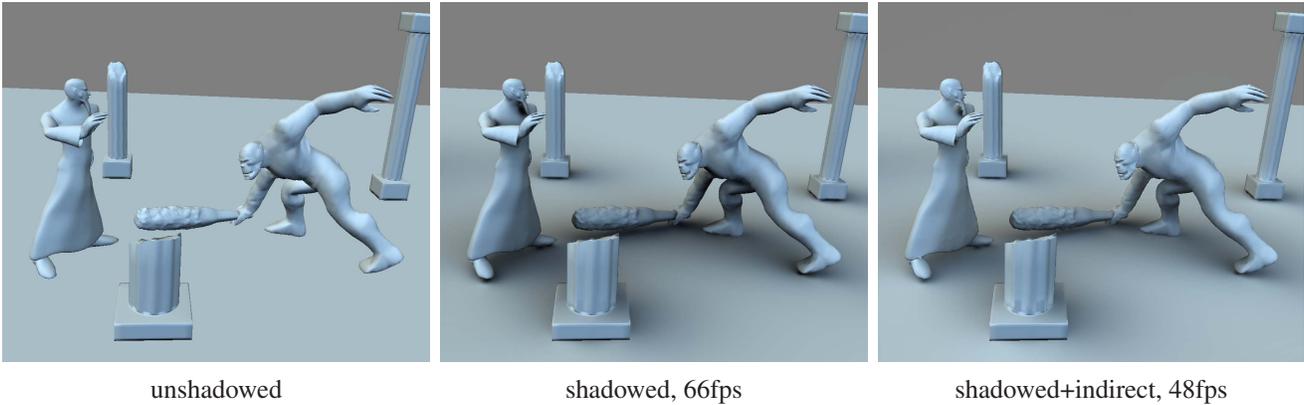


Image-Based Proxy Accumulation for Real-Time Soft Global Illumination

Peter-Pike Sloan Naga K. Govindaraju Derek Nowrouzezahrai John Snyder
Microsoft Corporation Microsoft Corporation University of Toronto Microsoft Research



Abstract

We present a new, general, and real-time technique for soft global illumination in low-frequency environmental lighting. It accumulates over relatively few spherical proxies which approximate the light blocking and re-radiating effect of dynamic geometry. Soft shadows are computed by accumulating log visibility vectors for each sphere proxy as seen by each receiver point. Inter-reflections are computed by accumulating vectors representing the proxy’s unshadowed radiance when illuminated by the environment. Both vectors capture low-frequency directional dependence using the spherical harmonic basis. We also present a new proxy accumulation strategy that splats each proxy to receiver pixels in image space to collect its shadowing and indirect lighting contribution. Our soft GI rendering pipeline unifies direct and indirect soft effects with a simple accumulation strategy that maps entirely to the GPU and outperforms previous vertex-based methods.

1. Introduction

Global illumination (GI) effects like soft shadows and inter-reflections greatly enhance realism but are challenging to render in real-time. Previous approaches capture only hard shadows from point or small area lights, or render soft effects only in static scenes. We target soft GI effects in dy-

amic scenes containing moving characters. Dynamic geometry is first approximated as a set of a few spheres used as proxies for how that geometry blocks and re-radiates light.

Monte Carlo techniques trace hundreds to thousands of rays around each receiver point to integrate direct shadowing of large light sources or indirect reflections. We instead consider all directions simultaneously using the spherical harmonic (SH) basis to represent proxy visibility and indirect radiance. Small SH vectors (e.g., order 4 yielding 16D vectors) make the computation practical and are sufficient for realistic soft effects. Each proxy’s SH vector can be computed using a simple formula. We obtain the total shadowed plus indirectly illuminated result by accumulating vectors over all proxies at each receiver point.

Proxies are accumulated in image space rather than in object space (i.e., over mesh vertices) or texture space (i.e., using a surface parameterization). Each pixel represents a receiver location for accumulating GI proxies. Shading computation thus takes place only where it is visible. Soft GI effects require much coarser sampling than the final display resolution which must support high-frequency textures and silhouettes. When upsampling the shading results, we apply bilateral filtering [15] to avoid interpolating over discontinuities in the receiver’s depth or normal.

Our contribution is a new, real-time method for soft GI effects on dynamic geometry. Unlike previous SH proxy accumulation approaches [17, 12], it is image-based and implementable entirely on the GPU. Its sampling rate is

also decoupled from the display’s. We introduce a new algorithm for accumulating indirect radiance from spherical proxies in environmental lighting, which ensures arbitrary proxy overlap does not yield unbounded radiance. Our rendering approach handles more GI effects, maps better to GPUs, provides higher-quality sampling, and is many times faster than previous object-based techniques.

2. Previous Work

Blocker Accumulation in Object Space [8] accumulates blockers by rasterizing them into bitmaps at each receiver vertex. This rapidly becomes impractical as the number of blockers and receivers grows beyond a small number. Our method uses simple, spherical proxies, for which accumulation can be done trivially and analytically in the SH basis.

