Wavelet Importance Sampling:
Efficiently Evaluating Products of Complex Functions

Petrik Clarberg*   Wojciech Jarosz†   Tomas Akenine-Möller‡   Henrik Wann Jensen§
Lund University / UC San Diego   UC San Diego   Lund University   UC San Diego

Figure 1: *From left to right: BRDF importance sampling, environment map importance sampling, and importance sampling of the combination of BRDF and environment. With our new technique we can efficiently sample the product of the BRDF and the environment map without evaluating the full product. Sampling the product results in a superior sampling distribution (shown in the right image) compared with sampling the individual functions (shown on the left). 100 samples were used for each image. The BRDF is a measured acrylic blue material, shown for a single normal and viewing direction, and the environment map is Grace cathedral. Sample points for the product were generated in 0.1 milliseconds.*

Abstract

We present a new technique for importance sampling products of complex functions using wavelets. First, we generalize previous work on wavelet products to higher dimensional spaces and show how this product can be sampled on-the-fly without the need of evaluating the full product. This makes it possible to sample products of high-dimensional functions even if the product of the two functions in itself is too memory consuming. Then, we present a novel hierarchical sample warping algorithm that generates high-quality point distributions, which match the wavelet representation exactly. One application of the new sampling technique is rendering of objects with measured BRDFs illuminated by complex distant lighting — our results demonstrate how the new sampling technique is more than an order of magnitude more efficient than the best previous techniques.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Shading, G.1.2 [Numerical Analysis]: Approximation—Wavelets, G.1.4 [Numerical Analysis]: Quadrature and Numerical Differentiation—Multidimensional Quadrature

Keywords: importance sampling, wavelets, complex products, Monte Carlo techniques, global illumination, rendering

---

*e-mail: petrik@cs.lth.se
†e-mail: wjarosz@ucsd.edu
‡e-mail: tam@cs.lth.se
§e-mail: henrik@cs.ucsd.edu

1   Introduction

In many areas of science, the integral of two or more complex functions needs to be evaluated efficiently. In computer graphics, we have the rendering equation [Kajiya 1986], which contains a product of the incident lighting and the material properties of a given object. To evaluate this integral it is common to use Monte Carlo sampling to sample unknown elements of the integral such as visibility. Monte Carlo sampling relies on random sampling of the integral, and is a widely used method for evaluating complex functions. Unfortunately, Monte Carlo methods are computationally costly, and to increase the efficiency, it is necessary to include as much information about the integral as possible in the sampling process. For this purpose importance sampling is a powerful technique. The goal of importance sampling is to distribute samples according to the known elements of the space being sampled in order to reduce the variance due to these elements. In the case of the rendering equation, efficient techniques exist for distributing the samples according to the reflection model being used, or the incident lighting, but not according to both. In the case of known complex lighting and sophisticated reflection models, it would be much more powerful to distribute the samples according to the product of both as shown in Figure 1, but currently there is no efficient method for doing this.

In this paper, we introduce a novel method for importance sampling the product of two or more complex functions. Our algorithm uses wavelets to represent the functions under consideration. Given two functions represented in the Haar wavelet basis, it has recently been shown that the wavelet decomposition of the product can be directly computed using tripling coefficients [Ng et al. 2004]. As a first contribution, we generalize this theory to products of higher dimensional spaces. Second, we introduce a novel hierarchical sample warping scheme that can be folded into the wavelet product evaluation. The new sampling scheme provides high-quality sample distributions as shown in Figure 1, but more importantly it enables the sampling of a complex product without the need for evaluating the full product.

This makes the sampling very fast (it can be used on-the-fly during rendering), and it makes it possible to sample according to the product of high-dimensional functions for which the actual product would take up too much space to be practically useful. Our results demonstrate that the new sampling technique provides superior sampling and quality when rendering models with measured material properties illuminated by complex distant high-dynamic range lighting.

2 Previous Work

In this section, we review previous work on importance sampling in computer graphics. Most of the previous work uses importance sampling in the context of the rendering equation [Kajiya 1986]:

$$L(\mathbf{x}, \vec{\omega}_o) = L_e(\mathbf{x}, \vec{\omega}_o) + \int_\Omega f_r(\mathbf{x}, \vec{\omega}_i, \vec{\omega}_o) L_i(\mathbf{x}, \vec{\omega}_i) \cos \theta_i d\vec{\omega}_i,$$

which is fundamental for rendering realistic images. Here, we need to evaluate the product of the BRDF, f_r, the incident radiance, L_i, and a cosine-term.

BRDF importance sampling is an important and commonly used technique for increasing the efficiency of ray tracing based algorithms. Several important BRDF models can be directly importance sampled, including the Phong model [Shirley 1991], the Ward model [1992], and the Lafortune model [1997]. See Pharr and Humphreys [2004] for more examples. Complex BRDF models such as the Torrance-Sparrow model cannot be analytically inverted and require numerical approximations. Both the Ward model and the Lafortune model have been used to approximate measured BRDF data.

Other work has addressed the problem of importance sampling measured BRDFs. Lalonde [1997] used a wavelet representation of the BRDF and presented a novel importance sampling scheme based on random sampling of the wavelet tree. Similar methods were used in [Claustres et al. 2003; Claustres et al. 2004]. Matusik [2003] also used wavelets to represent BRDFs and he presented a numerical method for sampling the BRDF data. Recently, Lawrence et al. [2004] introduced a technique for sampling BRDFs based on a factored representation. They represent the BRDF in Rusinkiewicz's parameterization [1998], which is compact and compresses well, and their technique can be used for directly importance sampling a 4D BRDF efficiently. BRDF importance sampling is an effective technique, but these methods do not take into account the lighting in the scene, and they become inefficient with complex lighting.

Environment map sampling is another powerful method for rendering objects under complex lighting captured in a high-dynamic range environment map [Debevec 1998]. LightGen [Cohen and Debevec 2001], Agarwal et al. [2003], Kollig and Keller [2003], and Ostromoukhov [2004] all resampled the environment map by placing pre-integrated directional lights at the brightest locations. This is an efficient method for rendering non-specular materials illuminated by environment map lighting, but as the materials get increasingly specular these methods need a very large number of lights to adequately represent the environment map. Cabral [1987] and Ramamoorthi and Hanrahan [2002] used spherical harmonics to directly filter the environment map according to the BRDF — these methods do not support efficient sampling,

and the spherical harmonics representation is efficient only when the BRDF is smooth and non-specular.

Monte Carlo rendering has a long history in computer graphics starting with the seminal work by Cook et al. [1984] and Kajiya [1986]. There are numerous Monte Carlo techniques for solving the rendering equation. See Dutré et al. [2003] for an overview. Most Monte Carlo techniques use the BRDF sampling methods or the environment map sampling methods just described to solve the rendering equation. Some methods such as path tracing [Kajiya 1986; Lafortune and Willems 1995] and photon mapping [Jensen 2001] have been extended to importance sampling of the product of the BRDF and the lighting. Both approaches are based on the use of a coarse representation combined with adaptive sampling in order to evaluate the rendering equation. Their efficiency is limited by the coarse representation and they are too costly in the case of complex lighting and specular BRDF models. Veach and Guibas [1995] presented a novel technique for combining estimators in Monte Carlo methods using multiple importance sampling. Multiple importance sampling is a powerful method for addressing the situation where either the lighting or the BRDF is complex, as it will pick the best of the available sampling techniques. When both lighting and the BRDF are complicated, multiple importance sampling provides less of an advantage, as it cannot account for the product of the two. It is likely to waste samples in regions with little or no influence on the final result.

