Approximate Reflectance Profiles for Efficient Subsurface Scattering

Per Christensen
Pixar Animation Studios

SIGGRAPH 2015, Los Angeles
Goal: subsurface scattering, fast+simple
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
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
Inspiration: Schlick’s Fresnel approx.

- Physics: Fresnel reflection formula -- reflection is average of parallel and perpendicular polarized:
  \[ R(\theta) = \frac{R_p + R_s}{2} \]

\[ R_p = \left| \frac{n_1 \cos \theta_t - n_2 \cos \theta_i}{n_1 \cos \theta_t + n_2 \cos \theta_i} \right|^2 = \frac{n_1 \sqrt{1 - \left( \frac{n_1}{n_2} \sin \theta_i \right)^2} - n_2 \cos \theta_i}{n_1 \sqrt{1 - \left( \frac{n_1}{n_2} \sin \theta_i \right)^2} + n_2 \cos \theta_i} \]

\[ R_s = \left| \frac{n_1 \cos \theta_i - n_2 \cos \theta_t}{n_1 \cos \theta_i + n_2 \cos \theta_t} \right|^2 = \frac{n_1 \cos \theta_i - n_2 \sqrt{1 - \left( \frac{n_1}{n_2} \sin \theta_i \right)^2}}{n_1 \cos \theta_i + n_2 \sqrt{1 - \left( \frac{n_1}{n_2} \sin \theta_i \right)^2}} \]
Inspiration: Schlick’s Fresnel approx.

- Physics: Fresnel reflection formula
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!
Outline of talk

• Subsurface scattering
• Physically-based subsurface scattering models
• Burley’s approximate model
• My reparameterization
• Results
Monte Carlo simulation

- Most general method: brute-force Monte Carlo

- But: very slow!
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

![Reflectance profiles (Monte Carlo)](image)

- Linear y axis
- Log y axis
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
Physically-based reflectance profiles

• Formulas:
Physically-based reflectance profiles

• Formulas:

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

- Formulas:

\[
R_d(r) = -D \frac{(\vec{n} \cdot \nabla \phi(x_\theta))}{d\Phi_i}
\]

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

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

- Formulas:

\[
R_d(r) = -D \left( \frac{\mathbf{n} \cdot \nabla \phi(x_s)}{d\Phi_i} \right)
= \frac{\alpha'}{4\pi} \left[ (\sigma_{tr} dr + 1) e^{-\sigma_{tr} dr} + \ldots (\sigma_{tr} dz + 1) e^{-\sigma_{tr} dz} \right]
\]

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

\[
L_o^{(1)}(x_o, \omega_o) = \sigma_s(x_o) \int_{2\pi} F_p(\omega'_i \cdot \omega'_o) \int_0^\infty e^{-\sigma t_c s} L_i(x_i, \omega_i) ds d\omega_i \tag{6}
\]

\[
= \int_A \int_{2\pi} S^{(1)}(x_i, \omega_i; x_o, \omega_o) L_i(x_i, \omega_i) (\mathbf{n} \cdot \omega_i) d\omega_i dA(x_i).
\]
Physically-based reflectance profiles

- Formulas:

\[
R_d(r) = -DR\left(\frac{\vec{n} \cdot \nabla \phi(x_0)}{d\Phi_i}\right)
\]

\[
= \frac{\alpha'}{4\pi} \left[ (\sigma_{tr} dr + 1) e^{-\sigma_{tr} dr} + \sigma_{sc} (\sigma_{sc} dr + 1) e^{-\sigma_{sc} dv} \right]
\]

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

\[
L_o^{(1)}(x_o, \omega_o) = \sigma_s(x_o) \int_{\omega'_o}^{\infty} e^{-\sigma_{sc}s} L_i(x_i, \omega'_i) ds d\omega'_i \quad (6)
\]

\[
= \int_A \int_{\omega'_o} S^{(1)}(x_i, \omega'_i; x_o, \omega_o) L_i(x_i, \omega'_i) (n \cdot \omega'_i) d\omega_i dA(x_i).
\]

\[
\frac{(\alpha')^2}{4\pi} \left[ \left( C^- E \frac{z_r^2}{d_r^2} + C_\phi \frac{1}{D} \right) e^{-\mu_{tr} dr} - \left( C^- E \frac{z_v^2}{d_v^2} + C_\phi \frac{1}{D} \right) e^{-\mu_{tv} dv} \right]
\]
Physically-based reflectance profiles

• Formulas:

\[ R_d(r) = -D \frac{(\vec{n} \cdot \nabla \phi(x_0))}{d\Phi_i} \]

\[ = \frac{\alpha'}{4\pi} \left[ (\sigma_{tr} dr + 1) e^{-\sigma_{tr} dr} + \ldots (\sigma_{tr} d_{rv} + 1) e^{-\sigma_{tr} d_{rv}} \right] \]