Shadow fields [17] tabulate an SH visibility vector in the 3D space surrounding each object. This requires large tables and slow SH products, limiting scenes to just a few rigid blockers. Recently, this method was extended to diffuse interreflection [7], by using source radiance fields (SRFs) based on a very low order (4 component) Fourier basis in texture space for each object’s indirect contribution.

Spherical harmonic exponentiation (SHEXP) [12] speeds up accumulation to allow many more, deformable blockers. It uses spherical proxies and accumulates their visibility in log space to convert expensive SH products to simple SH adds. We build on these two ideas but make a number of important new contributions. By splatting proxies into image space during accumulation, we obtain higher-quality sampling at reduced rendering cost. Our method also supports indirect reflections as well as shadowing. Methods and error analysis for approximating geometry by sphere sets is discussed in [16, 12].

Ambient Occlusion and Other Shadowing Methods

Ambient occlusion (AO) represents simple properties of occluders, such as their solid angle [4] or a spherical cap bound [9, 10]. Extremely soft shadows are obtained which provide a proximity cue but little response to lighting directionality (see Figure 1 in [12]). Shadowing from multiple blockers is problematic because AO cannot accurately estimate how much the solid angles of these blockers overlap. We account for such overlaps using products of the SH visibility functions, which retain (low-frequency) knowledge of which directions are occluded.

Multi-pass soft shadowing [13, 1, 11] and soft shadow volumes [3] slow down dramatically as light sources grow in extent and so are impractical for environmental lighting. These methods also neglect inter-reflection.

Image-Space Splatting Methods [5] forms point light sources by selecting a representative set over the image

as seen by a light source. Spherical or ellipsoidal regions around these lights are then splatted to accumulate their shading influence over receiving geometry. This method doesn’t handle environmental lighting or produce soft shadows, and the shading is accumulated at final display resolution. We share the idea of splatting a region of influence except that ours represents the blocking/re-radiating effect of a large surface region rather than indirect radiance emanating from a point light and bouncing off a single surface point.

[14] splats regions of influence of spherical blockers to accumulate ambient occlusion. Our main contribution is to extend this idea to higher visibility frequencies beyond simple AO in order to cast lighting-dependent soft shadows and to include indirect reflections. We also accelerate rendering by sampling at less than the display resolution. Finally, we smoothly clamp the proxy’s region of influence to zero to eliminate discontinuities.

Offline Rendering Acceleration Our aggressive visibility simplification for indirect illumination is similar to the near-field approximation in [2], but applied to spherical proxies rather than nearby triangles. Rather than simply summing over all proxies blocking the receiver, we also normalize by their total solid angle including overlap.

3. Terminology and Review

As shown in Figure 1, proxy accumulation and shading takes place at each receiver point, denoted p . Receiver points are represented in image space as pixels containing a position and normal vector. A set of m proxy spheres are used to block and shed indirect lighting at p ; sphere i has center c_i and radius r_i . The distance from p to the proxy center is $d_i = \|c_i - p\|$ and associated direction $\hat{d}_i = (c_i - p)/d_i$. The distance to the closest point on the sphere is denoted $\tilde{d}_i = d_i - r_i$. The half-angle subtended by the proxy at p is

$$\sin \theta_i = r_i/d_i = \alpha_i, \tag{1}$$

and its corresponding solid angle is denoted

$$\omega_i = 2\pi (1 - \cos \theta_i) = 2\pi \left(1 - \sqrt{1 - \alpha_i^2}\right). \tag{2}$$

Proxy Visibility The proxy visibility function as seen from p is denoted $V_i(s)$, defined via:

$$V_i(s) = \begin{cases} 1, & \text{if } s \cdot \hat{d}_i \geq \cos \theta_i, \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

where $s \in S = \{(s_x, s_y, s_z) \mid s_x^2 + s_y^2 + s_z^2 = 1\}$ is a point on the sphere S , representing a direction emanating from

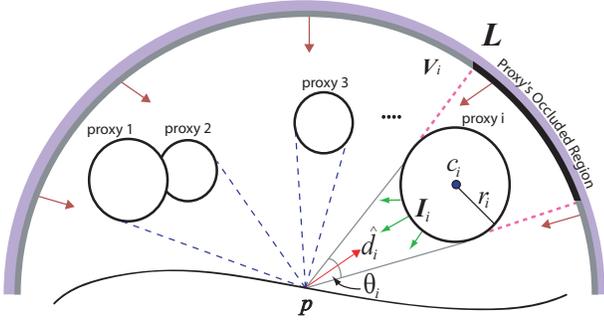


Figure 1. Proxy accumulation. A set of spherical proxies representing dynamic geometry block environmental lighting L through visibility function V_i and re-radiate indirect illumination I_i to a receiver point p .

p . The SH vector representing visibility is denoted V_i . We use bold face to denote SH vectors throughout the paper. We also define *occlusion* for each proxy which represents visibility of the distant environment rather than the proxy itself, denoted

$$O_i = \mathbf{1} - V_i = (\sqrt{4\pi}, 0, \dots, 0) - V_i$$

where the SH vector $\mathbf{1}$ represents the constant 1 over the sphere and has only its first (DC) coefficient nonzero, denoted $\mathbf{1}_0 = \sqrt{4\pi}$. Note that $\omega_i = V_i \cdot \mathbf{1} = (V_i)_0 \sqrt{4\pi}$.

