Affine Double- and Triple-Product Wavelet Integrals for Rendering

BO SUN
Columbia University
and
RAVI RAMAMOORTHI
University of California, Berkeley

Many problems in computer graphics involve integrations of products of functions. Double- and triple-product integrals are commonly used in applications such as all-frequency relighting or importance sampling, but are limited to distant illumination. In contrast, near-field lighting from planar area lights involves an affine transform of the source radiance at different points in space. Our main contribution is a novel affine double- and triple-product integral theory; this generalization enables one of the product functions to be scaled and translated. We study the computational complexity in a number of bases, with particular attention to the common Haar wavelets. We show that while simple analytic formulae are not easily available, there is considerable sparsity that can be exploited computationally. We demonstrate a practical application to compute near-field lighting from planar area sources, that can be easily combined with most relighting algorithms. We also demonstrate initial results for wavelet importance sampling with near-field area lights, and image processing directly in the wavelet domain.

1. INTRODUCTION

Integrations of products of functions are common in computer graphics and applied mathematics. For example, the reflection equation can be viewed as either a triple-product integral [Ng et al. 2004] consisting of three factors: the lighting, BRDF, and visibility, or a double-product integral where the BRDF and visibility are combined into the light transport function. A common assumption is that the illumination is distant, and each factor is represented in basis functions such as spherical harmonics or wavelets.

Local area lights have long been used as not only a practical modeling tool, but also an indispensable artistic device to set up a mood for a scene, for example, comfortable couches in a living room illuminated by a ceiling lamp (Figure 1(a)), or stretch chairs illumi-

nated by the light from a circular light source (Figure 1(b)). Standard double- and triple-product integrals, however, are not suitable for near-field relighting. The basic problem is that the incident lighting varies across the scene, and needs to be evaluated at all points in space. With proper parameterization, however, light propagation from planar area sources can be formulated as an affine transformation of the original source radiance, as shown in Figure 1(c). We call the transformed integrals affine double- or triple-product integrals (Section 3).

We present the first theoretical and computational analysis of affine double- and triple-product integrals in computer graphics. For actual computations, we focus our analysis on Haar wavelets, which have gained considerable attention in relighting. Haar wavelets are simple, and superior in compactly representing all-frequency effects
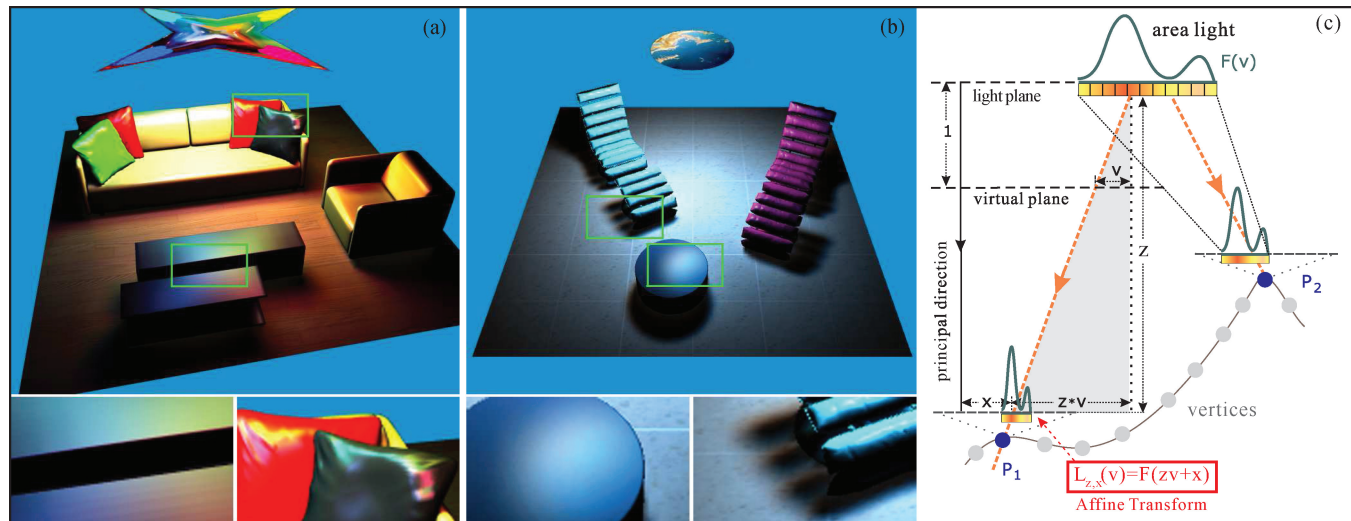
Fig. 1. (a) and (b): Our method enables relighting of scenes lit with near-field illumination. A planar area source can be moved, retextured, and reshaped at real-time rates. Important effects like spatially varying shading on the floors in 1(a) and 1(b), soft shadows under the chairs in the close-up of 1(b), and colored specularities on the cushions and tables in the close-ups of 1(a) and 1(b) are clearly visible. These effects are difficult to capture using only distant lighting, as shown in Figure 8(c). (c): Area lighting can be formulated as an affine transform. For simplicity, we parameterize the light field using a spatial coordinate x and an angular direction, given by the intercept v on a virtual plane, as shown in the diagram. The original area light is denoted as $F(v)$, and the vertex's incident radiance as $L_{z,x}(v)$, where z and x, respectively, are the vertical and horizontal coordinates of the vertex P_1. The intensity at P_1, $L_{z,x}(v)$, is then given by an affine transform ($F(zv + x)$) from simple trigonometry. Our main contribution is a novel theory of affine double and triple product wavelet integrals that enables near-field area lighting to be used in almost any precomputation-based relighting framework.

such as natural lighting, specular BRDFs, and intricate shadowing, often using only $1 \sim 2\%$ of coefficients [Ng et al. 2003]. Our analysis approach can also be extended to non-Haar wavelets, as discussed briefly in Section 7. Fundamentally, the problem we are trying to solve is to find an efficient representation for wavelets that are affinely transformed (scaled and translated). Wavelets, however, are known for their lack of even translation invariance [Strang 1989]. For example, simply translating a Haar wavelet basis function one pixel to the left would change its coefficients dramatically, causing its power to spread across many different sub-bands.

Our main technical contribution is a novel affine double- and triple-product integral theory for Haar wavelets, which is presented in Section 4. The theory is developed primarily on 1D signals; since 2D and higher dimensional wavelets are simply products of 1D basis functions, a direct extension to higher dimensions is possible (Section 4.4). Note that we focus on 1D affine transforms, namely, scaling and translations, and therefore do not consider general rotations and shears in 2D or 3D. As seen in our practical applications, this provides a solid basis for a more general wavelet framework for many rendering problems. In Section 7, we also discuss extensions to non-Haar wavelets and nonlinear transformations.

The standard theory of double- and triple-product integrals is expressed in terms of standard coupling and tripling coefficients, respectively. Our theory studies the affine analogs, that must now account not only for the different basis functions being coupled or integrated, but also the scale and translation in the affine transform. Affine coupling and tripling coefficients therefore gain two more degrees of freedom and are, respectively, 4D and 5D functions for 1D signals. In Section 4.2, we show that these coefficients can be boiled down to an intuitive 2D analytic core function, which we call the canonical coupling coefficient M. The canonical coupling coefficient exposes the inherent sparsity of the affine transform, which

can be exploited to develop efficient computational methods. This is analogous to how standard tripling coefficients are theoretically complex, but actually sparse in Haar wavelets [Ng et al. 2004].

Our theoretical development enables fast practical algorithms for affine transforms in Haar wavelets. This overturns a commonly held view that operations like shifts or scales are difficult in the wavelet domain. One practical application of our theory is relighting, and we take a significant step towards generalizing wavelet-based relighting methods to near-field settings with planar area light sources (Section 5; Figures 1, 8, and 9). There are also applications to a variety of other problems that depend on wavelet representations. Section 6 describes initial solutions for wavelet importance sampling [Clarberg et al. 2005] with near-field area lights, and image processing (dilation and translation) directly in the wavelet domain. Section 7 briefly discusses extensions to non-Haar wavelets and nonlinear transformations, and addresses relighting with out-of-plane rotation and arbitrarily shaped 3D lights. Readers more interested in implementation may want to first familiarize themselves with the basic concepts introduced in Section 3, and then focus on the applications in Sections 5 and 6, skimming through the development of the theory in Section 4 as needed.

2. PREVIOUS WORK

Light transport analysis. Recent papers [Durand et al. 2005; Ramamoorthi et al. 2007] have conducted a comprehensive analysis of light transport in Fourier and gradient representations. As noted in Durand et al. [2005], one of the main aspects is the propagation of light from an area source, which can be written as an affine transform (in much the same form as Figure 1(c)). Ramamoorthi et al. [2007] characterized the basic mathematical operations of light transport, noting that linear or affine transforms are a key element, but there is

no simple formula in wavelets. By developing a framework for affine double- and triple-product wavelet integrals, we take a significant step towards a full computational framework for rendering in the wavelet domain.

Double and triple product integrals. Much of the work in relighting [Sloan et al. 2002; Ng et al. 2003] can be seen as double product integrals of the illumination and the light transport function. These integrals usually reduce to simple dot products in orthonormal bases like spherical harmonics and Haar wavelets. Subsequently, Ng et al. [2004] developed the triple-product integral framework to consider the integration of the lighting, BRDF, and visibility, as needed for changing both view and illumination. The same mathematics can be applied to efficiently multiplying two wavelet signals [Beylkin 1992a]. Most recently, these results have been generalized to products of multiple functions [Sun and Mukherjee 2006]. Our work can be viewed as an important generalization of the *standard* double- and triple-product integral framework to *affine* double- and triple-product integrals. We also briefly discuss nonlinear transformations in Section 7.

Affine transforms of basis functions. Affine transforms of Fourier basis functions are well known [Bracewell et al. 1993]. Spherical harmonics can be analytically rotated, as often used for environment maps [Sloan et al. 2002]. However, the standard affine transform usually considered in the spatial domain has no simple analog in the spherical domain; therefore, we do not consider spherical harmonics in this article. Researchers have approximated the affine transform using a combination of spherical rotations and a spherical scaling operation [Wang et al. 2006a]. This approximation is limited to only midrange illumination, since the distortion tends to be too severe in the near field.

Wavelets lack translation invariance, and have no simple formula for affine transforms. Beylkin [1992b] and others have studied the concurrent wavelet decomposition of all integer (not continuous) circulant shifts of a signal. In comparison, our goals are different and more ambitious in that we want to consider a general continuous affine transform (and not just all integer shifts). Nevertheless, we are inspired by the sparsity indicated by Beylkin [1992b] and Wang et al. [2006b] who studied wavelet rotations. We have developed a fast algorithm for wavelet affine transforms, even when no simple analytic formula exists.

Near field relighting and image processing. Relighting techniques have been developed from the basic approach introduced by Nimeroff et al. [1994] and Dorsey et al. [1995] to much recent work on Precomputed Radiance Transfer (PRT) [Sloan et al. 2002]. In terms of our application, the most closely related works are methods extending PRT to near-field and dynamic settings. Spherical harmonic gradients [Annen et al. 2004] and scaling [Wang et al. 2006a] try to approximate the effects of midrange illumination. Spherical harmonic exponentiation [Ren et al. 2006] can render near-field soft shadowing effects in real time, but has to use sphere sets to approximate geometries so that affine transforms are avoided. As a precursor, Mei et al. [2004] proposed decomposing the illumination into directional lights and searching through precomputed spherical radiance transport maps to render dynamic scenes. Zhou et al. [2005] then developed precomputed shadow or source radiance fields[1] to support all-frequency effects, but this cannot support very high sam-

---

[1]The radiance function of Zhou et al. is 5D, consisting of all possible spatial locations and angular directions. This is just the light field from the source, and a simpler 4D representation, including possibly directly in the wavelet domain, should be possible. However, the 4D representation is very difficult

Notations	Descriptions	Acronyms
v, u	Integration variables	
s, o	Scaling and offsetting variables	
l_i, t_i	Level and offset of a wavelet $\psi_i(v)$	
$F(v), G(v)$	Functions in pixel domain	
$H(v), W(v)$	Functions in pixel domain	
$\Phi_i(v)$	Orthonormal basis function	
$\psi_i(v)$	Wavelet basis function	
C_{ij}	Standard Coupling Coefficient	SCC
C_{ijk}	Standard Tripling Coefficient	STC
$D_{ij}(s, o)$	Affine Coupling Coefficient	ACC
$D_{ijk}(s, o)$	Affine Tripling Coefficient	ATC
$M(r, c)$	Canonical Coupling Coefficient	CCC
r, c	Radius and center of the transformed mother wavelet in the CCC	

Fig. 2.   Notations used in the article.

pling rates nor general changes in the lighting (such as editing the pattern of an area source in lighting design). Kristensen et al. [2005] also extend PRT to local lighting using unstructured light clouds. Overall, these methods have to assume predetermined lights, allowing changes of only light positions or scale intensities. In contrast, our method first assumes planar area sources, allowing near-field relighting with dynamic editable lights, and eliminating the need for lighting-dependent precomputations and storage. In Section 7, we discuss a possible extension of our algorithm to general out-of-plane rotation and arbitrarily shaped 3D lights.