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

\[ L^{(1)}_o(x_o, \vec{\omega}_o) = \sigma_s(x_o) \int_{2\pi}^{2\pi} F_p(\vec{\omega}'_i \cdot \vec{\omega}_o') \int_0^\infty e^{-\sigma_{tc} s} L_i(x_i, \vec{\omega}_i) \, ds \, d\vec{\omega}_i \]

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

\[ \nabla \phi'_d = \frac{\Phi}{4\pi D} \frac{e^{-\sigma_{tr} r}}{r^3} \left( \vec{\nu} \cdot 3D (1 + \sigma_{tr} r) - \vec{x} (1 + \sigma_{tr} r) \right. \]

\[ \left. - \vec{x} \frac{3D}{r} \frac{3(1 + \sigma_{tr} r) + (\sigma_{tr} r)^2}{\cos \theta} \right). \]
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:
Physically-based reflectance profiles

• More formulas:

\[ \phi(r) = \frac{e^{-\mu_t r}}{4\pi r^2} + \frac{\mu_s}{2\pi^2 r} \int_0^\infty \frac{\arctan^2 u}{u - \alpha \arctan u} \sin(r \mu_t u) \, du. \]
Physically-based reflectance profiles

• More formulas:

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

\[
\phi(r) = \frac{e^{-\mu_t' r}}{4\pi r^2} + \frac{1}{4\pi} \frac{3\mu_s' \mu_t'}{2\mu_a + \mu_s'} \frac{e^{-\sqrt{\frac{\mu_g}{D}} r}}{r}
\]
Physically-based reflectance profiles

• More formulas:

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

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

\[
\int_0^\infty G_3D(v, \sqrt{r^2 + z^2}) Q(z) \, dz = \frac{1}{2} \mu_s' f(\mu_t'^2 v/2) G_{2D}(v, r),
\]
Physically-based reflectance profiles

• More formulas:

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

\[
\phi(r) = \frac{e^{-\mu'_t r}}{2} + \frac{1}{2\mu_a + \mu'_s} \frac{3\mu'_s \mu_t}{r}
\]

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

\[
\frac{1}{4\pi D} \frac{e^{-r\sqrt{\frac{\mu_a}{D}}}}{r} = \int_0^\infty \frac{c}{(4\pi Dct)^{3/2}} e^{-\mu_act} e^{-r^2/(4Dct)} \, dt
\]
Physically-based reflectance profiles

• More formulas:

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

\[
\phi(r) = \frac{e^{-\mu_t r}}{2} + \frac{3\mu'_s \mu'_t}{2\mu_s + \mu'_s} \frac{e^{-\sqrt{\frac{\mu_s}{D}} r}}{r}
\]

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

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

\[
w_\phi(v, z_1, z_2, m) = \int_{z_1}^{z_2} \frac{e^{-\frac{(z + m)^2}{2v}}}{\sqrt{2\pi v}} \alpha' \mu_{i} e^{-\mu_{r} z} \, dz = \frac{\alpha' \mu_t}{2} e^{m \mu_t + \frac{\mu_t^2 v}{2}} \left( \text{erf} \left[ \frac{m + \mu_t v + z}{\sqrt{2v}} \right] - \text{erf} \left[ \frac{m + \mu_t v + z_1}{\sqrt{2v}} \right] \right).
\]
Physically-based reflectance profiles

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

- Some other formulas:
Physically-based reflectance profiles

• Some other formulas:

\[
R_d \approx \frac{1}{N} \sum_{i=1}^{N} \frac{f(\tilde{x}, \tilde{\omega}, t_i)w_{\text{exp}}(t_i, x)}{w_{\text{exp}}(t_i, x) \text{pdf}_{\exp}(t_i) + w_{\text{eq}}(t_i, x) \text{pdf}_{\text{eq}}(t_i | \tilde{x}, \tilde{\omega})} \\
+ \frac{1}{N} \sum_{j=1}^{N} \frac{f(\tilde{x}, \tilde{\omega}, t_j)w_{\text{eq}}(t_j, x)}{w_{\text{exp}}(t_j, x) \text{pdf}_{\exp}(t_j) + w_{\text{eq}}(t_j, x) \text{pdf}_{\text{eq}}(t_j | \tilde{x}, \tilde{\omega})}
\]
Physically-based reflectance profiles

• Some other formulas:

