Approximate Reflectance Profiles for Efficient Subsurface Scattering

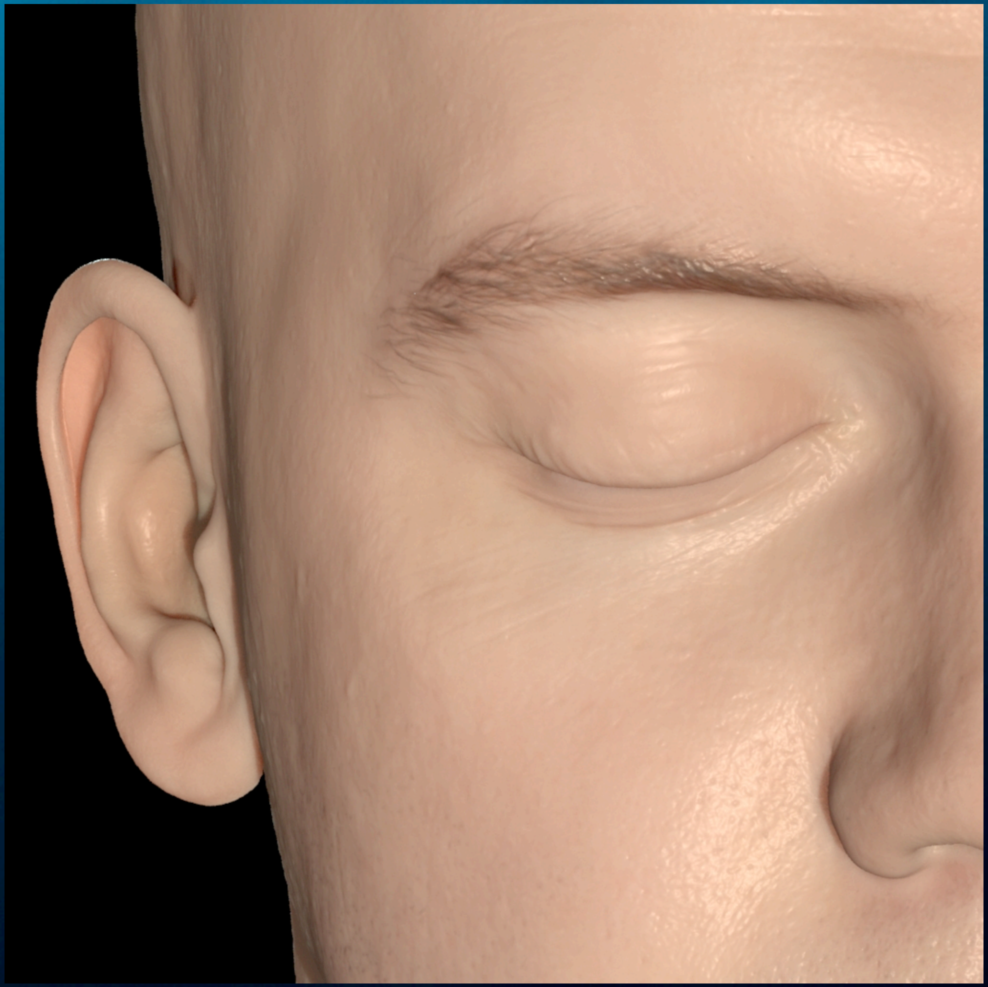
Per Christensen

Pixar Animation Studios

SIGGRAPH 2015, Los Angeles

PIXAR

Goal:
subsurface scattering, fast+simple



PIXAR

Overview

- Simple subsurface scattering model
- New parameterization allows comparison with physically-based models
- Matches Monte Carlo references very well -- better than physically-based models
- Useful for ray-traced (and point-based) subsurface scattering

PIXAR

Advantages

- Faster evaluation, simpler code
- Built-in single-scattering term
- No need for numerical inversion of user-friendly parameters (surface albedo and scattering length) to physical parameters (volume scattering and absorption coeffs)
- Bonus: simple cdf for importance sampling

PIXAR

Inspiration: Schlick's Fresnel approx.

- Physics: Fresnel reflection formula -- reflection is average of parallel and perpendicular polarized: R(theta) = $(R_p + R_s) / 2$

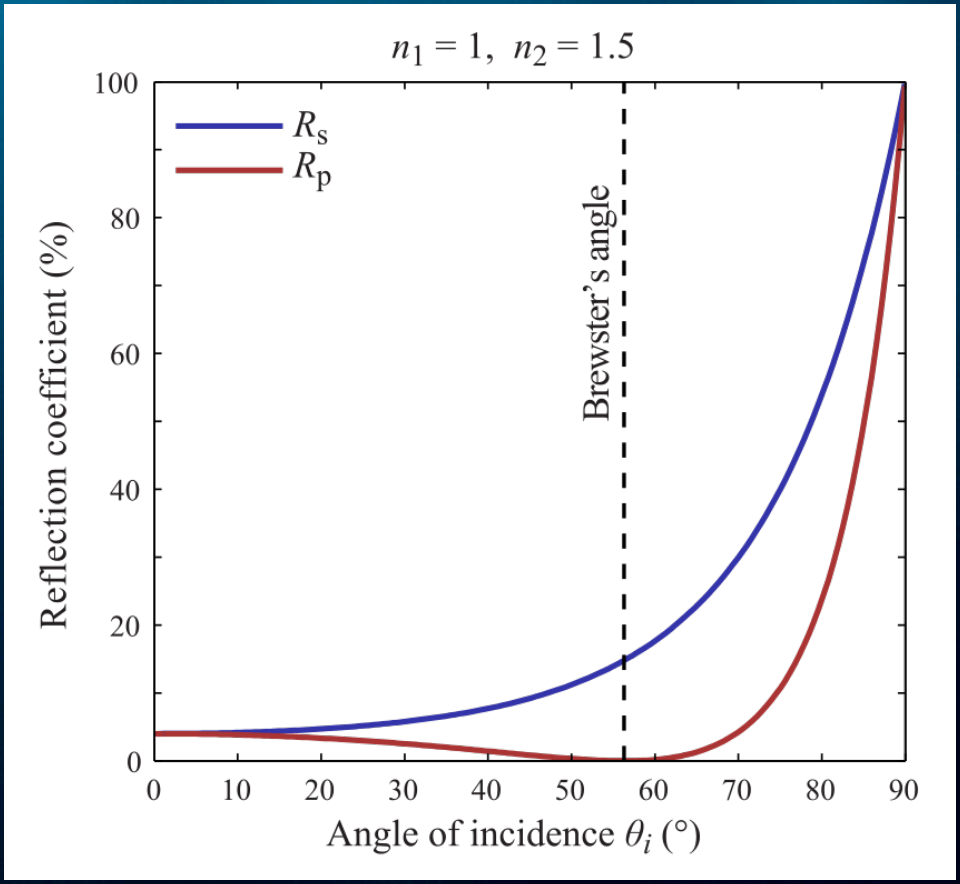
$$R_{\mathrm{p}} = \left|\frac{n_1\cos\theta_{\mathrm{t}} - n_2\cos\theta_{\mathrm{i}}}{n_1\cos\theta_{\mathrm{t}} + n_2\cos\theta_{\mathrm{i}}}\right|^2 = \left|\frac{n_1\sqrt{1-\left(\frac{n_1}{n_2}\sin\theta_{\mathrm{i}}\right)^2} - n_2\cos\theta_{\mathrm{i}}}{n_1\sqrt{1-\left(\frac{n_1}{n_2}\sin\theta_{\mathrm{i}}\right)^2} + n_2\cos\theta_{\mathrm{i}}}\right|^2.$$

$$R_{\mathrm{s}} = \left|\frac{n_1\cos\theta_{\mathrm{i}} - n_2\cos\theta_{\mathrm{t}}}{n_1\cos\theta_{\mathrm{i}} + n_2\cos\theta_{\mathrm{t}}}\right|^2 = \left|\frac{n_1\cos\theta_{\mathrm{i}} - n_2\sqrt{1-\left(\frac{n_1}{n_2}\sin\theta_{\mathrm{i}}\right)^2}}{n_1\cos\theta_{\mathrm{i}} + n_2\sqrt{1-\left(\frac{n_1}{n_2}\sin\theta_{\mathrm{i}}\right)^2}}\right|^2,$$

PIXAR

Inspiration: Schlick's Fresnel approx.

- Physics: Fresnel reflection formula



PIXAR

Inspiration: Schlick's Fresnel approx.

- [Schlick94]: Simple approximation as polynomial

$$R(\theta) = R_0 + (1 - R_0)(1 - \cos\theta)^5$$

- No visual difference
- We want similar simple approximation for subsurface scattering!

PIXAR

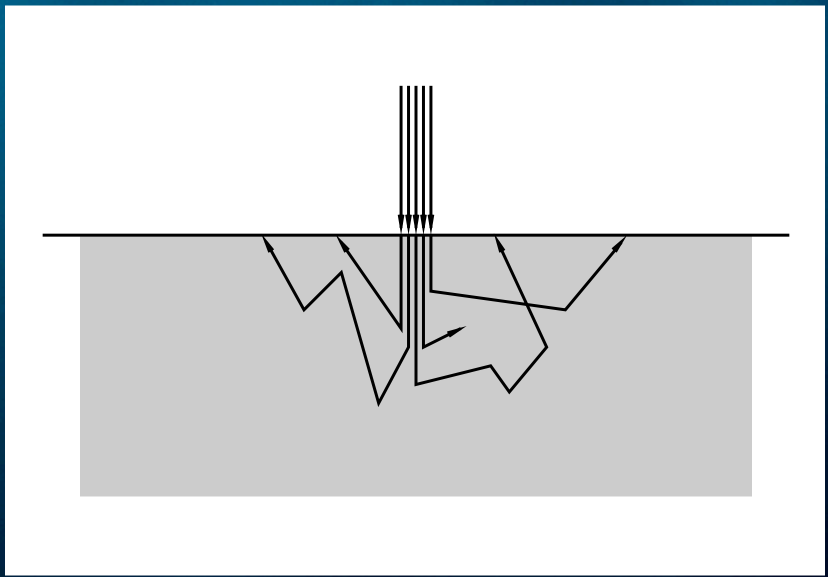
Outline of talk

- Subsurface scattering
- Physically-based subsurface scattering models
- Burley’s approximate model
- My reparameterization
- Results

PIXAR

Monte Carlo simulation

- Most general method: brute-force Monte Carlo



- But: very slow!