Recently, Burke et al. [2004] presented a novel technique for rendering objects with complex materials illuminated by an environment map. Their technique uses importance sampling of either the environment map or the BRDF and applies rejection sampling to discard samples if the product of the BRDF and the lighting is not large enough to motivate sampling. This helps reduce the number of samples, but the method is very costly due to the rejection sampling scheme. If both the BRDF and the lighting are complex, Burke et al. reports that more than 90% of the samples are rejected, requiring further evaluation of the functions to locate good samples.

In contrast to the previous work our method is capable of efficiently importance sampling the product of the lighting and the BRDF. In the following sections we will describe how this is done by using a compact wavelet representation combined with a generalized wavelet product, and a novel hierarchical sample warping scheme.

3 Overview

Figure 2 gives an overview of the elements of our new sampling technique for the particular example of sampling the product of an environment map and a BRDF. First, we take a high-quality point distribution. Second, we perform a hierarchical evaluation of the wavelet product. As this product is evaluated, the point distribution is warped hierarchically according to the evaluated product. The wavelet product is only completed for regions with one or more samples. The final output is a sample distribution that exactly matches the product of the wavelets without the need of evaluating the full product.

4 Wavelets and Wavelet Products

In this section, we first review the Haar basis notation, since it is used throughout the paper. Other bases can be used
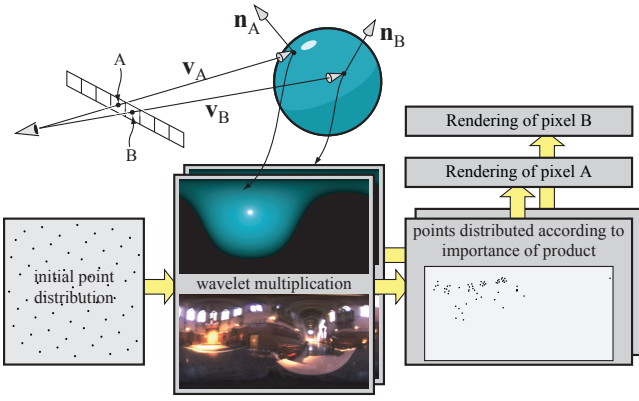
Figure 2: *This figure shows the main steps of our technique. An initial point distribution is warped according to the wavelet product of BRDF and environment map. For each pixel, a new well-distributed sampling is computed, taking both BRDF and environment map into account. The generated sample distribution will be different for two different pixels A and B, since the BRDF changes across the surface.*

as well, but the product tends to get much more complex. The wavelet product in two dimensions is described in Section 4.2. Finally, in Section 4.3, we present a new contribution: a generalized wavelet product, which works in higher dimensions.

4.1 The Haar basis

For the normalized Haar basis, the one-dimensional mother scaling function, $\phi(x)$, and the mother wavelet function, $\psi(x)$, are defined as [Mallat 1998; Stollnitz et al. 1996]:

$$\phi(x) := \begin{cases} 1, & \text{for } 0 \leq x < 1 \\ 0, & \text{otherwise} \end{cases}, \quad \psi(x) := \begin{cases} 1, & \text{for } 0 \leq x < 1/2 \\ -1, & \text{for } 1/2 \leq x < 1 \\ 0, & \text{otherwise.} \end{cases}$$

The normalized scaling and wavelet basis functions are:

$$\begin{aligned} \phi_t^l(x) &:= 2^{l/2}\phi(2^l x - t), \\ \psi_t^l(x) &:= 2^{l/2}\psi(2^l x - t), \end{aligned}$$

where l is the level and t the translation of the functions. The Haar basis is orthonormal, meaning that the inner product of two basis functions is zero except when they are the same, in which case the inner product is one.

Expanding a one-dimensional image, H, in the Haar basis is described as:

$$H(x) = H_{0,0}^0 \phi_0^0 + \sum_l \sum_t H_{t,1}^l \psi_t^l = \sum_\mathbf{i} H_\mathbf{i} \Psi_\mathbf{i}, \qquad (1)$$

where the second subscript for the basis coefficients, $H_{t,f}^l$, is zero for the scaling function, and one for the wavelet basis functions. Thus, $H_{0,0}^0$ is the scaling coefficient, and $H_{t,1}^l$ are the detail coefficients. The last step in Equation 1 shows the shorthand notation that we will use, where l, t, and the second subscript, indicating the type of basis function, have been "baked" into a single vector parameter $\mathbf{i}$, as done by Ng et al. [2004]. Note that this notation generalizes to higher dimensions, such as for two-dimensional images.

As can be seen in Equation 1, only a single scaling coefficient plus scaling function are used. However, as shown in
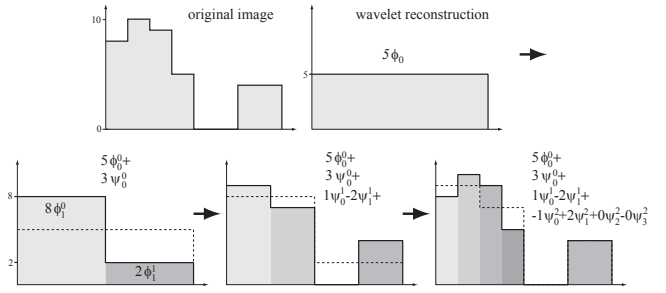

Figure 3: *The 1D image (upper, left) is:* $[8, 10, 9, 5, 0, 0, 4, 4]$, *and its unnormalized (used here because it is simpler to display) Haar representation is:* $[5, 3, 1, -2, -1, 2, 0, 0]$*. The image is then reconstructed one level at a time as follows:* $[5] \rightarrow [5+3, 5-3] = [8, 2] \rightarrow [8+1, 8-1, 2-2, 2+2] = [9, 7, 0, 4]$ *and so on.*

Figure 3, the scaling coefficients at other levels are computed as a part of the decompression process. The scaling coefficient for a certain level l and translation t holds the average of all pixels under the support of the scaling function. This is a key observation, also used by Lalonde [1997], that we will use in Section 5. In the example figure, the two scaling coefficients for, e.g., level 1 are 8 and 2.

It should also be noted that a good approximation is obtained if only the n largest coefficients in Equation 1 are kept. This provides lossy compression.

4.2 Two-Dimensional Wavelet Product

As a side result in their research on triple products, Ng et al. [2004] showed that for a product $G = E \cdot F$ of two-dimensional images, a wavelet representation of $G = \sum G_\mathbf{i} \Psi_\mathbf{i}$ can be directly computed from the wavelet representations of $E = \sum E_\mathbf{j} \Psi_\mathbf{j}$ and $F = \sum F_\mathbf{k} \Psi_\mathbf{k}$, that is, without decompressing them. We will briefly review the theory behind such *wavelet products* here.

The wavelet product we want to compute is described by:

$$G = E \cdot F \Leftrightarrow \sum G_\mathbf{i} \Psi_\mathbf{i} = \left(\sum E_\mathbf{j} \Psi_\mathbf{j}\right) \cdot \left(\sum F_\mathbf{k} \Psi_\mathbf{k}\right) \quad (2)$$

Taking the inner product of $\Psi_\mathbf{i}$ with the equation above yields the $\mathbf{i}^{\text{th}}$ basis coefficient for G:

$$\begin{aligned} G_\mathbf{i} &= \sum_\mathbf{j} \sum_\mathbf{k} C_{\mathbf{ijk}} E_\mathbf{j} F_\mathbf{k}, \text{ where} & (3) \\ C_{\mathbf{ijk}} &= \int \int \Psi_\mathbf{i}(\mathbf{x}) \Psi_\mathbf{j}(\mathbf{x}) \Psi_\mathbf{k}(\mathbf{x}) d\mathbf{x}. & (4) \end{aligned}$$

The terms $C_{\mathbf{ijk}}$ are called tripling coefficients, and the Ψ are two-dimensional basis functions.

