Factoring of Multiple Function Wavelet Production Sampling

ZHONGWEI CHEN, KUN GAO, GUOWEI HUANG
School of Computer Science and Information Technology
Zhejiang Wanli University
Ningbo, Zhejiang 315100
P. R. CHINA
http://www.zwu.edu.cn

Abstract: - Importance sampling methods of wavelet products can deal with some direct rendering applications with only two functions. For multiple functions sampling that are more useful in global rendering, we present a wavelet-based factor method to simplify multiple function integral into triple function product issues. Then an optimal wavelet product representation is introduced. Major algorithms with pseudo code are in detail to reduce computation by a strategy for hierarchically sampling a wavelet tree.

Key-Words: - Monte Carlo Integral; Wavelet Product; Importance Sampling; Dynamic Rendering

1 Introduction

Many interesting developments have occurred in the field of realistic rendering. Many researchers have worked to develop realistic rendering algorithms that can run efficiently in arbitrarily complex scenes [1][2]. The use of a good sampling strategy for illumination is critical when integrating image-based lighting, such as environment maps, into a rendering system [3]. The problem of efficient sampling of the illumination is compounded when the scene contains materials with high frequency BRDFs [4][5]. High fidelity images based on a whole range of reflection phenomena described by the rendering equation often take hours or days to compute [6].

The performance can be improved if we incorporate knowledge about the function being integrated into the sampling process. The idea is to concentrate samples to parts of the function where it is likely to be large. This technique is called *importance sampling*, and can vastly reduce the variance in Monte Carlo techniques [7].

Several researchers have recently worked on this problem, by either combining samples drawn independently according to the importance of the illumination and the BRDF [8][9], or more recently, by drawing samples from the product distribution of the illumination and the BRDF [10]. These approaches produce high quality images with a small number of samples in unoccluded regions.

Light source sampling and BRDF sampling. Many importance sampling techniques for direct lighting concentrate on either sampling the light source or the BRDF. For example, Burke [11] described two methods to distribute samples according to the brightness of an environment map, based on cdf inversion and the alias method. Other research has tackled direct lighting from environment maps by approximating the illumination with a set of point lights (sampling directions) [12] [13]. More recently, Debevec presented a simple technique to generate point lights to approximate environment lighting using a summed area table of the environment map [14]. Our algorithm also uses a summed area table, but we are able to approximate the product distribution, not just the incident light term.

Methods that generate samples based solely on illumination do not work well for highly specula surfaces. In many situations it is more efficient to sample according to the BRDF function rather than the incident illumination. Some analytical BRDFs can be directly importance sampled, including the Blinn model [1], the Ward model [15], the Lafortune model [16], and the Ashikhmin model [17]. Besides analytical BRDFs, importance sampling can be employed with sampled BRDFs. Lawrence, Rusinkiewicz and Ramamoorthi [18] described a factored, tabular representation for BRDFs that is both compact and amenable to importance sampling. Other new intelligent way also developed in sampling way by Teh, C. S. [19].

Sampling a product distribution. Sampling according to one of the terms of the lighting equation can reduce variance, but it is more advantageous to generate samples based on all of the terms, rather than just one. Multiple importance sampling (MIS) can sample the BRDF and lighting simultaneously, but the resulting distribution is more akin to the average of the terms rather than the product [20]. Ideally, an importance sampling algorithm should be able to generate samples according to the product distribution.

Burke, Ghosh and Heidrich [5] described a technique called bidirectional importance sampling (BIS) that can sample the product of an environment map and the BRDF, based on rejection sampling. The rejection sampling can be costly, however, and requires an unknown number of tries to produce samples from the product distribution. A second form of BIS that can produce samples in a deterministic amount of time replaces rejection sampling with re-sampling, but the resulting samples are only approximately distributed according to the product. Talbot, Cline and Egbert [21] generalized this second form of BIS, placing it into the more general category of resampled importance sampling (RIS). Resampling methods can also be costly, however, since they rely on taking a large number of tentative samples, most of which will be discarded.

Direct rendering applications with wavelet sampling. Recently, Clarberg et al. [22] presented an algorithm called Wavelet Importance Sampling (WaIS) that samples products of wavelet functions. Their algorithm uses a property of wavelets that allows a wavelet product to be evaluated in a top-down fashion. However, WaIS addressed only aspects of direct illumination and static ones. Prayoth K. and Kitti A.[23] studied the comparative performance of multiwavelet-based image watermaking schemes.

Sun, W. [24] designed an interactive lighting design system, providing real-time feedback with realistic all-frequency shadows for dynamic glossy objects. Their system supports interactive manipulation of objects (such as cloning, translating and scaling) with all-frequency shadows. The systemalso allows the designer to adjust environment lighting and change view conditions. The system captures all-frequency view-dependent lighting effects, such as intricate cast shadows from neighboring objects and specular highlights. Their approach is based on the pre-computed information of individual scene entities, and is not limited to the pre-animated models. The approach is flexible.

Discussion. Among these techniques, wavelets sampling exhibit most of the required features. Indeed, an interesting property is the compact support of most wavelets, leading to a fast local reconstruction (logarithmic time according to the number of samples). The discrete wavelet transform produces the same number of coefficients as samples in the original dataset, but many of them are close to zero. Flexible lossy compression is obtained by zeroing those that are below a certain threshold. At last, wavelets can handle all-frequency lighting and shadowing effects. For these reasons we propose a optimized way for wavelet production sampling for and implement the reconstruction and filtering stages on the fragment processor of GPU, hence providing BRDF-based local illumination with both high-quality and real-time rendering. We also profit from the multi-resolution (reconstruction at different levels of accuracy) .

Our contribution. In this paper, we represented an improvement of WaIS for real-time rendering with dynamic objects under global illumination (situation as *figure 1c*). We explicitly incorporate dynamic occlusions into the shading integral to account for cast shadows from neighboring objects. At each vertex, shading is formulated as the product integral of multiple functions, involving the lighting, BRDF, local visibility and global occlusions. We show that multi-function product integral in the primal corresponds to the summation of the product of integral coefficients and basis coefficients. The major contribution of our work is efficiently integrating the product of n functions (n > 3). In the paper, we employee a novel generalized Haar integral coefficient theorem to evaluate arbitrary Haar integral coefficients, which work together with the wavelet importance sampling to render dynamic global objects efficiently.