PIXAR

BSSRDF

- Function that describes how light enters an object, bounces around, then leaves: BSSRDF (bidirectional surface scattering reflectance distribution function) S
- Often simplified as:

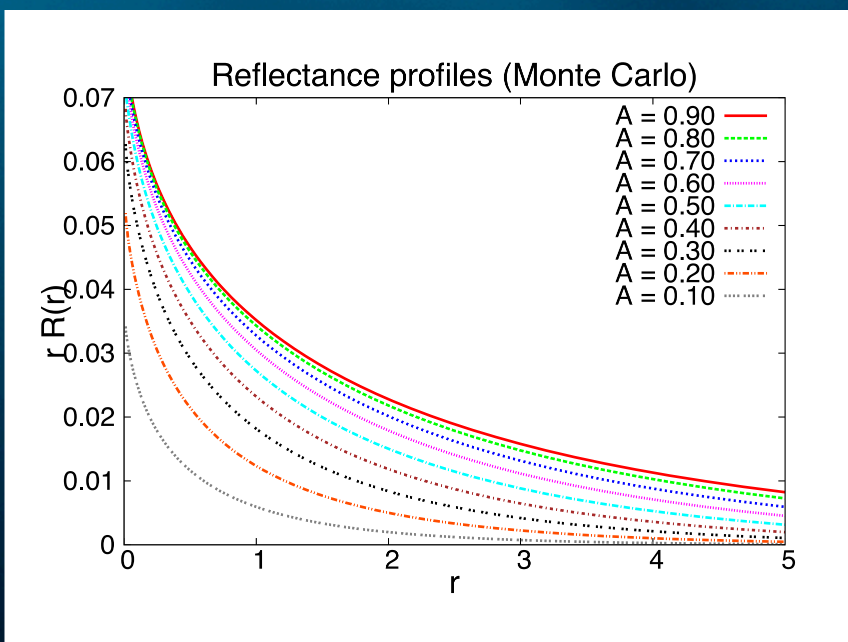
$$S(x_i, w_i; x_o, w_o) = C\, F_t(x_i, w_i)\, R(|x_o - x_i|)\, F_t(x_o, w_o)$$

reflectance profile

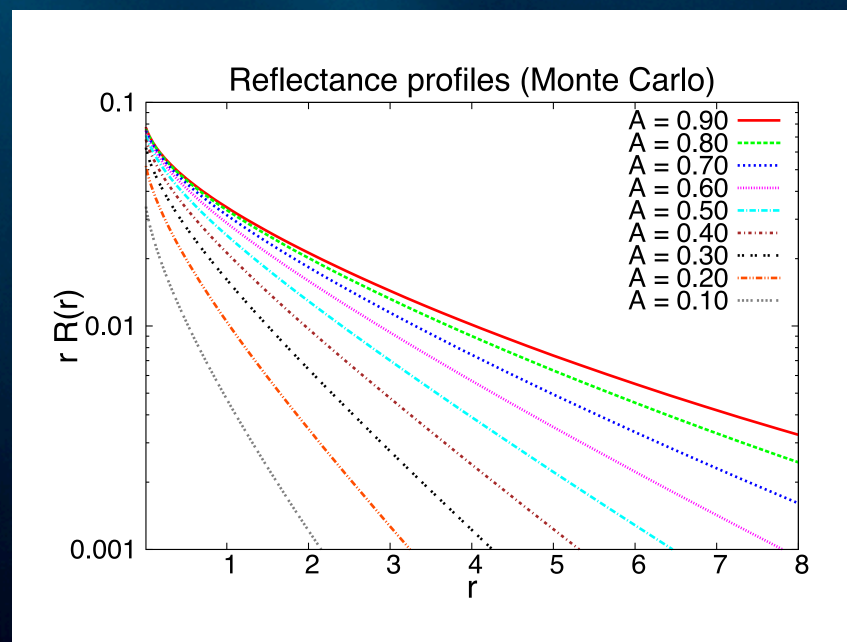
Fresnel transmission terms

Reflectance profiles: reference

- Brute-force Monte Carlo simulation
- Reflectance profile R(r); A = surface albedo



linear y axis



log y axis

PIXAR

Physically-based reflectance profiles

- Dipole diffusion [Jensen01,02]
  - simple, fast, widely used; but: blurry “waxy” look
- Better dipole diffusion [d’Eon12]
- Directional dipole diffusion [Frisvad14]
  - can handle oblique incident angles

PIXAR

Physically-based reflectance profiles

- Formulas:

Physically-based reflectance profiles

- Formulas:

$$
\begin{aligned}
R_d(r) &= -D \frac{(\vec{n} \cdot \vec{\nabla} \phi(x_s))}{d\Phi_i} \\
&= \frac{\alpha'}{4\pi} \left[(\sigma_{tr} d_r + 1) \frac{e^{-\sigma_{tr} d_r}}{\sigma'_t d_r^3} + z_v (\sigma_{tr} d_v + 1) \frac{e^{-\sigma_{tr} d_v}}{\sigma'_t d_v^3} \right].
\end{aligned}
$$

Physically-based reflectance profiles

- Formulas:

$$R_d(r) = -D \frac{(\vec{n} \cdot \vec{\nabla} \phi(x_s))}{d\Phi_i}$$

$$= \frac{\alpha'}{4\pi} \left[(\sigma_{tr} d_r + 1) \frac{e^{-\sigma_{tr} d_r}}{\cdots} + z_v (\sigma_{tr} d_v + 1) \frac{e^{-\sigma_{tr} d_v}}{\cdots} \right]$$

$$R_d = 2\pi \int_0^\infty R_d(r)\, r\, dr = \frac{\alpha'}{2} \left(1 + e^{-\frac{4}{3} A \sqrt{3(1-\alpha')}} \right) e^{-\sqrt{3(1-\alpha')}} .$$

PIXAR

Physically-based reflectance profiles

- Formulas:

$$R_d(r) = -D\frac{(\vec{n}\cdot\vec{\nabla}\phi(x_s))}{d\Phi_i}$$

$$= \frac{\alpha'}{4\pi}\left[(\sigma_{tr}d_r + 1)\, e^{-\sigma_{tr}d_r} + z_v\,(\sigma_{tr}d_v + 1)\, e^{-\sigma_{tr}d_v}\right].$$

$$R_d = 2\pi\int^{\infty} R_d(r)\, r\, dr = \frac{\alpha'}{2}\left(1 + e^{-\frac{4}{3}A\sqrt{3(1-\alpha')}}\right)e^{-\sqrt{3(1-\alpha')}}.$$

$$L_o^{(1)}(x_o,\vec{\omega}_o) = \sigma_s(x_o)\int_{2\pi} F\, p(\vec{\omega}_i'\cdot\vec{\omega}_o')\int_0^{\infty} e^{-\sigma_{tc}s} L_i(x_i,\vec{\omega}_i)\, ds\, d\vec{\omega}_i \quad (6)$$

$$= \int_A\int_{2\pi} S^{(1)}(x_i,\vec{\omega}_i;x_o,\vec{\omega}_o)\, L_i(x_i,\vec{\omega}_i)\,(\vec{n}\cdot\vec{\omega}_i)\, d\omega_i\, dA(x_i).$$

PIXAR

Physically-based reflectance profiles

- Formulas:

$$R_d(r) = -D\frac{(\vec{n}\cdot\vec{\nabla}\phi(x_s))}{d\Phi_i}$$
$$= \frac{\alpha'}{4\pi}\left[(\sigma_{tr}d_r + 1)\, e^{-\sigma_{tr}d_r} + z_v\,(\sigma_{tr}d_v + 1)\, e^{-\sigma_{tr}d_v}\right]$$

$$R_d = 2\pi\int^{\infty} R_d(r)\, r\, dr = \frac{\alpha'}{2}\left(1 + e^{-\frac{4}{3}A\sqrt{3(1-\alpha')}}\right)e^{-\sqrt{3(1-\alpha')}}\,.$$

$$L_o^{(1)}(x_o,\vec{\omega}_o) = \sigma_s(x_o)\int_{2\pi} F\, p(\vec{\omega}_i'\cdot\vec{\omega}_o')\int_0^{\infty} e^{-\sigma_{tc}s} L_i(x_i,\vec{\omega}_i)\, ds\, d\vec{\omega}_i \quad (6)$$
$$= \int_A\int_{2\pi} S^{(1)}(x_i,\vec{\omega}_i;x_o,\vec{\omega}_o)\, L_i(x_i,\vec{\omega}_i)\,(\vec{n}\cdot\vec{\omega}_i)\, d\omega_i\, dA(x_i).$$

$$\frac{(\alpha')^2}{4\pi}\left[\left(C_{\vec{E}}\frac{z_r(\mu_{tr}d_r+1)}{d_r^2} + \frac{C_\phi}{D}\right)\frac{e^{-\mu_{tr}d_r}}{d_r} - \left(C_{\vec{E}}\frac{z_v(\mu_{tr}d_v+1)}{d_v^2} + \frac{C_\phi}{D}\right)\frac{e^{-\mu_{tr}d_v}}{d_v}\right]$$

