DOI: 10.1111/cgf.15153

Eurographics Symposium on Rendering 2024
E. Garces and E. Haines
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 43 (2024), Number 4

Lossless Basis Expansion for Gradient-Domain Rendering

Q. Fang[1] and T. Hachisuka[1]

[1] University of Waterloo, Canada

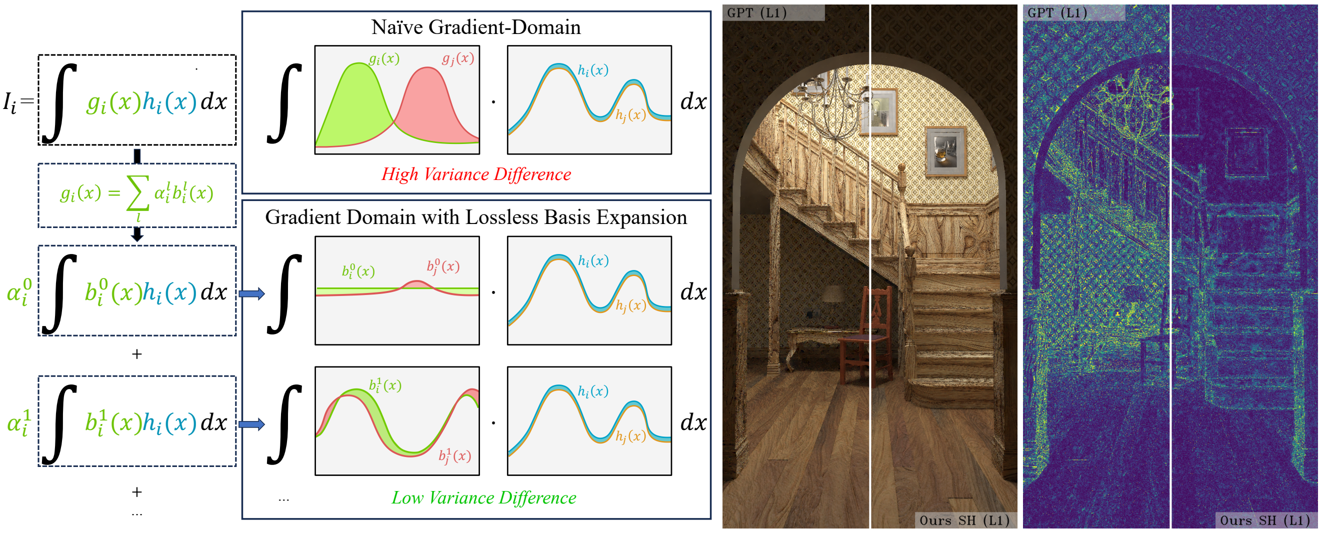


Figure 1: *Left: Overview of our method. We propose to expand a part of the integrand $g(x)$ that significantly varies across pixels into a weighted sum of lossless basis functions. The resulting basis expansion is designed to satisfy the similarity assumption between neighboring pixels i, j, making difference estimates effective with shift mapping. Right: Comparison between vanilla gradient-domain rendering and our method for a scene with highly varying materials across pixels. Our method accurately captures variation among pixels without blurring sharp features after L^1 Poisson reconstruction.*

Abstract

Gradient-domain rendering utilizes difference estimates with shift mapping to reduce variance in Monte Carlo rendering. Such difference estimates are effective under the assumption that pixels for difference estimates have similar integrands. This assumption is often violated because it is common to have spatially varying BSDFs with material maps, which potentially result in a very different integrand per pixel. We introduce an extension of gradient-domain rendering that effectively supports such per-pixel variation in BSDFs based on basis expansion. Basis expansion for BSDFs has been used extensively in other problems in rendering, where the goal is to approximate a given BSDF by a weighted sum of predefined basis functions. We instead utilize lossless basis expansion, representing a BSDF without any approximation by adding the remaining difference in the original basis expansion. This lossless basis expansion allows us to cancel more terms via shift mapping, resulting in low variance difference estimates even with per-pixel BSDF variation. We also extend the Poisson reconstruction process to support this basis expansion. Regular gradient-domain rendering can be expressed as a special case of our extension, where the basis is simply the BSDF per pixel (i.e., no basis expansion). We provide proof-of-concept experiments and showcase the effectiveness of our method for scenes with highly varying material maps. Our results show noticeable improvement over regular gradient-domain rendering under both L^1 and L^2 reconstructions. The resulting formulation via basis expansion essentially serves as a new way of path reuse among pixels in the presence of per-pixel variation.

CCS Concepts

- ***Computing methodologies*** → ***Ray tracing;***

1. Introduction

Gradient-domain rendering [LKL*13, KMA*15, HGP*19] is an efficient rendering algorithm for scenes with smooth features. The main idea is to estimate the differences between neighboring pixels, in addition to the pixels themselves, and reconstruct the image based on those estimates. The original gradient-domain rendering method has been extended to nearly all popular Monte Carlo rendering methods [MKA*15, HGNH17, GHV*18, SSC*17, BPE17, MKD*16], providing significant variance reduction over regular Monte Carlo counterparts in many cases.