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

\[ + \frac{1}{N} \sum_{j=1}^{N} \frac{f(\bar{x}, \bar{\omega}, t_j) w_{\text{eq}}(t_j, x)}{w_{\text{exp}}(t_j, x) \text{pdf}_i} \]

\[ R^d_E(\bar{x}, t) = C_E \frac{\alpha'}{4\pi} \left[ \frac{z_r(t) (1 + \sigma_{tr} d_r(t)) e^{-\sigma_{tr} d_r(t)}}{d^3_r(t)} + \frac{(z_r(t) + 2z_b) (1 + \sigma_{tr} d_v(t)) e^{-\sigma_{tr} d_v(t)}}{d^3_v(t)} \right], \]
Physically-based reflectance profiles

• Some other formulas:

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

\[
R^d_E(\vec{x}, t) = C_E \frac{\alpha'}{4\pi} \left[ \frac{z_r(t)(1 + \sigma_{tr} d_r(t)) e^{-\sigma_{tr} d_r(t)}}{d^3_r(t)} + \frac{(z_r(t) + 2z_h)(1 + \sigma_{tr} d_v(t)) e^{-\sigma_{tr} d_v(t)}}{l^3_v(t)} \right],
\]

\[
r^{(1)}(\vec{x}, \vec{x}_r(t)) = \frac{f_s(\vec{\omega} \cdot \vec{\omega}_r \vec{x}) e^{-\sigma_r(d_r(t)))}}{d^2_r(t)} \frac{F_i(\theta_o, \eta) F_i(\theta_i, 1/\eta) \cos \theta_o}{l^3_v(t)},
\]
Approximate reflectance profiles

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

![Graph showing approximate reflectance profiles](image)
Approximate reflectance profiles

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

![Graph showing Monte Carlo references (searchlight) with different values of $A$ ranging from 0.10 to 0.90.](image)
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?
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 $d$:
  – $d = \text{mfp} / s$ (s depends on A)

• With a translation we can compare normalized diffusion with physically-based diffusion models
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.
Translation from mfp to $d$

For $mfp=1$:

- Find $s$ that minimizes difference between $R(r)$ and Monte Carlo reference for same $A$

  \[ R_{l=1}(r) = A \frac{e^{-sr} + e^{-sr/3}}{8 \pi r} \]

- For example: with optimal $s$ for $A = 0.2, 0.5, 0.8 \ldots$
Comparisons: surface albedo 0.2

A = 0.2

- MC reference
- our approx.
- Gauss2 approx.
- dipole diffusion + 1scat
- better dipole + 1scat
- beam diffusion + 1scat

Graph showing comparisons of different approximations for surface albedo 0.2.
Comparisons: surface albedo 0.5

$A = 0.5$

Graph showing comparisons for different approximations:
- MC reference
- Our approx.
- Gauss2 approx.
- Dipole diffusion + 1scat
- Better dipole + 1scat
- Beam diffusion + 1scat
Comparisons: surface albedo 0.8

- A = 0.8
- MC reference
- our approx.
- Gauss2 approx.
- dipole diffusion + 1scat
- better dipole + 1scat
- beam diffusion + 1scat
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
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
Translation from mfp to d

- Data points and fitted curve:

\[ s = 1.85 - A + 7|A - 0.8|^3 \]
Translation from mfp to d

• Error wrt. MC references is ~5.5%

• Small error compared to approximations and assumptions built into MC references: semi-infinite homogeneous volume, flat surface, ...
Diffuse surface transmission

- Searchlight configuration
  - milk, juice, oily skin, ...

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

vs.
Diffuse surface transmission

Monte Carlo references (diffuse)
Translation from mfp to d (diffuse)

- Data points and fitted curve:

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

![Graph showing data points and fitted curve with equation](image)
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
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)
Translation from dmfp to d

• Data points and fitted curve:

\[ s = 3.5 + 100 (A - 0.33)^4 \]
Translation from dmfp to d

- Error wrt. MC references is $\sim 7.7\%$
- In practical use: dmfp might be more intuitive than mfp; hence standard parameter of our previous scattering models
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!
Practical detail: importance sampling

• Importance sampling of distance $r$ between light entry and exit points: need $\text{cdf}(r)$

• For physically-based BSSRDFs the cdf has to be computed with numerical integration: slow

• Burley’s normalized diffusion has simple cdf:

$$\text{cdf}(r) = 1 - \frac{1}{4} e^{-sr/\ell} - \frac{3}{4} e^{-sr/(3\ell)}$$
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
Result: comparison w/ beam diffusion

beam diffusion + 1scatter  our approx
Result: comparison w/ beam diffusion

beam diffusion + 1scatter
our approx

(Head data: Infinite Realities)
Results

marble

fruits

plastic
Result: still life

image credit: Dylan Sisson
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
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?
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
Acknowledgments

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