Other applications shown at the end of the article include wavelet importance sampling the product of the area lighting and general BRDFs for Monte Carlo offline rendering systems, which have previously only been applied to distant environment maps [Clarberg et al. 2005]. In addition, we also explore image processing. A number of image operations such as additions and multiplications can already be performed directly in the wavelet domain [Drori and Lischinski 2003; Kutil 2005]. Our algorithms extend these operations to dilations and translations, which can be cast as affine transforms.

3.   AFFINE DOUBLE AND TRIPLE PRODUCT INTEGRALS

In this section, we introduce affine coupling and tripling coefficients. We use 1D wavelets for simplicity in our analysis (we will see that our results extend directly to higher dimensions in Section 4.4). Notation used in this article is shown in Figure 2.

3.1   Standard Coupling and Tripling Coefficients

Double- and triple-product integrals can be written respectively as

$$H(v) = \int F(v)G(v)dv, \tag{1}$$

$$H(v) = \int F(v)G(v)W(v)dv, \tag{2}$$

where H, F, G, and W are functions in the spatial or angular domain. For example, in relighting applications, v could be the incident angle, H could be the reflected radiance, and F, G, and W

---

to unpack in real time for relighting applications that may require the full incident illumination field at any given pixel. By directly propagating lighting from the area light source, we bypass the need for these higher-dimensional representations.

could correspond to the lighting, visibility, and BRDF, respectively. For compact representation, it is common to expand them in basis functions. The double-product integral becomes

$$H(v) = \int \left(\sum_i F_i \Phi_i(v) \right) \left(\sum_j G_j \Phi_j(v) \right) dv$$

$$= \sum_i \sum_j F_i G_j \int \Phi_i(v) \Phi_j(v) dv$$

$$= \sum_i \sum_j F_i G_j C_{ij}, \tag{3}$$

$$C_{ij} = \int \Phi_i(v) \Phi_j(v) dv, \tag{4}$$

where $\Phi(v)$ are some set of orthonormal basis functions. We denote C_{ij} as the *Standard Coupling Coefficient*, or SCC.

Similarly, the triple-product integral becomes

$$H(v) = \int \left(\sum_i F_i \Phi_i(v) \right) \left(\sum_j G_j \Phi_j(v) \right) \left(\sum_k W_k \Phi_k(v) \right) dv$$

$$= \sum_i \sum_j \sum_k F_i G_j W_k \int \Phi_i(v) \Phi_j(v) \Phi_k(v) dv$$

$$= \sum_i \sum_j \sum_k F_i G_j W_k C_{ijk}, \tag{5}$$

$$C_{ijk} = \int \Phi_i(v) \Phi_j(v) \Phi_k(v) dv. \tag{6}$$

We denote C_{ijk} as the *Standard Tripling Coefficient*, or STC.

3.2 Affine Coupling and Tripling Coefficients

The SCC and STC only apply well to functions that are "fixed" and "static." We now consider functions that are affinely transformed. Without loss of generality, we assume $F(v)$ is scaled and translated to $F(sv + o)$. With respect to the illustration in Figure 1(c), the offset o corresponds to the horizontal position x, and the scale s to the vertical distance z. This leads to an important variation of the standard double-product integral, which we call the affine double-product integral:

$$H(s, o; v) = \int F(sv + o) G(v) dv$$

$$= \int \left(\sum_i F_i \Phi_i(sv + o) \right) \left(\sum_j G_j \Phi_j(v) \right) dv$$

$$= \sum_i \sum_j F_i G_j D_{ij}(s, o), \tag{7}$$

$$\boxed{D_{ij}(s, o) = \int \Phi_i(sv + o) \Phi_j(v) dv.} \tag{8}$$

We denote $D_{ij}(s, o)$ as the *Affine Coupling Coefficient*, or ACC. Just as standard triple-product integrals are used for multiplication, the same machinery as Eq. 8 is useful for affinely transforming a function. In fact, Eq. 7 is equivalent to an affine transform of F followed by a standard double-product integral with G.

Similarly, the affine triple-product integral can be written as

$$H(s, o; v) = \int F(sv + o) G(v) W(v) dv$$

$$= \int \left(\sum_i F_i \Phi_i(sv + o) \right) \left(\sum_j G_j \Phi_j(v) \right) \left(\sum_k W_k \Phi_k(v) \right) dv$$

$$= \sum_i \sum_j \sum_k F_i G_j W_k D_{ijk}(s, o), \tag{9}$$

$$\boxed{D_{ijk}(s, o) = \int \Phi_i(sv + o) \Phi_j(v) \Phi_k(v) dv.} \tag{10}$$

We denote $D_{ijk}(s, o)$ as the *Affine Tripling Coefficient*, or ATC.

3.3 Discussion of Properties

Properties of SCC and STC. Because of the orthonormal relation between two different basis functions, the SCC reduces to a Kronecker delta function, $C_{ij} = \delta_{ij}$, and has exactly N nonzero terms, where N is the total number of basis functions. The STC is slightly more complicated. For general orthonormal bases, the complexity (i.e., number of nonzero coefficients) of the STC is $O(N^3)$. Sparsity exists for bases with special structures [Ng et al. 2004], for example, the complexity is $O(N)$ for pixel bases, $O(N^2)$ for 2D Fourier series, and $O(N \log N)$ for Haar wavelets. The complexities of the SCC and STC are recapped in Section 4.3. In addition, note that both the SCC and STC are symmetric. We have

$$C_{ij} = C_{ji}, \qquad C_{ijk} = C_{perm(ijk)}, \tag{11}$$

where *perm(ijk)* is any permutation of the triplet (i, j, k).

Properties of ACC and ATC. By contrast, the ACC and ATC both gain two more degrees of freedom, since there are two new arguments: the scale s and the offset o. In fact, the SCC and STC are special cases of the ACC and ATC, when the scale is 1 and the offset is 0.

$$C_{ij} = D_{ij}(1, 0), \qquad C_{ijk} = D_{ijk}(1, 0) \tag{12}$$

The ACC and ATC do not preserve the same symmetries of the SCC and STC. For example, the ACC follows this more complex identity.

$$D_{ij}(s, o) = \frac{1}{s} D_{ji} \left(\frac{1}{s} - \frac{o}{s} \right). \tag{13}$$

Relation between ACC and ATC.

LEMMA 1. *For any orthonormal basis, the ATC can be represented using the the ACC and STC as*

$$D_{ijk}(s, o) = \sum_l D_{il}(s, o) C_{ljk}. \tag{14}$$

The proof is in Appendix A, and follows from expanding $\Phi(sv + o)$ in terms of $\Phi(v)$ and then using associativity. Lemma 1 indicates that the computational complexity of the affine tripling coefficient $D_{ijk}(s, o)$ relies on those of the ACC and STC. Lemma 1 also suggests a way of evaluating the ATC using the ACC and STC.

4. COMPLEXITY OF AFFINE COUPLING AND TRIPLING COEFFICIENTS (ACC & ATC)

In this section, we study the computational complexity of the affine coupling and tripling coefficients (ACC and ATC), determining their numbers of nonzero terms in a number of bases. In wavelets, we focus on the Haar basis for its simplicity. We present the main results

that are essential for understanding the key insights and implementing the theory, leaving many detailed mathematical derivations for the appendices.

4.1 General, Pixel, and Fourier Bases

In general orthonormal bases, the complexities of the ACC and ATC are $O(N^2)$ and $O(N^3)$, respectively (they can be lower for some values of s and o). This can be contrasted to the linear complexity of the SCC, and highlights the fact that the original orthonormal relation between different basis functions no longer holds under an affine transform.

Pixel basis functions, in mathematical terms, are disjoint and discrete step functions. A unique aspect of the pixel basis is that both the integration and product of multiple *different* pixels is zero. This leads directly to the linear $O(N)$ complexity of the ACC and ATC. A detailed account of the complexity analysis is in Appendix B. Despite its simplicity, the pixel basis is a poor basis for compression, which often outweighs its efficiency in calculation.

The Fourier series is mostly preferred in theoretical analysis and has a well-known 2D affine theorem [Bracewell et al. 1993]. Adapting this theory to 1D (Appendix B), we show that the ACC in the Fourier basis can be mapped to the SCC, and thus has the same linear $O(N)$ complexity. The specific mapping depends on the scale s and offset o. Similarly, the complexity of the ATC in the Fourier basis can be determined as $O(N^2)$ using Lemma 1 (Appendix B).

Spherical harmonics are the extension of the Fourier basis to the sphere. They are not considered in this discussion, since no standard operation in the spherical domain directly maps to the affine transform.

4.2 Haar Wavelets

We present our main theoretical contribution in this section, deriving the complexities of affine coupling and tripling coefficients in Haar wavelets. Readers interested in implementation may wish to skip to the summary of complexities in Section 4.3 on a first reading.

4.2.1 *Wavelet and Scaling Functions.* The normalized 1D Haar basis [Stollnitz et al. 1996] is defined as

—The mother scaling and wavelet functions are

$$\phi(v) = \begin{cases} 1, & \text{for } 0 \le v < 1 \\ 0, & \text{otherwise,} \end{cases}$$

and

$$\psi(v) = \begin{cases} 1, & \text{for } 0 \le v < 1/2 \\ -1, & \text{for } 1/2 \le v < 1 \\ 0, & \text{otherwise.} \end{cases}$$

—A normalized wavelet basis $\psi_j(v)$ at level l_j and offset t_j is

$$\psi_j(v) = 2^{l_j/2}\psi(2^{l_j}v - t_j),$$

which is a scaled and dilated copy of the mother wavelet.

4.2.2 *Haar Canonical Coupling Coefficient (CCC).* The ACC is a 4D function of the two subscripts i and j, the scale s, and the offset o. Similarly, the ATC is a 5D function. Since according to Lemma 1 the ATC can be reduced to the ACC, we first focus our analysis on the ACC. To reduce the dimensionality of the ACC, we invoke the standard property of Haar wavelets as a multiresolution series and write wavelet basis functions in terms of the mother wavelet $\psi(v)$.

$$\begin{aligned} D_{ij}(s, o) &= \int \psi_i(sv + o)\psi_j(v)dv \\ &= 2^{\frac{l_i+l_j}{2}}\int \psi(2^{l_i}sv - t_i + 2^{l_i}o)\psi(2^{l_j}v - t_j)dv \quad (15) \end{aligned}$$

To simplify Eq. (15), we make the substitution $u = 2^{l_j}v - t_j$ so that transformations of the two mother wavelets can be merged into a single combined transformation. We have

$$\begin{aligned} D_{ij}(s, o) &= 2^{\frac{l_i-l_j}{2}}\int \psi(2^{l_i-l_j}su - t_i + 2^{l_i}o + st_j2^{l_i-l_j})\psi(u)du \\ &= 2^{\frac{l_i-l_j}{2}}\int \psi\left(\frac{u}{2r} - \frac{c}{2r} + \frac{1}{2}\right)\psi(u)du \end{aligned}$$

$$\boxed{D_{ij}(s, o) = 2^{\frac{l_i-l_j}{2}}M(r, c)}, \quad (16)$$

where

$$\boxed{M(r, c) = \int \psi\left(\frac{u}{2r} - \frac{c}{2r} + \frac{1}{2}\right)\psi(u)du}, \quad (17)$$

$$r = \frac{2^{l_j-l_i-1}}{s}, \quad \text{and} \quad c = \frac{2^{l_j-l_i}t_i - 2^{l_j}o + 2^{l_j-l_i-1}}{s} - t_j. \quad (18)$$

We call $M(r, c)$ the *Canonical Coupling Coefficient*, or CCC. The CCC encapsulates the core structure of the ACC. Variables r and c dictate the combined transformation in wavelet $\psi(\frac{u}{2r} - \frac{c}{2r} + \frac{1}{2})$. Intuitively, they are, respectively, the *radius* and *center* of the transformed wavelet $\psi(\frac{u}{2r} - \frac{c}{2r} + \frac{1}{2})$, as shown in Figure 3(a).

Property 1. The evaluation of the 4D affine coupling coefficient reduces to a combination of a few simple function computations and an estimate of the 2D analytic function $M(r, c)$ called the canonical coupling coefficient, as described by Eq. 16. Variables r and c are given by Eq. 18. $M(r, c)$ is a sparse, piecewise linear, and symmetric function.