The total occlusion function over all proxies is denoted by $O_p(s)$, and its SH projection O_p . As shown in [12], the total occlusion vector accounting for overlap can be computed from the sum of the individual log occlusion vectors, denoted O_i^{\log} , via:

$$O_p \approx O_1 * O_2 * \dots * O_m \approx \exp\left(\sum_{i=1}^m O_i^{\log}\right). \quad (4)$$

This is an instance of the standard engineering trick that converts a chain of products into a sum of logs. In this case, visibility is accumulated using $m - 1$ simple vector adds instead of expensive SH products, at the cost of evaluating a single SH exponential. The benefits increase as the number of proxies m grows. We adopt the HYB method for SH exponential discussed in [12].

Indirect accumulation also requires the summed solid angle of the proxies, which we denote $\omega_p = \sum_{i=1}^m \omega_i$.

SH Review Let $f(s)$ be a spherical function, represented by the doubly indexed SH vector f_{lm} . It is often convenient to consider these vectors as singly indexed as well, as in f_i . SH vectors of order n have n^2 components. Let $g(s)$ be a

circularly symmetric function about $z = (0, 0, 1)$. Then g can be represented by a zonal harmonic (ZH) vector g_l having nonzero coefficients only for $m=0$. Projecting a spherical function $f(s)$ to an SH vector f involves integration against the SH basis functions y :

$$f_{lm} = \int_{s \in S} f(s) y_{lm}(s) ds. \quad (5)$$

Evaluating f at the spherical point s is computed via

$$f(s) = \sum_{lm} f_{lm} y_{lm}(s). \quad (6)$$

Convolving f by g yields:

$$(f * g)_{lm} = \sqrt{\frac{4\pi}{2l+1}} f_{lm} g_l = C_g(f) \quad (7)$$

where C_g is a diagonal matrix that repeats l times each component g_l , scaled by $\sqrt{\frac{4\pi}{2l+1}}$. Rotating g from its canonical center at z to an arbitrary one z' is computed via

$$\text{rot}(g, z')_{lm} = \sqrt{\frac{4\pi}{2l+1}} g_l y_{lm}(z'). \quad (8)$$

The product of two spherical functions represented by two SH vectors is denoted $f_1 * f_2$, and involves applying the order-3 triple product tensor to the two input vectors [12].

Shading in SH Distant environment light is represented by the spherical function $L(s)$ and SH vector L . Unshaded shading from this light onto a diffuse receiver (i.e., irradiance) is denoted $\tilde{L} = L * g_{\cos} = C_{g_{\cos}} L$. It can be computed via (7) by convolving with the clamped cosine function around z : $g_{\cos}(s) = \max(0, s_z)$. Indirect radiance from proxy i as seen by p is denoted $I_i(s)$ and SH vector I_i . Total indirect radiance at p is denoted I_p and is computed by normalizing the summed indirect radiance $\tilde{I}_p = \sum_i I_i$.

Final incident radiance at p is shadowed direct illumination plus indirect:

$$L_p = O_p * L + I_p. \quad (9)$$

This involves two accumulations over the proxy set at each receiver: one for log occlusion O_p and one for indirect radiance I_p . These accumulation steps will be further detailed in the next sections.

Given a receiver point p and a spherical proxy, O_i^{\log} is computed using a 1D table which stores log visibility ZH vectors for a canonically positioned sphere (centered along the z axis) as a function of $\alpha_i = \sin \theta_i$. Equation 8 then allows the tabulated vector corresponding to the angle subtended by the proxy at p to be rotated to its actual direction $z' = \hat{d}_i$. See [12, equation 34 and Section 5] for the precise calculation. Self-shadowing by proxies that contain p

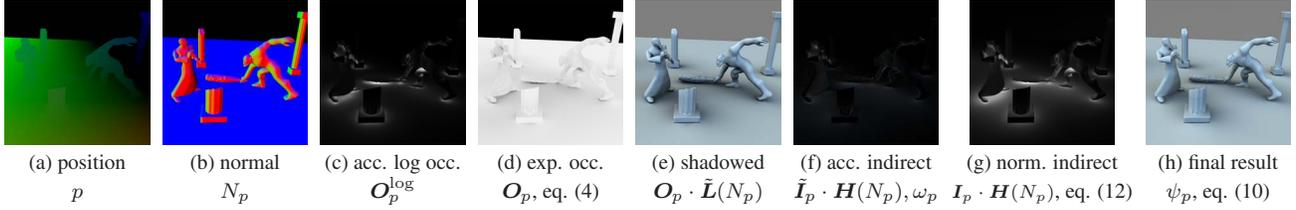


Figure 2. Rendering pipeline: successive passes are visualized from left to right.

or intersect its tangent plane are handled with special replacement rules [12].

Assuming a diffuse surface with normal vector N_p , the shade at p is given by

$$\psi_p = \mathbf{H}(N_p) \cdot \mathbf{L}_p = \mathbf{O}_p \cdot \tilde{\mathbf{L}}(N_p) + \mathbf{I}_p \cdot \mathbf{H}(N_p) \quad (10)$$

where $\mathbf{H}(N_p) = \text{rot}(\mathbf{g}_{\cos}, N_p)$ represents the diffuse BRDF: a clamped cosine around the normal.

