Efficient Rendering of Human Skin
CS6630

Sunling Yang Tim Langlois

Cornell University

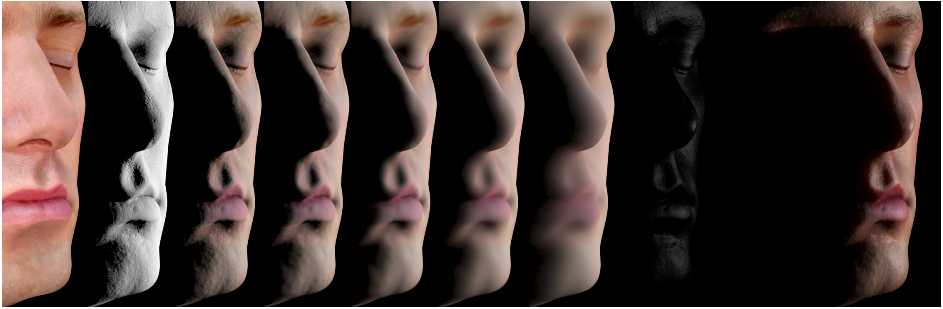
April 5, 2012

Outline

Assumptions/Approximations

1. Flat surface approximation
2. Ignore single scattering
3. Use 4-8 Gaussians to approximate the diffusion profile at each pixel point (original $R(r)$ is not a separable kernel but the approximation is)
4. Use texture space diffusion to approximate highly diffuse local scattering
5. Extend translucent shadow maps to approximate global scattering for highly curved texture space and close Euclidean space (e.g., ear)

Diffusion Profiles



From left to right: Albedo (1st) and irradiance (2nd) combine to give subsurface irradiance which is then convolved with each Gaussian basis profile (3rd through 7th) and combined in a final render pass with specular (8th) to produce the final image (9th). Convolutions are performed in off-screen 2D textures but shown here mapped onto the face.

Theory - extending from Jensen et al. 2001

- Single layer dipole approximation extended to develop multipole approximation for multiple thin layers
- Multiple bounces (reflection and transmission) across the different layers
- Accounting for rough surfaces
- This can be used to render the appearance of paint, paper, and human skin

Figure 7: *A translucent marble statue with surface roughness 0.1 on the left, 0.5 in the middle, and 1.0 on the right. The smaller images show the subsurface scattering component and the roughness component of the smooth (0.1) and the rough (1.0) translucent statues. Note how the smooth version is more shiny and brighter due to a higher subsurface scattering component. As the surface gets more rough the surface reflection increases, which reduces the amount of subsurface scattering, and the overall result is a desaturation of the color of the marble material.*

Extending from Dipole to Multipole

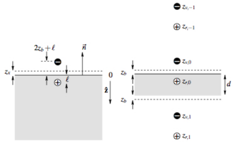


Figure 1: *Dipole configuration for semi-infinite geometry (left), and the multipole configuration for thin slabs (right).*

$$\alpha' = \frac{\sigma_s'}{\sigma_t'},$$
$$\text{r} = \|x_0 - x_i\|_2,$$
$$\sigma_{tr} = \sqrt{3\sigma_a \sigma_t'},$$
$$\text{D}=\frac{1}{3\sigma_t'},$$
$$\text{A} = \frac{1+F_{dr}}{1-F_{dr}},$$
F_{dr} = Fresnel diffuse reflectivity

- *Dipole case*: diffuse reflectance profile =
  $$\text{R(r)} = \frac{\alpha' z_r (1+\sigma_{tr} d_r) e^{-\sigma tr d_r}}{4\pi d_r^3} - \frac{\alpha' z_v (1+\sigma tr d_v) e^{-\sigma tr d_v}}{4\pi d_v^3}$$
- Equation to solve : $\phi(r) - 2AD\frac{\delta\phi(r)}{\delta r} = 0$ at z=0
- *Multipole* case (∞ approximated with 2n+1):
  $$\text{R(r)} = \Sigma_{i=-n}^{n} \frac{\alpha' z_{r,i} (1+\sigma_{tr} d_{r,i}) e^{-\sigma tr d_{r,i}}}{4\pi d_{r,i}^3} - \frac{\alpha' z_{v,i} (1+\sigma tr d_{v,i}) e^{-\sigma tr d_{v,i}}}{4\pi d_{v,i}^3}$$
- $$\text{T(r)} = \Sigma_{i=-n}^{n} \frac{\alpha' z_{r,i} (d-z_{r,i})(1+\sigma_{tr} d_{r,i}) e^{-\sigma tr d_{r,i}}}{4\pi d_{r,i}^3} - \frac{\alpha' z_{v,i} (d-z_{v,i})(1+\sigma tr d_{v,i}) e^{-\sigma tr d_{v,i}}}{4\pi d_{v,i}^3}$$
- Equations to solve:
  $\phi(r) - 2A(0)D\frac{\delta\phi(r)}{\delta r} = 0$ at z=0
  $\phi(r) - 2A(d)D\frac{\delta\phi(r)}{\delta r} = 0$ at z=d
- Note that the locations of the multipoles are *not* symmetric about the layers $\rightarrow$ they are scaled by the refractive indices of the media