Gradient-domain rendering improves Monte Carlo rendering mainly because of low-variance difference estimates with correlated samples. Correlated sampling is achieved by a technique called *shift mapping*, which maps a sample in one pixel to a similar sample in a neighboring pixel. Difference estimates by shift mapping will be effective when neighboring pixels have similar integrands since samples before and after shift mapping have similar contributions and thus they cancel each other in the estimator. In many practical scenarios, however, this assumption is often violated because it is common to have spatial variation of materials and shading normals across neighboring pixels. While gradient-domain rendering can still render images even in such cases, its performance degrades. A practical trick of factoring out albedo [ZSWL21] does not work if the variation does not come from albedo (e.g., shading normals).

We propose an extension of gradient-domain rendering based on basis expansion, which allows us to handle per-pixel variation across pixels while maintaining the benefits of gradient-domain rendering. Our extension considers an integrand a product of two functions - one that can vary significantly across pixels and the other that is assumed to be similar among pixels. An example of such an integrand is a product of a BSDF and incident illumination, where the BSDF can vary rapidly among pixels, but incident illumination is generally smooth over pixels. We propose to apply *lossless* basis expansion to the fast-varying part of the product to capture a variation among pixels as a variation of coefficients on the bases. Unlike regular basis expansion, our lossless basis expansion results in an exact recovery of a given function since it adds back the remaining differences (residuals) to the basis functions. The product of this lossless basis expansion and the slowly varying part of the integrand will become similar among neighboring pixels, effectively allowing us to cancel more terms via shift mapping in gradient-domain rendering. We also re-designed the Poisson reconstruction in gradient-domain rendering to support this basis expansion and reconstruct the final image by simultaneously taking into account the results of all the basis integrals. Under our formulation, gradient-domain rendering is a special case where no basis expansion is performed. Our extended gradient-domain rendering can resolve common artifacts in gradient-domain rendering with per-pixel variations, making it applicable to a larger set of practical scenes. Figure 1 shows one of the comparisons of our method with order 2 spherical harmonics against regular gradient-domain rendering. Our result accurately captures per-pixel variation while keeping the benefits of gradient-domain rendering.

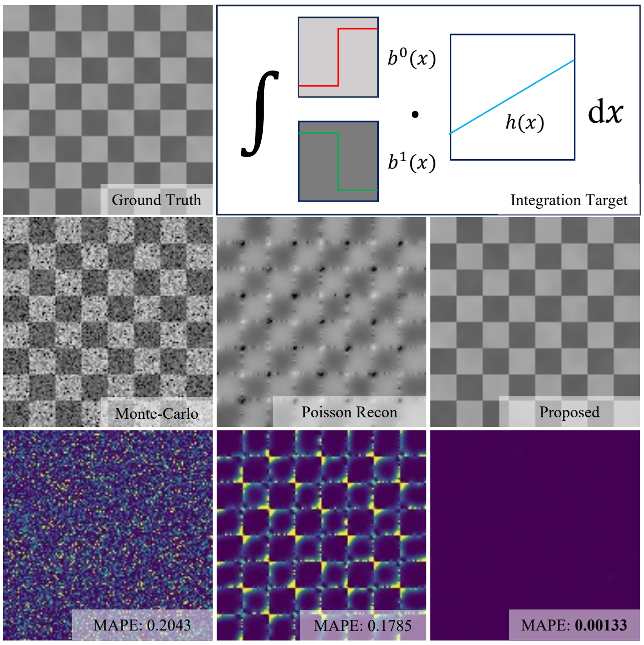


Figure 2: *Failure mode of gradient-domain rendering. The top right visualizes the integrands for lighter and darker blocks of the checkerboard pattern, and the middle and bottom rows show the reconstruction using per-pixel Monte Carlo integration, regular gradient-domain rendering, and our method with their errors compared to the ground truth. Gradient-domain rendering suffers from artifacts due to high variance in difference estimates around edges. Our method almost perfectly reconstructs the ground truth.*

2. Motivation

Let us consider a problem of solving an array of integrals I_i of functions $f_i(x)$ indexed by an integer i as

$$I_i = \int_{\mathcal{X}_i} f_i(x)\mathrm{d}x. \tag{1}$$

where $\mathcal{X}_i$ is the domain of integration. A prominent example is rendering based on the rendering equation [Kaj86] where we need to solve an integral I_i of the product function f_i of the incoming radiance and the cosine-weighted bidirectional scattering distribution function (BSDF) at each pixel i.

Monte Carlo integration estimates each integral I_i by taking the average of stochastic evaluations of the integrand f_i as in

$$I_i \approx \langle I_i \rangle_{MC} = \frac{1}{N}\sum_{k=1}^{N}\frac{f_i(x_{i,k})}{p_i(x_{i,k})} \tag{2}$$

where N is the number of samples and the samples $\{x_{i,k}\}$ are distributed according to a probability density function $p_i(x)$. While we can solve each integral I_i independently with this approach, we often have domain-specific knowledge regarding how integrals I_i and I_j $(i \neq j)$ differ from each other. For example, integrals for neighboring pixels I_i and I_j in rendering are usually similar except for those around discontinuities on the image (e.g., geometric discontinuities). Independent estimates for I_i and I_j do not take such similarities into account.

