Rendering with Environment Maps

Jaroslav Křivánek, KSVI, MFF UK
Jaroslav.Krivanek@mff.cuni.cz
Acknowledgement

- Mostly based on Ravi Ramamoorthi’s slides available from http://inst.eecs.berkeley.edu/~cs283/fa10
Goal

- Real-time rendering with complex lighting, shadows, and possibly GI
- Infeasible – too much computation for too small a time budget

Approaches
- Lift some requirements, do specific-purpose tricks
  - Environment mapping, irradiance environment maps
  - SH-based lighting
- Split the effort
  - Offline pre-computation + real-time image synthesis
  - “Pre-computed radiance transfer”
Environment mapping (a.k.a. image-based lighting)

Miller and Hoffman, 1984
Later, Greene 86, Cabral et al, Debevec 97, …
Assumptions

- Distant illumination
- No shadowing, interreflection
Image-based lighting

• Illuminating CG objects using measurements of real light (=light probes)

Eucaliptus grove
Grace cathedral
Uffizi gallery

© Paul Debevec
Point lighting
Image-based lighting
Image-based lighting
Image-based lighting
Image-based lighting
Video

- Rendering with natural light
- Fiat Lux
Sampling strategies

<table>
<thead>
<tr>
<th>BRDF IS</th>
<th>EM IS</th>
<th>MIS</th>
</tr>
</thead>
<tbody>
<tr>
<td>600 samples</td>
<td>600 samples</td>
<td>300 + 300 samples</td>
</tr>
<tr>
<td>Diffuse only</td>
<td>Ward BRDF, $\alpha=0.2$</td>
<td>Ward BRDF, $\alpha=0.05$</td>
</tr>
</tbody>
</table>
Real-time rendering

- Mirror surfaces easy (just a texture look-up)

- What if the surface is rougher...

- Or completely diffuse?
Reflection Maps

- **Phong model** for rough surfaces
  - Illumination function of reflection direction $R$
- **Lambertian diffuse** surface
  - Illumination function of surface normal $N$
- Reflection Maps [Miller and Hoffman, 1984]
  - Irradiance (indexed by $N$) and Phong (indexed by $R$)
Reflection Maps

- Can’t do dynamic lighting
  - Slow blurring in pre-process
SH-based Irradiance Env. Maps

Incident Radiance (Illumination Environment Map) → Irradiance Environment Map

R

N
Lambertian surface acts like low-pass filter

$$E_{lm} = A_l L_{lm}$$

Ramamoorthi and Hanrahan 01
Basri and Jacobs 01

$$A_l = 2\pi \frac{(-1)^{\frac{l}{2}-1}}{(l+2)(l-1)} \left[ \frac{l!}{2^{l} \left(\frac{l}{2}\right)!} \right] \quad l \text{ even}$$
9 Parameter Approximation

RMS error = 25 %
9 Parameter Approximation

RMS Error = 8%

Exact image

Order 1
4 terms

$Y_{lm}(\theta, \varphi)$
9 Parameter Approximation

RMS Error = 1%

For any illumination, average error < 3% [Basri Jacobs 01]
Real-Time Rendering

- Simple procedural rendering method (no textures)
  - Requires only matrix-vector multiply and dot-product
  - In software or NVIDIA vertex programming hardware

- Widely used in Games (AMPED for Microsoft Xbox), Movies (Pixar, Framestore CFC, ...)

```c
surface float1 irradmat (matrix4 M, float3 v) {
  float4 n = {v , 1} ;
  return dot(n , M*n) ;
}
```
SH-based Irradiance Env. Maps

Images courtesy Ravi Ramamoorthi & Pat Hanrahan
- Video – Ramamoorthi & Hanrahan 2001
SH-based Arbitrary BRDF Shading 1

- [Kautz et al. 2003]
- Arbitrary, dynamic env. map
- Arbitrary BRDF
- No shadows

SH representation
- Environment map (one set of coefficients)
- Scene BRDFs (one coefficient vector for each discretized view direction)
- **BRDF Representation**
  - BRDF coefficient vector for a given $\omega_o$, looked up from a texture (use e.g. paraboloid mapping to map $\omega_o$ to a texture coordinate)
  - BRDF coefficients pre-computed for all scene BRDFs (SH projection)
Rendering: for each vertex / pixel, do

\[
L_o(\omega_o) = \int_{\Omega} L_i(\omega_i) \cdot BRDF(\omega_i, \omega_o) \cdot \cos \theta_i \cdot d\omega_i
\]

\[= \text{coeff. dot product}\]

\[L_o(\omega_o) = \Lambda_{\text{intp}}(p) \cdot F(p, \omega_o)\]
SH-based Arbitrary BRDF Shading

- BRDF is in local frame
- Environment map in global frame
- Need coordinate frame alignment -> **SH rotation**

- SH closed under rotation
  - rotation matrix
  - Fastest known procedure is the $zxzxz$-decomposition
    [Kautz et al. 2003]

$$R_{SH} = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\
0 & x & x & x & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\
0 & x & x & x & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\
0 & x & x & x & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\
0 & x & x & x & X & X & X & X & X & X & \cdots \\
0 & 0 & 0 & 0 & X & X & X & X & X & X & \cdots \\
0 & 0 & 0 & 0 & X & X & X & X & X & X & \cdots \\
0 & 0 & 0 & 0 & X & X & X & X & X & X & \cdots \\
0 & 0 & 0 & 0 & X & X & X & X & X & X & \cdots \\
0 & 0 & 0 & 0 & X & X & X & X & X & X & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\end{bmatrix}.$$
Figure 3: Brushed metal head in various lighting environments.

(a) varying exponent  
(b) varying anisotropy

Figure 4: Spatially-Varying BRDFs.
Video: Kautz 2003
Environment Map Summary

- Very popular for interactive rendering
- Extensions handle complex materials
- Shadows with precomputed transfer
- But cannot directly combine with shadow maps
- Limited to distant lighting assumption