4. General Accumulation Pipeline

Our algorithm’s pipeline is illustrated in Figure 2. It begins with a *receiver setup pass* which rasterizes the 3D receiver point $p = (p_x, p_y, p_z)$ along with its normal N_p at each pixel of a low-resolution image called the *receiver buffer*, as well as to the higher display resolution. Typically, the receiver buffer is 1/2 or 1/4 of the display’s resolution in both x and y . A shader transforms the point/normal pair (p, N_p) to two separate buffers. One stores the receiver point p and the other a 4D vector (N_p, z_p) where $z_p = (p - e) \cdot E$ represents the eye-space depth of p given eye point e and image plane normal E . The first buffer is necessary to accumulate the proxies; the second to compute the shading. It also allows robust upsampling of the coarsely sampled shading using bilateral filtering, as will be explained in Section 4.2.

Two *accumulation passes* then splat the proxies to the receiver buffer to sum their shadowing and indirect reflection. They form the bottleneck of our computation and are detailed in later sections. A *coverage oracle*, based on a conservative sphere of influence for each proxy, ensures proxies are splatted only where necessary and is discussed in the next section.

An *exponentiation pass* then computes the SH exponential of the accumulated log visibility to obtain \mathbf{O}_p . This result is dotted with the lighting, yielding the first term in (10). Indirect radiance is then accumulated and added to this result.

4.1. Coverage Oracle

The coverage oracle bounds each proxy’s *sphere of influence*, representing the region of receiver points that can

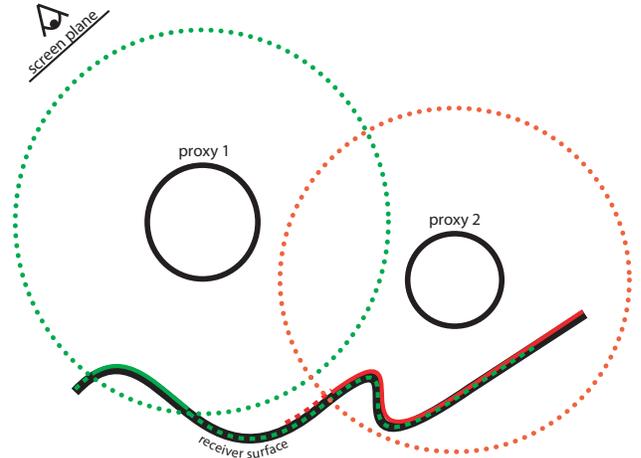


Figure 3. Splatting a proxy’s sphere of influence. Red and green lines over the receiver surface represent two proxy splats. Solid lines represent points which pass the depth test and so are inside the sphere of influence. Dashed lines show where the splat fails the depth test.

possibly receive a contribution from it. While this region is technically infinite, a proxy’s contribution falls off rapidly with distance. We thus approximate it by expanding the proxy’s radius by a fixed factor denoted η . Splatting the proxy’s sphere of influence then accumulates its contribution at all relevant receiver points, as shown in Figure 3.

For shadowing on diffuse surfaces, a conservative expansion factor is obtained assuming lighting via a delta function, which is the “peakiest” light representable at the given SH order n . The receiver normal is further assumed to point directly toward the strongest light direction. The light vector is normalized so that unshadowed shading maps to a unit result (i.e., just reaches saturation). Given this conservative configuration, the sphere of influence is chosen so that receiver points outside it yield no more than a small difference between shadowed and unshadowed shading. For SH order 4, an expansion factor of roughly $\eta=15$ (equivalent to a half-

angle $\theta=3.8^\circ$) yields a shading difference on the order of 1 in 256, suitable for 8-bit per channel rendering.

This idea can be specialized to the actual lighting environment instead of assuming the worst case. The same idea of determining the largest blocker with sufficiently small shading difference can be applied. For environments like a cloudy sky, a smaller expansion factor can be used which decreases splat area and thus rendering cost.

Similarly, a conservative expansion factor for indirect accumulation can be derived by analyzing shading differences. Since we use only SH order 3, an expansion factor of $\eta=10$ suffices.

To splat the proxy, we rasterize a coarsely tessellated sphere bounding its sphere of influence, as seen by the viewpoint. The depth of each covered pixel is then tested to see whether the corresponding receiver point is inside the sphere of influence. Only pixels inside need be processed further to accumulate shadowing or indirect illumination.

Smooth Clamping for the Sphere of Influence We force the proxy’s contribution to 0 smoothly at the boundary of its sphere of influence. This removes a discontinuity that would otherwise arise by abruptly eliminating the proxy’s blocking or inter-reflection effect. Given the radius expansion factor η , the smallest proxy supported at the receiver has half-angle sine of $\alpha_i = \sin \theta_i = 1/\eta$. We can smoothly clamp the proxy’s contribution by re-weighting the table of canonical ZH log occlusion vectors, $\mathbf{O}^{\log}(\alpha_i)$, for shadow accumulation or ZH visibility vectors, $\mathbf{V}(\alpha_i)$, for indirect accumulation. We choose the 4 table entries around $1/\eta$ and weight their corresponding ZH vectors smoothly to 0. Vectors for even smaller half angles are forced to 0.