2 Product importance sampling for direct rendering

2.1 Global illumination rendering equation

As Kajiya described [6], high fidelity images based on a whole range of reflection phenomena described by the rendering equation as:

$$L_o(x,\vec{\omega}_o) = L_e(x,\vec{\omega}_o) + \int_{\Omega} f_r(x,\vec{\omega}_i,\vec{\omega}_o) L_i(x,\vec{\omega}_i)\cos(\theta_i) d\vec{\omega}_i \quad (1)$$

The equation can be split into several components:

$$L_o(x,\vec{\omega}_o) = L_e(x,\vec{\omega}_o) + L_e(x,\vec{\omega}_o) + L_e(x,\vec{\omega}_o) \quad (2)$$

where L_{dir} is the direct illumination, L_{ind} is the indirect illumination, and L_e is the self-emitted radiance.

2.2 Monte Carlo integration and sampling

A common approach to evaluate the global lighting equation (1) is to use Monte Carlo integration, which replaces the continuous integral with the average of N Monte Carlo samples.

Monte Carlo integration is a probabilistic method for integrating difficult functions. It is commonly used in Global Illumination, in preference to more standard integration techniques like numerical quadrature, due to its generality, ease of use, and robustness in high dimensions and to discontinuous functions.

Monte Carlo integration is based on the fact that the integral of f can be approximated with the following estimator:

$$\int_{\Omega} f(\omega)d\omega \approx \frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)}{q(x_i)} \qquad (3)$$

where the samples $\{x_1, ..., x_N\}$ are drawn from the sampling distribution with pdf (probability density function) q [7]. Note that q must be normalized. This estimator provides an unbiased estimate of the integral. Also, the estimator works for general, non-continuous, high dimensional integrals. It only requires that f be evaluated. This generality makes Monte Carlo integration ideal for Global Illumination where the integrands are seldom well-behaved and are almost always high-dimensional.

The variance of the Monte Carlo integration estimator is

$$V\left(\sum_{i=1}^{N}\frac{f(x_i)}{q(x_i)}\right)=\frac{1}{N}V\left(\frac{f(\omega)}{q(\omega)}\right) \qquad (4)$$

we see that the variance of the Monte Carlo integration estimator is dependent on the variance of the ratio $f(\omega)/q(\omega)$.

If the variance of this ratio can be decreased, the overall variance can be reduced without increasing the number of samples.

Importance sampling refers to the technique of choosing the sampling distribution q to minimize the variance of the ratio. Ideally, if $q \propto f$, then the ratio is a constant for all w. In this case, the variance is zero and there is no error in the estimate. Unfortunately, finding such a q is typically impractical, since it requires integrating f, which is the very problem we are trying to solve. Instead, we try to find a sampling distribution that mimics f. We are greatly limited in our choices of distributions since the distribution must be normalized and it must be easy to generate samples from the distribution.

Although importance sampling can achieve remarkable results, it suffers from a major shortcoming: the sampling density, q, must be chosen specifically for each f. This means that, according to our definition, importance sampling is not robust enough.

2.3 Direct illumination and BRDF

Here, the direct illumination is given by the integral:

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

(5)

where the incident radiance, $L(x,\vec{\omega}_i)$, is provided by light sources in the scene, and $v(x,\vec{\omega}_i)$ is the visibility of a light source in direction $\vec{\omega}_i$. In order to apply realistic lighting to a virtual scene, it is common to capture real lighting in a high-dynamic range environment map [25], and use that for L during rendering.

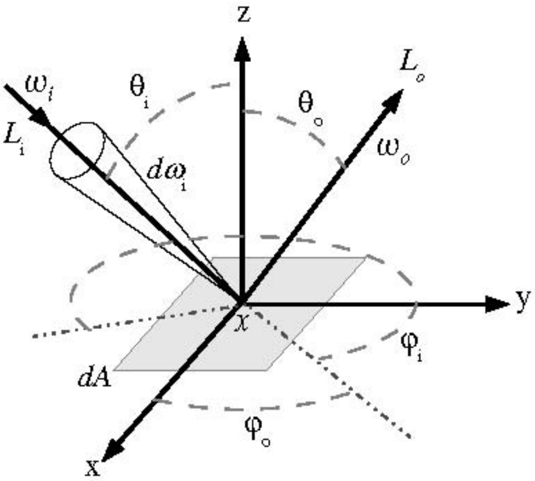

Fig. 1: The bidirectional reflectance distribution function (BRDF) describes the ratio of outgoing radiance to incident radiance (irradiance).

In order to deal with the direct illumination items in equation (1), the BRDF (bidirectional reflectance distribution function) was first formally defined by Nicodemus et al. [34]. The BRDF characterizes the reflection of light on a surface. In radiometric terms, the BRDF is the surface radiance divided by the surface irradiance, i.e., the incident light flux per unit illuminated surface area:

$$f_r(x,\vec{\omega}_i,\vec{\omega}_o)=\frac{dL_o(x,\vec{\omega}_o)}{dL_i(x,\vec{\omega}_i)}=\frac{dL_o(x,\vec{\omega}_o)}{L_i(x,\vec{\omega}_i)\cos(\theta_i)d\vec{\omega}_i} \quad (6)$$

Where x is the surface position, $\vec{\omega}_i$ is theincident direction, $\vec{\omega}_o$ is the outgoing (viewing) direction, $L_i(x,\vec{\omega}_i)$ is outgoing radiance, and $L_o(x,\vec{\omega}_o)$ is incident radiance (irradiance). See Fig.1 for an illustration.

Theoretically, the BRDF $f_r(x,\vec{\omega}_i,\vec{\omega}_o)$ also depends on other variables such as wavelength and polarization of the light, but we are usually considering only unpolarized light of one specific wavelength at the time. Therefore, the BRDF can be written as a four-dimensional function. The function is typically parameterized over the spherical coordinates for the incident direction $\vec{\omega}_i = (\theta_i, \varphi_i)$ and outgoing direction $\vec{\omega}_o = (\theta_o, \varphi_o)$.

2.4 Product importance sampling estimator

Many importance sampling techniques for direct lighting concentrate on either sampling the light source or the BRDF. BRDF importance sampling is better suited for specular materials, while environment map importance sampling is better for diffuser BRDFs.