Gradient-domain rendering [LKL*13, HGP*19] exploits similarities among neighboring I_i and I_j by coupling Monte Carlo estimates of differences $I_j - I_i$ through a Poisson image reconstruction process. The differences $I_j - I_i$ are estimated by correlating Monte Carlo samples for I_j and I_i via *shift mapping*. Shift mapping assumes that the integrand f_i is similar to a neighboring integrand f_j for similar x, and this assumption can be informally written as

$$x_i \approx x_j \Rightarrow f_i(x_i) \approx f_j(x_j). \tag{3}$$

where $x_j = T_{ij}(x_i)$ is a shift function $T_{ij}(x)$ [LKL*13] that maps x_i in the pixel i to x_j in the pixel j. The difference of the sample contributions $f_i(x_i)$ and $f_j(x_j)$ will be small when shift mapping results in $x_i \approx x_j$ and if the above assumption is satisfied.

The difference for $I_j - I_i$ is equal to the summation of two integrals symmetrically to properly account for shift operation from both directions:

$$\begin{aligned}\Delta_{ij}I = &\int_{\mathcal{X}_i} w_{ij}(x)\left(f_j(T_{ij}(x))\left|\frac{\mathrm{d}T_{ij}(x)}{\mathrm{d}x}\right| - f_i(x)\right)\mathrm{d}x \\ &+ \int_{\mathcal{X}_j} w_{ji}(x)\left(f_j(x) - f_i(T_{ji}(x))\left|\frac{\mathrm{d}T_{ji}(x)}{\mathrm{d}x}\right|\right)\mathrm{d}x\end{aligned} \tag{4}$$

where $\left|\frac{\mathrm{d}T_{ij}}{\mathrm{d}x}(x)\right|$ denote the Jacobian determinant for T_{ij} with respect to x. The weights w_{ij}, w_{ji} [KMA*15] ensure both efficient estimates in overlapping domains and prevent invalid or non-reversible shifts from contributing to the final estimate.

One can obtain unbiased estimates by sampling both integrals according to their respective sampling distribution p_i, p_j

$$\begin{aligned}I_j - I_i &\approx \langle \Delta_{ij} I\rangle_{MC} \\ &= \frac{1}{N}\sum_{k=1}^{N} w_{ij}(x_{i,k})\frac{f_j(T_{ij}(x_{i,k}))\left|\frac{\mathrm{d}T_{ij}(x_{i,k})}{\mathrm{d}x_{i,k}}\right| - f_i(x_{i,k})}{p_i(x_{i,k})} \\ &+ \frac{1}{N}\sum_{k=1}^{N} w_{ji}(x_{j,k})\frac{f_j(x_{j,k}) - f_i(T_{ji}(x_{j,k}))\left|\frac{\mathrm{d}T_{ji}(x_{j,k})}{\mathrm{d}x_{j,k}}\right|}{p_j(x_{j,k})}\end{aligned} \tag{5}$$

where N denotes the number of samples. When the assumption in Equation (3) holds and shift mapping results in $x_{i,k} \approx T_{ij}(x_{i,k})$, $x_{j,k} \approx T_{ji}(x_{j,k})$, this difference estimator $\langle \Delta_{ij} I\rangle$ will have low variance because

$$\begin{gathered}\left|\frac{\mathrm{d}T_{ij}(x_{i,k})}{\mathrm{d}x_{i,k}}\right| \approx 1 \quad \text{and} \quad \left|\frac{\mathrm{d}T_{ji}(x_{j,k})}{\mathrm{d}x_{j,k}}\right| \approx 1 \\ f_j(T_{ij}(x_{i,k}))\left|\frac{\mathrm{d}T_{ij}(x_{i,k})}{\mathrm{d}x_{i,k}}\right| - f_i(x_{i,k}) \approx 0.\end{gathered} \tag{6}$$

Gradient-domain rendering then reconstructs all I_i based on the difference estimates $\langle \Delta_{ij} I\rangle$. With L^2 loss, it can be expressed as

$$\langle I\rangle_G = \min_I \sum_{i,j} \|I_j - I_i - \langle \Delta_{ij} I\rangle_{MC}\|^2 + \lambda \sum_i \|I_i - \langle I_i\rangle_{MC}\|^2. \tag{7}$$

The first term aims to minimize the gap between the reconstructed difference and the difference estimate. The second term with the parameter λ regularizes the reconstruction towards the primal per-pixel Monte Carlo estimates for each I_i.

