Interactive Rendering of Translucent Objects

Hendrik Lensch, Michael Goesele, Philippe Bekaert, Jan Kautz, Marcus Magnor, Jochen Lang, Hans-Peter Seidel

2003

Presented By: Mark Rubelmann
Outline

- Motivation
- Background
- Preprocessing
- Rendering
- Results
Motivation

• Translucent objects = subsurface scattering
• Calculating subsurface scattering is expensive
• Observation: multiple scattering blurs and smoothes radiance
Motivation

- Low frequency can be taken advantage of
  - Global response
    - Long distance
    - Lots of scattering
    - Radiance can be calculated sparsely and interpolated
  - Local response
    - Short distance
    - Little scattering
    - Need to maintain detail for small neighborhood
Background

• Full BSSRDF: 8 dimensions
  – $S(x_i, \omega_i, x_o, \omega_o)$

• Diffuse subsurface scattering reflectance function: 4 dimensions
  – $R_d(x_i, x_o)$
Background

• $R_d$ relates incoming flux to outgoing diffuse radiance:

$$
L^\to(x_o, \omega_o) = \frac{1}{\pi} F_t(\eta, \omega_o) B(x_o)
$$

$$
B(x_o) = \int_S E(x_i) R_d(x_i, x_o) dx_i
$$

$$
E(x_i) = \int_{\Omega_+ (x_i)} L^\to(x_i, \omega_i) F_t(\eta, \omega_i) |N_i \cdot \omega_i| d\omega_i
$$
Background

- $R_d$ is very similar to $G$ in radiosity
  - Both are *throughput factors* (discrete version in Galerkin radiosity is *form factor*)
- $G$ only encodes geometric information; storage costs are too high for relighting
- $R_d$ maintains light transport properties between any two points and can handle dynamic lighting
Preprocessing

• Need discrete formulation of $B(x_o)$
• Actually use 2 formulations with two sets of basis functions
  – Global basis: hat functions at object vertices
  – Local basis: Piecewise-constant functions corresponding to surface texels
Preprocessing - Geometry

- Split mesh up into chunks of nearly-planar triangles and build 2D texture atlas
Preprocessing – Global Response

- Scattering over long distances is smooth
- Vertex-to-vertex throughput factors are used

\[ F_{ij} \approx R_d(v_i, v_j) \cdot \int_s \psi_i(x)dx \cdot \int_s \tilde{\psi}(y)dy = \frac{A_i}{3} R_d(v_i, v_j) \]

\[ B_j^g = \sum_i E_i F_{ij} \]
Preprocessing – Local Response

- Use texel-to-texel throughput factors to preserve details
- Modeled as 7 x 7 filter kernel

\[ K_{(u,v)}(s,t) = A(u,v)R_d(x_c(u,v), x_c(s,t)) \]
Preprocessing – Blending Local and Global

- Adding local and global results in twice the correct amount in direct illumination areas.
Preprocessing – Blending Local and Global

• Direct illumination found along diagonal of form factor matrix $F$
• $F^0$ is $F$ without direct illumination
• $B(x)$ found by introducing $B^d$

\[
B(x) = B^l(x) + B^d(x) + B^{g_0}(x)
\]

\[
B^{g_0}_j = \sum_i E^g_i F^{0}_{ij}
\]
Preprocessing – Blending Local and Global

• Also need to blend border between local and global
• Calculate “correct” radiosity by generating 9 x 9 kernel
• Adjust weighting of global radiosity to minimize difference
Rendering

- Compute direct illumination map
  - Implemented with vertex shader
- Split processing into two branches: global and local
- Global and local responses combined by multi-texturing in hardware
Rendering – Global Response

- Find irradiance at each vertex
- $B^g(y)$ at intermediate surface point $y$ is calculated by linear interpolation
- Surface radiosity can be modulated by texture, $T_p$

$$B^T_i = \frac{B^g_i}{T_p(v_i)}$$
Rendering – Local Response

• Convolve illumination map with filter kernel of every texel

\[ B^l(x) = K_{(u,v)}(s,t) \otimes E(s,t) = \sum_{(s,t) \in 7 \times 7} K_{(u,v)}(s,t)E(s,t) \]

• Initial implementation done in software
Results

- Renderings done on dual 1.7 GHz Xeon with 1 GB RAM and GeForce3 video card

<table>
<thead>
<tr>
<th>model</th>
<th># vertices</th>
<th>#form factors</th>
<th>fps</th>
<th>illummap</th>
<th>local</th>
<th>global</th>
<th>display</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>horse</td>
<td>10000</td>
<td>16441460</td>
<td>2.3</td>
<td>29</td>
<td>149</td>
<td>33</td>
<td>371</td>
<td>431</td>
</tr>
<tr>
<td>horse textured</td>
<td>10000</td>
<td>12409116</td>
<td>2.7</td>
<td>29</td>
<td>145</td>
<td>302</td>
<td>33</td>
<td>364</td>
</tr>
<tr>
<td>bust</td>
<td>8574</td>
<td>4946764</td>
<td>5</td>
<td>24</td>
<td>147</td>
<td>144</td>
<td>28</td>
<td>199</td>
</tr>
<tr>
<td>bird</td>
<td>4000</td>
<td>1750862</td>
<td>5.6</td>
<td>16</td>
<td>139</td>
<td>86</td>
<td>26</td>
<td>180</td>
</tr>
</tbody>
</table>
Results

Middle: simple blending   Right: optimized blending
Results

Local response  Global response  Combined
Results

With and without modulating texture
Results

Skim milk?