4.2. Bilateral Upsampling

After splatting the proxies and exponentiating, the receiver buffer is filled with occlusion vectors, \mathbf{O}_p . Similarly, a separate pass accumulates indirect shading, $\mathbf{I}_p \cdot \mathbf{H}(N_p)$, per receiver. These must be upsampled to display resolution before summing in (10).

During this upsampling, the interpolation kernel should not straddle discontinuities in the receiver geometry; e.g. across silhouettes or over large differences in normal. We detect and handle this situation using a variant of bilateral filtering [15]. Interpolated source samples at a coarse resolution are weighted based on their difference with the depth and normal of the (finer-resolution) target pixel. In other words, only source samples that are similar to the target’s receiver plane can contribute. A similar solution for upsampling indirect shading searches for the closest position/normal in each 3×3 pixel neighborhood [6].

Given the receiver plane information at the 2×2 block of coarse source samples indexed by $i = 1, 2, 3, 4$ and the

target sample at p , we interpolate using the 4 weights

$$w_i = \frac{\tilde{w}_i}{\sum_{j=1}^4 \tilde{w}_j} \text{ where } \tilde{w}_i = w_i^b w_i^z w_i^N. \quad (11)$$

The individual weight factors are defined as

$$w_i^N = (N_p \cdot N_i)^{32} \text{ and } w_i^z = \frac{1}{\epsilon + |z_p - z_i|}$$

and w_i^b are the standard bilinear weights determined by the 2D position of the target relative to the source samples.

It can happen that no coarse sample sufficiently matches the target; in other words, all four unnormalized weights, \tilde{w}_i , are nearly zero. This problem typically arises at very few pixels along silhouettes, assuming no objects are completely unsampled in the receiver buffer. A robust solution is to perform an additional pass which shades these few from scratch given the entire list of proxies. A cheaper solution is simply to interpolate shaded results rather than occlusion vectors which gives rise to fewer artifacts when interpolated.

5. Accumulating Indirect Radiance

Indirect lighting from each proxy is modeled assuming it is directly illuminated by the environment \mathbf{L} without shadows. We describe a progression of three methods: a simple approximation which pastes radiance sampled at the proxy center over its entire visible disk, a more accurate approximation that averages shading across the entire proxy, and a linear operator producing the exact radiance distribution.

Given the indirect lighting over each proxy, we compute the total by summing over all proxies, which ignores inter-proxy occlusion effects. However, we normalize the result to make sure lighting does not over-saturate when the proxies overlap, via:

$$\mathbf{I}_p = \xi_p \tilde{\mathbf{I}}_p \text{ where } \xi_p = \frac{(\mathbf{V}_p)_0}{\omega_p / \sqrt{4\pi}}. \quad (12)$$

The normalization factor ξ_p represents the ratio of the solid angle of all proxies combined (DC component of total visibility vector $\mathbf{V}_p=1 - \mathbf{O}_p$ where \mathbf{O}_p is computed using (4)), to total solid angle summed over all proxies i ignoring overlap. The indirect accumulation pass thus sums the r,g,b components of $\mathbf{I}_i \cdot \mathbf{H}(N_p)$ as well as the scalar solid angle ω_p . The result is then normalized and added to the previously accumulated shadowed radiance in a final pass.

Point-Sampled Indirect Radiance The simplest method samples radiance in the direction opposite to the proxy center, $-\hat{d}_i$, and assumes its entire visible disk emits this constant radiance value. The radiance vector for proxy i can

thus be computed as

$$\mathbf{I}_i = \tilde{\mathbf{L}}(-\hat{d}_i) \mathbf{V}_i = \left(\tilde{\mathbf{L}} \cdot \mathbf{y}(-\hat{d}_i) \right) \mathbf{V}_i. \quad (13)$$

Averaged Indirect Radiance The second method also assumes a constant radiance over the proxy's disk, but more accurately averages radiance across it instead of simply sampling at the center.

Assume a ray in direction s emanating from p outside the proxy first hits it at point $q(s)$ (which implies $V_i(s) = 1$):

$$q(s) = p + s \left(-s \cdot p_i - \sqrt{(s \cdot p_i)^2 - \|p_i\|^2 + r_i^2} \right)$$

where $p_i = p - c_i$. Denote the normalized direction from the proxy center to $q(s)$ as $\hat{q}(s) = (q_s - c_i)/r_i$. Then unshadowed radiance at $q(s)$ due to the environment is $\tilde{\mathbf{L}}(\hat{q}(s))$ since the sphere's unit normal vector at $q(s)$ is equal to $\hat{q}(s)$. Average radiance across the proxy's disk is then given by the following integral:

$$D[\tilde{\mathbf{L}}] = \frac{1}{\omega_i} \int_{s \in S} V_i(s) \tilde{\mathbf{L}}(\hat{q}(s)) ds.$$

This is a linear operator on $\tilde{\mathbf{L}}$ producing a scalar, so it can itself be represented as an SH vector.