PIXAR

Physically-based reflectance profiles

- Formulas:

$$R_d(r) = -D\frac{(\vec{n}\cdot\vec{\nabla}\phi(x_s))}{d\Phi_i} = \frac{\alpha'}{4\pi}\left[(\sigma_{tr}d_r + 1)\frac{e^{-\sigma_{tr}d_r}}{\dots} + z_v(\sigma_{tr}d_v + 1)\frac{e^{-\sigma_{tr}d_v}}{\dots}\right].$$

$$R_d = 2\pi\int_0^\infty R_d(r)\,r\,dr = \frac{\alpha'}{2}\left(1 + e^{-\frac{4}{3}A\sqrt{3(1-\alpha')}}\right)e^{-\sqrt{3(1-\alpha')}}.$$

$$L_o^{(1)}(x_o,\vec{\omega}_o) = \sigma_s(x_o)\int_{2\pi}F\,p(\vec{\omega}_i'\cdot\vec{\omega}_o')\int_0^\infty e^{-\sigma_{tc}s}L_i(x_i,\vec{\omega}_i)\,ds\,d\vec{\omega}_i \quad (6)$$

$$= \int_A\int_{2\pi}S^{(1)}(x_i,\vec{\omega}_i;x_o,\vec{\omega}_o)\,L_i(x_i,\vec{\omega}_i)\,(\vec{n}\cdot\vec{\omega}_i)\,d\omega_i\,dA(x_i).$$

$$\dots\frac{d_v+1)}{d_v^2} + \frac{C_\phi}{D}\right)\frac{e^{-\mu_{tr}d_v}}{d_v}\Bigg]$$

$$\nabla\phi_d' = \frac{\Phi}{4\pi D}\frac{e^{-\sigma_{tr}r}}{r^3}\Bigg(\vec{\omega}_{12}\,3D(1+\sigma_{tr}r) - \boldsymbol{x}\,(1+\sigma_{tr}r) - \boldsymbol{x}\,3D\frac{3(1+\sigma_{tr}r)+(\sigma_{tr}r)^2}{r}\cos\theta\Bigg). \quad (19)$$

PIXAR

Physically-based reflectance profiles

- Quantized diffusion [d'Eon11]
  - Improved diffusion theory
  - Extended source term (instead of just two points)
  - Sharper edges -- not "waxy" looking

Physically-based reflectance profiles

- More formulas:

PIXAR

Physically-based reflectance profiles

- More formulas:

$$\phi(r) = \frac{e^{-\mu_t r}}{4\pi r^2} + \frac{\mu_s}{2\pi^2 r} \int_0^\infty \frac{\arctan^2 u}{u - \alpha \arctan u} \sin(r\,\mu_t\,u)\,du.$$

PIXAR

Physically-based reflectance profiles

- More formulas:

$$\phi(r) = \frac{e^{-\mu_t r}}{4\pi r^2} + \frac{\mu_s}{2\pi^2 r} \int_0^\infty \frac{\arctan^2 u}{u - \alpha \, \text{arcta}\ldots} \sin(r \, \mu_t \, u) \, du.$$

$$\phi(r) = \frac{e^{-\mu_t' r}}{4\pi r^2} + \frac{1}{4\pi} \frac{3\mu_s' \mu_t'}{2\mu_a + \mu_s'} \frac{e^{-\sqrt{\frac{\mu_a}{D}} r}}{r}$$

Physically-based reflectance profiles

- More formulas:

$$\phi(r) = \frac{e^{-\mu_t r}}{4\pi r^2} + \frac{\mu_s}{2\pi^2 r} \int_0^\infty \frac{\arctan^2 u}{u - \alpha\, \mathrm{arcta}\ldots} \sin(r\,\mu_t\,u)\,du.$$

$$\phi(r) = \frac{e^{-\mu_t' r}}{\ldots} + \frac{1}{\ldots} \frac{3\mu_s'\mu_t'}{2\mu_a + \mu_s'} \frac{e^{-\sqrt{\frac{\mu_a}{D}}\,r}}{r}$$

$$\int_0^\infty G_{3D}(v, \sqrt{r^2 + z^2}) Q(z) dz = \frac{1}{2}\, \mu_s'\, f(\mu_t'^2 v/2)\, G_{2D}(v, r),$$

Physically-based reflectance profiles

- More formulas:

$$\phi(r) = \frac{e^{-\mu_t r}}{4\pi r^2} + \frac{\mu_s}{2\pi^2 r} \int_0^\infty \frac{\arctan^2 u}{u - \alpha \operatorname{arcta}\cdots} \sin(r\,\mu_t\,u)\,du.$$

$$\phi(r) = \frac{e^{-\mu_t' r}}{\cdots} + \frac{1}{\cdots} \frac{3\mu_s'\mu_t'}{2\mu_a + \mu_s'} \frac{e^{-\sqrt{\frac{\mu_a}{D}}r}}{r}$$

$$\int_0^\infty G_{3D}(v, \sqrt{r^2 + z^2})Q(z)dz = \frac{1}{2}\mu_s' f(\mu_t'^2 v/2)\, G_{2D}(v, r),$$

$$\frac{1}{4\pi D} \frac{e^{-r\sqrt{\frac{\mu_a}{D}}}}{r} = \int_0^\infty \frac{c}{(4\pi Dct)^{3/2}} e^{-\mu_a ct} e^{-r^2/(4Dct)}\,dt$$

Physically-based reflectance profiles

- More formulas:

$$\phi(r) = \frac{e^{-\mu_t r}}{4\pi r^2} + \frac{\mu_s}{2\pi^2 r} \int_0^\infty \frac{\arctan^2 u}{u - \alpha \arcta\ldots} \sin(r\,\mu_t\,u)\,du.$$

$$\phi(r) = \frac{e^{-\mu_t' r}}{\ldots} + \frac{1}{\ldots} \frac{3\mu_s'\mu_t'}{2\mu_a + \mu_s'} \frac{e^{-\sqrt{\frac{\mu_a}{D}}r}}{r}$$

$$\int_0^\infty G_{3D}(v, \sqrt{r^2 + z^2}) Q(z) dz = \frac{1}{2} \mu_s' f(\mu_t'^2 v/2)\, G_{2D}(v, r),$$

$$1 \quad e^{-r\sqrt{\frac{\mu_a}{D}}} \quad \int^\infty \quad c \quad \ldots e^{-r^2/(4Dct)}\,dt$$

$$\int_{z_1}^{z_2} Q(z) G_{3D}(v, \sqrt{r^2 + (z+m)^2}) dz = G_{2D}(v, r) w_\phi(v, z_1, z_2, m),$$

$$w_\phi(v, z_1, z_2, m) = \int_{z_1}^{z_2} \frac{e^{-\frac{(-z+m)^2}{2v}}}{\sqrt{2\pi v}} \alpha' \mu_t e^{-\mu_t z} dz =$$

$$\frac{\alpha' \mu_t}{2} e^{m\,\mu_t + \frac{\mu_t^2 v}{2}} \left(\operatorname{erf}\left[\frac{m + \mu_t v + z_2}{\sqrt{2v}}\right] - \operatorname{erf}\left[\frac{m + \mu_t v + z_1}{\sqrt{2v}}\right] \right).$$

Physically-based reflectance profiles

- Photon beam diffusion [Habel13]
  - As accurate as quantized diffusion, but faster
  - Accurate single-scattering model
  - Can handle oblique incident angles

PIXAR

Physically-based reflectance profiles

- Some other formulas:

PIXAR

Physically-based reflectance profiles

- Some other formulas:

$$R_d \approx \frac{1}{N}\sum_{i=1}^{N} \frac{f(\vec{x},\vec{\omega},t_i)w_{\mathrm{exp}}(t_i,x)}{w_{\mathrm{exp}}(t_i,x)\,\mathrm{pdf}_{\mathrm{exp}}(t_i)+w_{\mathrm{eq}}(t_i,x)\,\mathrm{pdf}_{\mathrm{eq}}(t_i|\vec{x},\vec{\omega})} + \frac{1}{N}\sum_{j=1}^{N} \frac{f(\vec{x},\vec{\omega},t_j)w_{\mathrm{eq}}(t_j,x)}{w_{\mathrm{exp}}(t_j,x)\,\mathrm{pdf}_{\mathrm{exp}}(t_j)+w_{\mathrm{eq}}(t_j,x)\,\mathrm{pdf}_{\mathrm{eq}}(t_j|\vec{x},\vec{\omega})},$$

PIXAR

Physically-based reflectance profiles

- Some other formulas:

$$R_d \approx \frac{1}{N}\sum_{i=1}^{N} \frac{f(\vec{x},\vec{\omega},t_i)\,w_{\text{exp}}(t_i,x)}{w_{\text{exp}}(t_i,x)\,\text{pdf}_{\text{exp}}(t_i) + w_{\text{eq}}(t_i,x)\,\text{pdf}_{\text{eq}}(t_i|\vec{x},\vec{\omega})}$$

$$+ \frac{1}{N}\sum_{j=1}^{N} \frac{f(\vec{x},\vec{\omega},t_j)\,w_{\text{eq}}(t_j,x)}{w_{\text{exp}}(t_j,x)\,\text{pdf} \ldots}$$

$$R^d_{\vec{E}}(\vec{x},t) = C_{\vec{E}}\,\frac{\alpha'}{4\pi}\left[\frac{z_r(t)\,(1+\sigma_{tr}d_r(t))\,e^{-\sigma_{tr}d_r(t)}}{d_r^3(t)} + \frac{(z_r(t)+2z_b)\,(1+\sigma_{tr}d_v(t))\,e^{-\sigma_{tr}d_v(t)}}{d_v^3(t)}\right],$$