Problem Statement. While gradient-domain rendering works well when Equation (3) holds, this assumption may not hold universally, thereby constraining its application in practice. For example, it is common to have a spatial variation of albedo and/or normals that may result in different integrands per pixel. Figure 2 shows a simple example where f_i varies like a checkerboard pattern over i. In this example, f_i is a product of two functions

$$f_i(x) = g_i(x)h_i(x) \tag{8}$$

where $g_i(x)$ varies like a checkerboard (i.e., differs among pixels around edges) but $h_i(x) = x$ is identical for all i (i.e., no variation among pixels). This example models a typical scenario in rendering where $g_i(x)$ is a spatially varying feature such as the BSDF term per pixel and $h_i(x)$ is a mostly smooth feature such as incident illumination. The integrand $f_i(x)$ does not satisfy the assumption in Equation (3) whenever pixels I_i and I_j have different checkerboard colors, and only a part of the integrand $h_i(x)$ satisfies the assumption. As such, gradient-domain rendering fails to handle even this simple example and results in large errors whenever the integrand $f_i(x)$ varies among neighboring pixels. We present an extension of gradient-domain rendering that can efficiently handle a larger set of problems including this simple example.

3. Basis Expansion for Gradient-Domain Rendering

To introduce our idea, let us work through the example in Figure 2. Suppose that we represent $g_i(x)$ as a weighted sum of two *basis* functions $b_i^0(x)$ and $b_i^1(x)$ as in

$$g_i(x) \overset{\text{def}}{=} \alpha_i^0 b_i^0(x) + \alpha_i^1 b_i^1(x) \tag{9}$$

where α_i^0 and α_i^1 are the weights. Let us choose the basis functions $b_i^0(x)$ and $b_i^1(x)$ such that each one of them is equal to $g_i(x)$ within the corresponding checkerboard block. The weights α_i^0 and α_i^1 in this case can be trivially defined as

$$\alpha_i^0 = \begin{cases} 0 & (i \in \text{darker block}) \\ 1 & (i \in \text{lighter block})\end{cases} \quad \alpha_i^1 = 1 - \alpha_i^0 \tag{10}$$

which effectively captures the checkerboard pattern via the weights. Note that $b_i^0(x)$ and $b_i^1(x)$ are the *same* across all the pixels (i.e., does not vary over i), and the per-pixel variation is now encoded by the weights α_i^0 and α_i^1.

Under this basis expansion of $g_i(x)$, the integral I_i could also be decomposed into a weighted sum:

$$\begin{aligned}I_i &= \int_{\mathcal{X}_i} h_i(x)g_i(x)\mathrm{d}x = \int_{\mathcal{X}_i} h_i(x)\left(\alpha_i^0 b_i^0(x) + \alpha_i^1 b_i^1(x)\right)\mathrm{d}x \\ &= \alpha_i^0 \int_{\mathcal{X}_i} h_i(x)b_i^0(x)\mathrm{d}x + \alpha_i^1 \int_{\mathcal{X}_i} h_i(x)b_i^1(x)\mathrm{d}x = \alpha_i^0 I_i^0 + \alpha_i^1 I_i^1\end{aligned} \tag{11}$$

where we wrote $I_i^0 = \int_{\mathcal{X}_i} h_i(x)b_i^0(x)\mathrm{d}x$ and similarly for I_i^1. The problem changed from solving *one* integral of $h_i(x)g_i(x)$ to *two* integrals for $h_i(x)b_i^0(x)$ and $h_i(x)b_i^1(x)$ weighted by α_i^0 and α_i^1. The new integrands $h_i(x)b_i^0(x)$ and $h_i(x)b_i^1(x)$ both satisfy the similarity assumption in Equation (3) because $b_i^0(x)$, $b_i^1(x)$, and $h_i(x)$ are in fact *exactly the same* for all the pixels i in this particular example. Therefore, gradient-domain rendering can be successfully applied for each of $h_i(x)b_i^0(x)$ and $h_i(x)b_i^1(x)$ independently, and we can reconstruct I_i^0 and I_i^1 to estimate I_i as the weighted sum $\alpha_i^0 I_i^0 + \alpha_i^1 I_i^1$ per pixel. We generalize this process as follows.

Step 1: Decompose $g_i(x)$ into a weighted sum of basis functions that satisfy the assumption, i.e. $b_i^l(x_a) \approx b_j^l(x_b)$ for $x_a \approx x_b$:

$$g_i(x) \overset{\text{def}}{=} \sum_{l=1}^{L} \alpha_i^l b_i^l(x). \tag{12}$$

Note that the basis functions do not need to be exactly the same for all the pixels as long as they satisfy the similarity assumption. There is also no assumption on the weights α_i^l across pixels.