To address this problem, Veach and Guibas [3] presented a novel technique for combining estimators in Monte Carlo methods using multiple importance sampling, which is a powerful method for the case where either the lighting or the BRDF is complex, as it will pick the best of the available sampling techniques. However, when both the lighting and the BRDF are complicated, their technique provides a smaller advantage. Burke et al. [5] introduced a technique for rendering objects with complex materials illuminated by an environment map.

In their work, the aim is to perform importance sampling using the *product* of the incident light distribution and the BRDF as the importance function:

$$p(\vec{\omega}_i) = \frac{f_r(\vec{\omega}_i,\vec{\omega}_o)L_i(\vec{\omega}_i)\cos(\theta_i)}{\int\limits_{\Omega} f_r(\vec{\omega}_i,\vec{\omega}_o)L_i(\vec{\omega}_i)\cos(\theta_i)d\vec{\omega}_i} \qquad (7)$$

Observe that the normalization term in the denominator is the direct illumination integral with the visibility term $v(\vec{\omega}_i)$ omitted. In other words, this term is the exitant radiance in the absence of shadows. Burke et al. refer to it as L_{ns} : radiance, no shadows.:

$$L_{ns} := \int\limits_{\Omega} f_r(\vec{\omega}_i,\vec{\omega}_o)L_i(\vec{\omega}_i)\cos(\theta_i)d\vec{\omega}_i \qquad (8)$$

If sample directions $\vec{\omega}_i^{(j)}$ are drawn according to the product distribution in Equation 8, then Equation 7 can be used to estimat $L_{N,p}(\vec{\omega}_o)$ as:

$$\begin{aligned} & L_{N,p}(\vec{\omega}_o) \\ & = \frac{1}{N}\sum_{j=1}^{N}\frac{f_r(\vec{\omega}_i^{(j)},\vec{\omega}_o^{(j)})L_i(\vec{\omega}_i^{(j)})v(\vec{\omega}_i^{(j)})\cos(\theta_i)}{p(\vec{\omega}_i^{(j)})} \qquad (9) \\ & = \frac{L_{ns}}{N}\sum_{j=1}^{N}v(\vec{\omega}_i^{(j)}) \end{aligned}$$

$L_{N,p}(\vec{\omega}_o)$ is referred to as the *bidirectional estimator* for the direct illumination integral.

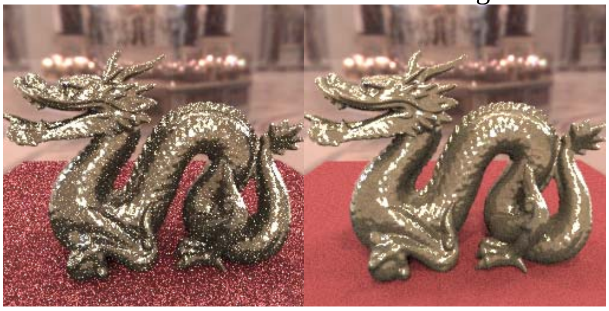

Fig.2: Dragon model in an indoor HDR EM. Left: Importance sampling from BDRF, 200 samples/pixel. Right: Bidirectional importance sampling.

As showed in Fig.2, note that the variance of the bidirectional estimator for the reflected radiance is proportional to the variance in the visibility function, $\vec{\omega}_i^{(j)} \propto p(\vec{\omega}_i), j = 1,...,N$. This is an improvement over sampling techniques that only consider either the illumination or the BRDF in the sampling process. This is because the variance of these techniques depends in addition on the variance in the function that they do not sample from, BRDF or illumination respectively.

3 Factoring of multifunction product integral for dynamic rendering

Although the previous works on above described techniques improved the static rendering a lot, it is more practical to deal with dynamic rendering situations such as much more objects in the environment, various lights , even just positional changes of the same object in the same scene.

3.1 Multi-function product integral

Consider equation 5 as the situation with only on static object, then given *n* distinct objects in a dynamic scene, the exitant radiance *B* at a surface point *x* along view direction *θ* due to distant environment lighting *L* is the product integral over all incident directions sampled at a surrounding cubemap [24]:

$$
\begin{aligned}
&B(x,\vec{\omega}_o)\\
&=\int_\Omega L(\vec{\omega}_i)v_1(x,\vec{\omega}_i)\prod_{j=2}^{n} v_j(x,\vec{\omega}_i)f_r(x,\vec{\omega}_i,\vec{\omega}_o)(\vec{N}\cdot\vec{\omega}_i)d\vec{\omega}_i \qquad (10)\\
&=\int_\Omega L(\vec{\omega}_i)\tilde{v}_1(x,\vec{\omega}_i)\prod_{j=2}^{n} \tilde{v}_j(x,\vec{\omega}_i)f_r(x,\vec{\omega}_i,\vec{\omega}_o)d\vec{\omega}_i
\end{aligned}
$$

Where $\vec{\omega}_i$ is the incident direction, $\vec{N}$ is the normal at x, f_r is the BRDF, v_1 is the *local visibility* at x due to self-occlusion. $v_i(2\leq j\leq n)$ is the *dynamic occlusion* at x occluded by the j^{th} neighboring object in the scene. In order to eliminate the dependance of the BRDF on the normal, the cosine term $(\vec{N}\cdot\vec{\omega}_i)$ is combined with the self visibility v_1 as $\tilde{v}_1$ as:

$$\tilde{v}_1(x,\vec{\omega}_i)=v_1(x,\vec{\omega}_i)(\vec{N}\cdot\vec{\omega}_i) \qquad (11)$$

For a fixed vertex x and view direction $\vec{\omega}_o$, equation (10) can be simplified as:

$$B=\int L(\vec{\omega}_i)\tilde{v}_1(x,\vec{\omega}_i)\prod_{j=2}^{n}\tilde{v}_j(x,\vec{\omega}_i)f_r(x,\vec{\omega}_i)d\vec{\omega}_i \qquad (12)$$

It is exactly the product integral of (n+2) functions :

$$B=\int\prod_{j=1}^{n+2}L_j(\vec{\omega}_i)d\vec{\omega}_i \qquad (13)$$

where $L_j(\vec{\omega}_i)$ is described in equation 5.