Light bouncing across different layers

- The transmission profile T and the reflectance profile R are computed recursively across all the layers, T = $((T1 * T2) * T3) * ...$, where $*$ stands for convolution
- By taking T and R into frequency space using FFT, $*$ becomes multiplication and T from medium 1 to medium 2 $= T_{12}^{+} = T_1^{+} T_2^{+} + T_1^{+} R_2^{+} R_1^{-} T_2^{+} + ... =$ $T_1^{+} T_2^{+} (1 + (R_2^{+} R_1^{-}) + (R_2^{+} R_1^{-})^2 + ...) = \frac{T_1^{+} T_2^{+}}{1 - (R_2^{+} R_1^{-})}$, where $+$ and $-$ stand for forward-scattering and backward-scattering, respectively (Donner and Jensen 2005). This is the Kubelka-Munk equation in frequency space.

Accounting for Rough Surfaces

- Before there is transmission from medium 1 to medium 2, random walk, and back to medium 1, $S_d(x_i, \omega_i, x_o, \omega_o) = \frac{1}{\pi} F_i(x_i, \omega_i) R(\|x_i - x_o\|_2) F_i(x_o, \omega_o)$, $F_i =$ Fresnel transmittance
- Now, the Fresnel term is replaced with Cook-Torrance BRDF term averaged by Monte Carlo sampling, $S_d(x_i, \omega_i, x_o, \omega_o) = \frac{1}{\pi} \rho_{dt}(x_i, \omega_i) R(\|x_i - x_o\|_2) \rho_{dt}(x_o, \omega_o)$, $\rho_{dt}(x, \omega_o) - 1.0 - \int_{2\pi} f_r(x, \omega_o, \omega_i)(\omega_i \cdot n) d\omega_i$, $f_r(x, \omega_o, \omega_i) = \frac{D(x,\omega_o,\omega_i) G(x,\omega_o,\omega_i) F(x,\omega_i,\omega_o)}{4(\omega_i \cdot n)(\omega_o \cdot n)}$
- A $= \frac{1+\rho_d}{1-\rho_d}$, $\rho_d =$ average diffuse reflection factor computed by Monte Carlo sampling

Diffusion profile = Gaussian convolution = Gaussian blur

- $\frac{\delta C}{\delta t} = D \frac{\delta^2 C}{\delta x^2}$
- Taylor expansion : $C_{i+1} = C_i + \delta x \frac{\delta C}{\delta x} + \frac{1}{2} \delta x^2 \frac{\delta^2 C}{\delta x^2} + \ O(\delta x^3)$
  $C_{i-1} = C_i - \delta x \frac{\delta C}{\delta x} + \frac{1}{2} \delta x^2 \frac{\delta^2 C}{\delta x^2} + \ O(\delta x^3)$
  $C_{i+1} - C_{i-1} = 2 \delta x \frac{\delta C}{\delta x}$
  $\frac{\delta C}{\delta x} = \frac{C + i + 1 - C_{i-1}}{2 \delta x}$
  $C_i - C_{i-1} = \delta x \frac{\delta C}{\delta x} - \frac{1}{2} \delta x^2 \frac{\delta^2 C}{\delta x^2}$
  $(C_{i+1} - C_i) - (C_i - C_{i-1}) = \delta x^2 \frac{\delta^2 C}{\delta x^2}$
  $\frac{\delta^2 C}{\delta x^2} = \frac{(C_{i+1} - C_i) - (C_i - C_{i-1})}{\delta x^2}$
- $\frac{\delta C}{\delta t} = \frac{D}{\delta x^2} ((C_{i+1} - C_i) - (C_i - C_{i-1}))$
- $\frac{\delta C}{\delta t} = C_{i,n+1} - C_{i,n} \approx \frac{D}{\delta x^2} (C_{i+1,n} + C_{i-1,n} - 2 C_{i,n})$
  $C_{i,n+1} = (1 - 2\lambda) C_{i,n} + \lambda C_{i+1,n} + \lambda C_{i-1,n}, \ \lambda = \frac{D}{\delta x^2}$

