Eurographics Symposium on Rendering - Experimental Ideas & Implementations (2015)
J. Lehtinen and D. Nowrouzezahrai (Editors)

Practical Shading of Height Fields and Meshes using Spherical Harmonic Exponentiation

Aude Giraud and Dered Nowrouzezahrai

Université de Montréal

Abstract

Interactively computing smooth shading effects from environmental lighting, such as soft shadows and glossy reflections, is a challenge in scenes with dynamic objects. We present a method to efficiently approximate these effects in scenes comprising animating objects and dynamic height fields, additionally allowing interactive manipulation of view and lighting. Our method extends spherical harmonic (SH) exponentiation approaches to support environmental shadowing from both dynamic blockers and dynamic height field geometry. We also derive analytic expressions for the view-evaluated BRDF, directly in the log-SH space, in order to support diffuse-to-glossy shadowed reflections while avoiding expensive basis-space product operations. We illustrate interactive rendering results using a hybrid, multi-resolution screen- and object-space visibility-marching algorithm that decouples geometric complexity from shading complexity.

Categories and Subject Descriptors (according to ACM CCS): Computer Graphics [I.3.3]: Three-Dimensional Graphics and Realism—Display Algorithms

1. Introduction

Shadows provide important lighting and depth cues, especially in the presence of complex illumination. Computing shadows from large area and environmental light sources is a difficult and long-standing problem in image synthesis.

Precomputed radiance transfer (PRT) efficiently approximates shading and shadows from environmental light sources by representing incident light in a compact basis-space (e.g., spherical harmonics (SH)), and then shading with potentially spatially-varying BRDFs entirely in this space. These approaches, however, typically require static geometry in order to precompute the complex inter- and intra-object visibility relationships that capture the shadowing effects.

We combine two techniques for efficient visibility determination, one for dynamic blockers and another for dynamic height field geometry, in order to compute visibility *interactively* in a PRT-context for scenes with dynamically animated meshes and height fields. We extend a logarithmic SH formulation [RWS*06] to couple both the visibility *and* the view-evaluated BRDF, allowing us to interactively compute final shading completely in the log basis-space, unlike existing log-SH techniques. Shading is performed on the GPU using a hybrid image- and object-space marching approach.

2. Previous Work

Precomputed Radiance Transfer. PRT methods render soft shadows from large area and environmental light sources [SKS02]. The transfer function that combines the view-evaluated BRDF and visibility at each point of an object is projected onto a compact spherical basis, and the shading integral is evaluated as the inner-product of the basis-projected transfer and incident lighting. The precomputation involves a costly ray-tracing for static objects in order to resolve visibility relationships, forming a bottleneck that precludes the use of standard PRT in dynamic scenes.

Ren et al. [RWS*06] lift this constraint by approximating dynamic and static objects with spheres, and then using *spherical harmonic exponentiation* (SHExp) to replace expensive SH product accumulation of blocker visibilities with cheap additions in SH log-space. At each receiver point, they accumulate log-SH over a hierarchy of blocking spheres, and exponentiate to evaluate the final SH visibility. The spheres' log-SH visibility vectors are calculated individually from a simple tabulation. We similarly accumulate log-SH for spherical blockers, but we additionally derive analytic formulations for wedge-visibility accumulation in log-SH space for dynamic height fields (see Section 4).

Blocker Accumulation. We approximate dynamic objects with spheres and height fields, accumulating their visibility

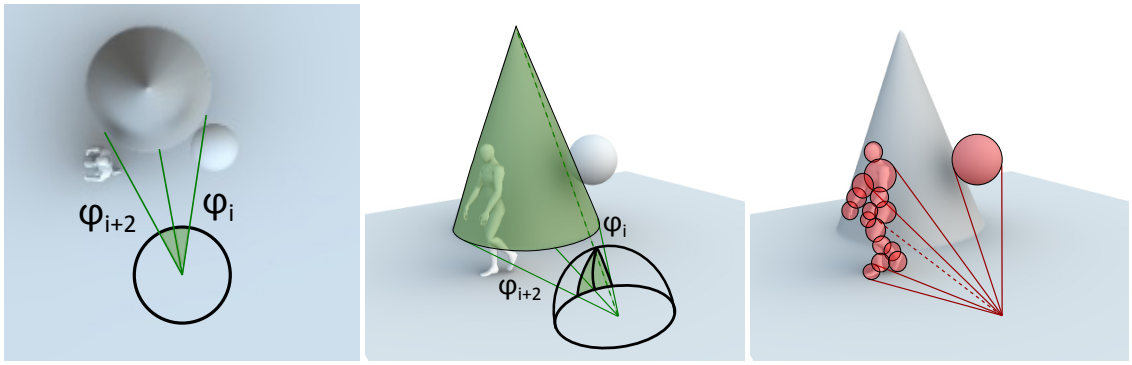


Figure 1: Accumulation of the visibility vectors of the blocker spheres and of the height field (consisting of the plane and the cone).

in log-SH space in order to compute the blocker visibility. Unlike the sphere fitting used by Ren et al. [WZS*06], we simply apply a k-means clustering in order to approximate our dynamic (non-height field) objects with spherical blockers. Conversely, Kautz et al. [KLA04] rasterize blocker geometry directly into hemispherical bitmaps at each receiver point, but their method is quickly limited by the scene complexity, which they alleviate to a certain degree using simplified blocker geometry during visibility accumulation. Zhou et al. [ZHL*05] precompute volumetric basis-space shadow fields for each rigid object in the wavelet and SH bases to render soft shadows, where the complete visibility across objects is combined using expensive basis-space products. As such, they are limited in the number of dynamic rigid objects, and they do not support deformations from, e.g., skinned mesh animations. For these reasons, we were motivated to build atop the approach of Ren et al.