4.3 Generalized Wavelet Product

In this section, we will generalize Equation 4, in order to compute the $\mathbf{i}^{\text{th}}$ basis coefficient of the product, $G = E \cdot F$, where G and E have n dimensions each, and F has m dimensions, and $0 < m \leq n$. Equation 4 and the related *Haar tripling coefficient theorem* [Ng et al. 2004] only work for $n = m = 2$. In the following, we assume that the non-standard decomposition technique [Stollnitz et al. 1996] is used.

For the generalized case, Equation 3 still holds, but the computation of tripling coefficients (Equation 4) is different and depends on m and n as shown below. Assume

we have the following vectors $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, $\overline{\mathbf{x}} = (x_1, x_2, \ldots, x_m)$, $m \leq n$, and that we want to compute the $\mathbf{i}^{\text{th}}$ basis coefficient of $G(\mathbf{x}) = E(\mathbf{x}) \cdot F(\overline{\mathbf{x}})$. The derivation can be found in Appendix A, and the main result is summarized in Equation 5.

$$
\begin{aligned}
C_{\mathbf{ijk}} &= \underbrace{\int \cdots \int}_{n} \Psi_{\mathbf{i}}(\mathbf{x}) \Psi_{\mathbf{j}}(\mathbf{x}) \Psi_{\mathbf{k}}(\overline{\mathbf{x}}) d\mathbf{x} \\
&= \prod_{q=1}^{m} c_{\alpha(\mathbf{ijk},q)} \times \prod_{p=m+1}^{n} \Delta_{\alpha(\mathbf{ij},p)}
\end{aligned}
\tag{5}
$$

Here, $c_{\alpha(\mathbf{ijk},q)}$ is used to denote a *one-dimensional* tripling coefficient, and the $\Delta_{\alpha(\mathbf{ij},p)}$ is, what we call, a one-dimensional *non-standard* coupling coefficient. The functions $\alpha(\mathbf{ijk}, q)$ and $\alpha(\mathbf{ij}, q)$ pick out the relevant parameters from $\mathbf{i}$, $\mathbf{j}$, and $\mathbf{k}$ for the q:th dimension. See again Appendix A for the details on notation, and on how to compute the non-standard coupling coefficients for the Haar basis.

To be able to evaluate Equation 5, we need the following theorem.

One-dimensional Haar Tripling Coefficient Theorem   *The integral, $c_{\alpha(\mathbf{ijk},q)}$, of three one-dimensional Haar basis functions is non-zero if and only if the support of the basis functions overlap and either of these two cases hold:*

1. *Two are identical basis functions, and the third basis function, at level l, is either a scaling function sharing their level, or the third basis function is at a strictly coarser level, $l \Rightarrow c_{\alpha(\mathbf{ijk},q)} = \pm 2^{l/2}$.*

2. *There is one scaling function at level l_1, and both the other basis functions are at strictly coarser levels, l_2 and $l_3 \Rightarrow c_{\alpha(\mathbf{ijk},q)} = \pm 2^{(l_2+l_3-l_1)/2}$.*

The proof of the theorem can be found in Appendix B. Applying this theorem twice to the two-dimensional case yields the same results as the two-dimensional Haar tripling coefficient theorem as expected.

Developing techniques for computing tripling coefficients in higher dimensions in the same manner as Ng et al. [2004] did for two dimensions appears to be time-consuming as each dimension would probably need a separate derivation and theorem. Instead, we have presented the fundamental result in Equation 5 together with the theorem above that allows for product computations in any dimensions without decompressing the wavelet images. In summary, the generalized tripling coefficients are computed as a simple product of one-dimensional tripling coefficients, and it has the same sublinear properties as the triple product for lossy approximations.

5   Importance Sampling

Assume an n-dimensional wavelet-compressed function:

$$
H = \sum_{\mathbf{i}} H_{\mathbf{i}} \Psi_{\mathbf{i}}
\tag{6}
$$

Sampling the wavelet requires computing the probabilities of different regions of the wavelet tree. We define these recursively such that the probabilities of the child regions at each wavelet node sum to 1. Using the scaling coefficients for a given region, the child probablities are defined as:

$$
P_{\mathbf{i}}^l = \frac{H_{\mathbf{i},0}^l}{\sum_{\mathbf{t}} H_{\mathbf{t},0}^l}
\tag{7}
$$

The simplest method of importance sampling a wavelet-represented function would be to treat the wavelet as a decision tree for hierarchical random thresholding. Previous work on sampling wavelets have used this method [Lalonde 1997; Claustres et al. 2003]. Another technique could be to ignore the hierarchical structure and randomly threshold according to values of the squares at the finest level. However, neither technique produces high quality point distributions, and thresholding according to the finest level requires the computation of the full wavelet product. In the following section, we present an alternative hierarchical warping algorithm that rapidly produces high-quality multi-dimensional sample distributions without having to evaluate a full wavelet product.

5.1   Hierarchical Warping

Our hierarchical warping technique transforms a uniformly distributed set of samples into a warped set of samples according to the wavelet tree. The warping algorithm begins at the coarsest level and proceeds recursively through each level of the wavelet hierarchy. One level of our warping algorithm in two dimensions is illustrated in Figure 5.

When warping multi-dimensional points, we consider each dimension individually. Starting with the first dimension, we split the input point set into two halves. For the following dimension, we split each of these new point sets into two new sets, and so on. The process can be thought of as building a kD-tree of the sample points.

To get the correct distribution when splitting the point set along a certain dimension, we need to compute the total probability for each half of that dimension. These are simply the sum of the child region probabilities within each half, call these probabilities $P_{\mathbf{i},x-}^l$ and $P_{\mathbf{i},x+}^l$, where $P_{\mathbf{i},x+}^l = 1 - P_{\mathbf{i},x-}^l$. To perform a split, we first divide the input sample points about a splitting plane positioned so that the lower set contains $P_{\mathbf{i},x-}^l$ fraction of the volume and the upper set contains $P_{\mathbf{i},x+}^l$. Next, both of these point sets are scaled to fit back within the unit interval. The procedure is then repeated independently on these two point sets, but along the subsequent dimensions until all points have been warped along each dimension.