If we canonically orient the proxy so that $c_i = (0, 0, 1)$, $p = (0, 0, 0)$, and $r_i = \sin \theta$, then this linear operator is only a function of the angle subtended θ , which we denote

$$D(\theta) = \frac{1}{\omega(\theta)} \int_{s \in S} v_\theta(s) \mathbf{y}(\hat{q}_\theta(s)) ds. \quad (14)$$

For this canonical configuration,

$$q_\theta(s) = s \left(s_z - \sqrt{s_z^2 - \cos^2 \theta} \right)$$

and $\hat{q}_\theta(s) = (q_\theta(s) - (0, 0, 1)) / \sin \theta$. The canonical visibility function, $v_\theta(s)$, is 1 if and only if $s_z \geq \cos \theta$. D is circularly symmetric around z so its only nonzero coefficients are for $m=0$.

The formula in (8) rotates D from its canonical orientation along $(0,0,1)$ to the direction along the actual proxy, \hat{d}_i . This scalar result then multiplies the proxy's visibility vector \mathbf{V}_i , yielding the final formula

$$\mathbf{I}_i = \kappa_i \mathbf{V}_i \text{ where } \kappa_i = \text{rot}(D(\theta_i), \hat{d}_i) \cdot \tilde{\mathbf{L}}. \quad (15)$$

The visibility vector is given by rotating a canonical visibility vector \mathbf{V} (again oriented directly above the z axis) tabulated as a function of α_i :

$$\mathbf{V}_i = \text{rot} \left(\mathbf{V}(\alpha_i), \hat{d}_i \right).$$

As in the case of shadowed accumulation, the table $\mathbf{V}(\alpha)$ can be re-weighted given the radius expansion factor η to smoothly clamp the proxy's indirect contribution to 0 at the boundary of its sphere of influence.

An analytic formula for the order-3 operator is given by:

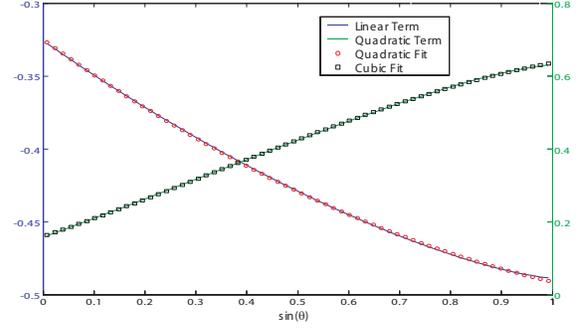


Figure 4. Linear and quadratic components of D and their polynomial fits as a function of $\sin \theta$. The linear component is fit with a quadratic with a relative squared error of 0.00016% while the quadratic component is fit with a cubic polynomial with an error of 0.00064%.

$$D(\theta) = \left(\frac{1}{2\sqrt{\pi}}, \frac{-\sqrt{3}(2d^3 - 2\check{d}d^2 - \check{d} + 1)}{6\sqrt{\pi} d(d - \check{d})}, \frac{\sqrt{5}(6d^5 - 6\check{d}d^4 - 3\check{d}d^2 - \check{d} + 4)}{20\sqrt{\pi} d^2(d - \check{d})} \right)$$

where $d = \csc \theta$ and $\check{d} = \sqrt{d^2 - 1}$. These three components are very smooth functions of $\alpha = \sin \theta$ and can be accurately approximated using low-order polynomial functions as shown in Figure 4. We use the following approximation:

$$\begin{aligned} D_0 &= 0.28209 \\ D_1 &\approx 0.08432 \alpha^2 - 0.25073 \alpha - 0.32494 \\ D_2 &\approx -0.22230 \alpha^3 + 0.22502 \alpha^2 + 0.47398 \alpha + 0.15950 \end{aligned}$$

Exact Indirect Radiance A linear operator on $\tilde{\mathbf{L}}$ yields the exact spherical function for radiance over the proxy. Assuming the proxy is in the same canonical orientation as in the previous subsection, the operator is solely a function of angle θ subtended by the proxy, defined by

$$M_{j k}(\theta) = \int_{s \in S} v_\theta(s) \mathbf{y}_j(s) \mathbf{y}_k(\hat{q}_\theta(s)) ds. \quad (16)$$

M can be evaluated analytically or by numerical integration as a preprocess and tabulated as a function of θ .

To apply this canonical operator, we must rotate the lighting into the canonical proxy's frame, apply the canonical operator, and then rotate the result back, via

$$\mathbf{I}_i = \mathbf{R}^{-1} M(\theta_i) \mathbf{R} \tilde{\mathbf{L}} \quad (17)$$

where \mathbf{R} rotates the direction \hat{d}_i to $(0,0,1)$ and \mathbf{R}^{-1} is the inverse rotation.