PIXAR

Physically-based reflectance profiles

- Some other formulas:

$$R_d \approx \frac{1}{N}\sum_{i=1}^{N} \frac{f(\vec{x},\vec{\omega},t_i)w_{\text{exp}}(t_i,x)}{w_{\text{exp}}(t_i,x)\,\text{pdf}_{\text{exp}}(t_i) + w_{\text{eq}}(t_i,x)\,\text{pdf}_{\text{eq}}(t_i|\vec{x},\vec{\omega})}$$

$$+ \frac{1}{N}\sum_{j=1}^{N} \frac{f(\vec{x},\vec{\omega},t_j)w_{\text{eq}}(t_j,x)}{w_{\text{exp}}(t_j,x)\,\text{pdf}\ldots}$$

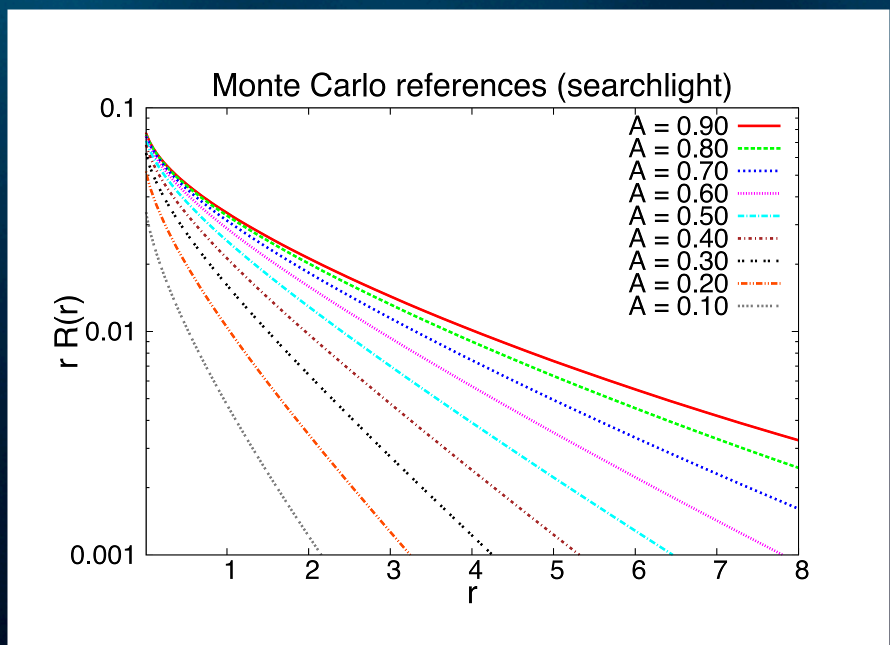
$$R^d_{\vec{E}}(\vec{x},t) = C_{\vec{E}}\,\frac{\alpha'}{4\pi}\left[\frac{z_r(t)\,(1+\sigma_{tr}d_r(t))\,e^{-\sigma_{tr}d_r(t)}}{d_r^3(t)} + \frac{(z_r(t)+2z_b)\,(1+\sigma_{tr}d_v(t))\,e^{-\sigma_{tr}d_v(t)}}{d_v^3(t)}\right],$$

$$r^{(1)}(\vec{x},\vec{x}_r(t)) = \frac{f_s(\vec{\omega}\cdot\vec{\omega}_{\vec{x}_r\vec{x}})e^{-\sigma_t(d_r(t))}F_t(\theta_o,\eta)F_t(\theta_i,1/\eta)\cos\theta_o}{d_r^2(t)}$$

PIXAR

Approximate reflectance profiles

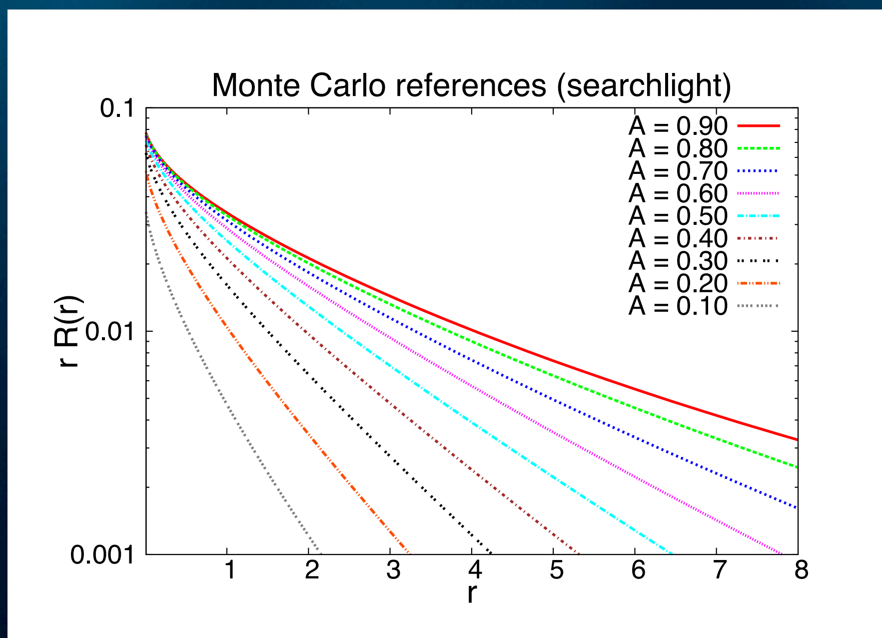
- Forget physics ... just approximate curves!
- Standard approach: sum of Gaussians



PIXAR

Approximate reflectance profiles

- Burley: curves look more like exponentials
- Sum of two exponentials (divided by distance r) is remarkably good approximation



PIXAR

Approximate reflectance profiles

- Normalized diffusion model [Burley]:

$$R(r) = \frac{e^{-r/d} + e^{-r/(3d)}}{8\pi d r}$$

- Multiply by A = surface albedo
- d controls width and height of curve ... but what is d ??
  - artistic control of subsurface “softness”
  - what is connection between d and physical params?

PIXAR

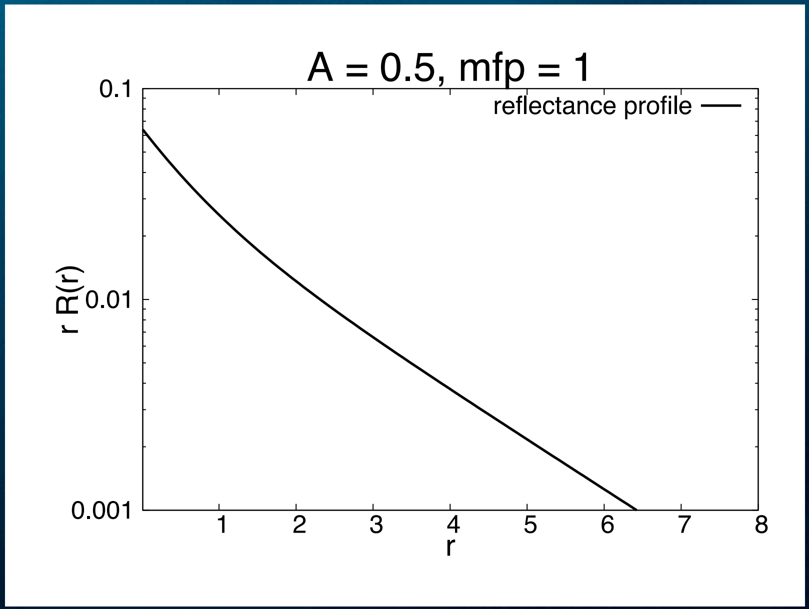
Translation from physical param to d

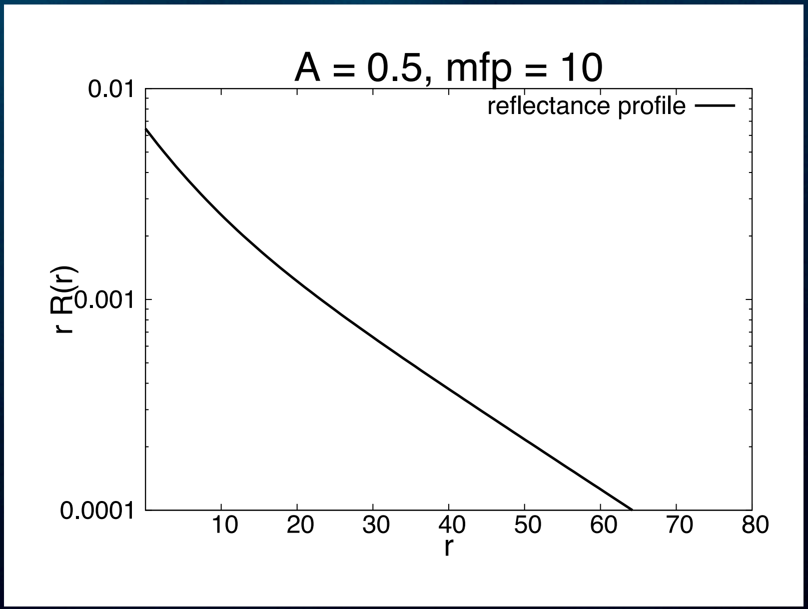
- Our usual way of expressing scattering distance is mfp or dmfp:
  - mean free path in volume
  - diffuse mean free path on surface
- Let's find a "translation" s between mfp and d:
  - d = mfp / s   (s depends on A)
- With a translation we can compare normalized diffusion with physically-based diffusion models