At the completion of one level of n-dimensional warping, the algorithm recurses on all of the child regions which have

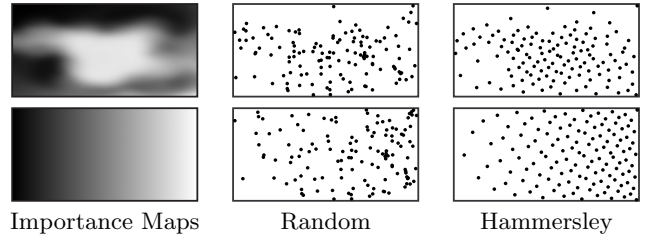| Importance Maps | Random | Hammersley |

Figure 4:   *Two example importance functions and 256 warped samples using uniform random points and Hammersley points. Our warping algorithm is able to preserve the quality of input point set.*
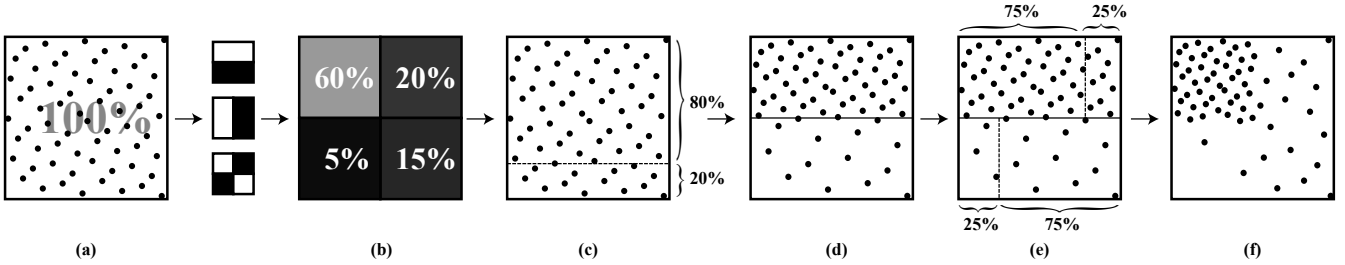
Figure 5: *Warping input points (a) according to one level of a wavelet-compressed importance map where the quadrant percentages (b) are derived from the wavelet coefficients for the current region using Equation 7. The initial point set is first partitioned into two rows with heights determined by their total probabilities (c) and then scaled to fit within the rows (d). Finally, each row is individually divided horizontally according to the probabilities of its child regions (e) and the points are again scaled to fit within the regions to arrive at the warped points for that level (f). The process repeats at step (a) for each child region using its allotted point set as input.*
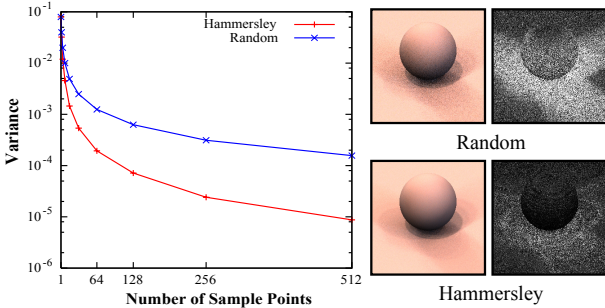

Figure 6: *Variance as a function of the number of samples used for rendering a simple scene illuminated by St. Peters cathedral. On the right are images rendered using 32 samples per pixel and their corresponding variance images. The warping scheme tends to preserve properties of the initial point distribution, hence the variance with Hammersley points is significantly lower than with uniform random points. Using 64 Hammersley points results in less variance than using 512 random points.*

at least one sample allocated. At each level of the hierarchical process, the expected number of points in each child region is proportional to the probability of that region. Hence, by induction, once recursion terminates the overall distribution of the warped points follows the energy distribution in the whole importance function.

Our warping algorithm is straightforward to use with low discrepancy points as generated by quasi-Monte Carlo methods [Niederreiter 1992]. Figure 4 demonstrates the effect that changing the input point set has on the quality of the output distribution. Consequently, our warping technique allows for variance reduction during rendering by using quasi-Monte Carlo sample points instead of uniform random points. This is shown in Figure 6.

5.2 Rapidly Sampling Wavelet Products

It is important to note that our sample warping technique only relies on the scaling coefficients at each level of the hierarchy. As shown in Section 4.2 and 4.3, coefficients of a wavelet product can be computed efficiently on-the-fly. In the Haar basis, it is trivial to reconstruct the necessary scaling coefficients from these wavelet coefficients. Hence, it is possible to rapidly warp points according to a wavelet product by computing product coefficients as needed.

In fact, our warping algorithm works particularly well for wavelet products because if no samples are placed in a particular region we do not have to further evaluate the product for that area. This gives significant savings because it

eliminates the need to compute coefficients for regions that are not being sampled. This is particularly true in our context where, for example, a highly specular BRDF is typically close to zero for a large portion of the integration domain.

6 Rendering

In this section, we will describe the application of wavelet-based importance sampling of products to the rendering of scenes with general BRDFs under complex direct illumination. The equation for evaluating direct illumination (derived from the rendering equation [Kajiya 1986]), is:

$$L(\mathbf{x}, \vec{\omega}_o) = \int_{\Omega} f_r(\mathbf{x}, \vec{\omega}_i, \vec{\omega}_o) L_i(\mathbf{x}, \vec{\omega}_i) v(\mathbf{x}, \vec{\omega}_i) \cos \theta_i d\vec{\omega}_i.$$

In our implementation, we are considering illumination $L_i(\mathbf{x}, \vec{\omega}_i)$ provided by a high-dynamic range environment map, and we fold the cosine term into the BRDF and work with the reflectivity, $\rho(\mathbf{x}, \vec{\omega}_i, \vec{\omega}_o)$. Therefore, a Monte Carlo estimator for the above integral can be written as:

$$\bar{L}(\mathbf{x}, \vec{\omega}_o) = \frac{1}{N} \sum_{i=1}^{N} \frac{\rho(\mathbf{x}, \vec{\omega}_i, \vec{\omega}_o) L_i(\mathbf{x}, \vec{\omega}_i) v(\mathbf{x}, \vec{\omega}_i)}{p(\vec{\omega}_i)}. \qquad (8)$$

By representing the BRDF and the environment map as wavelets, we can distribute samples according to their product, $H = \sum H_{\mathbf{i}} \Psi_{\mathbf{i}}$. The probability associated with a sample $\vec{\omega}_i$ is expressed in terms of wavelet scaling coefficients as:

$$p(\vec{\omega}_i) = c \frac{H_{\mathbf{t},0}^l}{H_{0,0}^0}, \qquad (9)$$

where $H_{\mathbf{t},0}^l$ is the scaling coefficient for the wavelet square in which the sample lies, and $c = 2^{nl/2}$ is a constant derived from the normalized Haar wavelet decomposition, assuming n-dimensional products.

Unbiased vs Biased Rendering We could use equation 8 to directly render an unbiased image by sampling the BRDF, environment map, visibility and divide by the sample probabilities. However, by accepting a small bias introduced by the wavelet approximation:

$$H_{\mathbf{t},0}^l \approx \rho(\mathbf{x}, \vec{\omega}_i, \vec{\omega}_o) L_i(\mathbf{x}, \vec{\omega}_i)/c, \qquad (10)$$

the rendering can be made significantly less noisy since two of the terms, which are a major source of variance, cancel out

in the division. Combining Equation 10 with Equations 8 and 9, we arrive at:

$$\bar{L}(\mathbf{x}, \vec{\omega}_o) \approx \frac{H_{0,0}^0}{N} \sum_{i=1}^{N} v(\mathbf{x}, \vec{\omega}_i). \qquad (11)$$

The scaling coefficient $H_{0,0}^0$ is already available, as it was needed during the warping step, and we are using the fact that it represents the pre-integrated value of the BRDF multiplied by the environment map. Thus, rendering is reduced to simple sampling of visibility. Areas with no occlusion will be noise-free since the pre-integrated value is known from the wavelet multiplication. In shadows, noise is inevitable, but since the samples are distributed according to the combination of the BRDF and the environment map, the sampling quickly converges towards a noise-free result.

6.1 2D Wavelet Importance Sampling

We represent a general 4D BRDF reparameterized about the reflected direction as in [Ramamoorthi and Hanrahan 2002]. This is stored as a 2D tabulation of 2D wavelets. Since the environment map has to be represented in the same coordinate space as the BRDF in order to perform a wavelet product, we replicate the 2D environment re-centered about an array of 2D directions. We found this representation more appropriate than having a 6D BRDF and a 2D environment map as in [Ng et al. 2004], since it stores less redundant data. Furthermore, most scenes contain many BRDFs but typically only one or a few environment maps.

