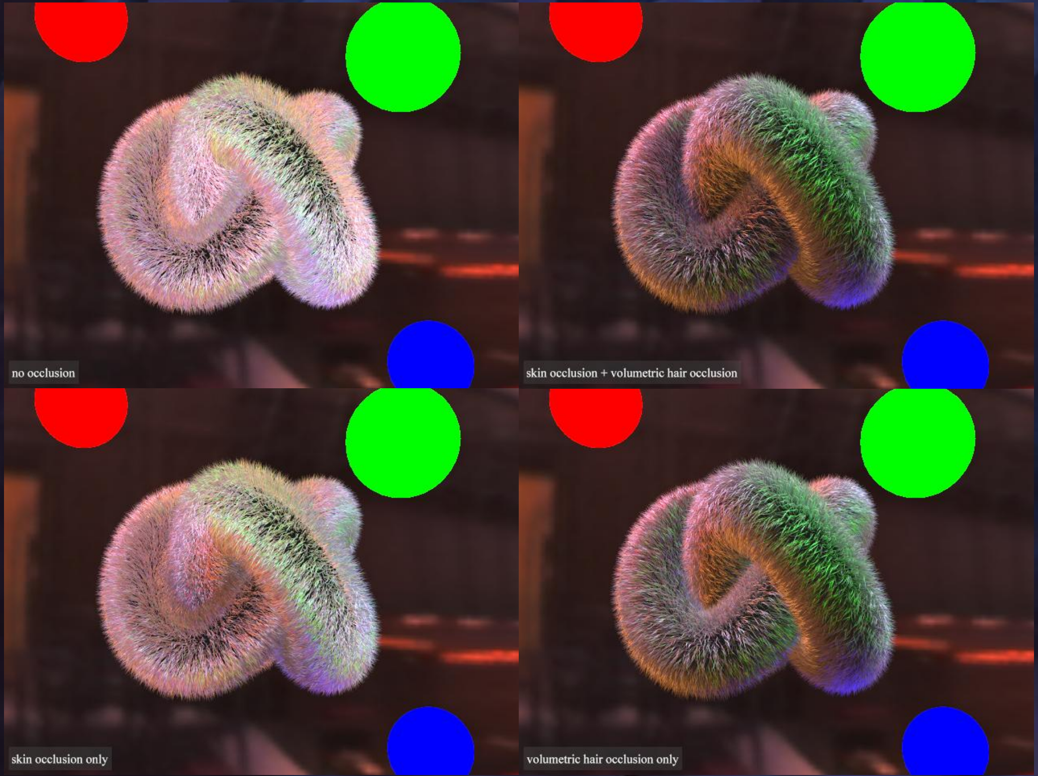
- Based on on SIGGRAPH 2004 sketch by Neulander
  - Each CP placed in virtual sphere based on $\mathbf{N}_O$, h_O
  - Occlusion depends on distance through sphere