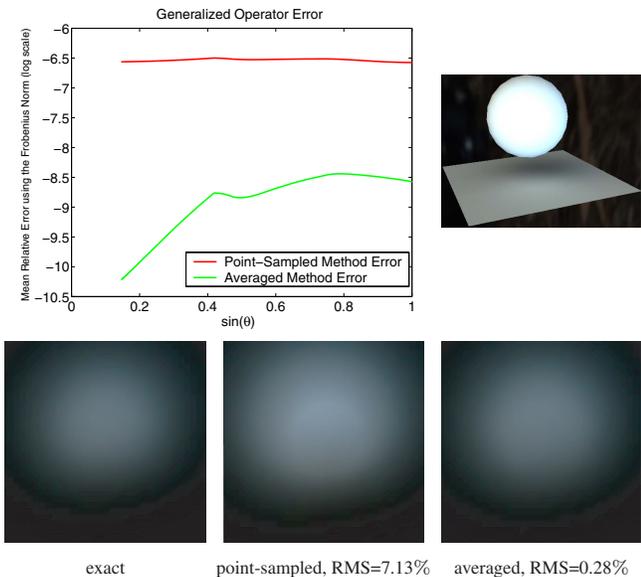


Figure 5. Indirect operator comparison. The graph shows operator error for the two non-exact operators as a function of $\sin \theta$. The bottom row shows image error for the test case shown in the top right (indirect lighting only as recorded on the ground plane).

Error Analysis We compare the point-sampled and averaged indirect radiance methods against the exact method. Error is measured in two ways. First, by considering each definition of I_i as a linear operator on the incident radiance L , we can compare operators using a Frobenius matrix norm on their difference with the exact operator. This is equivalent to averaging over all lighting environments, and produces the error curves in the top left of Figure 5. The rest of that figure shows an image comparison for a specific lighting environment, where errors are measured using relative squared error over the ground plane receiver.

The averaged operator provides an accurate approximation, while the point-sampled version incurs considerable error. The averaged operator can also be evaluated at very low cost using our simple polynomial approximation. On the other hand, the exact operator is much more expensive to evaluate, requiring two SH rotations and a complicated evaluation of the operator’s components.

6. Results

All renderings in the paper were performed on an Intel 2.4Ghz CoreDuo and Nvidia GeForce 8800GTX graphics card running DirectX9 and were computed at 1024×1024 resolution. Video results were computed at 512×512 resolution. A receiver buffer of 256×256 was used unless oth-

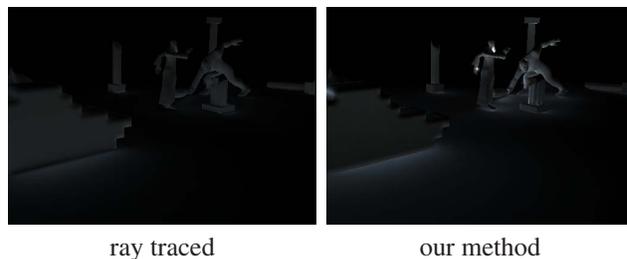


Figure 8. Indirect lighting comparison.

erwise noted. The figure on the first page shows the added realism produced by our method at real-time frame rates. Video results demonstrate real-time recordings on dynamic models.

Figure 6 compares our method with the vertex-based method of [12]. The left image shaded 60767 vertices whereas the right was shaded in image space using a receiver buffer of 256×256 . By shading in image space, our rendering is much smoother and better sampled while still attaining a 2x speedup.

Figure 7 compares upsampling of intermediate results using bilateral or bilinear interpolation with shading at display resolution. We can render without noticeable artifacts with bilateral upsampling by a factor of 4 (i.e., a 512×512 receiver buffer) whereas simple bilinear upsampling incurs artifacts at the silhouettes. At the higher upsampling rates of 64x there are artifacts which are much more severe with bilinear interpolation.

Figure 8 isolates the quality of our indirect lighting approximation. While our results are plausible, the primary sources of error are ignoring shadowing of lighting incident on the proxies and inaccuracy in the SH exponential in the presence of substantial blocker overlap. Nevertheless our approximation does reproduce realistic effects such as the color bleeding shown in Figure 9.

7. Conclusion

Soft GI effects can be efficiently handled by approximating geometry as a set of proxies and accumulating each proxy’s contribution in image space. Our method extends ambient occlusion splatting to higher frequencies in order to cast shadows that respond to lighting direction. It also accounts for inter-reflections as well as shadows. Finally, it obtains better performance and higher-quality sampling than previous vertex-based methods.

The basic method is quite general and can be extended to other GI effects. We are currently working on adaptive sampling methods based on image pyramids, handling spatially-varying albedo, and shadowed and multiple-bounce indirect radiance. Extension to non-diffuse (but not highly specular) receivers is straightforward by applying our low-frequency



vertex-based: 30fps
60767 vertices

our method: 63fps
256×256 receiver buffer

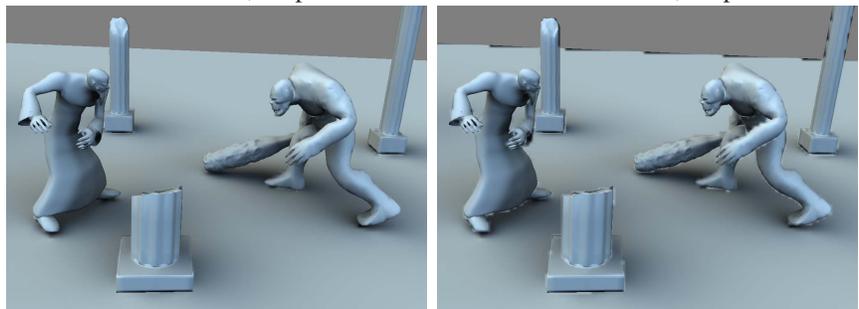
Figure 6. Vertex vs. pixel-based shading.