With our representation, two of the dimensions overlap between the environment map and the BRDF, allowing us to perform wavelet importance sampling on the product in 2D. We represent functions on the sphere in 2D as simple spherical maps sampled over (θ, ϕ), but other parameterizations, such as cube maps, would also work. Note that if a non-uniform parameterization is used, the solid angle each pixel represents must be taken into account. To prevent aliasing, we use super-sampling when creating the wavelet representations.

To get a smooth result, we bilinearly interpolate between the four nearest wavelets in both the environment map and the BRDF. Compression is achieved by discarding wavelet coefficients below a certain threshold. We handle color by storing wavelet coefficients as RGB triplets instead of single values. When sampling according to RGB wavelets, we distribute samples according to the luminance and multiply each sampled value by the normalized color.

6.2 4D Wavelet Importance Sampling

In addition to representing BRDFs as tabulated 2D wavelets, we have done some initial experiments with a true 4D×2D wavelet product using the theory in Section 4.3. Here, we store the environment map as a regular 2D map, and work with simple rotationally symmetric BRDFs represented as 4D functions in world-space. The first two dimensions specify the central BRDF direction, and the last two dimensions represent outgoing light direction.

This technique allows us to distribute 4D sample points according to the wavelet product as a preprocess, and eliminates the need to perform on-the-fly wavelet products during rendering. After warping, the wavelets need no longer to be kept in memory, and the warped 4D points are stored in a kD-tree with 2D keys and 2D values per key for efficient range searching during rendering. At render time, we find
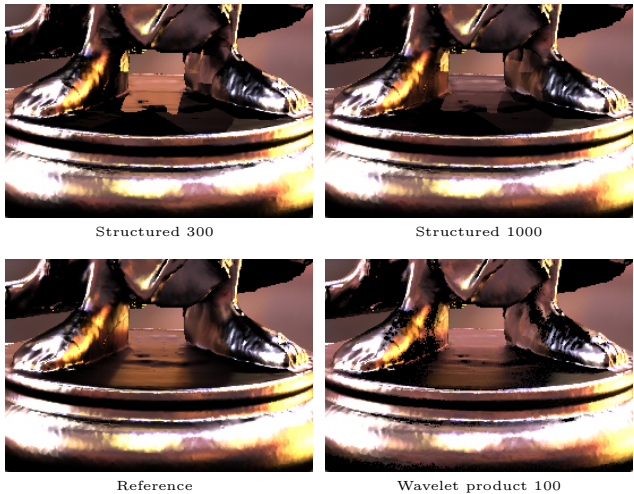

Figure 7: *Part of the Buddha rendered using structured importance sampling (without jittering) with 300 samples, 1000 samples, and wavelet sampling of the product using 100 samples per pixel and a wavelet resolution of 128×128. Even with a large number of samples, structured importance sampling fails to capture some of the more subtle details in the lighting. See for example the area between the model's feet.*

the n closest sample points in the first two dimensions, and use the last two dimensions of these sample points as world-space light sample directions. We also apply a weighting kernel to the chosen samples based on the deviation from the actual BRDF direction.

7 Results

This section demonstrates our results of the new sampling technique. All results have been generated on a 3.4GHz PC.

The first example, shown in Figure 8, is a rendering of a Buddha model with different measured BRDFs illuminated by a high-dynamic range environment map. From left to right, the selected BRDFs change from mostly diffuse to mostly specular. All images have been rendered with 30 visibility samples based on the product of the BRDF and the environment map. Note that the full range of BRDFs are practically noise-free, even with this low number of samples.

Previous work on rendering objects illuminated by high-dynamic range environment maps have focused on the sampling of either the environment map or the BRDF. In Figure 9, we compare our method to structured importance sampling [Agarwal et al. 2003]. We optimized the amount of jittering in structured importance sampling to reduce banding in the shadow, while avoiding excessive noise in the glossy reflection. We also compare our method to wavelet-based BRDF importance sampling, i.e., not using the wavelet product.

The wavelet product sampling technique produces images with low levels of noise at just 10–30 samples, while structured importance sampling needs 100 or more samples to produce similar results. In particularly difficult cases, even 1000 samples are not enough to accurately capture glossy effects (see Figure 7). BRDF importance sampling performs relatively well for the glossy Buddha, but very poorly for the diffuse floor, giving an overall quality worse than that of structured importance sampling. Performing the wavelet product sampling on-the-fly obviously adds some overhead. With our current implementation it is only possible to use

Figure 8: *The Buddha model in Grace cathedral rendered with different measured BRDFs. From left to right: latex fabric, silver paint, gold, blue metallic, brown wax, blue acrylic, green metallic, and copper. Wavelet importance sampling was performed on-the-fly, using 30 Hammersley samples per pixel. The BRDFs were stored with a wavelet sparsity of around 1.5%. The images were rendered in 53–66 seconds at resolution 400×800.*
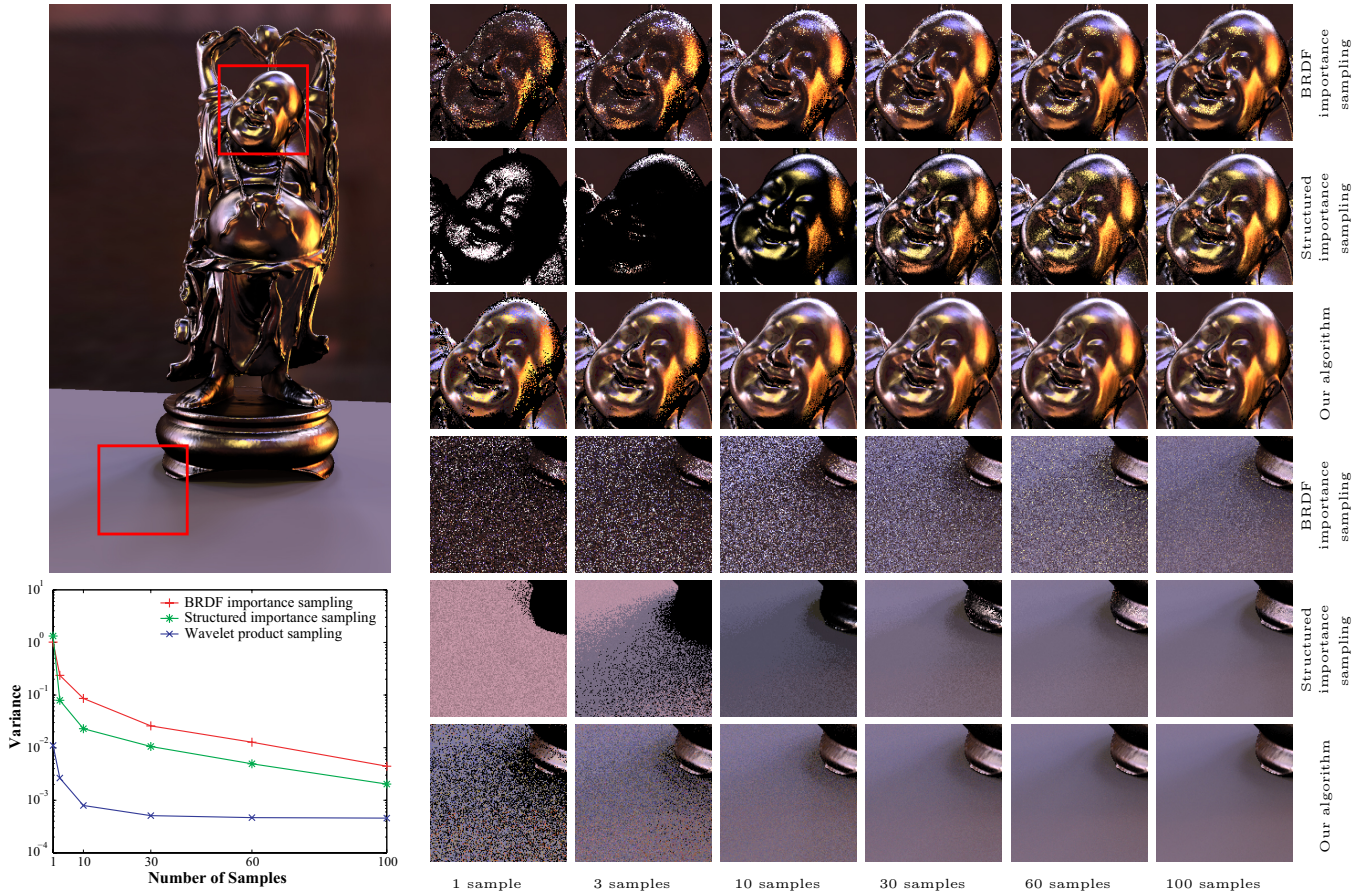