3.2 Factoring for dynamic radiance transfer

For dynamic radiance transferring, an effective approach to accelerating the evaluation of equation (12) is stated as follows:

$$
\begin{aligned}
B&=\int\left[\prod_{j=1}^{n+1}L_j(\vec{\omega}_i)\right]L_{n+2}(\vec{\omega}_i)d\vec{\omega}_i \qquad (14)\\
&=\left\langle T,L_{n+2}(\vec{\omega}_i)\right\rangle
\end{aligned}
$$

where the radiance transfer vector T is the product of n+1 functions as:

$$T=\prod_{j=1}^{n+1}L_j(\vec{\omega}_i) \qquad (15)$$

If L_1, L_2, · · ·, L_{n+1} are fixed, in other words, only L_{n+2} varies (i.e., dynamic instead of static), radiance transfer vector T needs to be computed only once. Therefore, shading integral reduces to a simple double function product integral of T and L_{n+2}, which can be approximated by the wavelet importance sampling method. Here we assume that only one function in the shading integral varies. This assumption is reasonable for lighting design systems, where normally the designer adjusts only one variable at a time, and real-time feedback is highly appreciated. For example, the designer may experiment with different lighting effects by fixing view conditions and objects. The designer may also render the scene from different view conditions by fixing the lighting and the objects. Another popular operation is to fix the lighting and view conditions, and relocate a single object in the scene. As long as there is only one (note that it can be any one) varying parameter, this approach can be used to generate all-frequency shadows in real-time.

In equation (4), the product of n+2 functions is factored into the product of two sets, one with n+1 functions, and the other with only one function. More generally, this factorization has the following form:

$$
\begin{aligned}
B&=\int\left[\prod_{j=1}^{k}L_j(\vec{\omega}_i)\right]\cdot\left[\prod_{j=k+1}^{n+2}L_j(\vec{\omega}_i)\right]d\vec{\omega}_i \qquad (16)\\
&=\left\langle T_1,T_2\right\rangle
\end{aligned}
$$

where $T_1=\prod_{j=1}^{k}L_j(\vec{\omega}_i)$

and $T_2=\prod_{j=k+1}^{n+2}L_j(\vec{\omega}_i)$.

As a result, the product of n+2 functions reduces to the double function product integral of two radiance transfer vectors.

Once factoring possible, we can use all previous works on importance sampling of double functions integral to dynamic rendering. This is our first contribution.

In our work, the wavelet ways are selected and improved to supply the efficient computing of real dynamic rendering. And that is the second contribution described in next section.

4 Optimal product importance sampling with wavelet

We focus on the efficient computation of the multi-function product integral and the product of multiple functions now. For this purpose, we choose Haar bases as the basis set ***B***.

Compared with the pixel domain representation, wavelets allow us to approximate signals at low distortion with a small number of significant coefficients. Haar bases have an interesting property that simplifies the computation as many of the integral coefficients are zero [26].

4.1 2D Haar bases

Nonstandard Haar wavelet transform decomposes a $2^n \times 2^n$ image into a 2D signal with $2^n \times 2^n$ coefficients. In each region <*j*, *k*, *l*>, four normalized 2D Haar basis functions are defined:

$\phi^j_{t_1,t_2}$ as *normalized Haar scaling basis function:*

$$\phi^j_{t_1,t_2}(x,y) = 2^j \phi^0(2^j x - t_1, 2^j y - t_2)$$

where ϕ^0 is the *mother scaling function*.

$\psi^j_{t_1,t_2}$ as *normalized Haar wavelet basis function*.

There are three types of wavelets defined in the region <*j*, *k*, *l*>:

$$\psi^j_{m,t_1,t_2}(x,y) = 2^j \psi^0_m(2^j x - t_1, 2^j y - t_2), m = 1,2,3$$

where $\psi^0_m, m = 1,2,3$, are three different mother wavelets, denoting the horizontal, vertical and diagonal differences [27].

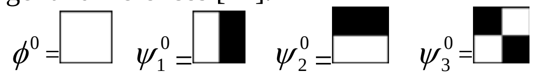

Fig.3: The mother scaling function and the three mother wavelet functions.

A two-dimensional image can be further expressed as a sum of the first scaling function plus the wavelet functions as:

$$F = F^0_{0,0}\phi_{0,0} + \sum_{k=0}^{l-1}\sum_t \sum_m F^m_{k,t}\psi^m_{k,t} = \sum_i F_i \psi_i \quad (17)$$

Here, set vector **t**= (t$_1$, t$_2$), *F* is a two-dimensional image with $2^l \times 2^l$ pixels.

4.2 General 2D wavelet product

Given two functions expressed in an orthonormal basis, it is possible to multiply them together and get the product expanded in the same basis Ng et al. [28]

Let $G = \sum G_j \psi_j$ and $H = \sum H_k \psi_k$ be the two images represented in the Haar basis. The wavelet product, $F = \sum F_i \psi_i$, of G and H is then given by:

$$F = G \cdot H \Leftrightarrow \sum F_i \psi_i = \sum G_j \psi_j \cdot \sum H_k \psi_k \quad (18)$$

By integrating against the i^{th} basis function, we can directly obtain the i^{th} coefficient for the wavelet representation of the product *F* as follows:

$$\begin{aligned} F_i &= \iint F(x)\psi_i(x)dx = \iint \psi_i(x)G(x)H(x)dx \\ &= \iint \psi_i(x)(\sum G_j \psi_j(x) \cdot \sum H_k \psi_k(x))dx \\ &= \sum_j \sum_k G_j H_k \iint \psi_i(x)\psi_j(x)\psi_k(x)dx \\ &= \sum_j \sum_k C_{ijk} G_j H_k \end{aligned} \quad (19)$$

$$C_{ijk} = \iint \psi_i(x)\psi_j(x)\psi_k(x)dx \quad (20)$$

Note that these equations are valid for any domain and suitable orthonormal basis, only the tripling coefficients will differ. Due to the compact support of the Haar basis functions, most of the tripling coefficients will be zero. The non-zero coefficients are given by the Haar tripling coefficient theorem by Ng R. [29].