Step 2: Write down the integral of $g_i(x)$ also as a weighted sum:

$$\begin{aligned} I_i &= \int_{\mathcal{X}_i} h_i(x) g_i(x) \mathrm{d}x = \int_{\mathcal{X}_i} h_i(x) \sum_{l=1}^{L} \alpha_i^l b_i^l(x) \mathrm{d}x \\ &= \sum_{l}^{L} \alpha_i^l \underbrace{\int_{\mathcal{X}_i} h_i(x) b_i^l(x) \mathrm{d}x}_{I_i^l} = \sum_{l=1}^{L} \alpha_i^l I_i^l. \end{aligned} \tag{13}$$

Step 3: Apply gradient-domain rendering on new integrands $b_i^l(x)h_i(x)$ independently, reconstruct I_i^l, and take the weighted sum to have $I_i = \sum_{l=1}^{L} \alpha_i^l I_i^l$. For the problems where $h_i(x)$ satisfies the similarity assumption, the integrand $b_i^l(x)h_i(x)$ should also satisfy the assumption.

For the example in Figure 2, in vanilla gradient-domain rendering, the difference estimate between pixels sharing the same color exhibits zero variance, i.e., $b^0(x)h(x) - b^0(x)h(x) = b^1(x)h(x) - b^1(x)h(x) = 0 = constant$ for arbitrary x, Conversely, the gradient estimate between pixels with different colors entails non-zero variance, i.e., $\pm(b^1(x)h(x) - b^0(x)h(x)) \neq constant$. Consequently, high-variance (error) information propagates to other pixels at the reconstruction step. This error manifests as "dipole artifacts" in Figure 2 under L^2 reconstruction. The difference image shows significant errors around the boundaries between different colors.

To summarize why basis expansion helps in this case, let us compare the difference between our formulation and vanilla gradient-domain. Assuming that $h_i(x) \approx h_j(x)$, $g_i(x) = \sum_l \alpha_i^l b_i^l(x)$, $g_j(x) = \sum_l \alpha_j^l b_j^l(x)$ and $b_i^l(x) \approx b_j^l(x)$ for all l:

- In gradient-domain rendering, the difference estimate used is

  $$\begin{aligned} \Delta_{ij} I &= \int_{\mathcal{X}} f_i(x) - f_j(x) \mathrm{d}x = \int_{\mathcal{X}} h_i(x) g_i(x) - h_j(x) g_j(x) \mathrm{d}x \\ &\approx \int_{\mathcal{X}} h_i(x) \left(g_i(x) - g_j(x)\right) \mathrm{d}x \end{aligned}$$

  which may have large variance when $g_i(x) \neq g_j(x)$.
- In our formulation, the difference estimate used is

  $$\begin{aligned} \Delta_{ij} I^l &= \int_{\mathcal{X}} h_i(x) b_i^l(x) - h_j(x) b_j^l(x) \mathrm{d}x \\ &\approx \underbrace{\int_{\mathcal{X}} h_i(x) \left(b_i^l(x) - b_j^l(x)\right) \mathrm{d}x}_{\text{(because } h_i(x) \approx h_j(x))} \approx 0 \end{aligned}$$

  which results in more cancellations.

3.1. Basis Expansion

Given a function g and a set of basis functions $\{b^l(x)\}_l$, basis expansion approximates g as a linear combination of b^l as

$$g(x) \approx \sum_{l=1}^{L} \alpha^l b^l(x) \tag{14}$$

where L is the number of basis functions of our choice and α^l defines the weight (or coefficients) for corresponding b^l. For orthonormal bases, each coefficient is simply a projection of function g onto the corresponding basis as

$$\alpha^l = \int_{\mathcal{X}} g(x) b^l(x) \mathrm{d}x \approx \sum_{k}^{K} \frac{g(x_k) b^l(x_k)}{p(x_k)}. \tag{15}$$

Basis expansion has been used in various applications in rendering. Sloan et al. [SKS23], for example, proposed to use a precomputed set of coefficients to represent light transport under an environment map. Ng et al. [NRH04] broke down the rendering equation for direct illumination into a triple product integral comprising the environment map, visibility, and cosine-weighted BSDF terms. This decomposition is akin to our decomposition of $g_i(x)$ and the resulting decomposition of the integral. A recent extension of this approach by Xu et al. [XZW*22] uses a neural network to train a more sophisticated set of basis functions for better approximation of each component.

Manzi et al. [MVZ16] also proposed to incorporate basis expansion into gradient-domain rendering, but the contexts of basis expansion fundamentally differ between our method and those of Manzi et al. Our method defines basis functions to decompose the integrand to have low-variance difference estimates. This process involves changes in both difference estimates and reconstruction. Manzi et al. consider a weighted sum of basis features (vectors) to regularize reconstruction only, which does not involve any change in difference estimates.

While those prior works are relevant in that they also use basis expansion, the contexts of how basis expansion helps are different and thus only remotely related to our work. For example, having higher-order basis expansion does not always help in our case because high-frequency signals tend to violate the similarity assumption as in Equation (3) when shift mapping slightly changes the sample location. In addition, unlike those prior uses of basis expansion, we would like to have an *exact* recovery of $g_i(x)$ for each pixel for consistent reconstruction. The example in Figure 2 is a case where the exact recovery is trivially available, but we would like to achieve this for more general cases. We thus propose slight modifications to this traditional basis expansion to align with our specific requirements.