Figure 9: *Wavelet importance sampling of the product compared to structured importance sampling of the environment map (Grace cathedral) and to wavelet-based BRDF importance sampling with varying number of samples. From left to right: 1, 3, 10, 30, 60, 100 samples per pixel. The larger image on the left represents ground truth and was rendered using brute force ray tracing. The light directions generated by structured importance sampling were jittered to avoid banding in the shadows. Note that structured importance sampling does not work well with very few number of samples, whereas wavelet product sampling quickly gives a good approximation. BRDF importance sampling works relatively well for the glossy Buddha, but very poorly for the diffuse floor. The rendering times for structured importance sampling were between 4–103 seconds, and for wavelet product sampling between 15–205 seconds, using a wavelet resolution of 64×64. The variance plot in the lower left corner clearly shows that our algorithm outperforms the other two.*

approximately half the number of samples for equal rendering times. As the number of samples grows, this difference becomes smaller.

Figure 10 shows the effect of using a sparse wavelet representation of the BRDF. We have found that for most measured BRDFs, a wavelet resolution of 64×64, and about 2%

of the wavelet coefficients are enough to produce renderings indistinguishable from reference images. This combined with reparameterization, allows us to store general BRDFs in around 300kB (resolution $16 \times 16 \times 64 \times 64$) or in higher resolution ($32 \times 32 \times 128 \times 128$) using 5MB. The memory usage could be further optimized, but it is not a major ob-
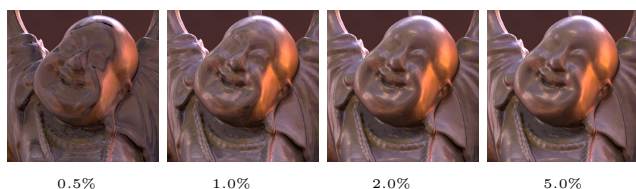
| 0.5% | 1.0% | 2.0% | 5.0% |

Figure 10: *Buddha with a measured BRDF (oxidized steel), rendered with different levels of BRDF compression. From left to right: 0.5%, 1%, 2%, and 5% sparsity. In practice, a sparsity of 1%–2% produces good results for a wide variety of materials.*
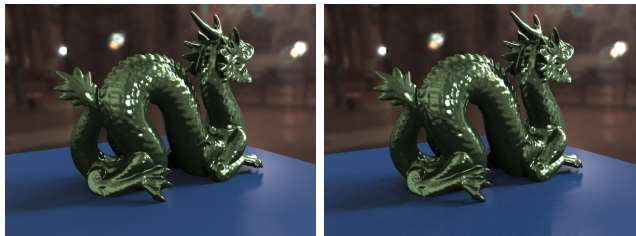

Figure 11: *Glossy dragon in Galileo's tomb rendered using 30 samples per pixel. The left image was rendered using on-the-fly 2D wavelet products and the right image used 4D wavelet products as a pre-process. The right image rendered 2.1 times as fast.*

stacle at this point. The computation time for creating the BRDFs is a few minutes, and is not included in the rendering times. For the pre-rotated lighting environment, we use a denser sampling of 64^4 or 128^4, as the lighting is typically more rapidly varying. In our implementation, the wavelet environment maps can take up to a couple of hours to create, mainly due to extensive super-sampling.

Figure 11 compares a dragon scene rendered using sample points warped on-the-fly by a tabulated set of 2D wavelet products and the same scene rendered using sample points warped as a pre-process by a full 4D×2D wavelet product. Both versions used 30 samples per pixel, but since no wavelet products were evaluated during render time with the 4D version, it rendered a factor 2.1 times faster.

Figure 12 shows a complex scene with 12 different measured BRDFs applied to two car models, illuminated by the eucalyptus grove light probe. This shows that with our technique, it is possible to render complex scenes under realistic lighting conditions with very few number of samples. The sampling handles highly glossy materials equally well as more diffuse ones.

8   Conclusions and Future Work

We have presented a new general tool for importance sampling products of complex functions. Our technique is not limited to a specific subset of functions, nor is it limited by the dimensionality of the functions. As an example we used the evaluation of the rendering equation, considering general BRDFs under direct illumination by an environment map. Wavelet importance sampling of the BRDF times the environment map proved to give superior sample distributions, enabling us to render essentially noise-free images using as few as 30–100 samples per pixel.

There are many possible applications for wavelet importance sampling of products. For example, our technique could be used for importance sampling of 4D surface light fields [Miller et al. 1998; Lalonde and Fournier 1999] mul-

tiplied by general BRDFs, or even by 6D bidirectional texture functions [Dana et al. 1999]. Another new application is importance sampling over the time domain for rendering animations more efficiently. For example, the product of a time-varying environment map and a BRDF could be used to generate samples that are well distributed in both time and space. We are confident our work will be useful in a wide variety of problems involving importance sampling of complex functions.

In the future, we are planning to apply wavelet importance sampling to some of these problems, and further investigate the effect of different parameters on the result. A deeper more theoretical analysis of how the properties of the initial point distribution are affected by the warping would be useful.

References

AGARWAL, S., RAMAMOORTHI, R., BELONGIE, S., AND JENSEN, H. W. 2003. Structured Importance Sampling of Environment Maps. *ACM Transactions on Graphics 22*, 3, 605–612.

BURKE, D., GHOSH, A., AND HEIDRICH, W. 2004. Bidirectional Importance Sampling for Illumination from Environment Maps. In *ACM SIGGRAPH Technical Sketches*.

CABRAL, B., MAX, N., AND SPRINGMEYER, R. 1987. Bidirectional Reflection Functions from Surface Bump Maps. In *Computer Graphics (Proceedings of ACM SIGGRAPH 87)*, 273–281.

CLAUSTRES, L., PAULIN, M., AND BOUCHER, Y. 2003. BRDF Measurement Modelling using Wavelets for Efficient Path Tracing. *Computer Graphics Forum 22*, 4, 701–716.

CLAUSTRES, L., BOUCHER, Y., AND PAULIN, M. 2004. Wavelet Projection for Modelling of Acquired Spectral BRDF. *Optical Engineering 43*, 10, 2327–2339.

COHEN, J., AND DEBEVEC, P., 2001. LightGen, HDRShop plugin. http://www.ict.usc.edu/ jcohen/lightgen/lightgen.html.