Generalized Haar Integral Coefficient Theorem. The nth-order Haar integral coefficient C^n has a non-zero value, if and only if the numbers of the three kinds of wavelets ψ_1, ψ_2 and ψ_3 at the finest scale have the same parity. In this case, the integral coefficient is $\pm 2^{\sum j - 2^{j0}}$, where $\sum j$ is the sum of the scales of all operand basis functions, and j$_0$ is the scale of the finest basis function. The sign of the integral coefficient is the multiplication of the signs of the subregions of all parent basis functions that the child basis function falls into[29].

According to the above theorem, the integral of three 2D Haar basis functions is non-zero if and only if one of the following three cases holds:

1. All three are the scaling function. In this case, $C_{ijk} = 1$.

2. All three functions occupy the same wavelet square, and all are of different wavelet types. $C_{ijk} = 2^l$, where the square is at level *l*.

3. Two are identical wavelets, and the third is either the scaling function or a wavelet that overlaps at a strictly coarser level. $C_{ijk} = \pm 2^l$, where the third function exists at level *l*.

The tripling coefficient theorem is written in general terms, and describes the cases where the tripling coefficients are non-zero. In this application, where we are looking at a specific basis function, ψ_i, the theorem can be rewritten to make the different cases more clear:

1. ψ_i is the mother scaling function:

(a) ψ_j and ψ_k are also the mother scaling function. $C_{ijk} = 1$.

(b) ψ_j and ψ_k are identical wavelets (at any level). $C_{ijk} = 1$.

2. ψ_i is a wavelet function at level *l*:

(a) All three functions occupy the same wavelet square and all are of different wavelet types. $C_{ijk} = 2^l$.

(b) ψ_j and ψ_k are identical wavelets under the support of. ψ_i and exist at a strictly finer level. $C_{ijk} = \pm 2^l$.

(c) One of the wavelets is identical to ψ_i, and the other is either the mother scaling function or a wavelet that overlaps at a strictly coarser level. $C_{ijk} = \pm 2^{l'}$, where the coarser function exists at level *l'*.

4.3 Wavelet importance sampling

For simplicity, the image $F(\mathbf{x})$ is defined to cover the unit square. Consider a wavelet square s = (*l*, **t**) at level *l* and translation t. The square has an area of A(s) = $2^{-l} \times 2^{-l} = 2^{-2l}$. The average function value *F(**s**)* over the square, can be found by integrating the function over s [22]. However, due to the constant and disjoint scaling functions, the average function value is given by the scaling coefficient for the square as follows:

$$F(s) = \iint\limits_{s} F(x)dx = 2^l \iint \phi_{l,t}^0(x) F(x) dx = 2^l F_{l,t}^0 \quad (21)$$

$$I = \iint F(x) dx = F_{0,0}^0 \qquad (22)$$

Thus, the probability density of the square s, is given by:

$$p(s) = \frac{F(s)}{I} = 2^l \frac{F_{l,t}^0}{F_{0,0}^0} \qquad (23)$$

The probability of placing a sample at a coordinate x within the square s, should be equal to:

$$p(x \in s) = p(s) A(s) = 2^{-2l} \frac{F(s)}{I} = 2^{-l} \frac{F_{l,t}^0}{F_{0,0}^0} \qquad (24)$$

For recursive algorithms, it is useful to know the conditional probabilities for each child square, given that the parent square is sampled. Let s be the parent square at level *l*, and let s_i, *i = 1 . . . 4*, be the four child squares at level *l+1*. The conditional probability for each of the four children can be expressed in the function values for the parent and child squares as:

$$p(x \in s_i | x \in s) = \frac{p(x \in s_i)}{p(x \in s)} = \frac{2^{-2(l+1)} F(s_i)/I}{2^{-2l} F(s)/I} = \frac{1}{4} \frac{F(s_i)}{F(s)} \quad (25)$$

and similarly expressed in scaling coefficients as:

$$p(x \in s_i | x \in s) = \frac{p(x \in s_i)}{p(x \in s)} = \frac{2^{-(l+1)} F_{l+1,t}^0 / F_{0,0}^0}{2^{-l} F_{l,t}^0 / F_{0,0}^0} = \frac{1}{2} \frac{F_{l+1,t}^0}{F_{l,t}^0} \quad (26)$$

4.4 Improved wavelet products sampling

For a simple case, the importance function f(x) is a product of only two faunctions, f(x) =g(x)h(x). We store approximations of g(x) and h(x) as images, G and H respectively, expressed as Haar wavelets. Then coefficients for the product F = G·H of the two wavelets can be computed using theory in 4.2.

In practice, as stated in last section, it is unnecessary to compute detail coefficients for the wavelet product, as only the scaling coefficients at each level are needed for sampling. So the general product in 4.2 could be simplified by direct product of only scaling coefficients. While replacing ψ_i with the specific scaling function $\phi_{l,t}$, the scaling coefficient for the product is then given by:

$$\begin{aligned} F_{l,t}^0 &= \iint F(x) \phi_{l,t}(x) dx = \iint \phi_{l,t}(x) G(x) H(x) dx \\ &= \iint \phi_{l,t}(x) (\sum G_j \psi_j(x) \cdot \sum H_k \psi_k(x)) dx \\ &= \sum_j \sum_k G_j H_k \iint \phi_{l,t}(x) \psi_j(x) \psi_k(x) dx \\ &= \sum_j \sum_k C'_{ijk} G_j H_k \end{aligned} \qquad (27)$$

where C'_{ijk} are modified tripling coefficients, defined as:

$$C'_{ijk} = \iint \phi_{l,t}(x) \psi_j(x) \psi_k(x) dx \qquad (28)$$

It turns out that the C'_{ijk} for a scaling function at level *l* are non-zero if and only if one of the following two cases holds:

1. ψ_j and ψ_k are either the mother scaling function or wavelets at strictly coarser levels, l_j and l_k. $C'_{ijk} = \pm 2^{l_j + l_k - l}$.

2. ψ_j and ψ_k are identical wavelets under the support of $\phi_{l,t}$, and exist at the same or finer levels. $C'_{ijk} = 2^l$.