In its analytic form, $M(r, c)$ is a branching function as shown in Figure 4. The calculation of $M(r, c)$ is equivalent to determining the overlapping relation between the original mother wavelet $\psi(u)$ (denoted as β) and the transformed one $\psi(\frac{u}{2r} - \frac{c}{2r} + \frac{1}{2})$ (denoted as α). Their overlapping relation depends on their *relative* positions (centers) and sizes (radii). We group their overlapping relations into six cases and show them in Figure 3. For brevity in exposition, we assume that α's radius is smaller than that of β, and α is located on the left. Our analysis will still hold when α's radius is actually larger or it is located on the right, since β can then be viewed as "the transformed wavelet" and exchange roles with α. In all cases, $M(r, c)$ can be computed in no more than 9 lines of code (Figure 4 also cross-references the six cases in Figure 3).

4.2.3 *Properties of Haar CCC.* $M(r, c)$ has special structures and important properties that can be exploited for computation. In Figure 5, we plot $M(r, c)$ in both r and c dimensions to better expose many such properties. We examine a few important ones here:

Sparsity. $M(r, c)$ is sparse. As shown in Figure 5, $M(r, c)$ is nonzero over only very limited ranges of the radius r and the center c. In particular, $M(r, c)$ is zero when the right endpoint of the transformed wavelet is less than 0 (meaning $c + r < 0$), or the left endpoint is greater than 1 (meaning $c - r > 1$). In these cases, the
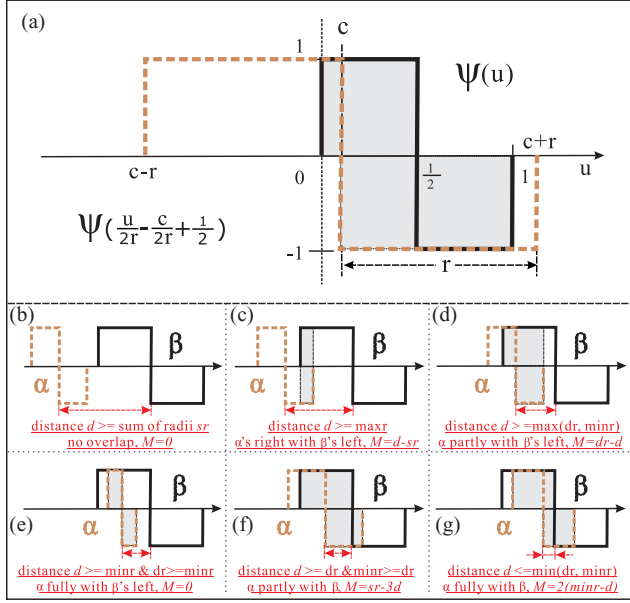
Fig. 3. The canonical coupling coefficient $M(r, c)$ is an integration of two wavelets denoted as α and β, with α being affinely transformed, as shown in (a). Variables r and c correspond to the radius and center of the transformed wavelet α. (b)–(g) show different overlapping relations of wavelets α and β. Variables $maxr$ and $minr$ are, respectively, the bigger and smaller of the radii of the two wavelets, and sr and dr their sum and difference. The red arrows show the distance d.

two wavelets do not overlap and their integration is 0. These two boundary conditions compactly combine to $|c - 0.5| > r + 0.5$.

$$M(r, c) = 0, \quad \text{when } |c - 0.5| > r + 0.5 \quad (19)$$

Since the ACC reduces to the CCC, the ACC will be sparse if many combinations of i, j, s, and o make r and c fall into the zero ranges. We will discuss the complexity of the ACC formally in Section 4.2.4.

Piecewise linearity. $M(r, c)$ is a piecewise linear function. As depicted in Figure 5, $M(r, c)$ has only a limited set of slopes $\{0, \pm1, \pm2 \pm 3\}$ in both r and c dimensions. This is because the partial derivatives of $M(r, c)$ with respect to r and c are just combinations of a few mother wavelets $\psi(u)$. These mother wavelets are valued at three break-points ($c - r$, $c + r$ and c, as in Figure 3(a)) of the transformed wavelet $\psi(\frac{u}{2r} - \frac{c}{2r} + \frac{1}{2})$. The derivation of the partial derivatives is in Appendix C.

$$\frac{\partial M(r,c)}{\partial c} = 2\psi(c) - \psi(c - r) - \psi(c + r) = \{0, \pm1, \pm2, \pm3\}, \quad (20)$$

$$\frac{\partial M(r,c)}{\partial r} = \psi(c - r) - \psi(c + r) = \{0, \pm1, \pm2\} \quad (21)$$

Consequently, the ACC is also piecewise linear, and its gradient is easily computed from that of $M(r, c)$ using the chain rule.

Symmetry. $M(r, c)$ is symmetric. In the c dimension, as shown in Figure 5(a), $M(r, c)$ is reflection symmetric about 0.5. This reflection symmetry is also used in computation in Figure 4 as the calculation depends not on c directly, but on $d = |c - 0.5|$. $d = |c - 0.5|$ is reflection symmetric about 0.5. In mathematical terms, we have

$$M(r, 0.5 + c) = M(r, 0.5 - c). \quad (22)$$

In the r dimension, $M(r, c)$ also has a certain degree of symmetry. When r is above 1, we can invert its value to below 1 by changing

Fig. 4. Analytic formula of $M(r, c)$ in pseudocode. Branches correspond to different overlappings between the two wavelets, as illustrated in Figure 3.

the integration variable in Eq. (16) to $w = \frac{u}{2r} - \frac{c}{2r} + \frac{1}{2}$.

$$M(r, c) = 2r M\left(\frac{1}{4r}, \frac{1}{2} + \frac{1}{4r} - \frac{c}{2r}\right) \quad (23)$$

Derivations of these symmetry properties are in Appendix C. Due to these symmetries, we only store a nonrepeating quarter ($[0 < r \le 1, -r \le c \le 0.5]$) of the nonzero range of $M(r, c)$ and save three-fourths of the storage space.

Boundedness. $M(r, c)$ is both upper and lower bounded, which makes it ideal for quantization and encoding in hardware textures.

$$max\big(M(r, c)\big) = 2min(r, 0.5) \le 1,$$
$$min\big(M(r, c)\big) = -min(r, 0.5) \ge -0.5$$

4.2.4 *Complexities of Haar ACC and ATC.* The number of nonzero ACC terms varies significantly with the scale and offset. We first show empirically in Figure 6 the $O(N \log N)$ complexity of the ACC in Haar wavelets. Complexities for general wavelets are discussed in Section 7, and are found to be similar to the Haar case. In Figure 6(a), we plot the numbers of nonzero ACC terms versus the total numbers of the wavelets for 50 randomly generated sets of scales and offsets. Seven representative curves out of the 50 are highlighted, and their numerical values are tabulated in Figure 6(c). To better illustrate the logarithmic behavior part of the $O(N \log N)$ complexity, we divide all curves by N and plot them again in Figure 6(b). Note that $O(N \log N)$ is just an upper bound. The actual complexity of the ACC is comparable to that of standard tripling coefficients (STCs) developed by Ng et al. [2004]. The $O(N \log N)$ complexity can also be mathematically proved by determining the upper bound of the total overlapping pairs between the two wavelet trees, one of which is affinely transformed. We give the detailed proof in Appendix D. In practice, indices of nonzero ACC terms can be picked either using Eq. (19) (the sparsity property of the CCC), or from a compact pretabulated table, as implemented in Section 5.3.2.

To compute the complexity of Haar ATC, we invoke the standard property of Haar wavelets that the product of two wavelets is either the finer wavelet up to a scale if they overlap, or zero otherwise. Therefore when wavelets overlap, the ATC reduces to the ACC, and subsequently to the CCC according to Property 1. When wavelets do not overlap, the ATC is simply zero. Based on this key observation, we derive Property 2 that relates the ATC to the CCC.

Property 2. The Haar ATC, as defined in Eq. (10), is evaluated using the canonical coupling coefficient as follows.
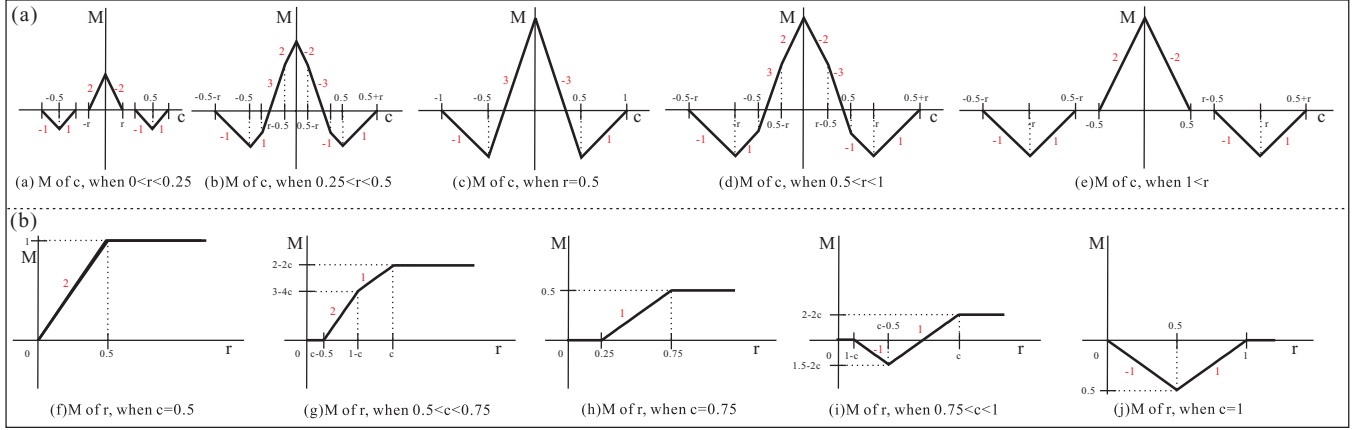
Fig. 5. (a) Canonical Coupling Coefficient $M(r, c)$ in the c dimension (the horizontal axis is $c - 0.5$ to better demonstrate the symmetry) for all different ranges of the radius r. The red labels represent line slopes, while the black labels are measurement marks along the axes; (b) $M(r, c)$ in the r dimension for $c \geq 0.5$. The horizontal axis is the radius r of the transformed wavelet. When $c \leq 0.5$, $M(r, c)$ can be computed using the symmetry $M(r, c - 0.5) = M(r, 0.5 - c)$. This graph illustrates a number of important properties of M, such as its sparsity, piecewise linearity, symmetries, and boundedness.
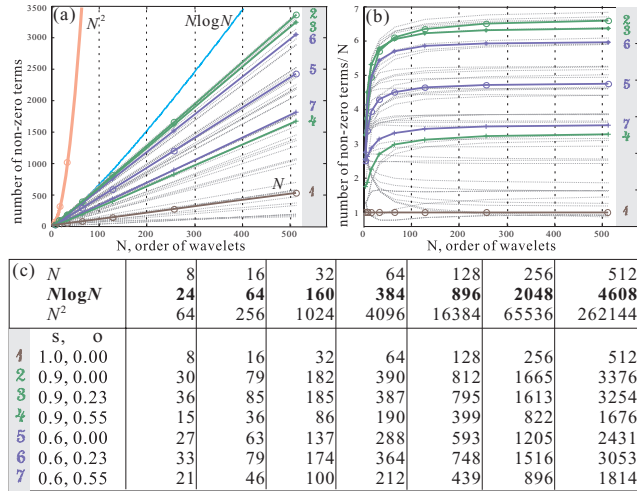

Fig. 6. Complexity of affine Haar wavelet coupling coefficients for 50 randomly generated sets of scales and offsets. Figure 6(a) empirically shows the $O(N \log N)$ complexity of the ACC. Seven representative curves out of 50 are highlighted and their values listed in Table (c). Note that in Figure 6(a) for most scales and offsets, the actual numbers of nonzero ACC terms fall well below $N \log N$ (cyan line), and some even below N (when significant offsets and scales transform a large portion of the wavelet tree out of [0, 1]). The formal proof of the complexity of the ACC is in Appendix D.

—In Eq. (10), if basis functions $\Phi_j(v)$ and $\Phi_k(v)$ overlap,

$$D_{ijk}(s, o) = 2^{\frac{l_i - l_m + l_n}{2}} M(r, c), \qquad (24)$$

where $n = min(j, k)$, $m = max(j, k)$, and

$$c = \frac{2^{l_m - l_i} t_i - 2^{l_m} o + 2^{l_m - l_i - 1}}{s} - t_m,$$

$$r = \frac{2^{l_m - l_i - 1}}{s}.$$

—Otherwise, $D_{ijk}(s, o) = 0$.

It can be shown that the complexity of the ATC in the Haar basis is $O(N \log^2 N)$. We leave the proof in Appendix D. For interested readers, Property 2 can also be verified using Lemma 1.