PIXAR

Translation from physical param to d

- To determine s it is sufficient to consider only curves for mfp=1 since the shape of reflectance profile curve for given A is independent of mfp





PIXAR

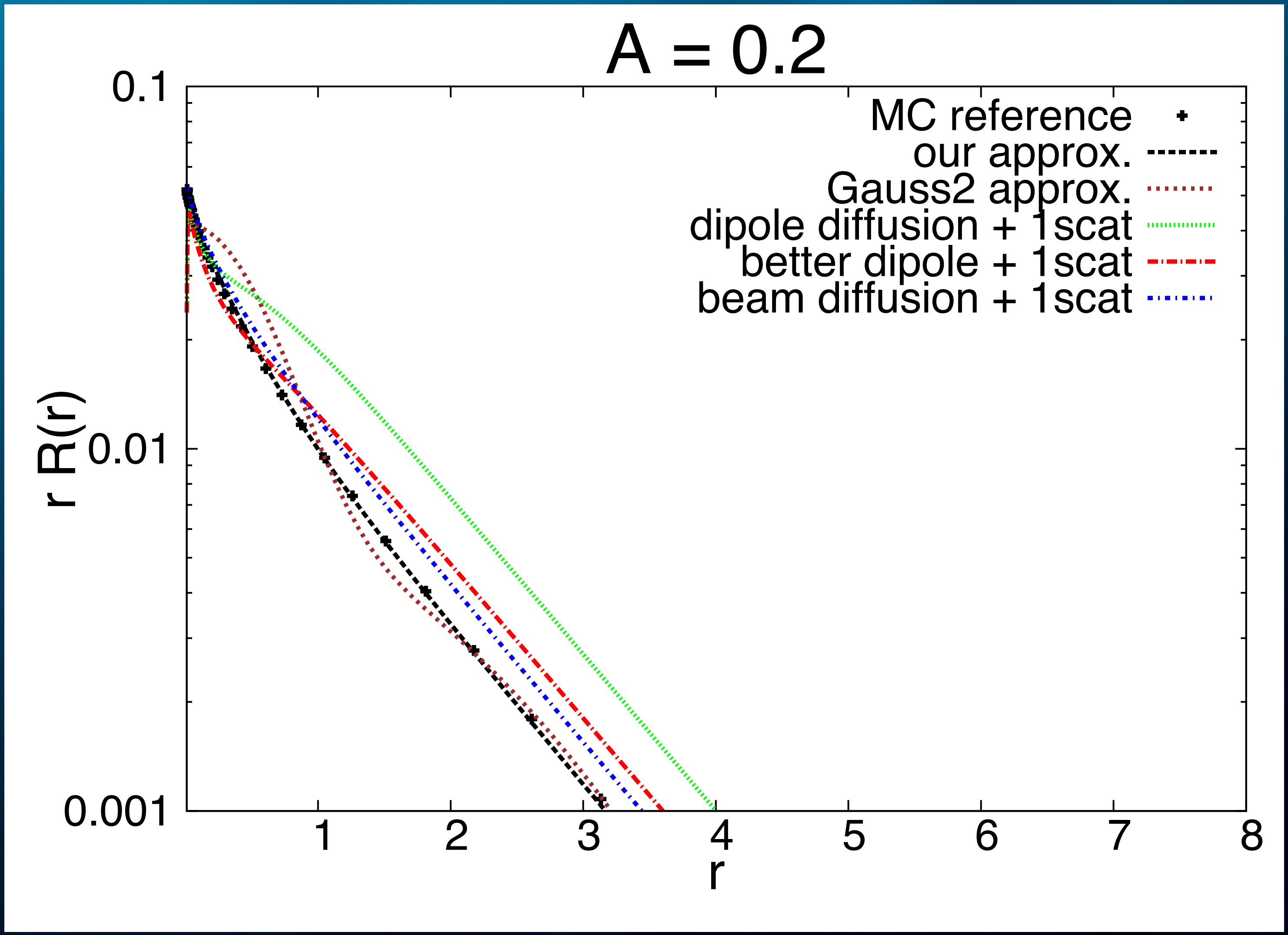
Translation from mfp to d

- For mfp=1 :

$$R_{\ell=1}(r) = A\, s\, \frac{e^{-sr} + e^{-sr/3}}{8\,\pi\, r}$$

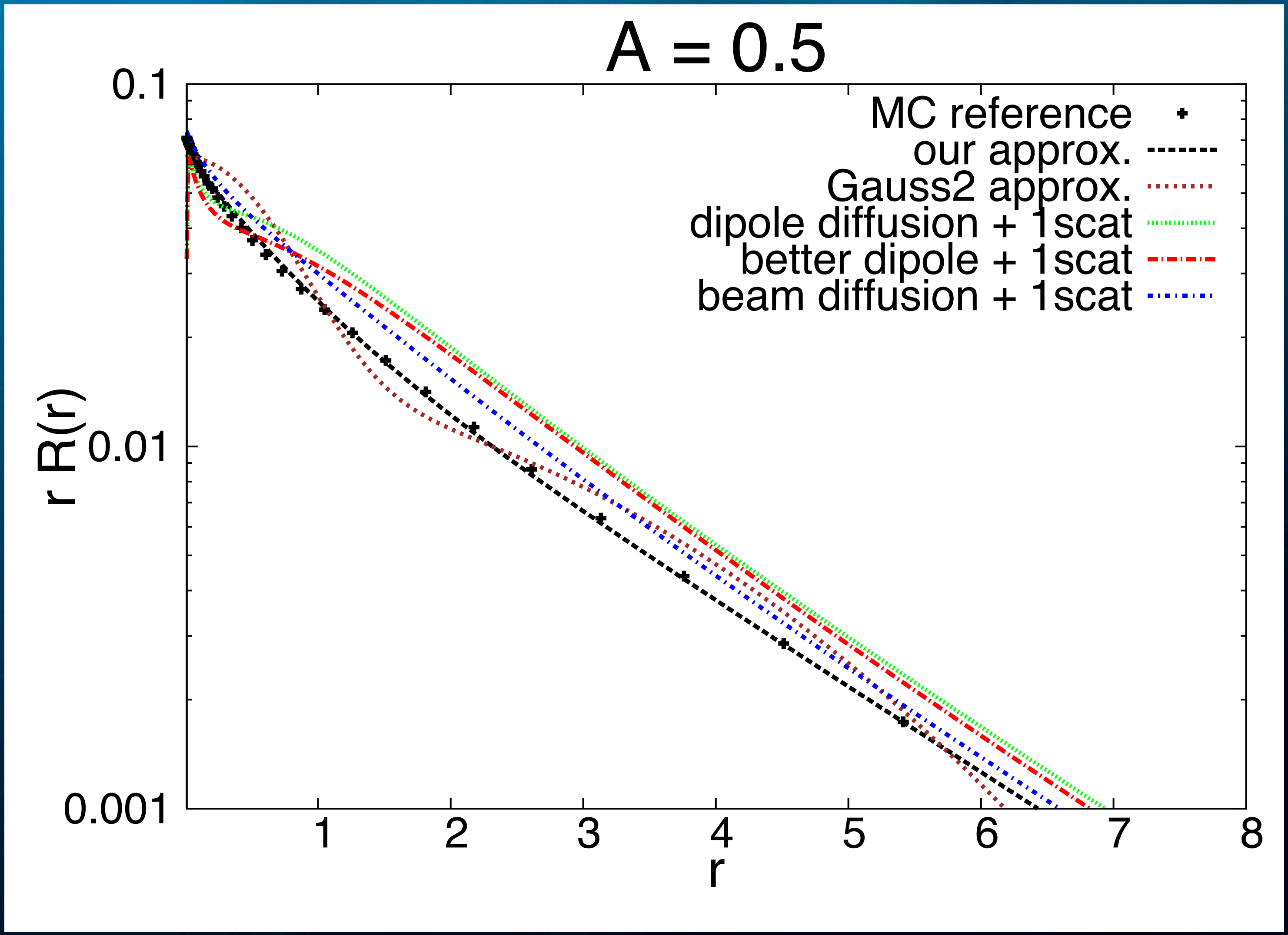
- Find s that minimizes difference between R(r) and Monte Carlo reference for same A
- For example: with optimal s for A = 0.2, 0.5, 0.8 ... :