COOK, R. L., PORTER, T., AND CARPENTER, L. 1984. Distributed Ray Tracing. In *Computer Graphics (Proceedings of ACM SIGGRAPH 84)*, 137–145.

DANA, K. J., VAN GINNEKEN, B., NAYAR, S. K., AND KOENDERINK, J. J. 1999. Reflectance and Texture of Real-World Surfaces. *ACM Transactions on Graphics 18*, 1, 1–34.

DEBEVEC, P. 1998. Rendering Synthetic Objects into Real Scenes: Bridging Traditional and Image-Based Graphics with Global Illumination and High Dynamic Range Photography. In *Proceedings of ACM SIGGRAPH 98*, 189–198.

DUTRÉ, P., BEKAERT, P., AND BALA, K. 2003. *Advanced Global Illumination*. A K Peters.

JENSEN, H. W. 2001. *Realistic Image Synthesis Using Photon Mapping*. A K Peters.

KAJIYA, J. T. 1986. The Rendering Equation. In *Computer Graphics (Proceedings of ACM SIGGRAPH 86)*, 143–150.

KOLLIG, T., AND KELLER, A. 2003. Efficient Illumination by High Dynamic Range Images. In *Eurographics Symposium on Rendering*, 45–50.

LAFORTUNE, E. P., AND WILLEMS, Y. D. 1995. A 5D Tree to Reduce the Variance of Monte Carlo Ray Tracing. In *Eurographics Workshop on Rendering*, 11–20.

LAFORTUNE, E. P. F., FOO, S.-C., TORRANCE, K. E., AND GREENBERG, D. P. 1997. Non-Linear Approximation of Reflectance Functions. In *Proceedings of ACM SIGGRAPH 97*, 117–126.

Figure 12: *Scene with 12 different measured BRDFs illuminated by the eucalyptus grove at UC Berkeley. The product of environment map and BRDF was sampled on-the-fly using 10 samples per pixel. The image was rendered in 804 seconds using ray tracing, at a resolution of 4000×1600.*

LALONDE, P., AND FOURNIER, A. 1999. Interactive Rendering of Wavelet Projected Light Fields. In *Proceedings of Graphics Interface '99*, 107–114.

LALONDE, P. 1997. *Representations and Uses of Light Distribution Functions*. PhD thesis, University of British Columbia.

LAWRENCE, J., RUSINKIEWICZ, S., AND RAMAMOORTHI, R. 2004. Efficient BRDF Importance Sampling using a Factored Representation. *ACM Transactions on Graphics 23*, 3, 496–505.

MALLAT, S. 1998. *A Wavelet Tour of Signal Processing*. Academic Press.

MATUSIK, W., PFISTER, H., BRAND, M., AND MCMILLAN, L. 2003. A Data-Driven Reflectance Model. *ACM Transactions on Graphics 22*, 3, 759–769.

MILLER, G. S. P., RUBIN, S. M., AND PONCELEON, D. B. 1998. Lazy Decompression of Surface Light Fields for Precomputed Global Illumination. In *Eurographics Workshop on Rendering*, 281–292.

NG, R., RAMAMOORTHI, R., AND HANRAHAN, P. 2004. Triple Product Wavelet Integrals for All-Frequency Relighting. *ACM Transactions on Graphics 23*, 3, 477–487.

NIEDERREITER, H. 1992. *Random Number Generation and Quasi-Monte Carlo Methods*. Society for Industrial and Applied Mathematics.

OSTROMOUKHOV, V., DONOHUE, C., AND JODOIN, P.-M. 2004. Fast Hierarchical Importance Sampling with Blue Noise Properties. *ACM Transactions on Graphics 23*, 3, 488–495.

PHARR, M., AND HUMPHREYS, G. 2004. *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann.

RAMAMOORTHI, R., AND HANRAHAN, P. 2002. Frequency Space Environment Map Rendering. *ACM Transactions on Graphics 21*, 3, 517–526.

RUSINKIEWICZ, S. M. 1998. A New Change of Variables for Efficient BRDF Representation. In *Eurographics Workshop on Rendering*, 11–22.

SHIRLEY, P. S. 1991. *Physically Based Lighting Calculations for Computer Graphics*. PhD thesis, University of Illinois at Urbana-Champaign.

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

VEACH, E., AND GUIBAS, L. J. 1995. Optimally Combining Sampling Techniques for Monte Carlo Rendering. In *Proceedings of ACM SIGGRAPH 95*, 419–428.

WARD, G. J. 1992. Measuring and Modeling Anisotropic Reflection. In *Computer Graphics (Proceedings of ACM SIGGRAPH 92)*, 265–272.

A Multi-Dimensional Wavelet Product

In this appendix, we will show how to derive Equation 5. Recall that we want to compute the $\mathbf{i}^{\text{th}}$ basis coefficient of the product, $G = E \cdot F$, where G and E have n dimensions each, and F has m dimensions, and $0 < m \leq n$. Furthermore, assume we have the following vectors $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, $\overline{\mathbf{x}} = (x_1, x_2, \ldots, x_m)$, $m \leq n$. For an orthonormal basis, the $\mathbf{i}^{\text{th}}$ basis coefficient of $G(\mathbf{x}) = E(\mathbf{x}) \cdot F(\overline{\mathbf{x}})$ is computed by projecting G onto the $\mathbf{i}^{\text{th}}$ basis function:

$$G_{\mathbf{i}} = \underbrace{\int \cdots \int}_{n} \Psi_{\mathbf{i}}(\mathbf{x}) G(\mathbf{x}) d\mathbf{x} = \int \cdots \int \Psi_{\mathbf{i}}(\mathbf{x}) E(\mathbf{x}) F(\overline{\mathbf{x}}) d\mathbf{x}$$

$$= \int \cdots \int \Psi_{\mathbf{i}}(\mathbf{x}) \left(\sum_{\mathbf{j}} E_{\mathbf{j}} \Psi_{\mathbf{j}}(\mathbf{x}) \right) \left(\sum_{\mathbf{k}} F_{\mathbf{k}} \Psi_{\mathbf{k}}(\overline{\mathbf{x}}) \right) d\mathbf{x}$$

$$= \sum_{\mathbf{j}} \sum_{\mathbf{k}} \left(E_{\mathbf{j}} F_{\mathbf{k}} \int \cdots \int \Psi_{\mathbf{i}}(\mathbf{x}) \Psi_{\mathbf{j}}(\mathbf{x}) \Psi_{\mathbf{k}}(\overline{\mathbf{x}}) d\mathbf{x} \right)$$

$$= \sum_{\mathbf{j}} \sum_{\mathbf{k}} C_{\mathbf{ijk}} E_{\mathbf{j}} F_{\mathbf{k}}, \text{ where}$$

$$C_{\mathbf{ijk}} = \underbrace{\int \cdots \int}_{n} \Psi_{\mathbf{i}}(\mathbf{x}) \Psi_{\mathbf{j}}(\mathbf{x}) \Psi_{\mathbf{k}}(\overline{\mathbf{x}}) d\mathbf{x}. \qquad (12)$$

Note that $\Psi_{\mathbf{i}}$ and $\Psi_{\mathbf{j}}$ are n-dimensional basis functions, and $\Psi_{\mathbf{k}}$ is an m-dimensional basis function. The above reasoning works for an arbitrary orthonormal basis even though the focus of our work is on the normalized Haar basis. A crucial insight to generalizing Haar wavelet products to higher dimensions is that higher dimension Haar basis functions are separable. For example, in the two-dimensional case this means:

$$\Psi_{\mathbf{i}} = \Psi_{\mathbf{t},\mathbf{f}}^l(x_1, x_2) = \begin{cases} \phi_{t_1}^l(x_1) \psi_{t_2}^l(x_2), & \text{if } \mathbf{f} = 01, \\ \psi_{t_1}^l(x_1) \phi_{t_2}^l(x_2), & \text{if } \mathbf{f} = 10, \\ \psi_{t_1}^l(x_1) \psi_{t_2}^l(x_2), & \text{if } \mathbf{f} = 11. \end{cases} \qquad (13)$$