Our approach does not require any costly precomputation, and we completely avoid triple-product shading by working almost exclusively in the log-SH domain: we are the first to explore BRDF representations in the log-SH domain, and our initial investigation yields some promising results (see Section 5).

We combine deformable objects with dynamic height fields. Many existing works have looked at the problem of computing shadows on height field geometry, starting with the seminal *horizon mapping* work [Max88, SC00] where azimuthal directions are discretized and object-space ray-marching is employed to precompute maximum blocking angles along these directions. At run-time, an azimuthal "slice" is queried depending on the light source position/direction, in order to determine whether the point is in (hard) shadow or not.

Snyder and Nowrouzezahrai [SN08] extend this approach to dynamic height fields and environmental lighting. Their method replaces naÃŕve ray-marching with a multi-resolution approach that takes non-uniform steps into pre-filtered geometric approximations of the height field. This reduces the number of required samples, but at the cost of biasing the resulting maximum blocking angle computation. They directly compute SH visibility vectors by composition across azimuthal wedges, where the per-wedge visibility is pre-tabulated. In 2009, the same authors extend their method to indirect illumination and an analytic formulation of per-wedge SH visibility [NS09]. We base ourselves on this technique and derive an analytic per-wedge visibility formulation *directly* in *log*-SH space (Section 4).

Timonen and Westerholm [TW10] use a GPU-based convex hull formulation to further accelerate the per-wedge blocking angle computation, completely avoiding any approximation bias. This allows them to more accurately represent visibility, avoiding low-frequency basis representation and capturing all-frequency shadows from dynamic environment maps. Timonen [Tim13] later applied this concept to the problem of screen-space ambient occlusion, treating the Z-buffer as a height field.

3. Overview and Terminology

We provide a quick overview of the mathematical concepts we will use throughout our paper.

Height fields are 3D surfaces defined by a height function h, where each 2D point on the plane of the height field (x,y) has height $z = h(x,y)$. Height fields are often represented by a texture, or height map, where each texel stores a height value. In general, these grey level values are multiplied by a coefficient to scale the heights. Height fields can represent macro-geometry, like mountains or oceans (as we do in our applications), but also micro-geometry like a skin's geometric texture.

Visibility is a spherical function at a shading point that determines, for each direction s on the unit sphere, whether an outgoing ray in that direction would be occluded or not. We define the visibility function v as:

$$v(\omega) = \begin{cases} 0 & \text{if the ray is occluded,} \\ 1 & \text{otherwise.} \end{cases} \quad (1)$$

In numerical log space, visibility v is expressed by

$$v^{log}(\omega) = \begin{cases} \log\varepsilon & \text{if the ray is occluded,} \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

Spherical Harmonics are an orthonormal, frequency-space basis used to represent spherical functions. They are useful for representing irradiance environment maps [RH01] or for computing soft shadows and glossy reflections [SKS02]. The SH projection coefficients of a spherical function f are defined as:

$$\boldsymbol{f}_l^m = \int_\Omega f(\omega)\, \boldsymbol{y}_l^m(\omega) d\omega \quad (3)$$

where $\boldsymbol{y}_l^m(\omega)$ is the m-th band-l SH basis function, evaluated in direction s on the unit sphere S. After projection, we can reconstruct an order-n band-limited approximation of f as:

$$\tilde{f}(\omega) = \sum_{l=0}^{n-1} \sum_{m=-l}^{l} \boldsymbol{f}_l^m\, \boldsymbol{y}_l^m(\omega) = \sum_{i=0}^{n^2} \boldsymbol{f}_i\, \boldsymbol{y}_i(\omega)\,, \quad (4)$$

where we often replace the (l,m)-double indices with a more compact single index $i = l(l+1)+m$.

Zonal Harmonics (ZH) are the $m=0$ circularly-symmetric (about the canonical $\boldsymbol{z}$ up-vector) subset of SH basis functions. An order-n ZH expansion requires only n projection coefficients, instead of n^2 for an SH expansion, however it can only represent a (band-limited) circularly-symmetric function aligned along $\boldsymbol{z}$. The SH projection coefficients of a circularly-symmetric aligned about an **arbitrary direction** d can, however, be obtained using the Funke-Hecke convolution theoreom [Slo08] as:

$$\boldsymbol{f}_l^m = \sqrt{\frac{4\pi}{2l+1}}\, \boldsymbol{z}_l\, \boldsymbol{y}_l^m(d)\,, \quad (5)$$

where $\boldsymbol{z}_l = \boldsymbol{y}_l^0$ are the ZH projection coefficients of the circularly-symmetric function aligned along the canonical z axis.