Comparisons: surface albedo 0.2



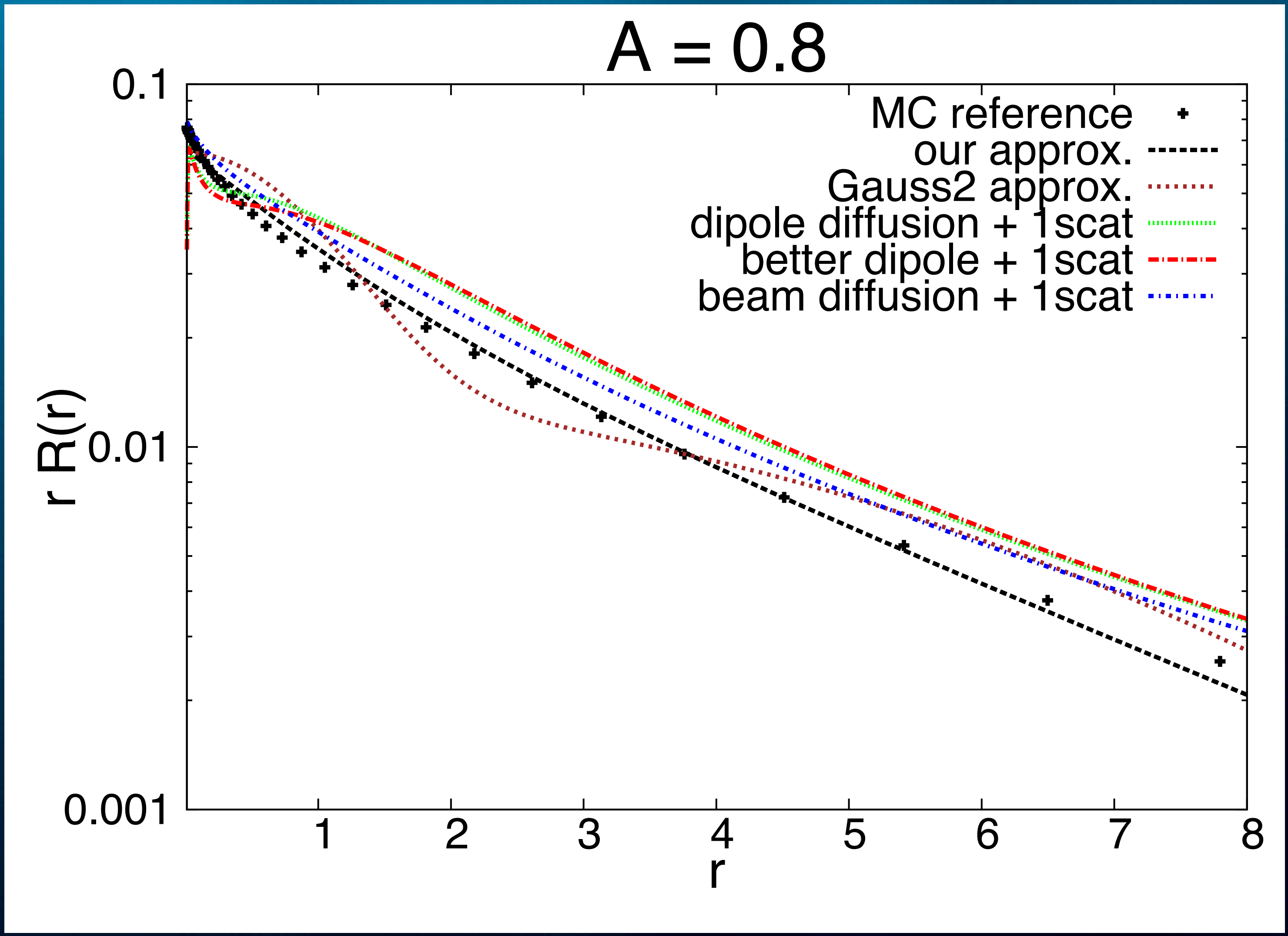
PIXAR

Comparisons: surface albedo 0.5



PIXAR

Comparisons: surface albedo 0.8



PIXAR

Comparisons: summary

- Normalized diffusion is closer to the MC reference points than dipole, better dipole, beam diffusion (w/ single scattering)
- Normalized diffusion (two exponentials) is a better approximation than two Gaussians

PIXAR

Translation from mfp to d

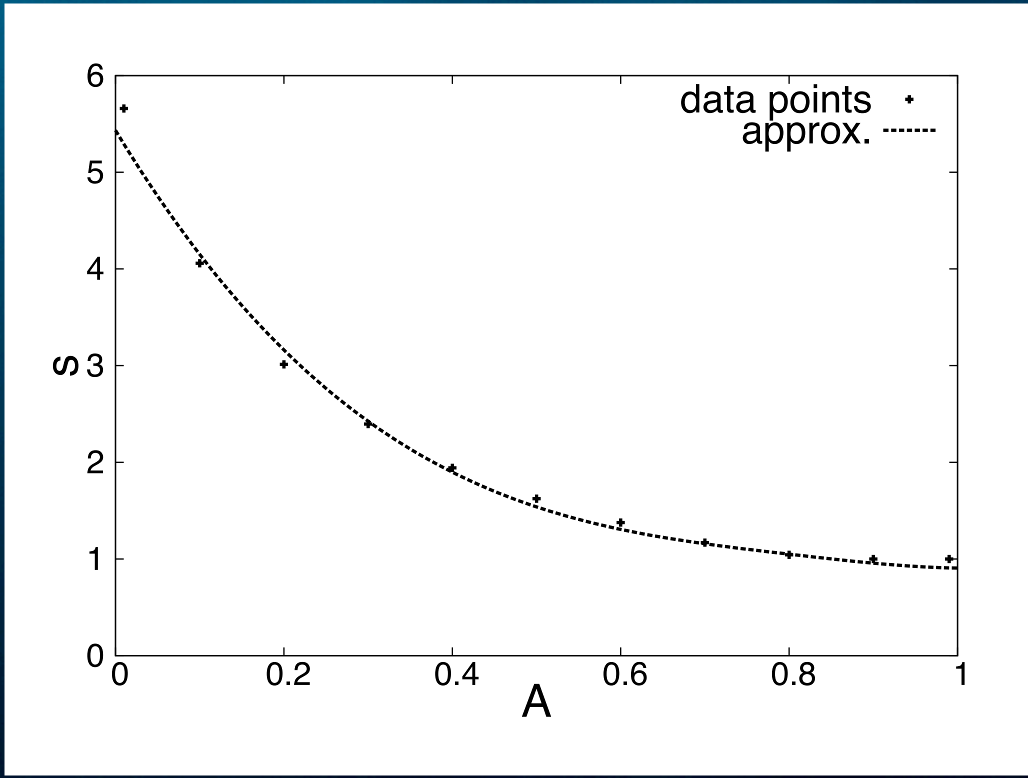
- Find s that minimizes difference between R(r) and Monte Carlo reference for all A in 0.01, 0.02, ... , 0.99
- Gives data points; fit simple polynomial

PIXAR

Translation from mfp to d

- Data points and fitted curve:

$$s = 1.85 - A + 7\,|A - 0.8|^3$$



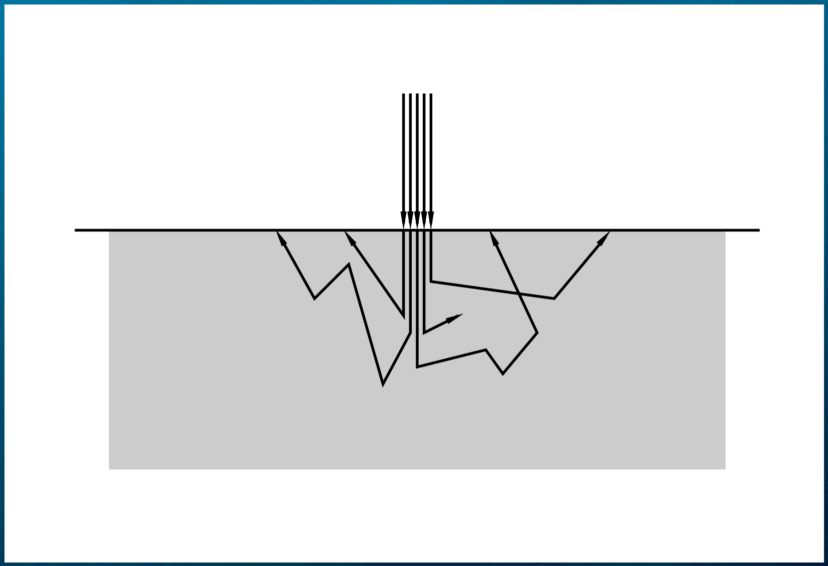
PIXAR

Translation from mfp to d

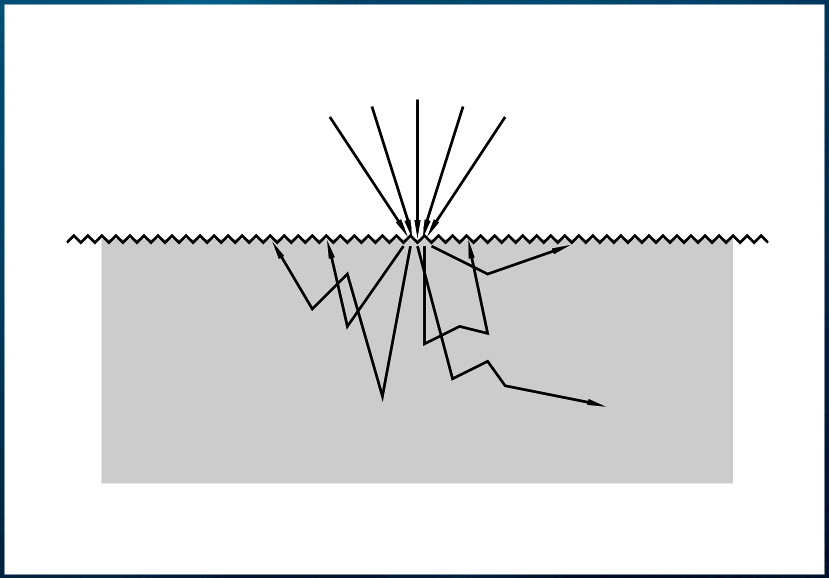
- Error wrt. MC references is ~5.5%
- Small error compared to appoximations and assumptions built into MC references: semi-infinite homogeneous volume, flat surface, ...

PIXAR

Diffuse surface transmission



vs.