Here, the index $\mathbf{i}$ includes all information from l, $\mathbf{t}$, and $\mathbf{f}$. Also, note that $\mathbf{t} = (t_1, t_2)$ is a vector of translations, and $\mathbf{f} = (f_1, f_2)$ is an 2-bit vector that determines which combination of basis functions should be used. For n dimensions, this generalizes to vectors of n elements. A n-dimensional image, H, is then described as $H = \sum_{\mathbf{i}} H_{\mathbf{i}} \Psi_{\mathbf{i}}$ (see Equation 1). To simplify notation, we introduce the following function:

$$\chi_{\alpha(\mathbf{i},q)}(x_q) := \begin{cases} \phi_{t_q}^l(x_q), & \text{if } f_q = 0, \\ \psi_{t_q}^l(x_q), & \text{if } f_q = 1, \end{cases} \tag{14}$$

where $\alpha(\mathbf{i}, q) = (l, t_q, f_q)$, i.e., it picks out the parameters related to the q:th dimension of $\mathbf{i}$. Thus, our shorthand for Equation 13 becomes $\Psi_{\mathbf{i}}(x_1, x_2) = \chi_{\alpha(\mathbf{i},1)}(x_1)\chi_{\alpha(\mathbf{i},2)}(x_2)$. This reasoning generalizes to higher dimensions as well.

The n-dimensional Haar basis functions can be written as products of one-dimensional scaling or wavelet functions: $\Psi_{\mathbf{i}}(\mathbf{x}) = \prod_{q=1}^{n} \chi_{\alpha(\mathbf{i},q)}(x_q)$, where q is simply a dimension index. This separability property is used below to prove that the tripling coefficient from Equation 12 can be computed as a product of one-dimensional integrals:

$$
\begin{aligned}
C_{\mathbf{ijk}} &= \underbrace{\int \cdots \int}_{n} \Psi_{\mathbf{i}}(\mathbf{x}) \Psi_{\mathbf{j}}(\mathbf{x}) \Psi_{\mathbf{k}}(\overline{\mathbf{x}}) d\mathbf{x} \\
&= \prod_{q=1}^{m} \left(\underbrace{\int \chi_{\alpha(\mathbf{i},q)}(x_q) \chi_{\alpha(\mathbf{j},q)}(x_q) \chi_{\alpha(\mathbf{k},q)}(x_q) dx_q}_{\text{1D tripling coefficient}} \right) \times \\
&\quad \prod_{p=m+1}^{n} \left(\underbrace{\int \chi_{\alpha(\mathbf{i},p)}(x_p) \chi_{\alpha(\mathbf{j},p)}(x_p) dx_p}_{\text{1D } \textit{non-standard} \text{ coupling coefficient}} \right) \\
&= \prod_{q=1}^{m} c_{\alpha(\mathbf{ijk},q)} \times \prod_{p=m+1}^{n} \Delta_{\alpha(\mathbf{ij},p)}.
\end{aligned} \tag{15}
$$

In the last line in the equation above, $c_{\alpha(\mathbf{ijk},q)}$ is used to denote a one-dimensional tripling coefficient, and $\Delta_{\alpha(\mathbf{ij},p)}$ is, what we call, a one-dimensional *non-standard* coupling coefficient. Similar to before, the functions $\alpha(\mathbf{ijk}, q)$ and $\alpha(\mathbf{ij}, q)$ pick out the relevant parameters from $\mathbf{i}$, $\mathbf{j}$, and $\mathbf{k}$ for the q:th dimension. For the Haar basis, $\Delta_{\alpha(\mathbf{ij},p)} = 1$ if the corresponding two basis functions, identified by $\alpha(\mathbf{i}, p)$ and $\alpha(\mathbf{j}, p)$, are exactly the same[1]. If the two basis functions are overlapping and the finest of the basis functions is a scaling function, then $\Delta_{\alpha(\mathbf{ij},p)} = \pm 2^{(l_1 - l_2)/2}$, where l_1 and l_2 are the levels of the two involved basis functions and $l_1 < l_2$. The sign is determined by the signs of the basis functions where they overlap.

B Proof of 1D Haar Tripling Coefficient Theorem

For this proof, we assume that the normalized Haar basis is used, and for that the wavelet basis functions have vanishing integrals. In the following, support of the three basis functions must overlap, otherwise $c_{\alpha(\mathbf{ijk},q)} = 0$.

Case 1: All basis functions at the same level Since the support of all basis functions at the same level are disjoint, the three basis functions must share the same translation, t. Due to orthonormality, the product of two identical basis functions must be a constant function, $2^{l/2} \times 2^{l/2} = 2^l$ with the same support as the terms in the product. The integral of a constant function times the third basis function is zero if the third basis function is a wavelet function due to vanishing integrals of the wavelets. If the third basis function is a scaling function, thus with height $2^{l/2}$, then $c_{\alpha(\mathbf{ijk},q)} = 2^l \times 2^{l/2} \times 2^{-l} = 2^{l/2}$, where l is the level of the three basis functions and 2^{-l} is the width of the basis functions.

Case 2: Exactly two basis functions at the same level We first assume that the basis functions sharing level are at a finer level than the third basis function. The third function will therefore be constant over the support of the two functions sharing level. Thus, the two functions sharing level need to be exactly the same due to orthonormality. Using similar reasoning as for case 1, the tripling coefficient becomes $c_{\alpha(\mathbf{ijk},q)} = \pm 2^{l/2}$, where l is the level of the third basis function (at a coarser level), and the sign is determined by the sign of the third basis function where the finer functions overlap. In all other cases, $c_{\alpha(\mathbf{ijk},q)} = 0$.

Next, we assume that the basis functions sharing level are at a coarser level than the third basis function. The product of the two coarser basis functions will be constant over the support of the finer basis function. Hence, due to vanishing integrals, $c_{\alpha(\mathbf{ijk},q)} = 0$, if the third basis function is a wavelet function. If the third basis function is a scaling function, then $c_{\alpha(\mathbf{ijk},q)} = \pm 2^{l_1/2} \times 2^{l_2/2} \times 2^{l_3/2} \times 2^{-l_1} = \pm 2^{(l_2 + l_3 - l_1)/2}$, where l_1 is the finest level, and $l_2 = l_3$ is the level of the basis functions sharing level.

Case 3: All three basis functions at different levels Due to orthonormality, the product of the two coarsest basis functions must be a constant function $\pm 2^{l_2/2} \times 2^{l_3/2}$ over the support of the finest basis function. Again, due to vanishing integrals, $c_{\alpha(\mathbf{ijk},q)} = 0$ if the basis function at the finest level is a wavelet function. Using similar reasoning to case 2, $c_{\alpha(\mathbf{ijk},q)} = \pm 2^{l_1/2} \times 2^{l_2/2} \times 2^{l_3/2} \times 2^{-l_1} = \pm 2^{(l_2 + l_3 - l_1)/2}$ when the basis function at the finest level is a scaling function. This concludes the proof.

---

[1] This is the same case as for the standard coupling coefficient, i.e., a Kronecker delta.