Hack 1 : approximating $R^{+,-}(r), T^{+,-}(r)$ at each layer with 4 Gaussians each

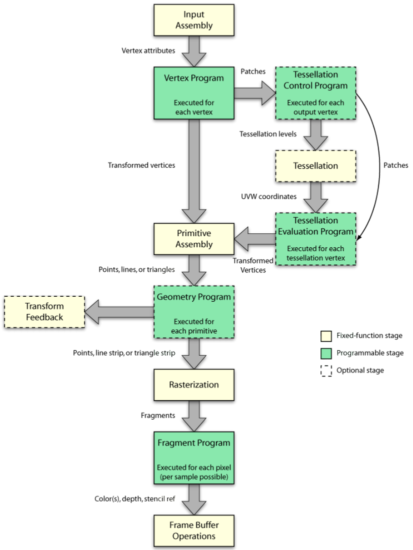
- Instead of FFT and multiplication in frequency space and inverse FFT back, a minimization of $\int_0^\infty r(R(r) - \Sigma_{i=1}^{k} w_i G(v_i, r))^2 dr$ is performed to find the diffusion profile R(r) with parameters weights w_i and variance v_i.
- $R^+ = \Sigma_{i=1}^{k_1} w_i G(\nu_i, r) * \Sigma_{j=1}^{k_2} w'_j G(\nu'_j, r) = \Sigma_{i=1}^{k_1} \Sigma_{j=1}^{k_2} w_i w'_j G(2\nu_i, r)$
  all initial slab profiles are fitted to powers of a single Gaussian of narrow variance
- Physical correctness requires infinite sum. In this case the sum goes until n, where n is found by computation where the Gaussian sum above converges towards the Kubelka-Munk equations with error < epsilon
  $\frac{T_1^+ T_2^+}{1-(R_2^+ R_1^-)}$ - $T_1^+ T_2^+ (\Sigma_{i=0}^{n} (R_2^+ R_1^-)^i) < \epsilon$

Approximate diffusion profiles as a linear combination of Gaussian basis functions = Why the Gaussian is the ultimate awesome function

- Convolution with Gaussian kernels is faster than FFT and inverse FFT
- 2D convolution can be split into two 1D convolutions
- Mean-free paths differ by frequency of spectral bands, so each diffusion profile R(r) has 3 components R, G, and B.
- Associative law of convolution to solve diffusion solution of many time steps as multiple time step convolutions
- The fact that $G(A + B) = G(A) * G(B)$ means diffusion across different layers =sum of different diffusion constants inside Gaussians

Graphics Hardware

- Many cores, very fast
  - Not as general as CPU
- Geometry → Processing → Pixels → Processing
- Shaders
  - Vertex
  - Fragment (Pixel)



Texture space diffusion

- Performs irradiance convolution
    1. Rasterize irradiance into a texture
        - Vertex shader
    2. Compute image filtering operations (convolutions) on that texture
        - Gaussian convolutions are blurs (fragment shader)
        - This is super fast due to the separable Gaussian kernel
    3. Texture map the result back onto the 3D mesh
- Assume single scattering is negligible