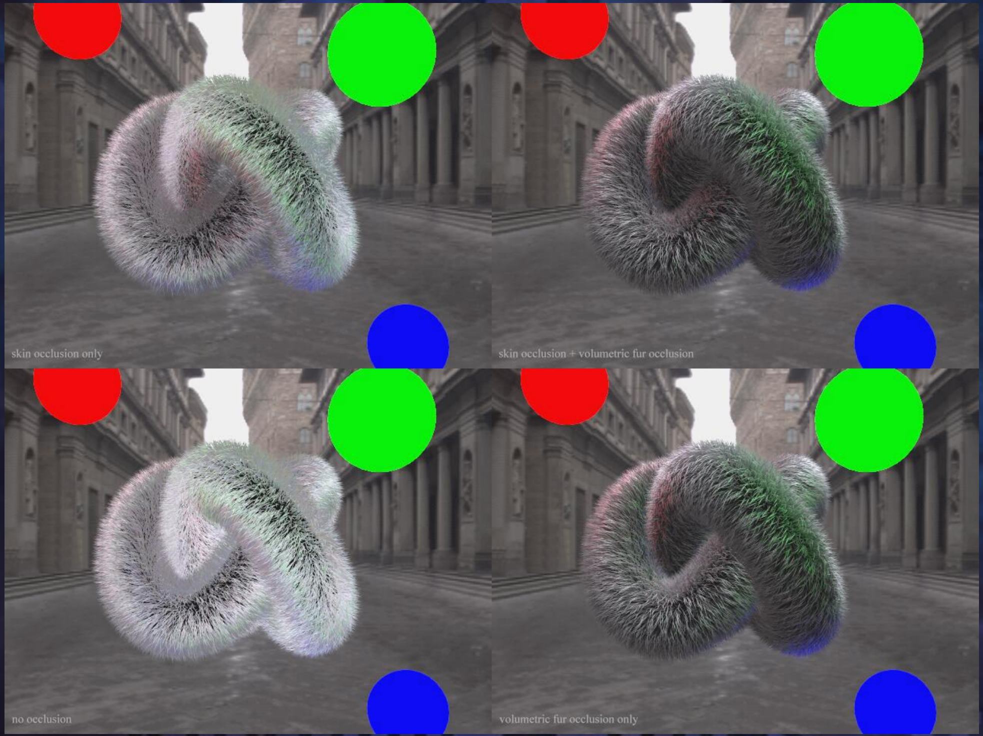
$$e^{-\rho\left(-h_O \mathbf{N_O}\cdot\mathbf{L} + \sqrt{1-h_O^2(1-\mathbf{N_O}\cdot\mathbf{L}^2)}\right)}$$

2) Volumetric Occlusion Model: Example

2) Volumetric Occlusion Model: Example



2) Volumetric Occlusion Model: Animation

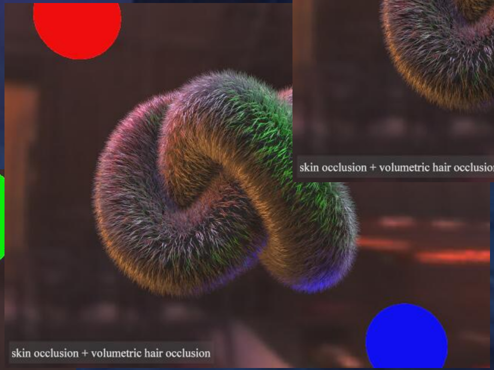


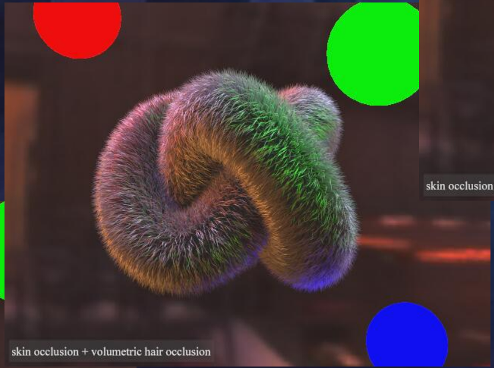
2) Volumetric Occlusion Model: Art Direction

- We can alter the apparent fur density:
  - Change exponent ρ
  - Geometry stays the same

2) Volumetric Occlusion Model: Art Direction


skin occlusion + volumetric hair occlusion


skin occlusion + volumetric hair occlusion


skin occlusion + volumetric hair occlusion


skin occlusion + volumetric hair occlusion

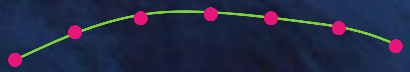

skin occlusion + volumetric hair occlusion

3) Reflection Cache

- Evaluate subset of shading requests
  - Cache a sparse set of points, interpolate
- Ward-style irradiance caching doesn't help
  - Irradiance varies abruptly between strands
    - but it varies gradually along any one strand