Lossless Basis Expansion. We need to choose $\{b_i^l\}$ such that the weighted sum perfectly recovers $g_i(x)$ while also satisfying the similarity assumption. Observing that approximation by basis expansion (e.g., spherical harmonics) effectively captures the general shape of g_i while simultaneously satisfying the similarity assumption (i.e., spherical harmonics are the same for all the pixels), we would like to still utilize such existing basis functions. To achieve this goal, we consider adding back the *residual* (i.e., approximation error) to the basis functions. Given a set of basis function $\{b^l(x)\}_l$ we define the residual as

$$r_i(x) = g_i(x) - \sum_{l=1}^{L} \alpha_i^l b^l(x). \tag{16}$$

To recover $g_i(x)$ exactly, we "distribute" $r_i(x)$ to all the given bases. We add $r_i(x)$ evenly based on either the absolute value or the squared value of the coefficient to each basis, which results in

$$\begin{aligned} g_i(x) &= \sum_{l=1}^{L} \alpha_i^l b^l(x) + r_i(x) = \sum_{l=1}^{L} \left(\alpha_i^l b^l(x) + \frac{\|\alpha_i^l\|^n}{\sum_{l'} \|\alpha_i^{l'}\|^n} r_i(x) \right) \\ &= \sum_{l=1}^{L} \alpha_i^l \left(b^l(x) + \frac{1}{\alpha_i^l} \frac{\|\alpha_i^l\|^n}{\sum_{l'} \|\alpha_i^{l'}\|^n} r_i(x) \right) \end{aligned} \tag{17}$$

where $n \in \{1,2\}$. We can redefine our *lossless basis* for pixel i, namely $\{b_i^l(x)\}_l$, as

$$b_i^l(x) \overset{\text{def}}{=} b^l(x) + w_i^l r_i(x) \tag{18}$$

where $w_i^l = \frac{1}{\alpha_i^l} \frac{\|\alpha_i^l\|^n}{\sum_{l'} \|\alpha_i^{l'}\|^n}$. This process keeps the basis for every pixel as close as possible to the given original basis while still maintaining the equality constraint. We empirically did not observe a clear advantage of one weighting scheme over the other with $n = 1$ or $n = 2$. Consequently, in our implementation, we adhere to the $n = 2$, maintaining consistency for evaluation purposes.

This formulation is straightforward to implement with an arbitrary choice of the starting global basis. Now the difference of $b_i^l(x)$ between pixel i and j essentially depends on the residual, so selecting global bases that result in smaller residuals helps in general. We have not explored an optimal choice of such global basis functions and will leave it as future work. It is worth noting that using $g_i(x)$ as the basis itself, as well as the BSDF integration pre-factorization approach [ZSWL21] are both special cases of our method. In the case of using $g_i(x)$, our method reduces to gradient-domain path tracing [MKA*15]. The case of BSDF integration pre-factorization approach [ZSWL21] is equivalent to our method with a single constant global basis followed by per-pixel residual distribution, resulting in a unit cosine-weighted BSDF as a single lossless basis in our case. Our formulation thus generalizes those prior approaches. Figure 3 shows our method with different orders of spherical harmonics as global bases. General material variation, including albedo, can be effectively handled with more bases. It is worth iterating that a higher-order basis does not always mean better reconstruction in our method and the performance of higher-order bases is empirically comparable to lower-order ones in some scenes.

3.2. Reconstruction

Once we decomposed g_i, we can estimate the difference estimator for $I_j^l - I_i^l \approx \left\langle \Delta_{ij} I^l \right\rangle$ as in Equation (5) using the same set of samples and the same shift mapping as in regular gradient-domain, except we replace $f_i(x)$ with $h_i(x) b_i^l(x)$. It is not necessary to have additional samples for multiple bases. We then reconstruct the contribution to the pixel coming from this basis integral I_i^l.

While directly reconstructing each I_i^l independently by following Equation (7) is straightforward, the results are rather undesirable because our end goal is to reconstruct I_i, not individual I_i^l. If the coefficient α_i^l of any particular basis l for pixel i is small, I_i^l is not as important for our final estimate I_i because of Equation (13). As a result, we could give less weight to regularization related to I_i^l and allow the reconstruction to be more focused on other indices.