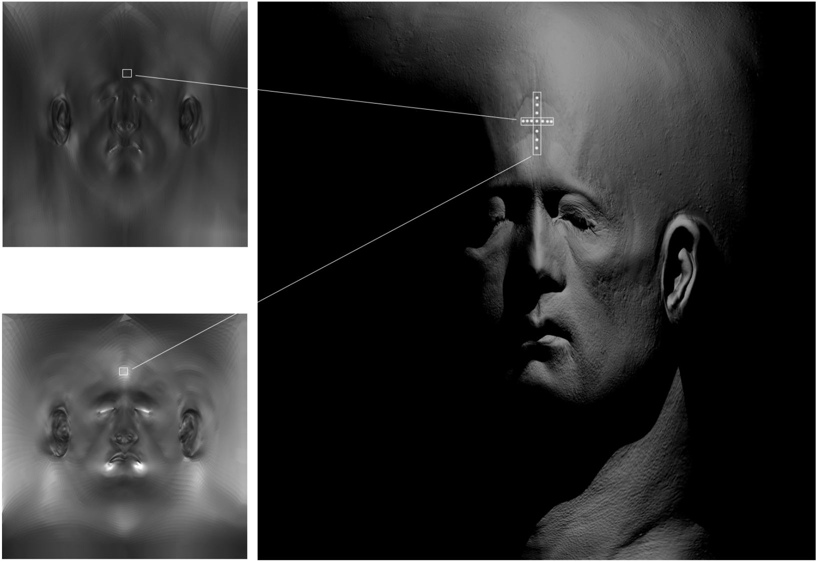
Texture space stretching

- When mapping to uv coordinates, texture distortion (stretching) occurs
- Diffusion between two points on the surface should depend on Euclidean distance
- Stretching needs to be accounted for during convolution
- Used to scale gaussian at each pixel (change weights of each pixel)
- Vertex shader provides derivatives from uv mapping, fragment shader computes stretch

Texture space stretching

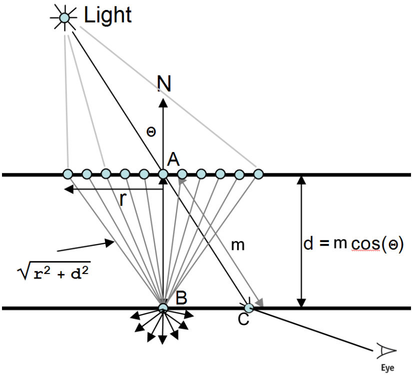
Without stretching (left) and accounting for stretching (right)

Texture space stretching



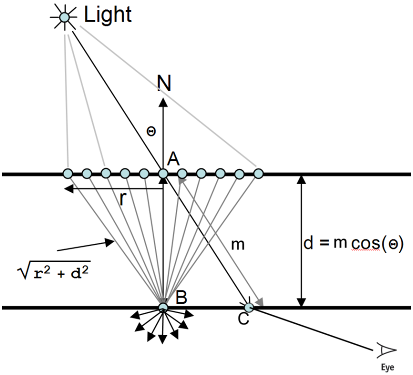
Translucent Shadow Maps (TSMs)

- Texture space diffusion captures local scattering, but not global scattering through thin regions (such as an ear)
  - These regions are close in Euclidean space but far in texture space
- For each pixel C of the shadow map, the TSM renders:
  - (u,v) coordinates of light facing surface
  - Depth of light facing surface



Translucent Shadow Maps (TSMs)

- Estimate scattered light at C
    - Convolution of irradiance at each light-facing point by profile R through the thickness of object
- Faster to do this at point B
- High-frequency changes in depth can cause artifacts
    - Convolve with depth also (use an average of depth)
- Global scattering term interpolated to 0 as point approaches light facing side



Translucent Shadow Maps (TSMs)

- A 3D convolution
  - Irradiance at each light facing point (2D convolution) with depth
- Gaussian kernels are separable
- TSMs can reuse the textures computed for local scattering
  - Weighted sum of k texture lookups
- SCORE!

Translucent Ear

Previous texture-space diffusion techniques (left). Modification to TSMs (center). Monte Carlo rendering (right).

Texturing

- Diffuse color map $\rightarrow$ infinitesimal, highly absorptive layer
  - Absorbs light once as it enters, once as it leaves
  - Two absorptions of $\sqrt{diffuseColor}$
  - Gives final skin tone

Apply specular terms

- Specular shading using precomputed specular texture map
  - Specular BRDF by Keleman and Szirmay-Kalos

Results

Donner 05

This Method

Performance

	resolution			no TSM		TSM	
	final render	irradiance	shadow maps	env only	1 light	1 light	2 lights
Full-Res	2500x1500*	2048x2048	2048x2048	60 fps	44 fps	31 fps	23 fps
Low-Res	1024x1024	1024x1024	1024x1024	115 fps	103 fps	75 fps	61 fps

*downsampled afterwards to display resolution

Results

Algorithm applied to color and normal maps captured from actors.

Results

No precomputation required for animated or deforming models.

Limitations and Future Work

- Texture stretching inaccurate for extreme curvature, and convolving across seams presents a problem
- TSMs
  - Bias needed to prevent speckling
  - Need one map per light (but environment lighting possible by importance sampling a few point lights)
  - Depth calculations are inaccurate for extreme curvature
- Have only shown low error fit of small number of Gaussians
  - No formal proof that the fits can be made arbitrarily accurate with more Gaussians
- Would like to support spatially varying/texture dependent diffusion profiles
  - Move $R(r)$ calculation to fragment shader
  - Would allow freckles, scars, makeup