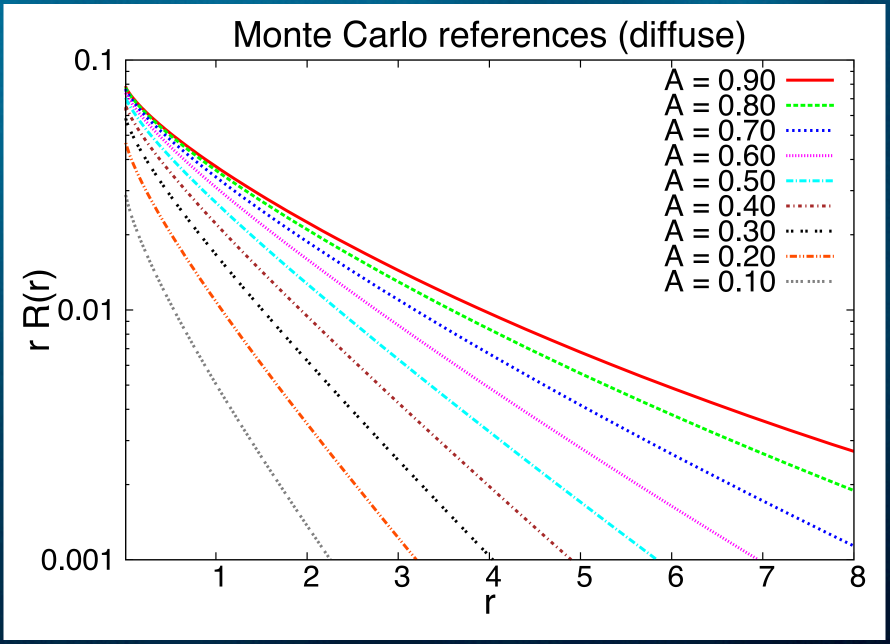


searchlight configuration
milk, juice, oily skin, ...

diffuse transmission
dry skin, make-up, ...

PIXAR

Diffuse surface transmission

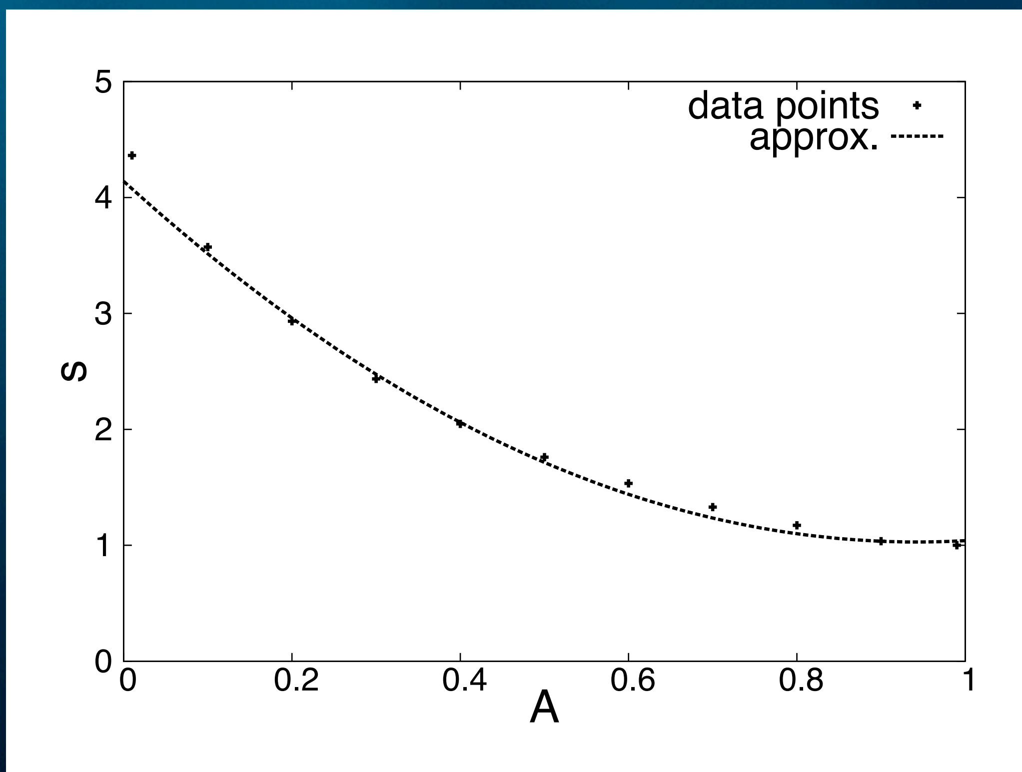


PIXAR

Translation from mfp to d (diffuse)

- Data points and fitted curve:

$$s = 1.9 - A + 3.5\,(A - 0.8)^2$$



PIXAR

Translation from mfp to d (diffuse)

- Error wrt. MC references is only ~3.9%
- In practical use: not much visual difference between searchlight approx and diffuse-transmission approx -- even though built on very different assumptions

PIXAR

Translation from dmfp to d

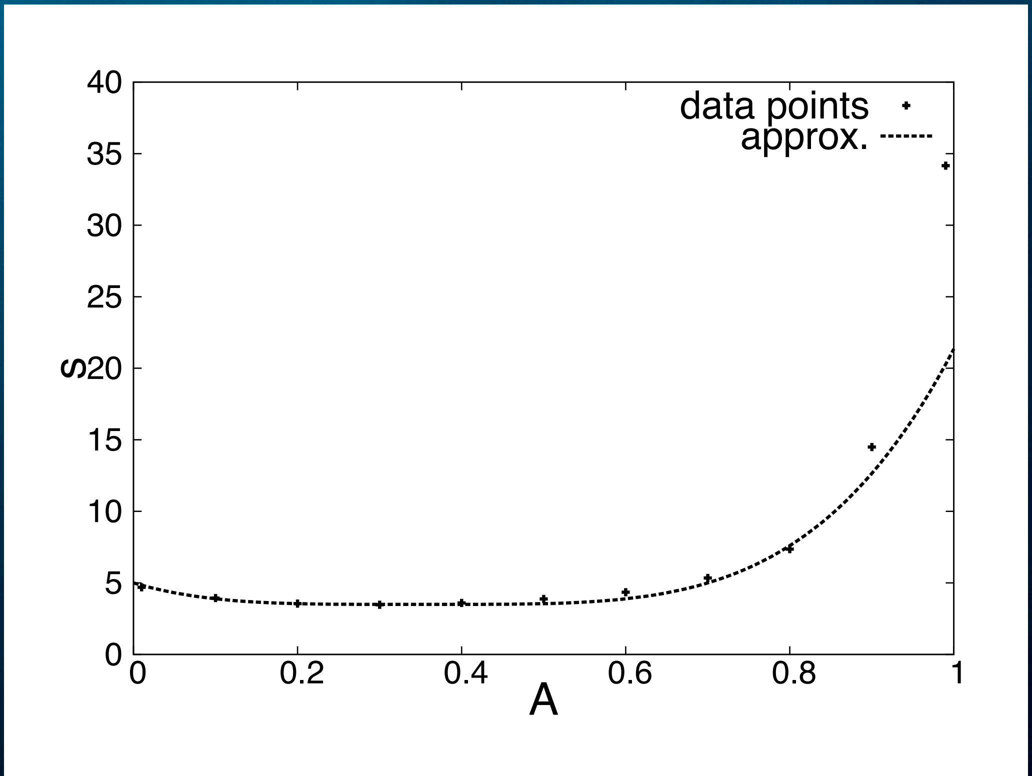
- Back to searchlight configuration
- Change parameterization of scattering distance: diffuse mean free path on surface (instead of mean free path in volume)

PIXAR

Translation from dmfp to d

- Data points and fitted curve:

$$s = 3.5 + 100\,(A - 0.33)^4$$



PIXAR

Translation from dmfp to d

- Error wrt. MC references is ~7.7%
- In practical use: dmfp might be more intuitive than mfp; hence standard parameter of our previous scattering models

PIXAR

Translation summary

- 3 ways to determine d in Burley's normalized diffusion formula:
  - mfp to d for searchlight configuration
  - mfp to d for diffuse transmission
  - dmfp to d for searchlight configuration
- 3 simple polynomials for s = s(A)
- Pick the one you like!