4.3 Summary of Complexities in All Bases

The following table summarizes the computational complexities of the SCC, STC, ACC, and ATC in different bases. For general orthonormal bases, there is no sparsity due to the lack of special structures. The pixel basis, despite its simplicity, is a poor basis for compression and hence undesirable for practical applications. The Fourier basis is widely used in theoretical analysis, but is not good at capturing all-frequency lighting (or visibility/BRDF) as shown in Ng et al. [2003]. Haar wavelets are preferred in all-frequency relighting and only need a handful of coefficients and basis functions to achieve good approximations. To distinguish from N, the total number of basis functions, we denote the number of terms retained after compression as n. n is usually around $1 \sim 2\%$ of N for wavelets. As shown in the following table, Haar wavelets have linear or close-to-linear complexities ranging from n to $O(n \log^2 N)$ across all columns. In practice, we would need to compute far fewer terms after compression than in the other bases. This makes Haar wavelets ideal for many operations in practical applications. We also show the complexities of ACCs and ATCs for general wavelets are $O(n \log N)$ and $O(n \log^2 N)$, respectively, in Section 7.

Bases	SCC	ACC	STC	ATC
General	N	$O(N^2)$	$O(N^3)$	$O(N^3)$
Pixel	N	N	N	N
Fourier	n	n	$O(n^2)$	$O(n^2)$
Haar	n	$O(n \log N)$	$O(n \log N)$	$O(n \log^2 N)$
General Wavelets	n	$O(n \log N)$	$O(n \log N)$	$O(n \log^2 N)$

4.4 Generalization to Higher Dimensions

A high-dimensional Haar basis can be viewed as a product of multiple 1D Haar wavelet basis functions for both standard and nonstandard decompositions, as utilized by Clarberg et al. [2005]. Similarly to a high-dimensional Fourier basis, a high-dimensional Haar basis

can be written as

$$\Psi_{\mathbf{i}}(\mathbf{v}) = \prod_{q=1}^{Q} \psi_{i_q}(v_q), \qquad (25)$$

where $\Psi_{\mathbf{i}}$ is a Q-dimensional basis function, variables $\mathbf{i}$ and $\mathbf{v}$ are Q-dimensional vectors, and i_q and v_q index into the qth dimension of vectors $\mathbf{i}$ and $\mathbf{v}$.

The ACC becomes

$$
\begin{aligned}
\mathbf{D_{ij}(s, o)} &= \underbrace{\int \ldots \int}_{Q} \Psi_{\mathbf{i}}(\mathbf{s}\mathbf{v} + \mathbf{o})\Psi_{\mathbf{j}}(\mathbf{v})d\mathbf{v} \\
&= \prod_{q=1}^{Q} \underbrace{\left(\int \psi_{i_q}(s_q v_q + o_q)\psi_{j_q}(v_q)dv_q \right)}_{\text{1D affine coupling coefficient}} \\
&= \prod_{q=1}^{Q} D_{i_q j_q}. \qquad (26)
\end{aligned}
$$

Similarly, the ATC becomes

$$
\begin{aligned}
\mathbf{D_{ijk}(s, o)} &= \underbrace{\int \ldots \int}_{Q} \Psi_{\mathbf{i}}(\mathbf{s}\mathbf{v} + \mathbf{o})\Psi_{\mathbf{j}}(\mathbf{v})\Psi_{\mathbf{k}}(\mathbf{v})d\mathbf{v} \\
&= \prod_{q=1}^{Q} \underbrace{\left(\int \psi_{i_q}(s_q v_q + o_q)\psi_{j_q}(v_q)\psi_{k_q}(v_q)dv_q \right)}_{\text{1D affine tripling coefficient}} \\
&= \prod_{q=1}^{Q} D_{i_q j_q k_q}, \qquad (27)
\end{aligned}
$$

where $\Psi_{\mathbf{i}}$, $\Psi_{\mathbf{j}}$, and $\Psi_{\mathbf{k}}$ are Q-dimensional basis functions, and variables $\mathbf{i}$, $\mathbf{j}$, $\mathbf{k}$, $\mathbf{s}$, and $\mathbf{o}$ are vectors of Q elements. Eq. (26) and Eq. (27) show that the high-dimensional Haar ACC and ATC, are products of multiple 1D ACCs and ATCs, respectively. They enable scales and translations in wavelets, but not rotations or shears.

The complexities of the high-dimensional ACC and ATC are respectively $O(n[\frac{\log N}{Q}]^Q)$ and $O(n[\frac{\log N}{Q}]^{2Q})$. If we denote the total number of basis functions in 1D as $\tilde{N}$, we obtain $N = \tilde{N}^Q$ for higher dimensions. Noting that $\log N = Q \log \tilde{N}$ and the complexities of 1D ACCs and ATCs are $O(\tilde{n} \log \tilde{N})$ and $O(\tilde{n} \log^2 \tilde{N})$, multiplying these complexities Q times respectively generates the complexities of the high-dimensional ACC and ATC. For relighting, we will be working with 2D ACCs, whose complexity is $O(n \log^2 N)$ from the previous analysis. However, we show in Section 5.3 that we can develop a more efficient algorithm with $O(n \log N)$ complexity.

5. INTERACTIVE NEAR-FIELD RELIGHTING

We now develop our main practical application, showing how to integrate our theory with the PRT framework to render near-field lighting effects at interactive rates. Later Section 6 will present initial results for wavelet importance sampling for near-field planar area lights and image processing directly in the wavelet domain.

5.1 Basic Relighting Framework

In the reflection equation, the exitant radiance is

$$
\begin{aligned}
B(\mathbf{q}, \boldsymbol{\omega}_o) &= \int_\Omega L(\mathbf{q}, \boldsymbol{\omega}_i)V(\mathbf{q}, \boldsymbol{\omega}_i)\rho(\mathbf{q}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_o)(\boldsymbol{\omega}_i \cdot \mathbf{n})d\boldsymbol{\omega}_i \\
&= \int_\Omega L(\mathbf{q}, \boldsymbol{\omega}_i)T(\mathbf{q}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_o)d\boldsymbol{\omega}_i, \qquad (28)
\end{aligned}
$$

where B is the reflected radiance as a function of the spatial location $\mathbf{q}$ and outgoing direction $\boldsymbol{\omega}_o$, L is the incoming lighting, $\boldsymbol{\omega}_i$ is the incident direction, V is the visibility, ρ is the BRDF, and $\mathbf{n}$ is the surface normal. Symbols in bold represent 2D vectors. Often the visibility V, BRDF ρ and cosine term $(\boldsymbol{\omega}_i \cdot \mathbf{n})$ are combined into the transport function T as shown in Eq. (28). Eq. (28) is a double product integral of the lighting and the transport function, and is often expanded in the Haar wavelet basis in actual computations.

In most relighting algorithms, distant illumination is assumed so that $L(\mathbf{q}, \boldsymbol{\omega}_i) = L(\boldsymbol{\omega}_i)$ is the same for all vertices $\mathbf{q}$. However, in the near-field setting considered here, we need to propagate light from the planar area source to each spatial location. We will show next that this corresponds to an affine transform of the original area source radiance. Thus, Eq. (28) becomes an affine double product integral, and can be efficiently computed on the fly for each vertex using the theory of Sections 3 and 4.

5.2 Light Propagation

We consider light propagation from an area light source in 1D as shown in Figure 1(c). Extension to 2D planar sources in Section 5.3 is straightforward, as explained in Section 4.4. Note that we consider propagation to surfaces only at the "front" of the light source; surfaces at the back of the source will not be illuminated, and this must be tested separately. The area light is $F(v)$, and the incident radiance at a vertex $L(z, x; v)$. Variables z and x are the vertical and horizontal coordinates of the vertex. v is the intercept on a virtual plane a unit distance away. From simple trigonometry, the incident radiance can be written as

$$\boxed{L(z, x; v) = F(zv + x),} \qquad (29)$$

which is an affine transformation of the original light $F(v)$. z and x respectively correspond to s and o previously used in the ACC.

It is worth making a note of the parameterization. In terms of the more familiar angular coordinates, $v = \tan\theta$, and we must include the correct angular/area measure for $dv/d\theta$ when changing the integration variable from θ to v. As is conventional, we incorporate this into the transport function $T(v)$. We emphasize that while our parameterization is similar to the linearization used in, for instance Durand et al. [2005], Eq. (29) is exact, and not an approximation.

The coefficients of the incident radiance $L_k(z, x)$ can be computed as

$$
\begin{aligned}
L_k(z, x) &= \int F(zv + x)\psi_k(v)dv \\
&= \int \left(\sum_i F_i \psi_i(zv + x) \right) \psi_k(v)dv \\
&= \sum_i F_i D_{ik}(z, x), \qquad (30)
\end{aligned}
$$

where F_i are the wavelet coefficients of F. Eq. (30) propagates light directly in the wavelet domain.

We emphasize that Eq. (29) and (30) simply express the incident radiance at a given spatial location. The PRT algorithm can be treated
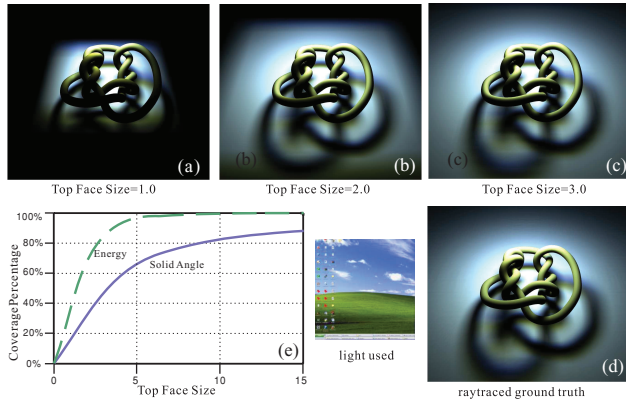
Fig. 7. (a), (b), and (c) are renderings using our method with different face sizes. (d) is the ground truth from a ray tracer, using a full representation of the lighting environment at each vertex. (e) shows the coverage ratios of the top face over the upper hemisphere with different sizes. In the graph, the blue solid curve is the top face solid angle covered, and the green dotted curve the energy of a centrally aligned Lambertian lobe. For a standard cube map, its top face size is 2. Top face sizes 3 or 4 can generate visually accurate results.

as a black box that takes this incident lighting L and applies the light transport function T. Therefore, our method can be incorporated into almost any PRT framework and representation for T, including those that are view dependent.

By substituting the angle ω_i with v using appropriate normalizing weights (numerical cubature), we can write Eq. (28) as

$$B = \int L(z, x; v) T(v) dv = \sum_i \sum_j F_i T_j D_{ij}(z, x), \quad (31)$$

where T_j is the wavelet coefficient of the transport function $T(v)$.

5.2.1 *Parameterization.* It is common to parameterize the sphere S^2 of directions using a cube map consisting of six faces. Propagations of light to the six faces would require six affine transformations. To speed up the computation, we make a trade-off by using only the top face[2], but expand its size to cover a larger solid angle. The top face is aligned with the light plane. Our simplification is motivated by the observation that in near-field settings for most vertices, at least one of the lighting, visibility, or BRDF terms would tend to peak at the top face and die out towards peripheral regions. In addition, the cosine term (cosine of the incident angle) reduces contributions towards grazing angles.

We make the top face adjustable so that a large or smaller solid angle can be covered, as needed. The top face in the standard cube map is assumed size 2. Figure 7 shows that midsized faces can cover a sufficient portion of the hemisphere. For example, a size 10 top face covers 82% solid angle of a hemisphere and captures 99% energy of a centrally aligned Lambertian cosine lobe. The surface normal is assumed to be normal to the planar light. Therefore, the top-face approximation works best for the ground plane and other surfaces oriented with their normals similar to that of the light source. Note that the lighting from the planar source will itself peak at the top face and reduce in intensity at other regions. This allows us to extend the approximation to surfaces oriented in other directions as well.

In our experiments, sizes 3 or 4 suffice to generate visually accurate results.

5.2.2 *Assumptions and Limitations.* We have assumed planar area light sources, where the angular distribution of light is uniform. In Section 7, we discuss extending our relighting algorithm to general 3D lights. We have not implemented, but show theoretically (Appendix E) how to handle lights with angular variations such as light fields using ATCs. As noted earlier, we use an expanded top-face parameterization that may omit light incident at grazing angles. Our method is general enough to allow interactive scaling, translation, and horizontal rotation of lights and general edits to the light textures. Horizontal rotations are achieved by simply rotating the light textures in the pixel domain, before applying the wavelet transform. As in most wavelet-based relighting algorithms, our algorithm does not support out-of-plane rotations of the lighting. However, in Section 7, we discuss possible approaches to incorporate out-of-plane rotation into our framework.

5.3 Rendering Algorithm

We present key computation steps and major rendering results in this section. All renderings and time measurements are done on a commodity 3.0 GHz Dual-Quad-Core PC with 4GB memory.