SH Product Operators can be defined in order to compute the (approximate) SH projection coefficients of the product of two SH-projected functions. This operation is useful for shading, where we require the product of the visibility, the view-evaluated BRDF, and the lighting prior to spherical integration. SH products can also be used to accumulate the spherical visibility across blockers, as in e.g., [ZHL*05]. The SH coefficients of the product of two functions $f(s)$ and $g(s)$ (with projection coefficient vectors $\boldsymbol{f}$ and $\boldsymbol{g}$) can be expressed as:

$$(\boldsymbol{f} * \boldsymbol{g})_i = \sum_{j=0}^{n^2} \sum_{k=0}^{n^2} \Gamma_{ijk}\, \boldsymbol{f}_j\, \boldsymbol{g}_k \quad \text{with } \Gamma_{ijk} = \int_\Omega \boldsymbol{y}_i\, \boldsymbol{y}_j\, \boldsymbol{y}_k\, d\omega, \quad (6)$$

where Γ is the SH tripling coefficient tensor [Slo08], and we temporarily omit parameters for brevity. Computing Equation 6 has $O(n^{5/2})$ time complexity in SH [NRH04], which becomes prohibitively slow in the case of many blockers (i.e., many product operations).

SH Logarithm and Exponentiation are used to approximate SH products. Ren et al. [RWS*06] compute numerically the log-ZH projection coefficients of a canonically oriented spherical blocker. In log-space, we must take care to treat the undefined behavior of the log operator for values below zero; Sloan et al. [SGNS07] simply clamp their functions to a small (positive) value ε prior to taking the log. Ren et al. improve upon this naive method by clipping the eigenvalues of the product matrix [RWS*06]. Another way to handle negative values is to use the log-modulus transformation [JD80], but this is used primarily in statistics for data visualization. We use the naive method, and explore the effect of changing ε on the accuracy of the log-SH representation, particularly in the context of our new analytic log-SH BRDF formulation (see Section 5). Concretely, the log-SH projection coefficients of a spherical function $f(s)$ are defined as

$$\boldsymbol{f}_i^{log} = \int_\Omega \log\left(\max\left(f(\omega), \varepsilon\right)\right)\, \boldsymbol{y}_i(\omega)\, d\omega\,. \quad (7)$$

SH Blocker Accumulation combines the visibility function across all blockers at a shading point p. We decompose our scenes into a set of m spherical blockers, approximating deformable objects, and a dynamic height fields (see Figure 1). Here, the visibility can be composed as a sequence of (SH) product operations over of the m blockers' SH visibility vectors $\boldsymbol{v}[i]$ and the height field's SH visibility vector $\boldsymbol{v}[\mathrm{HF}]$ as:

$$\boldsymbol{v} = \boldsymbol{v}[1] * \boldsymbol{v}[2] * \ldots * \boldsymbol{v}[m] * \boldsymbol{v}[\mathrm{HF}]\,. \quad (8)$$

In log-SH, however, these product operations are replaced with cheaper additions [RWS*06] of the log-SH visibility vectors, after which we take an SH exponentiation to obtain an approximation of the final accumulated visibility, as expressed in Equation 9. We note exp_* the exponential operator applied to SH vectors as in [RWS*06]. We use the hybrid variant, called HYB in [RWS*06].

$$\boldsymbol{v} \approx \exp_*(\boldsymbol{v}^{log}[1] + \ldots + \boldsymbol{v}^{log}[m] + \boldsymbol{v}^{log}[\mathrm{HF}])\,. \quad (9)$$

4. Visibility

We combine the effects of visibility from dynamic height fields and blockers. Here, we will describe how we perform this accumulation entirely in log-SH space by building atop the approaches of Ren et al. [RWS*06] and Nowrouzezahrai and Snyder [NS09]. We derive analytic expressions for per-wedge log-SH visibility, as well as devise a new ratio clamping heuristic that reduces ringing artifacts that arise from accumulated numerical imprecisions in log-SH representation.

4.1. Height Field Log-SH Visibility

We build atop the multi-resolution ray-marching approach of Nowrouzezahrai and Snyder [NS09] to compute log-SH visibility for points on a dynamic height field. Consider point $p = (p_x, p_y, p_z)$ on the height field, we define the *blocking angle* between p and any other point on the height field p' as:

$$\theta_b(p,p') = \arctan\left(\frac{p'_z - p_z}{\sqrt{(p'_x - p_x)^2 + (p'_y - p_y)^2}}\right). \quad (10)$$

This angle is measured from the horizon plane to the North pole, representing the portion of the sky hidden by the height field in a given direction (i.e., from p to p'). The visibility is zero below this elevation angle and one above it. It is convenient to adopt Max's [Max88] horizon angle definition:

$$\theta_h(p,p') = \pi/2 - \theta_e(p,p'), \quad (11)$$

where horizon angle θ_h is directly related to the *visible* portion of the great-circle (see Figure 2).

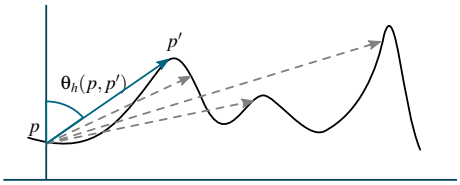


Figure 2: Horizon angle θ_h for point p on the height field.

We can use a compact representation of the elevation function to define a slice of the height field's visibility, as in [NS09]. We use the orthonormal Normalized Legendre Polynomial (NLP) basis to project the log-SH visibility of a single azimuthal slice. Zonal harmonics are related to NLP basis function, denoted $\hat{P}_l$, *via* the following expression:

$$\mathbf{y}_l^0(\theta,\phi) = \sqrt{\frac{1}{2\pi}}\,\hat{P}_l(\cos\theta). \quad (12)$$

The NLP functions are given by Rodrigues' formula:

$$\hat{P}_l(z) = \sqrt{\frac{2l+1}{2}}\frac{1}{2^l\,l!}\frac{d^l}{dz^l}\left[(z^2-1)^l\right], \quad (13)$$

where $z = \cos\theta$.