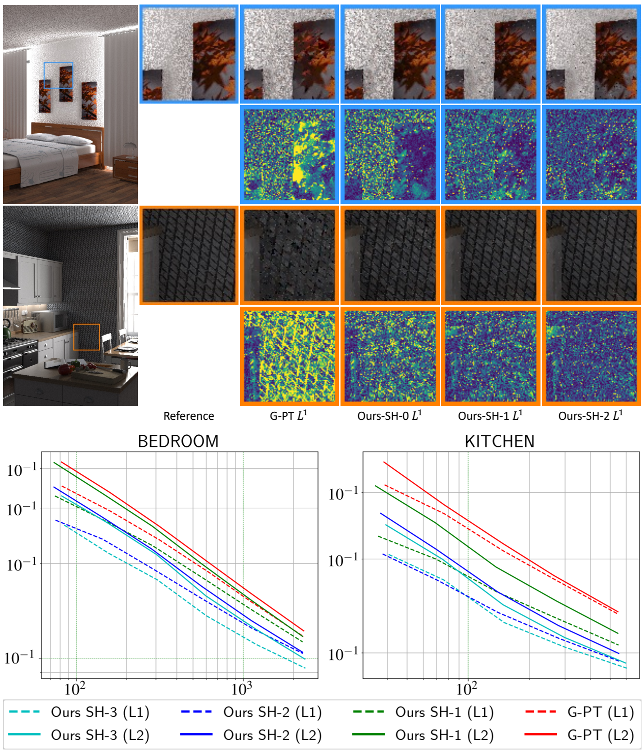


Figure 3: *Top: Comparison of our method with different orders of lossless basis alongside G-PT, with 128 samples per pixel under L^1 reconstruction. Bottom: Respective Convergence MAPE plot. Our method handles both the albedo and normal map better with higher-order bases.*

We introduce a new loss that takes the coefficients α into account. Under L^2 loss, our reconstruction is defined as

$$\begin{aligned} \left\{ \left\langle I^l \right\rangle_{G'} \right\}_l &= \min_{\{I^l\}_l} \sum_{i,j} \frac{1}{L} \sum_{l=1}^{L} \| I_j^l - I_i^l - \left\langle \Delta_{ij} I^l \right\rangle_{MC} \|^2 \\ &+ \lambda \sum_i \| \sum_{l=1}^{L} \alpha_i^l I_i^l - \langle I_i \rangle_{MC} \|^2 \end{aligned} \tag{19}$$

where $I_i^l = \int_{\mathcal{X}_i} h_i(x) b_i^l(x) \mathrm{d}x$ is as defined in Equation (13). The subscript G' is to differentiate the estimator for I_i^l from the naive Poisson reconstruction. The first difference term is the same as in Equation (7) except now it also considers averaging the loss from all the bases. The second term, instead of regularizing each basis directly with the primal estimate, regularizes the reconstructed image with the primal image estimate. While another feasible option is to regularize the weighted basis with the weighted primal estimate, we use the current approach because it has only one λ parameter that empirically affects the results in the same manner as in regular gradient-domain rendering. The final integral estimator is simply $\langle I_i\rangle_{GB} = \sum_{l=1}^{L} \alpha_i^l \left\langle I_i^l \right\rangle_{G'}$. Note that the estimated $\left\{ \left\langle I^l \right\rangle_{G'} \right\}_l$ are unbiased under L^2 because the expected values of $I_j^l - I_i^l - \left\langle \Delta_{ij} I^l \right\rangle_{MC}$ and $\sum_{l=1}^{L} \alpha_i^l I_i^l - \langle I_i\rangle_{MC}$ are both zeros by construction. Thus

$$\mathbf{E}\left[\langle I_i\rangle_{GB}\right] = \mathbf{E}\left[\sum_{l=1}^{L} \alpha_i^l \left\langle I_i^l \right\rangle_{G'}\right] = \sum_{l=1}^{L} \alpha_i^l \mathbf{E}\left[\left\langle I_i^l \right\rangle_{G'}\right] = \sum_{l=1}^{L} \alpha_i^l I_i^l = I_i \tag{20}$$

which implies that our estimator $\langle I_i\rangle_{GB}$ is unbiased under L^2 reconstruction.

3.3. Application to Rendering

We apply our framework to rendering based on the rendering equation [Kaj86] where we have

$$L_o(\mathbf{x},\phi) = L_e(\mathbf{x},\phi) + \int_\Omega \underbrace{L_i(\mathbf{x},\omega)}_{h(\omega)} \underbrace{f_r(\mathbf{x},\phi,\omega)|\mathbf{n}\cdot\omega|}_{g_\phi(\mathbf{x},\omega)} \mathrm{d}\omega \tag{21}$$

where $\mathbf{x}$ is the position, ϕ,ω are the outgoing and incoming directions respectively. We assigned incident radiance L_i to h and the cosine weighted BRDF term $f_r(\mathbf{x},\phi,\omega)|\mathbf{n}\cdot\omega|$ to g. This choice is justified by common scenarios where incident illumination varies smoothly (except for geometric discontinuities) and the BRDF term may vary rapidly over the image due to material variation. We ignore the emission term in our formulation for simplicity.

With a pinhole camera, we have a 2D array of integration problems for image I that involve the first hit vertices $\mathbf{x}_i$ for $i \in [1,W] \times [1,H]$:

$$\begin{aligned} I_i &= \int_\Omega L_i(\mathbf{x}_i,\omega) g_{\phi_i}(\mathbf{x}_i,\omega)\mathrm{d}\omega \\ &\overset{\text{def}}{=} \int_\Omega h_i(\omega) g_i(\omega)\mathrm{d}\omega \overset{\text{def}}{=} \int_\Omega f_i(\omega)\mathrm{d}\omega \end{aligned} \tag{22}$$

where we defined

$$h_i(\omega) \overset{\text{def}}{=} L_i(\mathbf{x}_i,\omega) \quad g_i(\omega) \overset{\text{def}}{=} g_{\phi_i}(\mathbf{x}_i,\omega) \quad f_i(\omega) \overset{\text{def}}{=} g_i(\omega)h_i(\omega) \tag{23}$$

to align with our notations h_i, g_i, and f_i. Anti-aliasing could be achieved by downsampling after rendering a higher-resolution image with this formulation.