5.3.1 *Log-Linear-Time Light Propagation.* To propagate light, we compute Eq. (30) in 2D,

$$L_{\mathbf{k}}(\mathbf{z}, \mathbf{x}) = \sum_{\mathbf{i}} F_{\mathbf{i}} D_{\mathbf{ik}}(\mathbf{z}, \mathbf{x}),$$

where $\mathbf{x}$ and $\mathbf{z}$ are 2D variables[3], and $\mathbf{i}$ and $\mathbf{k}$ are, respectively, 2D vectors of (i_1, i_2) and (k_1, k_2). Since we have to loop through all subscripts $\mathbf{i}$ and $\mathbf{k}$, we only evaluate the equation for nonzero 2D ACC $D_{ik}(z, x)$. Variables $\mathbf{z}$ and $\mathbf{x}$, respectively, correspond to the scale s and the translation o. Recall that we use N to denote the total number of basis functions and n for the number of terms retained after compression. The total cost appears to be the $O(n \log^2 N)$ complexity of the 2D ACCs, which is derived in Section 4.4.

In fact, a better $O(n \log N)$ algorithm exists if we separate 2D ACCs as products of 1D ACCs and perform the computations in each dimension in succession.

Step 1: $\forall \, i_2, k_1 \quad Z_{i_2 k_1}(z_1, x_1) = \sum_{i_1} F_{i_1 i_2} \cdot D_{i_1 k_1}(z_1, x_1)$

Step 2: $\forall \, k_1, k_2 \quad L_{k_1 k_2}(z_1, z_2, x_1, x_2) = \sum_{i_2} Z_{i_2 k_1} \cdot D_{i_2 k_2}(z_2, x_2)$

$Z_{i_2 k_1}$ is an intermediate variable that carries the accumulation result from the first dimension. Step 1 involves looping over subscripts i_1, i_2 and k_1. For any given i_2, over all i_1 and k_1, there are approximately $O(\sqrt{n} \log N)$ nonzero $D_{i_1 k_1}(z_1, x_1)$ since the complexity of 1D ACCs is about the square root of that of 2D ACCs. Multiplying the number of i_2 which is about $\sqrt{n}$ gives the cost for step 1 as $O(n \log N)$. Similarly, step 2 also takes $O(n \log N)$ time, and thus the total complexity of both steps is $O(n \log N)$. This computation is performed independently for all three color channels.

5.3.2 *Precomputation and Rendering.* Transport functions T are precomputed similarly to Ng et al. [2003], except using the expanded-top-face parameterization. Light propagation (Eq. (30)) involves computing the ACCs. There are two practical approaches,

---

[2]Similar plane-angle and plane-plane parameterizations have been used in light field representations [Levoy and Hanrahan 1996; Chai et al. 2000].

[3]Since each vertex has only one depth from the light source, vector $\mathbf{z}$'s two elements are the same $z_1 = z_2$.
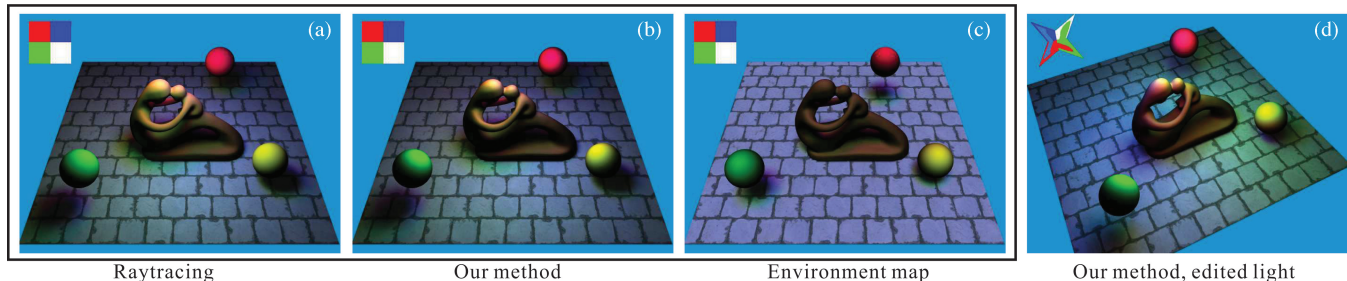
Fig. 8. A diffuse scene of a fertility sculpture and three spheres lit by a simple textured area light. The light textures are shown in the top left corners. The area sources are above the fertility sculpture, but cropped to save space. (a), (b), and (c): Images compare ray tracing, our method, and distant environment map lighting, respectively. Compared to distant lighting in (c), we see that our method in (b) correctly captures the spatially varying shading on the floor and sculpture and generates a result that is quite close to the ground truth. About 1% of source-level and 22% of target-level lighting coefficients are used in our method. (d) Finally, we can edit the light texture and shape, and rotate it in real-time to obtain a distinct appearance in (d). (Note that rotations are limited to horizontal in-plane rotations, and are performed simply in the pixel domain before transforming to wavelets. General out-of-plane rotations are not possible.) This important tool for lighting design would not be possible with previous techniques like precomputed shadow and source radiance fields.

one relying on memory lookups and the other favoring faster computation.

Memory based approach. It is easy to store all nonzero 1D ACCs $D_{ik}(s, o)$ compactly in a 4D table and look up their values during the computation. Due to the sparsity of ACCs, the cost to compute and store the 4D table is minor. As shown in the following table, nonzero ACCs for wavelet order 32 and at spatial resolution 256 only takes 1.06 seconds to precompute and 18.66MB to store. Note that the space and time cost scales close to linearly with the order of the wavelets, confirming our analysis in Section 4.2.4. Precomputed tables at all different orders and resolutions will be downloadable from our Web site (http://www1.cs.columbia.edu/cg/adtpi).

Precomputation and Storage Cost in Memory-Based Approach					
space (MB)	"Spatial" Resolutions (s, o)				
time (S)	64	128	256	512	1024
8	0.26MB	1.05MB	4.22MB	16.92MB	67.74MB
	0.011S	0.044S	0.176S	0.708S	2.760S
16	0.56MB	2.25MB	9.03MB	36.24MB	145.09MB
	0.029S	0.114S	0.457S	1.801S	7.093S
32	1.15MB	4.65MB	18.66MB	74.81MB	299.45MB
	0.066S	0.257S	1.059S	4.236S	16.871S
64	2.33MB	9.44MB	37.83MB	151.60MB	606.78MB
	0.147S	0.589S	2.368S	9.498S	37.768S

(Left side label: Wavelet Orders (i, j))

Because of the independence of lighting coefficients across vertices, light propagation can be easily parallelized on multicore machines or clusters. We implement both single-thread and multiple-thread rendering algorithms on a 3.0 GHz Dual-Quad-Core machine using the boost library. To ensure workload balance between threads, we choose a round-robin scheduling scheme among a pool of tasks, each carrying a small number (512) of vertices to compute. Compared to the single-thread, an eight-thread implementation generates the exact same rendering result, but improves the speed by about 6.5 times and easily obtains real-time rates. Most of our relighting results are generated using the multithread memory-based implementation.

Computational approach. For machines with faster computation, we tabulate $M(r, c)$ (canonical coupling coefficient defined by Eq. (17)) in a 2D texture and compute ACCs using Eq. (16). $M(r, c)$ is stored using simple angular discretization. Since $M(r, c)$ is sparse and symmetric (Section 4.2.2), the entire M table can fit into the L2

cache. A 256×256 M table in floating point only takes 0.25MB space. In our experiment settings, the computational approach is about half as fast as the memory-based approach.

5.3.3 *Nonlinear Lighting Approximation.* Realistic illumination can be well approximated using a handful of wavelet coefficients. Our model allows compression at two levels, of both the original light F (source level) and the local incident radiance L (target level), drastically speeding up the performance.

Source-level compression. At the source level, lighting coefficients F_i are ordered, and the most significant ones are picked, as in standard PRT. Using only 1% of source-level coefficients usually generates accurate renderings for our test scenes, as shown in Figures 8 and 9. In general, the compression depends on the BRDF of the receiver as well as the distance between emitter and receiver.

Target-level compression. Target lighting coefficients L_k vary across all vertices and change as the light source is dynamically updated. It is difficult to predetermine which target coefficients are important, and generating and sorting them in real time is too expensive. Instead, we assume a heuristic light $\tilde{F}_i$ and precompute lighting predictions $\tilde{L}_k$ for a number of z and x values. The light predictions $\tilde{L}_k$ are used to pick target lighting coefficients L_k at runtime.

$$\tilde{L}_{\mathbf{k}}(\mathbf{z}, \mathbf{x}) = \sum_{\mathbf{i}} \tilde{F}_{\mathbf{i}} D_{\mathbf{ik}}(\mathbf{z}, \mathbf{x}). \tag{32}$$

Rarely will the lighting predictions $\tilde{L}_k$ be exactly the same as the actual lighting L_k. They will, however, roughly track how power distributions of the wavelet coefficients under affine transformations change with respect to spatial locations. We use a simple constant light heuristic (a vector with only DC, $[1, 0, 0, \dots]$) in our experiments, and need about $20 \sim 30\%$ of target-level lighting coefficients for visually accurate results, as shown in Figures 8 and 9.

5.4 Results

We demonstrate three scenes: fertility (39,391 vertices, Figure 8), chairs (59,995 vertices, Figures 1(b) and 9(a)–(c)), and couches (68,046 vertices, Figures 1(a) and 9(d)–(f)). All scenes have static geometries and are rendered using PRT, with near-field lighting effects generated using our method. The chairs, tables, and couches have specular materials with precomputation done per vertex [Ng et al. 2003]. The fertility scene is diffuse and can be viewed from
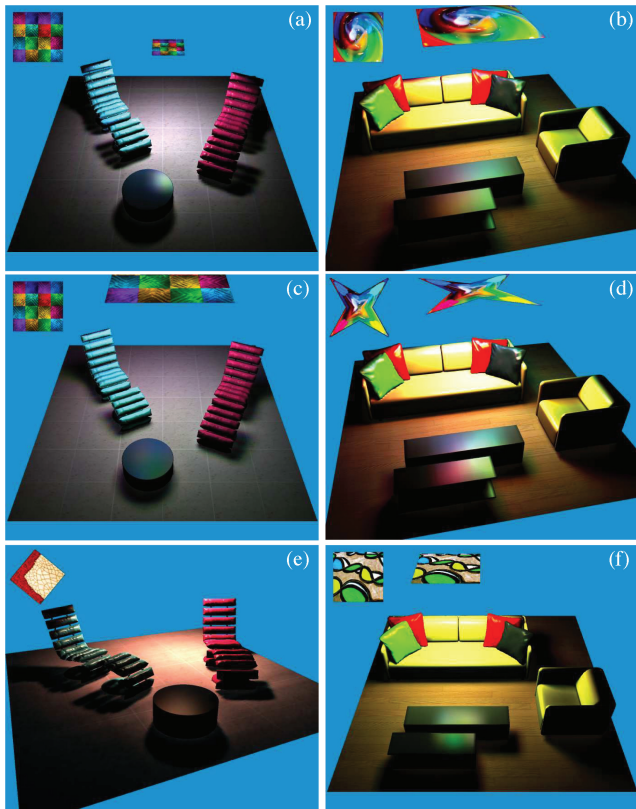
Fig. 9. Two specular scenes rendered with near-field sources using our method. Light textures used are shown in the upper left corners, and their sizes and positions vary. In the chair scene, note the sharper and larger shadows underneath the chairs when the light is small and close (a), and smaller and softer shadows when the light is larger and far away (b). Also note in the couch scene how light editing (change texture, reshape) from (d) to (e) influences specular reflections on the tables and cushions. Figure 1(a) follows (e) but resizes and rotates the light. (c) and (f) render both specular scenes from another view point with a different light texture. Specular rendering from a different viewpoint requires a separate precomputation. Close-ups can be found in Figure 1. Performance numbers are reported in Figure 11.
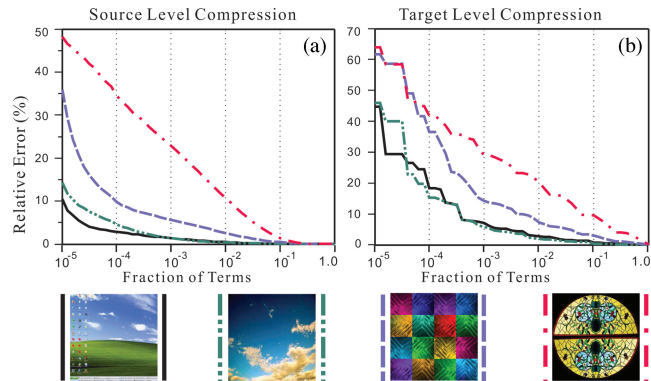

Fig. 10. Approximation errors of the source-(graph (a)) and target-(graph (b)) level compressions. Four representative light textures with different levels of high frequencies are used. The horizontal axes are on a log scale.

different angles. Please note that our technique is orthogonal to many PRT calculations and can certainly be used with view-dependent methods such as BRDF in-out factorization [Wang et al. 2004; Liu et al. 2004] or other approaches. In all cases, we can interactively edit the light.