For order 4, they can be analytically expressed by:

$$\hat{\boldsymbol{P}}(z) = \left[\sqrt{\frac{1}{2}}, \frac{\sqrt{3}z}{\sqrt{2}}, \frac{\sqrt{5}\sqrt{2}(3z^2-1)}{4}, \frac{\sqrt{7}\sqrt{2}(5z^3-3z)}{4}\right]. \quad (14)$$

The visibility vector for a slice with a horizon angle σ, projected in the NLP basis, is then given by:

$$\boldsymbol{v}^{log}(\sigma) = \int_{\pi/2-\sigma}^{\pi} \log\varepsilon\,\hat{\boldsymbol{P}}(\cos\theta)\,\sin\theta\,d\theta. \quad (15)$$

At order 4, their expression is the following:

$$\boldsymbol{v}^{log}(\sigma) = \left[\log\varepsilon\frac{\sin\sigma+1}{\sqrt{2}}, \log\varepsilon\frac{-3\cos^2\sigma}{2\sqrt{6}}, \log\varepsilon\frac{-5\sin\sigma\cos^2\sigma}{2\sqrt{10}}, \log\varepsilon\frac{7\cos^2\sigma(-4+5\cos^2\sigma)}{8\sqrt{14}}\right]. \quad (16)$$

To reconstruct the continuous function of the visibility, one has to do a dot product of the visibility vector and the basis vector, as in Equation 17.

$$v^{log}(\theta) = \boldsymbol{v}^{log}\cdot\hat{\boldsymbol{P}}(\cos\theta). \quad (17)$$

We compose log-SH visibility due to the height field by summing over the log-SH wedge visibility, similarly to [NS09] albeit with a few important modifications, discussed below. First, we must not only compute the height field's self-shadows, but also its shadows projected onto the other objects in the scene.

Height Field Self-shadowing. For the height field self-shadowing, we sample the (log) visibility on a discrete set of azimuthal directions, as in [NS09]. At each height field shading point, and for each azimuthal direction, we use a multi-resolution pyramid marching scheme [SN08] to march along the azimuthal direction and sample the (filtered) height field in order to approximate θ_h for each discrete φ. We linearly interpolate the visibility between these $n-1$ discrete azimuthal *wedges*: the first wedge is aligned canonically at $\varphi = 0$, and each successive wedge is rotated by $k\triangle\varphi$, with $\triangle\varphi = 2\pi/W$, with W the number of wedges and $k \in [0, W-1]$.

In the primal SH domain, wedge visibilities could be added together in order to obtain the final SH height field visibility [SN08], since visibility value outside a wedge is (implicitly) zero, as illustrated in Figure 3. In the log-SH domain, however, summation equates to multiplication in the primal domain (after exponentiation), and so we have to rethink the wedge accumulation strategy.

Specifically, we now treat the (log-)visibility as 1 outside the wedge; it now has effectively an implicit value of $\log(1) = 0$ in log-SH space. Thus, when working in log-SH space we can apply the same tactic as used in [NS09]. We construct a *canonical* log-SH wedge function from the two log-ZH azimuthal values on the borders of the wedge, using a composition operator that explicitly sets all the visibility outside the wedge to a value of 0 (see Figure 3). Assuming that $V_i(\theta,\phi)$ is the visibility function of the i-th wedge, Equation 18 gives its projection in log-SH space.

$$\boldsymbol{V}_i^{log} = \int_0^{\Delta\varphi}\int_0^{\pi} V_i^{log}(\theta,\phi)\,\mathbf{y}_i(\theta,\phi)\,\sin\theta\,d\theta\,d\phi, \quad (18)$$

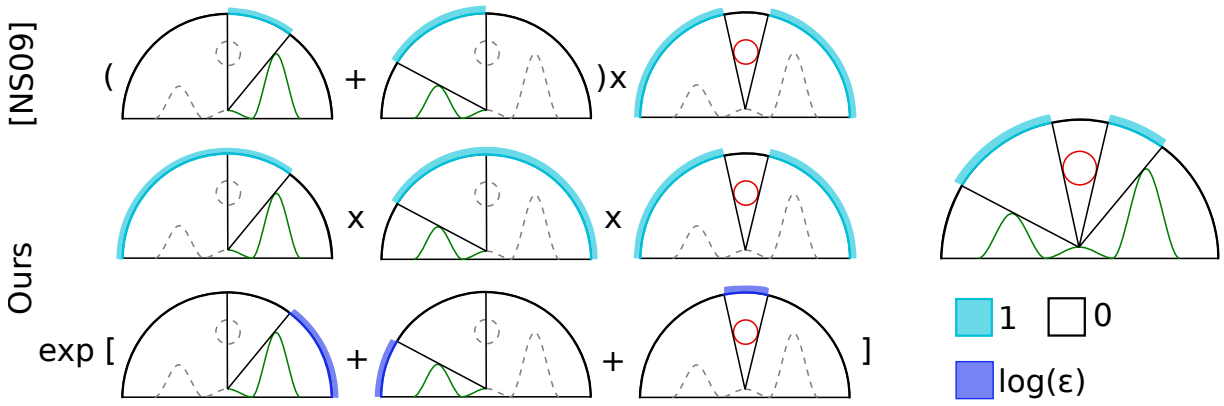


Figure 3: Combining visibilities. Top: The visibility wedges are added before we multiply them with the blockers' visibility vectors. Middle: Adding in log-SH space is equivalent to multiplying in SH space. Bottom: When using spherical harmonics exponentiation, we add the visibility vectors in log-SH space; the numerical values are inverted as $\log(1) = 0$, and we use a factor ε because $\log(0)$ is undefined. Right: The resulting visibility.