The first case corresponds to a multiplication of the scaling coefficients for G and H at level *l* that overlap $\phi_{l,t}$, scaled by 2^l, i.e., a multiplication of the scaling coefficients $G_{l,t}^0$ and $H_{l,t}^0$. Hence, scaling coefficients for the product as:

$$F_{l,t}^0 = 2^l G_{l,t}^0 H_{l,t}^0 + 2^l \sum_{l' \geq l, t' \in t, m} F_{l',t'}^m H_{l',t'}^m \qquad (29)$$

where the summation is over all wavelet coefficients that are under the support of $\phi_{l,t}$. the scaling coefficients $G_{l,t}^0$ and $H_{l,t}^0$ can easily be computed separately for the two functions, using standard wavelet reconstruction from their respective wavelet coefficients.

This simplified way is much more efficient than the general one. Once the product *F* can be computed, the importance sampling probability computing is as same as above equations for single function case described in 4.3.

4.5 Algorithms of wavelet products sampling

In the following algorithms, we often need to refer to the sign of a particular 2D Haar basis function over one of its four quadrants. Therefore, we introduce a function, sign(*m*, q_x, q_y), that returns the sign of a wavelet function $\psi_m^0, m = 1,2,3$ in the quadrant $(q_x, q_y) \in \{0,1\}$.

First, a helper function of parent sum (PSUM) is introduced, which was used by Ng et al. [10] in their computation of wavelet triple products. The parent sum of a square *s= (l, x, y)* in a wavelet representation *F*, is the reconstructed function value over the square, which we call *F(s)*. The reconstructed value is found by taking the sum of the coefficients of basis functions overlapping the square, scaled by the value of the corresponding basis functions over the square we are considering.

Table 1 Pseudo-code of PSUM

```
function PSUM(wavelet F, square s)
if s ∈ psum_table then
return psum_table(s)
else
if s = (0, 0, 0) then
value = $F_{0,0}^0$
else
s' = (l', x', y') = (l-1, $\lfloor x/2 \rfloor$, $\lfloor y/2 \rfloor$)
($q_x$, $q_y$) = (x−2x', y − 2y')
value = PSUM(F, s') +
$2^{l'} \sum_{m=1}^{3} F_{l',x',y'}^{m} .sign\ (m, q_x, q_y)$
psum_table−>insert(s, value)
return value
```

Second, another helper function is defined, which we call children sum (CSUM). The children sum of two wavelet representations, *G* and *H*, at a square s, is the sum of the product of the coefficients of identical basis functions that overlap the square at the same or finer levels. We start by computing all the non-zero children sums and storing them in a hash table using CsumHash function. The children sums are very sparse, since both coefficients for an identical basis function need to be non-zero for the result to be non-zero. Later, the children sums are accessed through a function CSUM(s) that simply returns the value stored in the hash map, or zero if there is no value associated with the square s.

Table 2 Pseudo-code of CsumHash

```
function CsumHash(wavelet G, wavelet H)
for $G_{l,x,y}^m \in G, G_{l,x,y}^m \neq 0$ do
if $H_{l,x,y}^m \neq 0$ then
    $c = G_{l,x,y}^m . H_{l,x,y}^m$
        s = (l ,x, y)
    while $l' \geq 0$ do
    csum_table(s) = csum_table(s) +c
    s=( l'-1, $\lfloor x/2 \rfloor$, $\lfloor y/2 \rfloor$)
```

Now, with the necessary support functions, PSUM and CSUM, it is straightforward to implement the wavelet product. We implement the wavelet multiplication as a function that takes two wavelets as input, and returns a specific wavelet coefficient for the product. The following function returns the wavelet coefficient of type m at a square s, for the product of two wavelets, *F = G.H*. We also need the first scaling coefficient for the product, which is given by the simple expression: $F_{0,0}^0 = G_{0,0}^0 H_{0,0}^0 + CSUM(s)$.

Table 3 Pseudo-code of PRODUCT

```
function PRODUCT(square s=(l,x,y), int m)
// All at the same square but of different type
m1 = ((m + 1) mod 4) + 1
m2 = ((m + 2) mod 4) + 1
$c1 = 2^l (G_{l,x,y}^{m1} H_{l,x,y}^{m2} + G_{l,x,y}^{m2} H_{l,x,y}^{m1}$
// The product at strictly finer levels
c2 = $2^l$ sign(m, 0, 0).CSUM(l+1, 2x, 2y)
+ $2^l$ sign(m, 1, 0).CSUM (l+1, 2x+1, 2y)
+ $2^l$ sign(m, 0, 1).CSUM (l+1, 2x, 2y+1)
+ $2^l$ sign(m, 1, 1).CSUM (l+1, 2x+1, 2y+1)
// One identical and the rest at coarser levels
$c3 = G_{l,x,y}^m .PSUM(H, s) + H_{l,x,y}^m .PSUM(G, s)$
// Return sum of the contributions
return (c1 + c2 + c3)
```

This function can be used in the general case for computing coefficients of the product of two wavelet representations. For importance sampling, we are only interested in computing scaling coefficients for the product.

Using the theory in Section 4.4, we arrive at the function described as following algorithm. This function directly returns the reconstructed average function value for a wavelet square s in the product of two wavelet represented functions, that is, the returned value is the scaling coefficient in s for the product, scaled by 2^l.

Table 4 Pseudo-code of optimized PRODUCT

```
function OPPRODUCT(square s=(l,x,y) )
// Both are scaling functions
c1 = PSUM(H, s) . PSUM(G, s)
// The product of wavelets at finer levels
c2 = $4^l$ · CSUM(s)
// Return sum of the contributions
return c1 + c2
```

5 Implement and results

5.1 Re-parameterization for 4D BRDF

In our application, the BRDF is given in local coordinates with respect to the reflection vector, while an environment map is commonly expressed in global coordinates. By rewriting the environment map as a four-dimensional function $L(\vec{\omega}, \vec{\omega}_r)$, where the direction $\vec{\omega}$ is given with respect to $\vec{\omega}_r$, the environment map is in the same local space as the BRDF.

By a change of variables, the BRDF can be transformed into a function that is more compact. There are many ways for such re-parameterizations. In our application, we need a parameterization that is suitable for both the BRDF and for the environment map. The BRDF is centered about the reflection vector $\vec{\omega}_r = (\theta_r, \varphi_r)$, instead of around the surface normal $\vec{N}$.

The environment map image can be preprocessed and evaluated by practical ways as Gribkov I.V. used in their works [30].