5.4.1 *Near-Field vs. Distant Lighting Effects.* In Figure 8(b), we light a diffuse fertility scene with near-field lights. Spatially varying shadings and colored soft shadows on the floor are clearly visible. In contrast, Figure 8(c) shows a rendering using the standard environment mapping technique,[4] which fails to capture the shading variations on the floor or sculpture that are critical to the mood of the scene.

In Figure 9, we render two specular scenes with a number of lights at different positions. In particular, when the light is closer to the floor in Figure 9(a) (as compared to Figure 9(b)), shadows of both

---

[4]The incident lighting at the center of scene is used as the "environment map" and fed to all vertices for shading computations.

chairs expand sideways and the specularity on the table focuses. In Figures 9(d)–(e), we can clearly observe how editing the light texture, as well as its shape and size, changes the specular highlights on the tables and cushions. These effects are hard to capture with distant illumination. Close-ups are found in Figures 1(a) and 1(b).

5.4.2 *Light Editing.* We develop a prototype light editing and design system, which allows artists to edit the lights in a more intuitive and interactive way. An artist can move, resize, or horizontally rotate the light. Note that these operations on the light source are all performed once per frame in the pixel domain, before transforming the lighting into wavelets and propagating to the vertices. Lights can also be textured. Image processing methods such as blending, filtering, warping, and painting can be easily applied to edit the light texture. Because the cost of compressing the edited light texture into wavelets in real time is minor, changes can be immediately reflected in the realistically rendered images using our algorithm. For example, starting from Figure 9(d), we first paint and reshape the light to obtain Figure 9(e), and then resize and rotate the light to generate the image in Figure 1(a). Note the generality of our system to handle textured and editable/reshapeable light sources in Figures 8 and 9, which cannot be addressed by previous near-field lighting techniques like precomputed source radiance fields [Zhou et al. 2005].

5.4.3 *Accuracy Analysis and Validation.* We compare the ground-truth image from a ray tracer in Figure 8(a) with our result in Figure 8(b). We also showed a ground truth comparison in our didactic example of a knot scene in Figure 7. Since we only use a finite plane to parameterize the upper hemisphere, light incident at grazing angles is omitted, resulting in dimmer shading for some boundary vertices and the lack of grazing angle specularities. Also, lighting coefficients are compressed at both the source and target levels. Thus, energies at certain frequencies may be lost. Nevertheless, Figures 7 and 8 clearly demonstrate that our results show little visual difference from the ground truth, but now can be rendered at real-time rates. Similar results hold for our other images.

Lighting approximation error. Figure 10 shows the approximation errors for different compressions at both source and target levels. A few representative light textures used in the renderings are included in the accuracy analysis. Note that the horizontal axes in Figure 10 are on a log scale. The approximation accuracy improves quickly with increasing numbers of coefficients used. In addition, the target-level compression is less efficient than the source-level, requiring more coefficients for the same level of accuracy. This

Scenes	#Verts	Threads	Source=1% Target=1%	Source=1% Target=20%	Source=5% Target=40%	Source=20% Target=70%	Source=100% Target=100%
diffuse fertility	39,391	8 threads	151.86fps	38.71fps	24.83fps	13.78fps	7.04fps
		1 thread	32.79fps	7.58fps	4.17fps	2.17fps	1.08fps
specular chairs	56,995	8 threads	107.33fps	25.87fps	12.48fps	7.87fps	4.86fps
		1 thread	20.43fps	4.52fps	2.03fps	1.15fps	0.75fps
specular couches	68,046	8 threads	101.64fps	23.34fps	11.98fps	7.31fps	4.80fps
		1 thread	18.60fps	3.91fps	1.94fps	1.10fps	0.71fps

Fig. 11. Rendering performance with different compressions. For each scene, the upper row contains the performance numbers using the multi-thread implementation, and the lower row for the single-thread implementation. All performance is measured on a 3.0 GHz Dual-Quad-Core PC with 4.0 GB memory. The second and third performance columns correspond to realistic compression levels for accurate rendering, and achieve real-time rates for both the single-thread and multithread algorithms.

inefficiency is due to the fact that we rely on a constant heuristic light to predict significant lighting coefficients after the affine transformation. Our experiments show that 1% of the source-level and $20 \sim 30\%$ of the target-level lighting coefficients (using a constant light heuristic) usually suffice to generate visually accurate results.

5.4.4 *Performance*. All images are rendered at 1200×900 resolution and a wavelet order of 32. The rendering speed depends on a number of factors such as the total number of vertices, numbers of the retained source- and target-level lighting coefficients, and the complexities of the scene materials. In Figure 11, we report the rendering performance for the fertility, chair, and couch scenes. As shown in the second and third performance columns (which correspond to the common compression usage), our algorithm provides real-time performance, as also seen in the supplemental video that can be accessed through the ACM Digital Library.

5.5 Discussion and Comparison to Previous Near-Field Relighting Methods

In comparison with previous techniques, our method offers significant design flexibilities and achieves effects that are otherwise hard to capture. Annen et al. [2004] and Wang et al. [2006a] pioneered rendering midrange illuminations, using, respectively, spherical harmonic gradients and scaling. Lights are assumed some distance away from the scene so that the lighting can be smoothly interpolated and its propagation (affine transformation) approximated. Our work can be seen as extending these methods to near-field settings, as shown in Figure 8. Moreover, our technique can also render specular scenes, such as the chair and couch scenes in Figure 9. Zhou et al. [2005] and Kristensen et al. [2005] made important advances in near-field rendering of both diffuse and specular scenes. The light content, however, is built into the precomputations and has to remain static during rendering. Designers can move the light or change its intensity, but cannot edit the light shape or texture. Our method can be seen as an important generalization of their techniques, and allows light editing to be fully integrated with any PRT framework. For example, interactively painting the lights and changing their shapes, as done in Figures 8(d) and 9, or quickly flipping through several arbitrary light textures during relighting, as demonstrated in the video, are all feasible with our method. Finally, Annen et al. [2004], Wang et al. [2006a], and Kristensen et al. [2005] base their methods on the spherical harmonic basis and cannot capture all-frequency effects.

6. OTHER APPLICATIONS

Besides our main problem of PRT near-field relighting, affinely transformed wavelets and wavelet integrals have many other potential applications in rendering, image and signal processing, and nu-
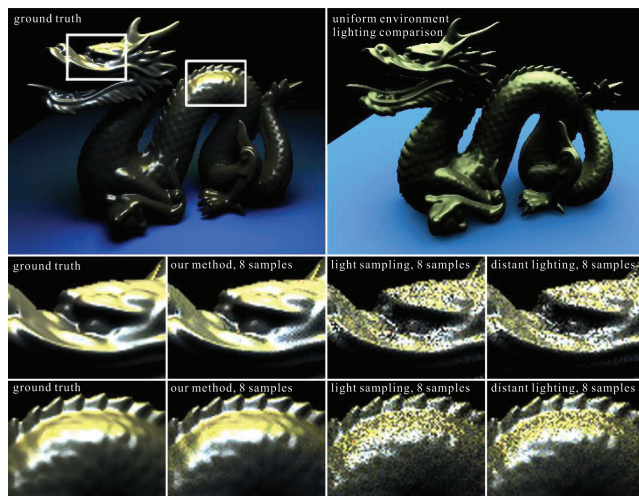
Fig. 12. A gold dragon under a near-field area source rendered using wavelet importance sampling. The ground-truth rendering is at the top left. To compare near-field and distant lighting effects, we also show an image rendered using uniform environment lighting at the top right. The color of the environment lighting is the average of the textured area light used in the near field. Shading variations and shadows are clearly different under the two lighting conditions. Under near-field lighting, the dragon exhibits high color contrast and strong directional specular reflections, while the shading under environment lighting is significantly diffused out. In the close-ups, for the same number of samples (8), we compare the result of our method with those by standard wavelet importance sampling but assuming that the light is distant (distant lighting is used to generate important samples, but shading is calculated using the area light), and light importance sampling (light sampling). Order 64 wavelets are used for all three sampling methods. Our method converges to the ground truth an order of magnitude faster, exhibiting substantially less noise at this sample count.

merical analysis. As a first proof of concept, we demonstrate initial examples of wavelet importance sampling with near-field lighting for offline Monte Carlo rendering, and image dilation and translation directly in the wavelet domain for image processing.

6.1 Wavelet Importance Sampling

Clarberg et al. [2005] have shown that importance sampling the product of the BRDF and the distant lighting can greatly reduce the variance in Monte Carlo rendering. However, their method is limited to distant environment map lighting, since lighting-BRDF products are computed using standard triple product integrals. Our theory enables a direct extension to the near-field setting, with planar area sources. We pretabulate BRDFs as 4D functions as done in Clarberg et al. [2005]. For each pixel, we affinely transform the original light source into the local incident radiance using Eq. (30) of Section 5, in a very similar fashion as for relighting. We then multiply the wavelet coefficients of the local radiance with those of the BRDF using the standard triple product wavelet integral [Ng et al. 2004]. Thereafter, we perform hierarchical wavelet warping to obtain importance samples, used for Monte Carlo estimation. Distinct from Clarberg et al. [2005], we use the standard wavelet decomposition and an expanded-top-face parameterization, as we do in Section 5 for relighting.

We demonstrate a near-field rendering of a gold dragon under an area light source in Figure 12. We observe visual effects such as highly contrasted and spatially varying shadings that are hard

| Original image before zooming | Pixel up-sampling time=0.121s | Our method, time=0.007s src=15%, dst=20% |

Our method, time=0.004s
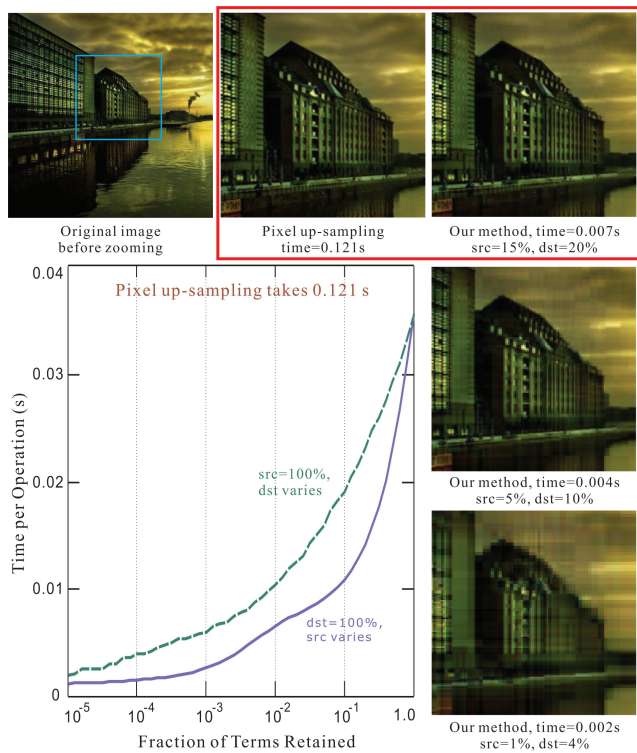src=5%, dst=10%

Our method, time=0.002s
src=1%, dst=4%

Fig. 13. Zooming in on a building in a picture of Berlin. The top left is the original image. The red box at the top contains zoomed images using up-sampling in the pixel domain and our method. All image resolutions are 512×512. Our method is visually as accurate as the pixel domain method, but more than 15 times faster. More results with different compressions at both the source and target levels (src and dst) are shown in the column beneath the red box. In the bottom left graph, we plot the performance curves of our method with different compressions. Each curve corresponds to either the source- or target-level compressions (one level per curve). The horizontal axis is on a log scale. Time measurements of the pixel method include decompression and recompression of images encoded in wavelets.

to obtain with distant lighting. In the close-ups, we compare the results of our method with those of standard wavelet importance sampling, but assuming the light is distant (distant lighting), and light importance sampling (light sampling). With only 8 samples per pixel, our method can significantly reduce the noise and generate results that are visually indistinguishable from the ground truth.

6.2 Image Dilation and Translation

Many image operations such as convolution, masking, and zooming involve basic operators like dilations (scales) and translations. Using our theory, images compressed using wavelets can be scaled and translated directly in the wavelet domain, instead of first requiring decompression into the pixel domain. If needed, for example, images are streamed from a remote server, and image dilations and translations can also be performed in a multiresolution fashion.

The basic equation for image scaling and translation remains the same as Eq. (30). $F(v)$, originally the area light, becomes the source image and $L(v)$, the incident radiance, becomes the target image. Variables z and x, respectively, describe the amount of dilation and translation. As before, compression of wavelet coefficients can be performed at both the source and target levels.

An example result is shown in Figure 13, where we zoom in on a building in a photograph of Berlin, and plot the performance curves with different compressions for the source and target. As shown in the red box, our method is visually as accurate as the brute-force up-sampling in the pixel domain, but more than 15 times faster. With more compression (two bottom right images), the speedup can be up to 60 times, at the cost of losing some details in the final images. Accuracy analysis can be referred to Figure 10 in Section 5. Along with other wavelet domain image operations such as additions and multiplications, we believe our work is a significant step towards a complete wavelet domain toolbox for image processing.