where V_i^{log} results from the linear interpolation of the slices' functions in the directions φ_i and φ_{i+1}:

$$V_i^{log}(\theta,\phi) = \left(\left(1-\frac{\phi-\varphi_i}{\Delta\varphi}\right)\boldsymbol{v}_i^{log} + \left(\frac{\phi-\varphi_i}{\Delta\varphi}\right)\boldsymbol{v}_{i+1}^{log}\right)\cdot\hat{\boldsymbol{P}}(\cos\theta)\,. \quad (19)$$

When substituting Equation 19 into Equation 18, we see that the integral can be expressed by a product between a matrix of linear interpolation, and the NLP log-SH vectors of the visibility of the discrete border directions of the wedge:

$$\boldsymbol{V}_i^{log} = \mathbb{M}_{lin}\left[\boldsymbol{v}_i^{log}\ \boldsymbol{v}_{i+1}^{log}\right]^{\mathrm{T}}, \quad (20)$$

where $\mathbb{M}_{lin}$ coefficients are calculated with

$$[\mathbb{M}_{lin}]_{i,j} = \int_0^{\Delta\phi}\int_0^{\pi}[\mathbb{I}_{lin}]_j\,\hat{\boldsymbol{P}}_j(\cos\theta)\,\boldsymbol{y}_i(\theta,\phi)\,\sin\theta\,d\theta\,d\phi\,, \quad (21)$$

and

$$\alpha_i = \frac{\phi-\varphi_i}{\Delta\varphi} \quad (22)$$

$$\mathbb{I}_{lin} = [1-\alpha_i, 1-\alpha_i, 1-\alpha_i, 1-\alpha_i, \alpha_i, \alpha_i, \alpha_i, \alpha_i]\,. \quad (23)$$

We precompute the wedge interpolation and composition operator $\mathbb{M}_{lin}$ which, for e.g., an order-4 SH expansion has dimensions $(4^2)\times(2\times4) = 16\times8$. After applying this operator to the two wedge boundary coefficients (Equation 21), we align the canonical log-SH wedge to its actual starting azimuthal coordinate using a fast z-rotation, as in [SN08,NS09]. We finally add the rotated wedges to get the complete visibility function, as in [NS09].

Height Field Cast Shadows. To compute the shadow of the height field onto another (non-height field) object's point, we use a similar approach: we first project the shading point of the object onto the plane of the height field, and we offset our maximum blocking angle calculation using the height of the point above the height field. We march along each azimuthal direction as before, accumulating and compositing log-ZH visibility slices, as illustrated in Figure 4.

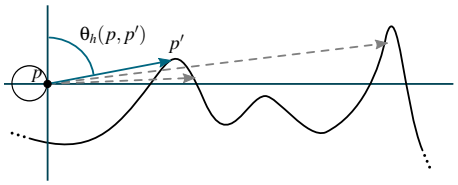


Figure 4: Horizon angle θ_h for one receiver point p of a blocker.

4.2. Dynamic Blocker Shadows

When treating spherical blockers, the log-SH visibility for a single blocker can be determined analytically: consider a spherical blocker with center c and radius r. If we first assume that c is aligned directly above our shading point p, the horizon angle formed by the projection of the sphere onto the point is:

$$\theta(p,c,r) = \arcsin\left(r/\|c-p\|\right). \quad (24)$$

The log-ZH of this canonical blocker visibility is simply:

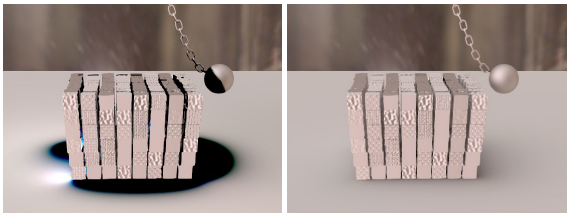
$$\boldsymbol{V}_l = \int_{\phi=0}^{2\pi}\int_{\theta=\theta(p,c,r)}^{\pi}\boldsymbol{y}_l^0(\theta)\,\sin\theta\,d\theta\,d\phi\,, \quad (25)$$

which we can quickly rotate to align along the *actual* direction between p and c using the Funke-Hecke theorem (see Section 3). We can similarly compute the log-ZH blocker visibility analytically as:

$$\boldsymbol{V}_l^{log} = \int_{\phi=0}^{2\pi}\int_{\theta=0}^{\theta(p,c,r)}\log\varepsilon\,\boldsymbol{y}_l^0(\theta)\,\sin\theta\,d\theta\,d\phi\,, \quad (26)$$

and repeat the rotation to obtain the log-SH visibility, which we will accumulate (with a sum) across all spherical blockers (and the accumulated height field log-SH visibility). All dynamics objects (except the height field) are approximated with a set of spheres [RWS*06], and we use a simple k-means clustering to place the sphere blockers and determine their radii. Equation 26 (and Equation 25, for that matter) has compact analytic forms that we hardcode into our shaders. Figure 1 shows how we accumulate the visibilities at a point p.