5.2 Wavelet transform and compression

As noted earlier, we use normalized Haar wavelets and the non-standard decomposition. Starting with an image with n × n pixels, where n is a power of two, the one-dimensional wavelet transform is applied first on rows and then on columns. The wavelet transform is done using the lifting scheme [31], which is a fast linear-time operation that does not require any additional temporary storage. Each pass splits the data into a set of scaling coefficients and a set of detail coefficients. The transform is applied recursively on the scaling coefficients, until only a single scaling coefficient remains.

After our data has been wavelet transformed, we apply lossy compression by thresholdning the wavelet coefficients. There are many ways to do that not only in wavelet ones but also combined with some fractacl ones as described by Radu D. et al [32].

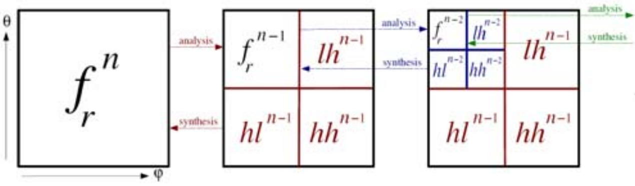

Fig.4: 2D Wavelet transform applied on each hemisphere of the original BRDF data.

A non-standard approach of wavelet transform for f_r is employed here (shown in Fig.4).

In practice, both the BRDF and the environment map are tabulated as a sparse 2D set of 2D wavelet compressed images[31]. The maps are stored at the resolution 64×64 or 128×128.

5.3 BRDF data and environment map data

In our renderings, we use isotropic BRDFs acquired from real materials. The BRDF data sets we use consist of dense reflectance measurements in color for over 50 different materials. The measurements were done for 90 × 90 × 180 discrete directions with a denser sampling around the specular highlight. A few of the BRDFs and source-code for loading them can be downloaded from website[1] . We resample the measured reflectance data into our BRDF representation. Each two-dimensional reflectance map is first created at a high resolution (256 × 256), and then sub-sampled to the desired resolution to avoid aliasing [33].

We use high-dynamic range environment maps , so called light probes, for rendering scenes under realistic illumination. Paul Debevec's Light Probe Gallery [2] has a selection of environment maps available for download. The light probes are stored in Ward's RGBE format. The maps represent the full sphere and use a simple angular parameterization. The light probes are re-sampled so that for each tabulated two-dimensional environment map, we loop over its pixels and sample the light probe in the corresponding direction. To avoid aliasing, the re-sampling is done at a higher resolution than the final result, usually 512×512 or 1024×1024, and the image is down-sampled to the desired resolution using a box filter. Each two-dimensional environment map is then wavelet compressed, and only the non-zero coefficients are stored.

5.4 Ray tracing rendering

A simple Monte Carlo ray tracer was implemented in order to evaluate how well wavelet importance sampling of products performs compared to other methods.

At the rendering time, the number of texture accesses for each fragment depends on the resolution chosen on-the-fly for the BRDF

[1] http://graphics.csail.mit.edu/_wojciech/BRDF/

[2] http://www.debevec.org/Probes/

reconstruction. If the surface is far away from the viewpoint, fewer levels are required to estimate the BRDF and the performance is enhanced.

A ray tracing rendering result is implemented and showed as fig.5.

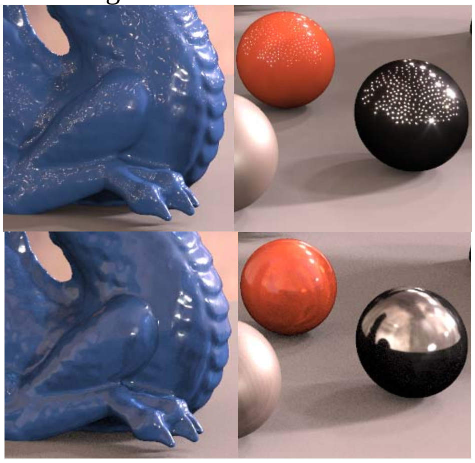

Fig. 5: Top: Structured sampling results; Bottom: Wavelet sampling rendering results

These results are achieved with GPU built-in hardware filtering (see fig.6) and by using the linearity and the multi-resolution of the wavelet encoding.

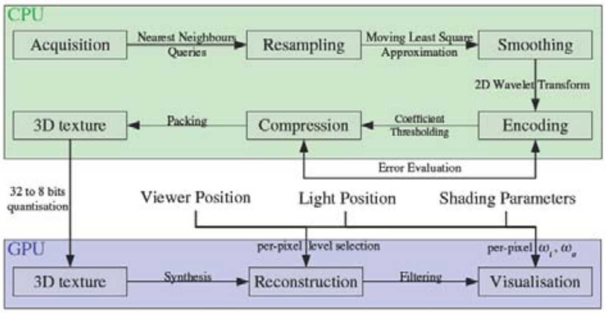

Fig.6: the GPU enhanced rendering pipeline.

6 Conclusions and future work

Wavelet product importance sampling is an efficient way for static direct illumination with complex environment mapping. According to the feature of its product sampling of two functions, the factoring scheme we developed makes shading integral reduce to a simple double function product integral. Such way is suitable for dynamic global lighting situations with multiple objects where normally only one variable is adjusted at a time, and real-time feedback is highly appreciated.

A GPU enabled pipeline is also used to accelerate the real-time rendering and worked well in practice.

Wavelet representations of BRDF and EM provide novel approaches for complex rendering. The way we proposed here can be used in other domains where the efficient computation and real-time generation are critical such as game, animation, and simulation.

More complicated scenes should be tested in future and much more efficient ways will be used.

References:

[1] Blinn, J. F.. Models of light reflection for computer synthesized pictures. In SIGGRAPH 77, 1977, pp.192–198.

[2] Phong, B. T.. Illumination for computer generated pictures. Commun. ACM, Vol.18, No.6, 1975, pp. 311–317,.

[3] Eric V. and Leonidas J. Guibas. Optimally Combining Sampling Techniques for Monte Carlo Rendering. In Proceedings of ACM SIGGRAPH 95, 1995, pp. 419–428.