display resolution, 8fps

bilateral 16x, 66fps

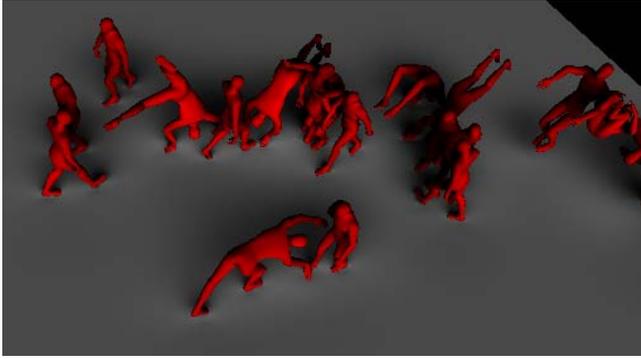
bilinear 16x, 88fps



bilateral 64x, 99fps

bilinear 64x, 171fps

Figure 7. Bilateral vs. bilinear upsampling.



without indirect, 55fps



with indirect, 41fps

Figure 9. Color bleeding with indirect accumulation.

incident radiance L_p to other BRDFs. We are also interested in gradient-based interpolation of soft GI intermediates like SH occlusion and indirect radiance. Finally, our method's ability to sample more finely reveals some artifacts from the self-shadowing replacement rules which could be improved.

References

- [1] M. Agarwala, R. Ramamoorthi, A. Heirich, and L. Moll. Efficient image-based methods for rendering soft shadows. In *Proc. of ACM SIGGRAPH*, pages 375–384, 2000.
- [2] O. Arikan, D. Forsyth, and J. O'Brien. Fast and detailed approximate global illumination by irradiance decomposition. *ACM Trans. Gr.*, 24(3):1108–1114, 2005.
- [3] U. Assarsson and T. Akenine-Möller. A geometry-based soft shadow algorithm using graphics hardware. *ACM Trans. Gr.*, 22(3):511–520, 2003.
- [4] M. Bunnell. Dynamic ambient occlusion and indirect lighting. In *GPU Gems 2: Programming Techniques for High-Performance Graphics and General-Purpose Computation*, pages 223–233. Addison-Wesley Professional, 2004.
- [5] C. Dachsbacher and M. Stamminger. Splatting indirect illumination. In *Symposium on Interactive 3D Graphics*, pages 93–100, 2006.
- [6] M. Hašan, F. Pellacini, and K. Bala. Direct-to-indirect transfer for cinematic relighting. *ACM Trans. Gr.*, 25(3):1089–1097, 2006.
- [7] K. Iwasaki, Y. Dobashi, N. Tamura, F. Yoshimoto, and T. Nishita. Precomputed radiance transfer for dynamic scenes with diffuse interreflection. *ACM SIGGRAPH Sketches*, 2006.
- [8] J. Kautz, J. Lehtinen, and T. Aila. Hemispherical rasterization for self-shadowing of dynamic objects. In *Proc. of Eurographics Symposium on Rendering*, pages 179–184, 2004.
- [9] J. Kontkanen and S. Laine. Ambient occlusion fields. In *Proc. of Symposium on Interactive 3D Graphics, SI3D*, pages 41–48, 2005.
- [10] M. Malmer, F. Malmer, U. Assarsson, and N. Holzschuch. Fast precomputed ambient occlusion for proximity shadows. Technical Report 5779, INRIA, 2005.
- [11] C. Mei, J. Shi, and F. Wu. Rendering with spherical radiance transport maps. *Eurographics (Computer Graphics Forum)*, 23(3):281–290, 2004.
- [12] Z. Ren, R. Wang, J. Snyder, K. Zhou, X. Liu, B. Sun, P. Sloan, H. Boa, Q. Peng, and B. Guo. Real-time soft shadows in dynamic scenes using spherical harmonic exponentiation. *ACM Trans. Gr.*, 25(3):977–986, 2006.
- [13] M. Segal, C. Korobkin, R. Van Widenfelt, J. Foran, and P. Haeberli. Fast shadows and lighting effects using texture mapping. In *Proc. of SIGGRAPH*, pages 249–252, 1992.
- [14] P. Shanmugam and O. Arikan. Hardware accelerated ambient occlusion techniques on GPUs. In *Proc. ACM Symposium on Interactive 3D Graphics and Games*, pages 73–80, 2007.
- [15] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Proceedings of International Conference on Computer Vision*, pages 839–846, 1998.
- [16] R. Wang, K. Zhou, J. Snyder, X. Liu, H. Bao, Q. Peng, and B. Guo. Variational sphere set approximation for solid objects. *Pacific Graphics (The Visual Computer)*, 22(9-11):612–621, 2006.
- [17] K. Zhou, Y. Hu, S. Lin, B. Guo, and H. Shum. Precomputed shadow fields for dynamic scenes. *ACM Trans. Gr.*, 24(3):1196–1201, 2005.