A common problem with log accumulation is that, given many blockers, numerical accumulation errors are compounded as the occluded regions of blockers overlap. For example, for a given direction s occluded by m blockers, the threshold $\log\varepsilon$ will be accumulated m times yielding an effective value of $\log(\varepsilon^m)$. Since $\varepsilon < 1$, $\log(\varepsilon^m)$ approaches infinity and the error (after exponentiation) can become unbounded. To mitigate this problem, we smoothly attenuate the influence of distant blockers to the shading point during log-SH accumulation. To avoid visible transitions, we multiply the log-SH vector of each sphere's visibility (Equation 26) by the ratio $r/\|c-p\|$. The farther and smaller a blocker, the lower its influence. This approach has limits, e.g., when a small sphere close to p is aligned with a big sphere far from p, but we found it gives visually pleasing results in most of the cases. Figure 5 illustrates the impact of our attenuation approach for a scene with 402 blockers. Moreover, for a shading point p with a normal $\boldsymbol{n}$, we only consider blockers b in direction $\boldsymbol{d}_b$ (from p) if $\boldsymbol{n}\cdot\boldsymbol{d}_b > 0$.



(a) Without ratio attenuation (b) With ratio attenuation

Figure 5: Use of a ratio criterion to decide if a blocker is considered or not.

5. BRDF

Log-SH Diffuse. Ren et al. [RWS*06] accumulate log-SH visibility, exponentiation, and then perform an expensive *triple-product* basis-space shading operation with the view-evaluated BRDF and the light source SH coefficients. This amounts to the application of an SH product and double-product integration.

We instead compute log-SH BRDF coefficients, accumulate them with the (accumulated) log-SH visibility, and *exponentiate*, prior to double-product integration with the lighting coefficients. This allows us to completely avoid SH products, and reduces shading to a simpler double-product integration. For a diffuse BRDF, we first can again compute canonical log-ZH coefficients as:

$$\boldsymbol{f}_i^{log} = \int_{\phi=0}^{2\pi}\int_{\theta=0}^{\pi} \log(\max(\cos\theta/\pi,\varepsilon)\,\boldsymbol{y}_l^0(\theta)\,\sin\theta\,d\theta\,d\phi\,.$$

We then rotate this vector in the direction of the shading point's normal.

Log-SH Phong. For a glossy Phong BRDF we again project the canonical log-ZH lobe as:

$$\boldsymbol{f}_i^{log} = \tag{27}$$
$$\int_{\phi=0}^{2\pi}\int_{\theta=0}^{\pi/2} \log\left(\max\left(\frac{r+1}{2\pi}\cos\theta^r,\varepsilon\right)\right)\boldsymbol{y}_l^0(\theta)\sin\theta\,d\theta\,d\phi$$
$$+\int_{\phi=0}^{2\pi}\int_{\theta=\pi/2}^{\pi} \log\varepsilon\;\boldsymbol{y}_l^0(\theta)\sin\theta\,d\theta\,d\phi\;,$$

where we have to take care to split the integral in case r is even, in order for the hemispherical clamping to $\log\varepsilon$ to hold. For a Phong term, we simply rotate (using Funke-Hecke again) the log-ZH lobe to align with the reflected view-vector.

ε Parameter Setting. We choose ε experimentally: for different values of ε, we compute the ground-truth ZH BRDF vector, as well as the exponentiated log-ZH BRDF vector, and we compute their L_2 error (see Figure 6). The closer ε is to one, the larger the error. For example, consider the diffuse BRDF, $\cos\theta/\pi < 1$, which means that when $\varepsilon = 0.9$, $\max(\cos(\theta)/\pi,\varepsilon) = \varepsilon$ most of the times. Thus, $\log\varepsilon$ is close to 0, and its exponentiation is close to 1. The same reasoning holds for our glossy BRDF.

Figure 7 shows a specular sphere with different exponents. When the exponent is high, we can see that the specular highlight on the sphere is not very strong. This is due to our choice of order 4, as low order SH cannot capture high frequency information [Slo08].

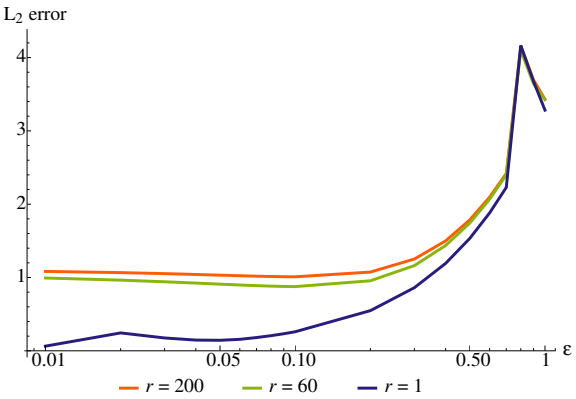


Figure 6: L_2 error between the SH BRDFs and the exponentiated log-SH BRDFs for various values of r and ε.

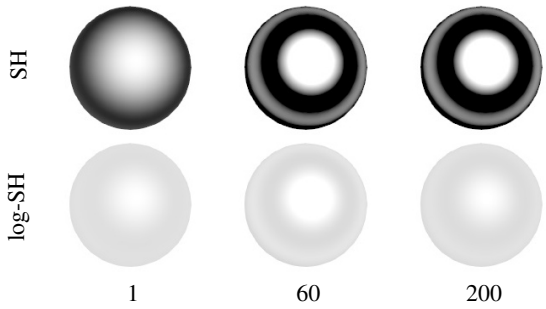


Figure 7: Spheres lit by a directional light with varying r. Our log-SH BRDF is a coarse approximation of the SH BRDF, but avoids a costly triple-product integration. log-SH uses $\varepsilon = 0.2$.