7. LIMITATIONS AND FUTURE EXTENSIONS

In this section, we address three important limitations of our model, and suggest future directions for extending our theory to general non-Haar wavelets and our relighting algorithm to out-of-plane rotation and 3D shaped lights. The implementation of these extensions is left as future work. We also briefly discuss the difficulty involved in nonlinearly transforming functions in wavelets and compare their complexities to that of the affine transform.

7.1 General Non-Haar Wavelets

In general, wavelet basis functions are defined as scaled and translated copies of their mother wavelets. By definition, they should have the same tree structure as the Haar wavelets. As a result, our analysis and derivation of the canonical coupling coefficient (CCC) in Section 4.2.2 for Haar holds for general wavelets. The affine transformation in general wavelets can be similarly reduced to a 2D analytic function $M(r, c)$. Since we only relied on the tree-structured and nonoverlapping properties of Haar wavelets, which are also true for general wavelets, in analyzing its ACC and ATC complexities (Appendix D), similar analysis quickly leads us to $O(n \log N)$ ACC and $O(n \log^2 N)$ ATC complexities for general wavelets. The constants in the complexities will be larger because of greater wavelet support. Moreover, depending on the structures of the mother wavelets, the properties of the CCCs for general wavelets such as sparsity, piecewise linearity, symmetry, and boundedness may also be different and their calculation is likely to be more expensive than that in Haar. However, our implementation in Haar wavelets, as explained in Section 5.3.2, is a general approach and can be similarly applied to general wavelets.

7.2 Relighting with Out-of-Plane Rotation and 3D Lights

There is a feasible approach to extend our technique to general out-of-plane rotation and arbitrarily shaped 3D lights. The theory and the algorithm we presented in Sections 4 and 5 allow us to propagate parallel planar lights to relight a scene. When the light is rotated out of the plane, we can first project the light back onto the original plane towards the shading vertex using the standard camera perspective projection model. We then propagate the projected light to the shading vertex using our theory of affine transformations. The perspective projection and affine transformation together yield the correct transformation for the rotated light. This approach has two more benefits. First, perspective projections can be quickly done on a GPU. To minimize texture read-back cost, multiple projected results can be packed into a larger texture and then read back all at once. Second, though the perspective projection depends on the shading vertex, we only need to sample the rotated light for a sparse set of angles to obtain an accurate estimation, for example, 10×10 sampling grid. Lighting in between the sampling points can be

interpolated. In essence, we create a sparse light field representation in the original plane for the rotated light. Relighting with a light field can be achieved by our theory and is described in detail in Appendix E. Using perspective projections and a sparse light field representation, we avoid the more expensive nonlinear transformation for out-of-plane rotation. The same projection approach can be extended to arbitrarily shaped 3D lights, such as a torus, an ellipsoid, or a shining buddha. The construction of the light field will slow down the rendering performance. However, the light field only needs to be constructed once per rotation. To further improve the speed, the sparse light field representation can be precomputed and tabulated for runtime lookups. In fact, light rotation around the vertical axis then is equivalent to "rotating" the set of projected textures in the azimuth angle.

7.3 Nonlinear Rotation in Wavelets

The out-of-plane rotation of lights actually corresponds to rotating the wavelet frame. For theoretical interest, we look at nonlinear rotations directly in the wavelet domain. Wavelet rotation is much more complex as it requires complicated coupling between the spatial and angular dimensions and their nonlinear transformations. We use a simple case as an example and assume a 1D light along the x-axis. Rotating the light around the zero point ($x = 0$) α degrees would achieve out-of-plane light rotation. Projecting the light back onto the x-axis yields a light field and requires computing $L(x, v) = F(\frac{x}{\cos\alpha + v\sin\alpha})$, where x is the spatial location and v is the intercept with the virtual parallel plane that is a unit distance away as before. When expanded in wavelets, the coefficients of the light field can be written as

$$
\begin{aligned}
L_{jk} &= \int L(x, v)\Phi_j(x)\Phi_k(v)dxdv \\
&= \int \left(\sum_i F_i \Phi_i \left(\frac{x}{\cos\alpha + v\sin\alpha} \right) \right) \Phi_j(x)\Phi_k(v)dxdv \\
&= \sum_i F_i R_{ijk}(x, v, \alpha),
\end{aligned}
\tag{33}
$$

where

$$
R_{ijk}(x, v, \alpha) = \int \Phi_i \left(\frac{x}{\cos\alpha + v\sin\alpha} \right) \Phi_j(x)\Phi_k(v)dxdv. \tag{34}
$$

The light field represented in wavelets can then be propagated using our theory. In this derivation, we clearly see that the tripling coefficient $R_{ijk}(x, v, \alpha)$ is much more involved and includes complicated coupling between x, v, and α. In general, if we denote a general nonlinear transformation as $T_n(x, v, \alpha)$, the nonlinear tripling coefficients in 1D can be written as

$$
R_{ijk}(x, v, \alpha) = \int \Phi_i(T_n(x, v, \alpha))\Phi_j(x)\Phi_k(v)dxdv. \tag{35}
$$

Depending on the transformation involved, these nonlinear tripling coefficients may or may not have analytic expressions. However, because of the tree structure of wavelets, we expect them to still be sparse and have complexities at roughly the same order as the ATC. In fact, Wang et al. [2006b] rotate an environment map over a sphere for relighting by computing coupling coefficients of a similar nature. Spherical harmonic scaling [Wang et al. 2006a] also relies on similar coefficients but computed in the spherical harmonic basis. As inspired by Wang et al. [2006a, 2006b], one possible approach for us to implement these nonlinear functions in wavelets is to precompute and tabulate $R_{ijk}(x, v, \alpha)$ sparsely, and then quickly interpolate their values at runtime. By studying the calculation and complexities

of more complicated coupling and tripling coefficients in different bases, we hope more general and complicated frequency and wavelet domain operations can be developed in the future.

8. CONCLUSIONS AND FUTURE WORK

We have presented a novel theory of affine double and triple product integrals. In particular, we have analyzed the sparsity of affine coupling and tripling coefficients in the Haar wavelet basis, showing that they have nearly linear complexity, which leads to efficient algorithms for computing affine transforms in Haar wavelets.

Besides being of substantial theoretical interest, our framework has important practical implications. We develop some of the first methods for including near-field lighting effects in all-frequency PRT algorithms. A planar area source can be translated, dilated, rotated in its plane, and have its texture edited, all while the scene is rendered in real time; local lighting is propagated to each vertex via an affine transform of the source radiance directly in wavelets. Our method can be viewed as an enabling tool that allows near-field lighting effects to be integrated with most PRT algorithms.

Future work in relighting can follow many avenues. Better heuristics can be experimented with for more efficient lighting compression. Relighting with more general illumination sources like light fields and dynamic textures can be explored. More general analytic local light models should be investigated.

Our theory can also be applied to many other applications that depend on wavelet representations. As a proof of concept, we demonstrate implementation of wavelet importance sampling with near-field lights for offline Monte Carlo renderers. We also show the practical utility of our theory in dilating and translating images directly in the wavelet domain for image processing. We believe that we have only scratched the surface of applications, and predict many future developments in graphics and applied mathematics.

Finally, our work reveals the fundamental sparsity of some basic wavelet operations (dilation and translation). This opens up a fresh perspective in approaching many other operations that have traditionally been viewed as difficult to compute directly with wavelets. Together with standard double- and triple product integrals (multiplication), more complicated operators can be built, for example, convolutions can be reduced to translations and multiplications. We hope the basic mechanism and computational machinery we employed in developing our theory will give some insight and serve as a stepping stone for constructing a more complete suite of wavelet domain operations.

APPENDICES

A. Proof of Lemma 1

By definition, the ATC is

$$
D_{ijk}(s, o) = \int \Phi_i(sv + o)\Phi_j(v)\Phi_k(v)dv. \tag{36}
$$

The integrand function $\Phi_i(sv + o)$ can be viewed as a function of s and o, and be expanded in basis $\Phi_l(v)$ as

$$
\begin{aligned}
\Phi_i(sv + o) &= \sum_l \left(\int \Phi_i(su + o)\Phi_l(u)du \right) \Phi_l(v) \\
&= \sum_l D_{il}(s, o)\Phi_l(v).
\end{aligned}
\tag{37}
$$

Plugging Eq. (37) into (36), we obtain

$$
\begin{aligned}
D_{ijk}(s, o) &= \int \left(\sum_l D_{il}(s, o)\Phi_l(v) \right) \Phi_j(v)\Phi_k(v)dv \\
&= \sum_l D_{il}(s, o) \int \Phi_l(v)\Phi_j(v)\Phi_k(v)dv \\
&= \sum_l D_{il}(s, o)C_{ljk}.
\end{aligned}
\tag{38}
$$

B. ACC and ATC in Pixel and Fourier Bases

Pixel basis. The ATC, according to Lemma 1, reduces to the ACC because in pixel basis C_{ijk} equals 1 when $i = j = k$, and otherwise is 0. As a result, the complexities of the ATC and ACC are the same, and we only need to analyze the ACC $D_{ij}(s, o)$. Calculating the ACC in the pixel basis is essentially equivalent to determining if the two pixels $\Phi_i(sv + o)$ and $\Phi_j(v)$ overlap. Without loss of generality, assume that the original pixel basis is defined in the range $[0, 1]$ and there are N pixel basis functions in total. Each pixel $\Phi_i(sv + o)$ covers an area of $1/(sN)$, and $\Phi_j(v)$ an area of $1/N$. Any given $\Phi_j(v)$, independent of the offset o, can overlap with at most $s + 1$ basis functions $\Phi_i(sv + o)$ due to the area constraint. Taking all $\Phi_j(v)$ into account, we obtain a maximum of $(s + 1)N$ overlapping pairs. Similarly, any given $\Phi_i(v)$ can overlap at most $s^{-1} + 1$ basis functions $\Phi_j(v)$. Considering all $\Phi_i(v)$ gives maximally $(s^{-1} + 1)N$ overlapping pairs. Combining these two maximums, we derive an upper bound $min\big((s + 1)N, (s^{-1} + 1)N\big)$ of the complexity, which can be proven tight easily. The tight upper bound $min\big((s + 1)N, (s^{-1} + 1)N\big)$ is a function of the scale s, and its maximum is $2N$ when $s = 1$. So we have proved the complexity of the ACC and ATC in the pixel basis is $O(N)$.

Fourier basis. The complex form ($I = \sqrt{-1}$) of 1D Fourier series basis functions on an azimuthal domain $[0, 2\pi]$ can be written as

$$
\phi_p(v) = (2\pi)^{-1/2}e^{Ipv}.
$$

Based on the 2D affine Fourier theorem [Bracewell et al. 1993], we recap the derivation for the 1D Fourier basis.[5]

$$
\begin{aligned}
D_{pq}(s, o) &= \int_0^{2\pi} \Phi_p(sv + o)\Phi_q^*(v)dv \\
&= e^{Ipo} \int_0^{2\pi} (2\pi)^{-1/2}e^{Ispv}(2\pi)^{-1/2}e^{-Iqv}dv \\
&= e^{Ipo}C_{sp,q} = e^{Ipo}\delta_{sp,q}
\end{aligned}
\tag{39}
$$

The ACC is mapped to a SCC. The specific mapping depends on the scale s and the offset o. Therefore same as the SCC, the complexity of the ACC in a 1D Fourier basis is $O(N)$.

To relate the ATC to the ACC, we start with Lemma 1.

$$
D_{npq}(s, o) = \sum_l D_{nl}(s, o)C_{lpq}
$$

Substituting D_{nl} using Eq. (39) and $C_{lpq} = \sqrt{\frac{1}{2\pi}}\delta_{l+p,q}$ [Ng et al. 2004], we obtain

$$
\begin{aligned}
D_{npq}(s, o) &= \sum_l e^{Ino}\delta_{sn,l}(2\pi)^{-1/2}\delta_{l+p,q} \\
&= (2\pi)^{-1/2}e^{Ino} \sum_l \delta_{sn,l}\delta_{l+p,q} \\
&= (2\pi)^{-1/2}e^{Ino}\delta_{sn+p,q},
\end{aligned}
\tag{40}
$$

where $\delta_{sn+p,q}$ is nonzero when $sn + p - q = 0$. Since one of the subscripts is uniquely determined by the other two, there are only two degrees of freedom as opposed to three. So the complexity of the ATC in the 1D Fourier basis is $O(N^2)$. All our analysis for the 1D Fourier basis can be easily extended to 2D and higher dimensions because high-dimensional Fourier bases are just products of multiple 1D bases.