3) Reflection Cache:
Our Approach

- Cache reflected radiance at primary rays along each strand



  - Diffuse and specular radiance cached separately
    - at separate user-defined densities
  - Sample count varies with pixel length

  - Caching is suppressed for very short strands

3) Reflection Cache: Temporal Coherence

- Problem: Shading may “pop” from frame to frame as sample count changes
- Solution: Compute shading on a fixed 2^n-1 lattice for
  - $n_1 = \lfloor \log_2 samples \rfloor$
  - $n_2 = \lceil \log_2 samples \rceil$

  - Interpolate between these (similar to mip-map)

3) Reflection Cache: Performance Gain

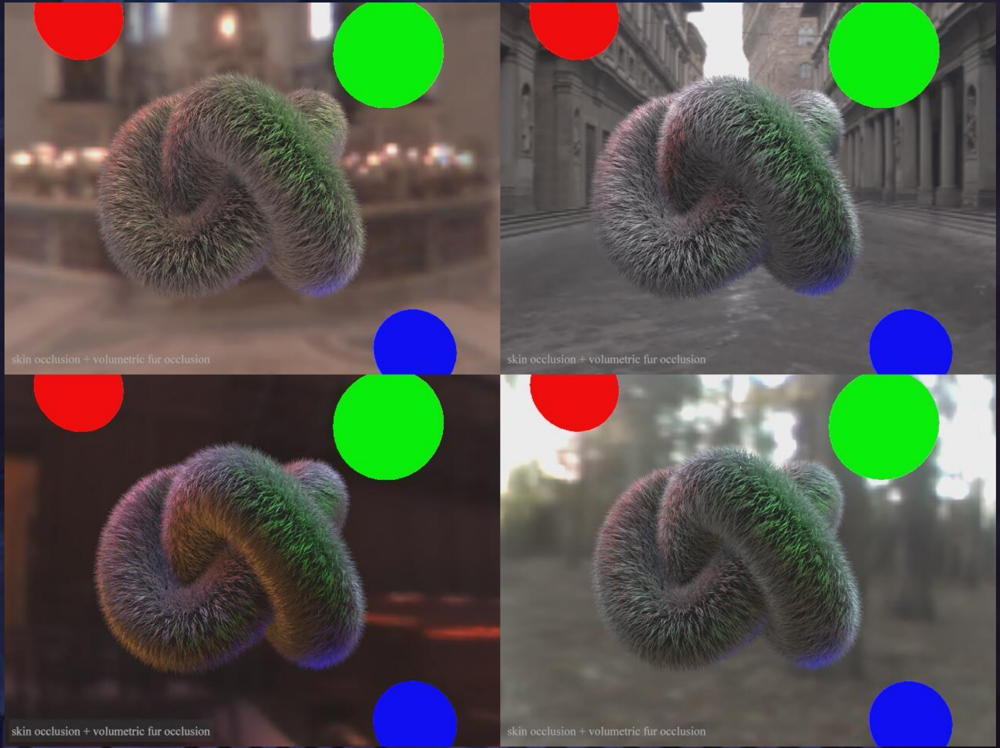
25 million rays; 260 s

6.2 million rays; 76 s

3) Reflection Cache: Results in animation:

Stats Per Frame

- resolution 640x480
- strands 53k
- 3.8 million rays
- CPU time on Athlon64 5600:
  ~ 53s



Production Results:

Knight and Day

Rhythm + Hues Studios

Future Work

- Approximate multiple scattering using skin-based multi-pole model
- Adaptively modulate cache density based on
  - Hair curvature
  - Perceptual metrics of color variation

Thank You

- Thanks to Rhythm & Hues
  - Josh Bryer, Ray Chih, Ryan Gillis, Toshi Kato, Dan Lazarow, Mike Sandrik
- Slides and abstract available on: www.rhythm.com/~ivan/pubs.html

Rhythm + Hues Studios