6. Implementation and Results

We generate G-buffers of our scenes in Blender for the positions, the normals, the UV coordinates and the type of each object (height field, water, diffuse or specular surfaces). We also use Mathematica to approximate our general meshes with spherical blockers. If a mesh is animated, we create a list of blockers for each separate frame and record the centers' positions and radii in a file. For the light, we pre-generate SH vectors representing the environment maps using Monte Carlo sampling.

We load the G-buffers, the height map, the lights' SH vectors, and a buffer with the blockers' informations on the GPU. We hardcode a table of pre-generated log-ZH coefficients for the BRDFs, that are rotated along the normal or the reflection vector depending on the type of material. We compute the shading interactively in a pixel shader. At each pixel, we determine the type of object. The material ID is used to choose which log-SH vector to use for the BRDF. We apply the ray-marching algorithm on the heightmap with or without an offset depending on whether the pixel is on the height field or a blocker. This gives us the log-SH visibility vector resulting from the height field geometry. We then process the list of blockers and compute their visible angles and directions on the fly. We rotate them and add their log-SH visibility vectors together. We add the height field's visibility log-SH vector, and the complete visibility is summed with the BRDF log-SH vector. We finally exponentiate the resulting log-SH visibility vector summed with the log-SH BRDF vector and do a dot product with the light's SH vector to get the final shading. The lights' SH vectors can be rotated according to the user input, who can manipulate the light interactively.

We render our scenes on a NVIDIA GTX480 card, at the resolution of 960×540 pixels. The whale in Figure 8 is approximated by 50 spheres and runs at 10fps. The reflection is computed using ray-tracing in Blender to pre-generate a special G-buffer for the water surface, that gives the positions, normals, and object's ID of the reflected point. The method is then applied on the water surface as if it were the reflected point, for example if sand is reflected, then we compute the shading as if we were on the sand's surface. The wrecking-ball scene in Figure 9 has 402 blockers and runs at about 15fps. The ball and half the cubes are specular. We retrieve the view direction from the moving camera using a python script in Blender and use this information to rotate the specular BRDF log-SH vector. Finally, the scenes with the walking man do not use G-buffers, and run at 68fps.

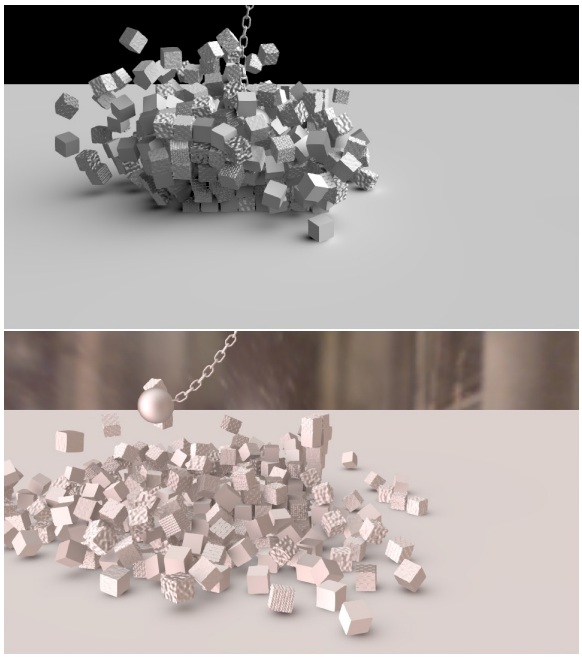


Figure 9: A glossy wrecking ball crashing into a block of diffuse and glossy cubes.

7. Conclusion and Future Work

Our method provides a unified framework to combine low-frequency visibility from dynamic objects and a dynamic height field. We derive a model to represent diffuse and specular BRDFs in the log-SH space, as well as height field's visibility. We combine those log-SH vectors with the accumulations of the spherical blockers' log-SH vectors and use the SH exponentiation to reduce the number of SH products. Our method can be applied as a post-processing step, and runs interactively.

In future work, we could add indirect illumination. Guerrero et al. use the spherical blockers as secondary light sources to compute real-time indirect illumination [GJW08], and we could combine this approach with indirect illumination for height fields [NS09]. These methods work only for diffuse surfaces, and for specular BRDFs we could adapt the approach of Pan et al. [PWXLPB07]. We could extend the work of [RWS*06] to find an optimal value for ε in a systematic way. The use of a hierarchy of blockers

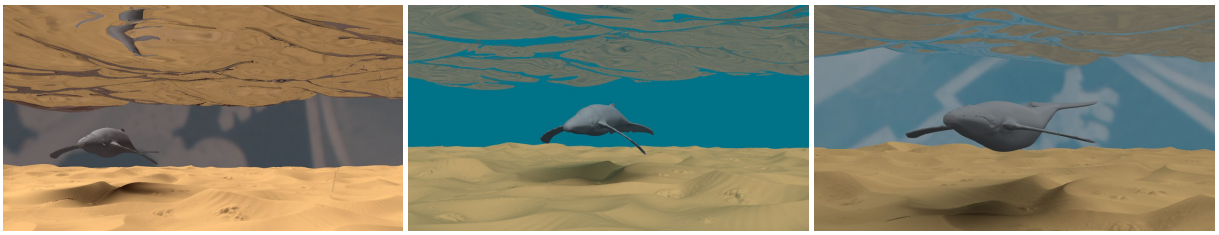


Figure 8: A whale under water with self- and cast-shadows.

as in [RWS*06] would probably speed up the rendering as fewer blockers would be considered, and it would reduce the artifacts of SH ringing without the need for a ratio criterion. Another approach would be to drop the blockers' approximation and use a semi-analytic visibility function for general meshes [NBMJ13].