[4] Burke D., Ghosh A., and Heidrich W.. Bidirectional Importance Sampling for Illumination from Environment Maps. In ACM SIGGRAPH Technical Sketches, 2004

[5] Burke D., Ghosh A., and Heidrich W.. Bidirectional importance sampling for direct illumination. Rendering Techniques 2005 Eurographics Symposium on Rendering (Aire-la-Ville, Switzerland) , 2005, Eurographics Association, 2005, pp. 139–146.

[6] James T. Kajiya. The Rendering Equation. Computer Graphics (Proceedings of ACM SIGGRAPH 86), 1986, pp. 143–150.

[7] Malvin H. Kalos and Paula A. Whitlock. Monte Carlo Methods, Volume 1: Basics. Wiley-Interscience, New York, 1986.

[8] Agarwal S., Ramamoor R., Belongie S., and Jensen H.W.. Structured importance sampling of environment maps. SIGGRAPH 03, 2003, pp.605–612.

[9] Philip Dutr´e, Philippe Bekaert, and Kavita Bala. Advanced Global Illumination. AK Peters, 2003.

[10] Kollig T., Keller A.. Efficient multidimensional sampling. Computer Graphics Forum, Vol.21, No.3, 2002, pp.557–563.

[11] Burke D.: Bidirectional importance sampling for illumination from environment maps. Master's Thesis, University of British Columbia, 2004.

[12] Kollig T., Keller A.: Efficient illumination by high dynamic range images. In EGRW '03: Proceedings of the 14th Eurographics workshop on Rendering (Aire-la-Ville, Switzerland, Switzerland,), Eurographics Association, 2003, pp. 45–50.

[13] Ostromoukhov V., Donohue C., Jodon P.: Fast hierarchical importance sampling with blue

noise properties. ACM Trans. Graph. Vol.23, No.3, 2004, pp.488–495.

[14] Debevec P.: A median cut algorithm for light probe sampling. SIGGRAPH 2005 Poster, 2005.

[15] Ward G. J.: Measuring and modeling anisotropic reflection. In SIGGRAPH '92: Proceedings of the 19th annual conference on Computer graphics and interactive techniques (NY, USA, 1992), ACM Press, 1992,pp. 265–272.

[16] Lafortune E. P. F., Foo S.-C., Torrance K. E., Greenberg D. P.: Non-linear approximation of reflectance functions. In SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques (NY, USA), ACM Press / Addison-Wesley Publishing Co., 1997, pp. 117–126.

[17] Ashikhmin M., Shirley P.: An anisotropic phong BRDF model. J. Graph. Tools, Vol.5, No.2, 2000, pp.25–32.

[18] Lawrence J., Rusinkiewicz S., and Ramamoorthi R.. Efficient BRDF Importance Sampling using a Factored Representation. ACM Transactions on Graphics, Vol.23, No.3, 2004, pp.496–505.

[19] Teh, C. S.,Lim, C. P.. A probabilistic SOM-kMER model for intelligent data analysis. WSEAS Transactions on Systems,Vol.5, No.4, 2006, pp.825-832

[20] Veache E., Guibas L. J.: Optimally combining sampling techniques for monte carlo rendering. In SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques (NY, USA), ACM Press, 1995, pp. 419–428.

[21] Talbot J. F., Cline D., Egbert P. K.: Importance resampling for global illumination. In Rendering Techniques 2005 Eurographics Symposium on Rendering (Aire-la-Ville, Switzerland), Bala K., Dutré P., (Eds.), Eurographics Association, 2005, pp. 139–146.

[22] Clarberg P., Jarosz W., Akenine-Moller T., and Jensen H. W.. Wavelet importance sampling: Efficiently evaluating products of complex functions. ACM Transactions on Graphics (SIGGRAPH '05),Vol.24, No.3, 2005, pp. 1166–1175

[23] Kumsawat, Prayoth; Attakitmongcol, Kitti; Srikaew, Arthit. Comparative performance of multiwavelet-based image watermaking schemes. WSEAS Transactions on Systems, Vol.5, No.5, 2006, pp. 1041-1047

[24] Sun, W., and Mukherjee, A.. Generalized wavelet product integral for rendering dynamic glossy objects. ACM Transactions on Graphics (SIGGRAPH 2006), Vol.25, No.3, 2006, pp.477–487.

[25] Matt Pharr and Greg Humphreys. Physically Based Rendering. Morgan Kaufmann, 2004.

[26] Mallat S. G.. A Theory for Multiresolution Signal Decomposition: The Wavelet Representation. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.11, No.7, 1989, pp. 674–693.

[27] Eric J. Stollnitz, Tony D. DeRose, and David H. Salesin. Wavelets for Computer Graphics: Theory and Applications. Morgan Kaufmann, 1996.

[28] Ng, R., Ramamoorthi, R., and Hanrahan, P.. All frequency shadows using non-linear wavelet lighting approximation. ACM TOG (SIGGRAPH 2003), Vol.22, No.3, 2003, pp. 376–381.

[29] Ng R., Ramamoorthi R., and Hanrahan P.. Triple Product Wavelet Integrals for All-Frequency Relighting. ACM Transactions on Graphics, Vol.23, No.3, 2004 ,pp. 477–487.

[30] Gribkov I.V., Koltsov, P.P., Kotovich N.V., et al.. Empirical evaluation of image processing methods using PICASSO 2 system. WSEAS Transactions on Systems, Vol.4, No.11, 2005, pp. 1923-1930

[31] W. Sweldens. The Lifting Scheme: A New Philosophy in Biorthogonal Wavelet Constructions. In Wavelet Applications in Signal and Image Processing III. Proceedings of SPIE, 1995, pp.68–79.

[32] Radu D., Matei D. Stefan M., Sebastian T.. A fractal estimator for comparing predictors performance in lossless image compression. WSEAS Transactions on Systems, Vol.5, No.5, 2005, pp. 989-994

[33] Ben-Artzi, A., Overbeck, R., and Ramamoorthi, R.. Real-time BRDF editing in complex lighting. ACM Transactions on Graphics (SIGGRAPH 06), Vol.25, No.3, 2006, pp.945–954.

[34] Nicodemus F. E., Richmond J. C., Hsia J. J., Ginsberg I. W., and Limperis T.. Geometric Considerations and Nomenclature for Reflectance, volume NBS Monograph 160. National Bureau of Standards, Washington, D.C., 1977.