PIXAR

Practical detail: importance sampling

- Importance sampling of distance r between light entry and exit points: need cdf(r)
- For physically-based BSSRDFs the cdf has to be computed with numerical integration: slow
- Burley's normalized diffusion has simple cdf:

$$\mathrm{cdf}(r) = 1 - \frac{1}{4}e^{-sr/\ell} - \frac{3}{4}e^{-sr/(3\ell)}$$

PIXAR

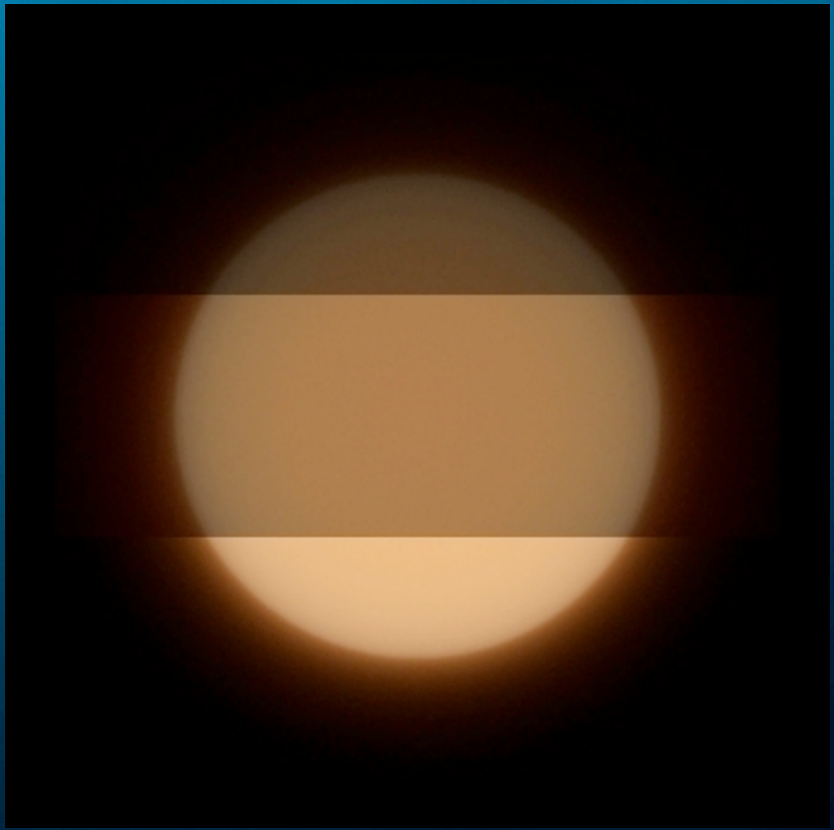
Discussion

- Much simpler than physically-based diffusion (e.g. quantized diffusion or beam diffusion)
- Many times faster*

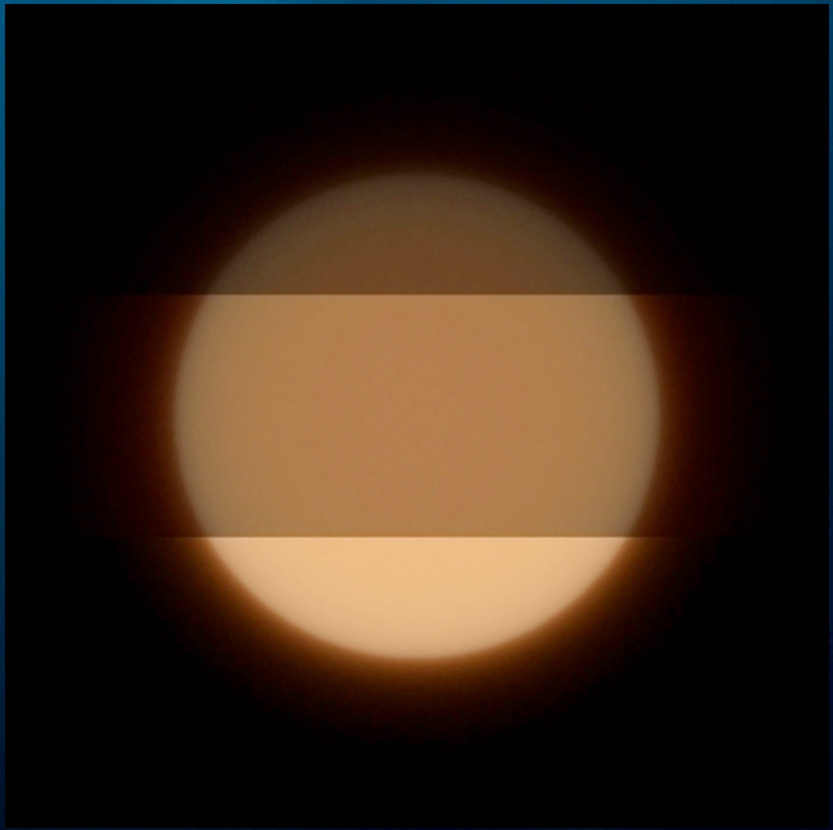
- *footnote: only a bit faster if careful table-based optimizations of physically-based

PIXAR

Result: comparison w/ beam diffusion



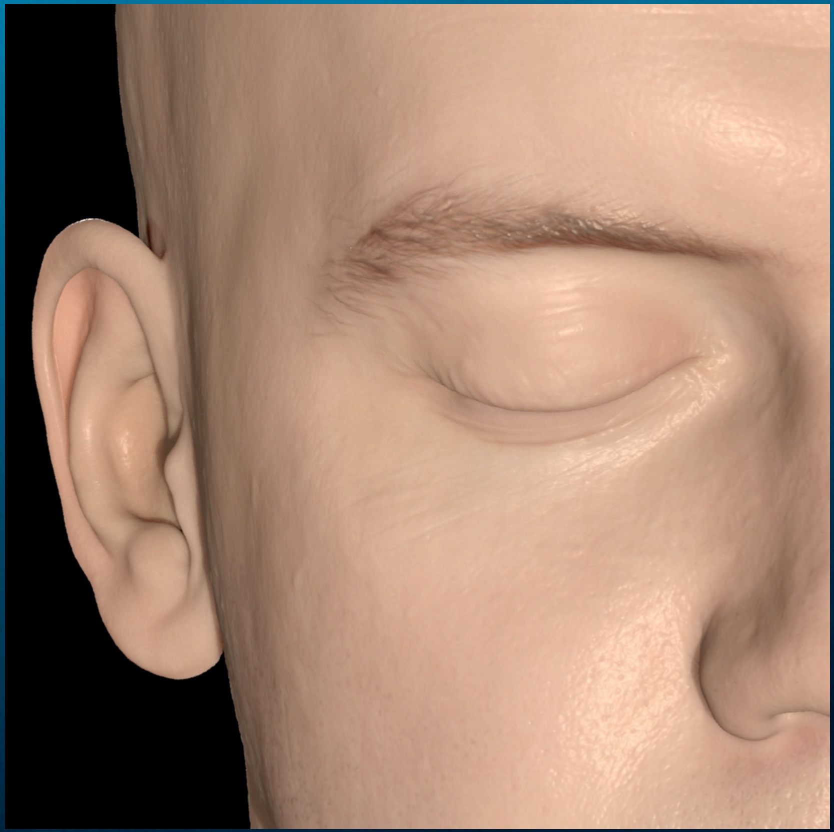
beam diffusion + 1scatter



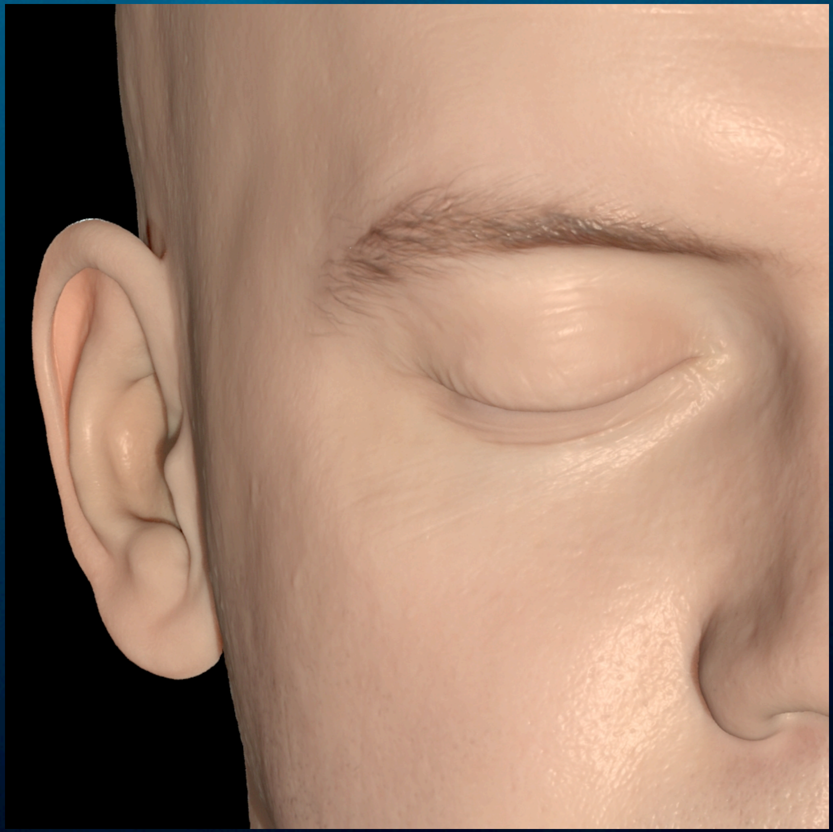
our approx

PIXAR

Result: comparison w/ beam diffusion



beam diffusion + 1scatter



our approx

(Head data: Infinite Realities)

PIXAR

Results

marble

fruits

plastic

PIXAR

Result: still life

image credit: Dylan Sisson

PIXAR

Conclusion

- Reparameterization of Burley's normalized diffusion approximation gives plug-in replacement of physically-based diffusion formulas -- same parameters
- Simpler, faster
- Error wrt. MC references is only a few percent
- More accurate than physically-based models
- One of the sss models built into RenderMan

PIXAR

Future work

- Oblique angles of incidence; non-symmetric scattering
  - maybe just s that depends on polar and relative azimuthal angle of incident illumination?
- Anisotropic scattering?

PIXAR

More information

- Burley, "Extending Disney's physically based BRDF with integrated subsurface scattering", Physically Based Shading Course
- Technical report: Christensen & Burley, graphics.pixar.com/library/ApproxBSSRDF

PIXAR

Acknowledgments

- HUGE thanks: Brent Burley
- Colleagues in RenderMan team
- Christophe Hery and Ryusuke Villemin (Pixar)
- Wojciech Jarosz and Ralf Habel (Disney)
- Images: Dylan Sisson, ...

PIXAR

Thank you!

PIXAR

PIXAR