C. Properties of the CCC

Piecewise linearity. We compute the partial derivatives of $M(r, c)$ with respect to c and r as follows.[6]

$$
\begin{aligned}
\frac{\partial M(r, c)}{\partial c} &= \frac{\partial}{\partial c} \int \psi\left(\frac{u}{2r} - \frac{c}{2r} + \frac{1}{2} \right)\psi(u)du \\
&= \frac{-1}{2r} \int \left(\delta\left(\frac{u}{2r}, \frac{c - r}{2r} \right) + \delta\left(\frac{u}{2r}, \frac{c + r}{2r} \right) \right. \\
&\quad \left. - 2\delta\left(\frac{u}{2r}, \frac{c}{2r} \right) \right)\psi(u)du \\
&= -\int \left(\delta(u, c - r) + \delta(u, c + r) - 2\delta(u, c) \right)\psi(u)du \\
&= 2\psi(c) - \psi(c - r) - \psi(c + r) \\
\frac{\partial M(r, c)}{\partial r} &= \frac{\partial}{\partial r} \int \psi\left(\frac{u}{2r} - \frac{c}{2r} + \frac{1}{2} \right)\psi(u)du \\
&= \frac{-1}{2r^2} \int \left(\delta\left(\frac{u}{2r}, \frac{c - r}{2r} \right) + \delta\left(\frac{u}{2r}, \frac{c + r}{2r} \right) \right. \\
&\quad \left. - 2\delta\left(\frac{u}{2r}, \frac{c}{2r} \right) \right)(u - c)\psi(u)du \\
&= \frac{-1}{r} \int (\delta(u, c - r) + \delta(u, c + r) - 2\delta(u, c))(u - c)\psi(u)du \\
&= \frac{-1}{r}(-r\psi(c - r) + r\psi(c + r)) \\
&= \psi(c - r) - \psi(c + r)
\end{aligned}
$$

Symmetry. We derive the symmetry properties of $M(r, c)$ in the c and r dimensions as follows.

$$
\begin{aligned}
M(r, c + 0.5) &= \int_{-\infty}^{\infty} \psi\left(\frac{u - c - 0.5}{2r} + \frac{1}{2} \right)\psi(u)du \\
&= -\int_{-\infty}^{\infty} \psi\left(\frac{u - c - 0.5}{2r} + \frac{1}{2} \right)\psi(1 - u)du \quad //\psi(u) = -\psi(1 - u) \\
&= \int_{\infty}^{-\infty} \psi\left(\frac{-w - c + 0.5}{2r} + \frac{1}{2} \right)\psi(w)dw
\end{aligned}
$$

$$
//\text{substitute } w = 1 - u
$$

---

[5]Since the Fourier basis is complex, complex conjugates of basis functions are used in ACCs and ATCs.

[6]Kronecker delta is denoted as $\delta(x, y)$ here which equals 0 when $x = y$.

$$= \int_{-\infty}^{\infty} \psi \left(\frac{w + c - 0.5}{2r} + \frac{1}{2} \right) \psi(w) dw$$
$$//\psi(0.5 + u) = -\psi(0.5 - u)$$

$$= M(r, 0.5 - c)$$

$$M(r, c) = \int \psi \left(\frac{u - c}{2r} + \frac{1}{2} \right) \psi(u) du$$

$$= 2r \int \psi(w) \psi \left(2r(w - 0.5) + c \right) dw$$
$$//\text{substitute } w = \frac{u - c}{2r} + \frac{1}{2}$$

$$= 2r M \left(\frac{1}{4r}, \frac{1}{2} + \frac{1}{4r} - \frac{c}{2r} \right)$$

D. ACC and ATC in Haar Wavelet Basis

ACC. In the Haar wavelet basis, the ACC is nonzero only when wavelets $\Phi_i(sv+o)$ and $\Phi_j(v)$ overlap with each other. Counting the number of nonzero ACC terms reduces to determining the number of overlapping wavelet pairs. Since wavelets are best organized in a tree structure, we denote the two wavelet trees using **I** and **J**. The basic structure of a wavelet tree is that two wavelets in the same tree will overlap only if they are in the same tree branch. We start by fixing a level and picking a wavelet for each wavelet tree: level l_i and wavelet $\Phi_i(sv + o)$ for tree **I**, and level l_j and wavelet $\Phi_j(v)$ for tree **J**. Since respectively at levels l_i and l_j there are 2^{l_i} and 2^{l_j} wavelets, the support of $\Phi_i(sv + o)$ and $\Phi_j(v)$ each must subtend $1/(s2^{l_i})$ and $1/(2^{l_j})$ in area. Any given wavelet $\Phi_j(v)$ at level l_j in tree **J** can overlap with at most $s \cdot 2^{l_i - l_j} + 1$ wavelets $\Phi_i(sv + o)$ at level l_i in tree **I** due to the area constraint. Summing over all levels in tree **I**, we get at most $\sum_{l_i}(s2^{l_i - l_j} + 1)$ overlapping pairs between the wavelet $\Phi_j(v)$ and wavelet tree **I**. Since the formula just derived is valid for any wavelet in tree **J**, we now sum over all wavelets $\Phi_j(v)$ in wavelet tree **J** and get a maximum of $\sum_{l_j} 2^{l_j} \sum_{l_i}(s2^{l_i - l_j} + 1)$ overlapping pairs between trees **I** and **J**. Similarly, we can start from a given wavelet $\Phi_i(sv + o)$ in tree **I** and obtain a maximum of $\sum_{l_i} 2^{l_i} \sum_{l_j}(2^{l_j - l_i}/s + 1)$ overlapping pairs between trees **I** and **J**. Combining these two maximums, we obtain an upper bound of $min(\sum_{l_i, l_j}(s2^{l_i} + 2^{l_j}), \sum_{l_i, l_j}(2^{l_j}/s + 2^{l_i}))$, which is tight for most s and o. The upper bound's maximum is $2N \log N$. So the complexity of the ACC in the Haar wavelet basis is $O(N \log N)$. Because of compression, only a small subset (n out of N) of wavelet coefficients suffice to generate accurate results, and thus the complexity reduces to $O(n \log N)$.

ATC. As discussed in Section 4.2.4, the ATC in the Haar wavelet basis reduces to an ACC when the nonzero segments of $\Phi_j(v)$ and $\Phi_k(v)$ overlap, or is 0 otherwise. For a fixed j, there are at most $O(\log N)$ overlapping pairs of wavelets $\Phi_j(v)$ and $\Phi_k(v)$ because to overlap k has to be either the same as j on the same level or on a higher level and subsumes j. Therefore the complexity of the ATC in the Haar basis is $O(n \log^2 N)$.

E. Relighting with Light Field

A general light field can be denoted as $F(v_1, v_2)$ and the local incident radiance $L(z, x; v)$ equals to $F(x + zv, v)$. Similarly, this relation can be written in wavelets using the ATC as

$$L_k(z, x) = \int L(z, x; v) \psi_k(v) dv$$
$$= \int F(zv + x, v) \psi_k(v) dv$$

$$= \int \left(\sum_i \sum_j F_{ij} \psi_i(zv + x) \psi_j(v) \right) \psi_k(v) dv$$
$$= \sum_i \sum_j F_{ij} D_{ijk}(z, x). \tag{41}$$

REFERENCES

ANNEN, T., KAUTZ, J., DURAND, F., AND SEIDEL, H.-P. 2004. Spherical harmonic gradients for mid-range illumination. In *Eurographics Symposium on Rendering*. 331–336.

BEYLKIN, G. 1992a. On the fast algorithm for multiplication of functions in the wavelet bases. In *International Conference on Wavelets and Applications*. 53–61.

BEYLKIN, G. 1992b. On the representation of operators in bases of compactly supported wavelets. *SIAM J. Numer. Anal. 29,* 6, 1716–1740.

BRACEWELL, R. N., CHANG, K.-Y., JHA, A. K., AND WANG, Y.-H. 1993. Affine theorem for two-dimensional fourier transform. *Electron. Lett. 29,* 3, 304.

CHAI, J.-X., CHAN, S.-C., SHUM, H.-Y., AND TONG, X. 2000. Plenoptic sampling. In *Proceedings of the SIGGRAPH International Conference on Computer Graphics and Interaction Techniques*, 307–318.

CLARBERG, P., JAROSZ, W., AKENINE-MÖLLER, T., AND JENSEN, H. W. 2005. Wavelet importance sampling: Efficiently evaluating products of complex functions. *ACM Trans. Graph. (SIGGRAPH'05 Proceedings) 24,* 3, 1166–1175.

DORSEY, J., ARVO, J., AND GREENBERG, D. 1995. Interactive design of complex time-dependent lighting. *IEEE Comput. Graph. Appl. 15,* 2, 26–36.

DRORI, I. AND LISCHINSKI, D. 2003. Fast multiresolution image operations in the wavelet domain. *IEEE Trans. Visualiz. Comput. Graph. 9,* 3, 395–411.

DURAND, F., HOLZSCHUCH, N., SOLER, C., CHAN, E., AND SILLION, F. X. 2005. A frequency analysis of light transport. *ACM Trans. Graph. (SIGGRAPH'05 Proceedings) 24,* 3, 1115–1126.

KRISTENSEN, A., MOLLER, T., AND JENSEN, H. W. 2005. Precomputed local radiance transfer for real-time lighting design. *ACM Trans. Graph. (SIGGRAPH'05 Proceedings) 24,* 3, 1208–1215.

KUTIL, R. 2005. Approximating linear image operations in the wavelet domain. Tech. rep. 2005-08, Department of Scientific Computing, University of Salzburg, Austria.

LEVOY, M. AND HANRAHAN, P. 1996. Light field rendering. In *Proceedings of the SIGGRAPH International Conference on Computer Graphics and Interaction Techniques*, 31–42.

LIU, X., SLOAN, P.-P. J., SHUM, H.-Y., AND SNYDER, J. 2004. All-Frequency precomputed radiance transfer for glossy objects. In *Euro-Graphics Symposium on Rendering'04*. 337–344.

MEI, C., SHI, J., AND WU, F. 2004. Rendering with spherical radiance transport maps. In *Eurographics'04*. 281–290.

NG, R., RAMAMOORTHI, R., AND HANRAHAN, P. 2003. All-Frequency shadows using non-linear wavelet lighting approximation. *ACM Trans. Graph. (SIGGRAPH'03 Proceedings) 22,* 3, 376–381.

NG, R., RAMAMOORTHI, R., AND HANRAHAN, P. 2004. Triple product wavelet integrals for all-frequency relighting. *ACM Trans. Graph. (SIGGRAPH'04 Proceedings) 23,* 3, 477–487.

NIMEROFF, J. S., SIMONCELLI, E., AND DORSEY, J. 1994. Efficient re-rendering of naturally illuminated environments. In *Eurographics Rendering Workshop*, 359–373.

RAMAMOORTHI, R., MAHAJAN, D., AND BELHUMEUR, P. 2007. A first order analysis of lighting, shading, and shadows. *ACM Trans. Graph. 26,* 1, 2.

REN, Z., WANG, R., SNYDER, J., ZHOU, K., LIU, X., SUN, B., SLOAN, P.-P., BAO, H., PENG, Q., AND GUO, B. 2006. Real-Time soft shadows in dynamic scenes using spherical harmonic exponentiation. *ACM Trans. Graph. (SIGGRAPH'06 Proceedings) 25,* 3, 977–986.

SLOAN, P.-P., KAUTZ, J., AND SNYDER, J. 2002. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. *ACM Trans. Graph. (SIGGRAPH'02 Proceedings) 21,* 3, 527–536.

STOLLNITZ, E. J., DEROSE, T. D., AND SALESIN, D. H. 1996. *Wavelets for Computer Graphics: Theory and Applications.* Morgan Kaufmann.

STRANG, G. 1989. Wavelets and dilation equations: A brief introduction. *SIAM Rev. 31,* 4, 614–627.

SUN, W. AND MUKHERJEE, A. 2006. Generalized wavelet product integral for rendering dynamic glossy objects. *ACM Trans. Graph. (SIGGRAPH'06 Proceedings) 25,* 3, 955–966.

WANG, R., TRAN, J., AND LUEBKE, D. 2004. All-Frequency relighting of non-diffuse objects using separable BRDF approximation. In *Eurographics Symposium on Rendering*, 345–354.

WANG, J., XU, K., ZHOU, K., LIN, S., HU, S., AND GUO, B. 2006a. Spherical harmonics scaling. *The Visual Comput. 22,* 9-11, 713–720.

WANG, R., NG, R., LUEBKE, D., AND HUMPHREYS, G. 2006b. Efficient wavelet rotation for environment map rendering. In *Eurographics Symposium on Rendering*. 173–182.

ZHOU, K., HU, Y., LIN, S., GUO, B., AND SHUM, H. 2005. Precomputed shadow fields for dynamic scenes. *ACM Trans. Graph. (SIGGRAPH'05 Proceedings) 24,* 3, 1196–1201.