Acknowledgements

Derek Nowrouzezahrai acknowledges funding from the Natural Sciences and Engineering Research Council of Canada's Discovery program, as well as the MITACS Accelerate program. We thank Maurice Ko, Stephane Morichere-Matte, David Lucas and members of the team at Microsoft's Game Studios in Vancouver for helpful discussion during the preliminary phases of the project. We also thank Bruce Luo and BlenderGuru.com for providing the wrecking ball scene data, users *rolmo* and *der_On* on BlendSwap for the sand dunes mesh and the whale animation sequence. Finally, we appreciate the many thorough comments and suggestions provided by the reviewers.

References

[GJW08] GUERRERO P., JESCHKE S., WIMMER M.: Real-time indirect illumination and soft shadows in dynamic scenes using spherical lights. *Computer Graphics Forum 27*, 8 (2008), 2154–2168. 7

[JD80] JOHN J. A., DRAPER N. R.: An alternative family of transformations. *Journal of the Royal Statistical Society. Series C (Applied Statistics)* (1980), 190–197. 3

[KLA04] KAUTZ J., LEHTINEN J., AILA T.: Hemispherical rasterization for self-shadowing of dynamic objects. In *Proceedings of the Fifteenth Eurographics Conference on Rendering Techniques* (2004), EGSR'04, Eurographics Association, pp. 179–184. 2

[Max88] MAX N. L.: Horizon mapping: shadows for bump-mapped surfaces. *The Visual Computer 4*, 2 (1988), 109–117. 2, 4

[NBMJ13] NOWROUZEZAHRAI D., BARAN I., MITCHELL K., JAROSZ W.: Visibility silhouettes for semi-analytic spherical integration. *Computer Graphics Forum* (2013). 8

[NRH04] NG R., RAMAMOORTHI R., HANRAHAN P.: Triple product wavelet integrals for all-frequency relighting. In *ACM SIGGRAPH 2004 Papers* (2004), SIGGRAPH '04, ACM, pp. 477–487. 3

[NS09] NOWROUZEZAHRAI D., SNYDER J.: Fast global illumination on dynamic height fields. In *Proceedings of the Twentieth Eurographics conference on Rendering* (2009), EGSR'09, Eurographics Association, pp. 1131–1139. 2, 3, 4, 5, 7

[PWXLPB07] PAN M., WANG XINGUO LIU R., PENG Q., BAO H.: Precomputed radiance transfer field for rendering interreflections in dynamic scenes. In *Computer Graphics Forum* (2007), vol. 26, Wiley Online Library, pp. 485–493. 7

[RH01] RAMAMOORTHI R., HANRAHAN P.: An efficient representation for irradiance environment maps. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (2001), SIGGRAPH '01, ACM, pp. 497–500. 3

[RWS*06] REN Z., WANG R., SNYDER J., ZHOU K., LIU X., SUN B., PIKE SLOAN P., BAO H., PENG Q., GUO B.: Real-time soft shadows in dynamic scenes using spherical harmonic exponentiation. *ACM Trans. Graph 25* (2006), 977–986. 1, 3, 6, 7

[SC00] SLOAN P.-P. J., COHEN M. F.: Interactive horizon mapping. In *Proceedings of the Eurographics Workshop on Rendering Techniques 2000* (2000), Springer-Verlag, pp. 281–286. 2

[SGNS07] SLOAN P.-P., GOVINDARAJU N. K., NOWROUZEZAHRAI D., SNYDER J.: Image-based proxy accumulation for real-time soft global illumination. In *Proceedings of the 15th Pacific Conference on Computer Graphics and Applications* (2007), PG '07, IEEE Computer Society, pp. 97–105. 3

[SKS02] SLOAN P.-P., KAUTZ J., SNYDER J.: Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. In *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques* (2002), SIGGRAPH '02, ACM, pp. 527–536. 1, 3

[Slo08] SLOAN P.-P.: Stupid spherical harmonics (SH) tricks, 2008. 3, 6

[SN08] SNYDER J., NOWROUZEZAHRAI D.: Fast soft self-shadowing on dynamic height fields. In *Proceedings of the Nineteenth Eurographics conference on Rendering* (2008), EGSR'08, Eurographics Association, pp. 1275–1283. 2, 4, 5

[Tim13] TIMONEN V.: Line-Sweep Ambient Obscurance. *Computer Graphics Forum (Proceedings of EGSR 2013) 32*, 4 (2013), 97–105. 2

[TW10] TIMONEN V., WESTERHOLM J.: Scalable Height Field Self-Shadowing. *Computer Graphics Forum (Proceedings of Eurographics 2010) 29*, 2 (2010), 723–731. 2

[WZS*06] WANG R., ZHOU K., SNYDER J., LIU X., BAO H., PENG Q., GUO B.: Variational sphere set approximation for solid objects. *Vis. Comput. 22*, 9 (2006), 612–621. 2

[ZHL*05] ZHOU K., HU Y., LIN S., GUO B., SHUM H.-Y.: Precomputed shadow fields for dynamic scenes. *ACM Trans. Graph. 24*, 3 (2005), 1196–1201. 2, 3