Note that $h(\omega)$ is an integral over all possible paths $\bar{\mathbf{p}}$ traced from $\mathbf{x}$ towards direction ω. Shift mapping $T_{ij}(\omega)$ as defined in Equation (4) in a more complete context changes both ω and path $\bar{\mathbf{p}}$. Although the full path formulation is more complete, we focus on the formulation in the angular domain rather than the path integral domain for simplicity, treating $h_i(\omega)$ as a function that returns a stochastic value. As long as the similarity assumption can be designed to hold with path-space shift mapping (i.e., the value return from h_i and h_j satisfies $h_i(\omega) \approx h_j(T_{ij}(\omega))$ for $T_{ij}(\omega) \approx \omega$), our current formulation still satisfies the similarity assumption. There are several popular choices for shift function T_{ij} and we refer readers to the survey by Hua et al. [HGP*19] for more details.

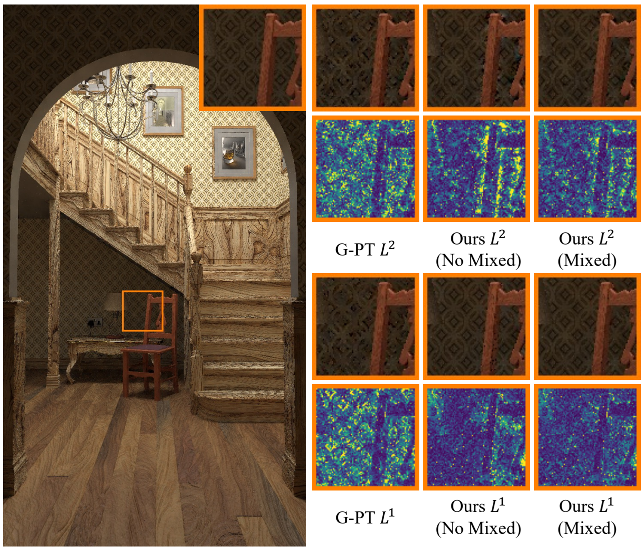


Figure 4: *Our result (128 spp) with/without mixing with G-PT. Our method works well for continuous surfaces with high-frequency texture maps. Artifacts that appear around geometry discontinuities could be effectively handled by combining with G-PT. Note that our method under L^1 reconstruction performs well even without such a combination.*

The reconstruction step is as explained already. Our framework does not need much change over the existing pipeline of gradient-domain rendering, except that we need to incorporate the basis expansion. Our method generally works well for high-frequency albedo and normal maps. However, our difference estimators may still have high variance around geometric discontinuities where incoming radiance h_i changes rapidly, which invalidate our assumptions. Fortunately, because shift mapping already produces samples that can be used for regular-gradient domain rendering, we optionally mix the result of our estimator and regular gradient-domain estimator with inverse variance weight estimated with at least two batches of partial results. We found that this optional combination with G-PT empirically works well. Figure 4 shows the comparison of our method with and without combining with G-PT. It is worth noting that in L^1 reconstruction, our method without such a combination still outperforms regular G-PT because the propagation of error in L^1 is typically not as significant as in L^2.

4. Implementation

Baseline. We implemented our own CPU gradient-domain path tracer (G-PT) [KMA*15] in LuisaRenderer [ZZC*22] and our method on top of the same system. Our implementations are based on the method by [KMA*15] and we chose to omit next-event estimation for simplicity. We employed the reversible path-space shift mapping techniques to follow the suggested implementation by [KMA*15]. We only made slight modifications to regular gradient-domain path tracing to avoid some redundant calculations for our method: splitting the throughput into two parts and storing them: Basis/BSDF evaluation at the primary hit, and incoming throughput after the primary hit. When a path hits a light source, we only need to multiply the basis/BSDF evaluation of the first hit with the throughput after the hit once. Primal estimates and difference estimates for each basis are stored in a buffer for the later reconstruction stage. We use the same sample to evaluate all lossless bases. The only overhead during the rendering stage compared to vanilla G-PT is basis evaluation, which is about 8% of the total rendering time with 9 basis functions.

Reconstruction. We used PyTorch for both L^1 and L^2 reconstructions. Since we deal with convex optimization, we used the L-BFGS optimizer due to its efficiency. It has been observed that L^1 reconstruction often yields visually appealing results but may introduce bias in gradient-domain rendering. This bias is typically manifested as an overall darker image compared to unbiased reconstruction or local blurring when high-frequency details are present. We have observed that employing L^1 loss for differences and L^2 loss for regularization addresses the brightness issue while still producing visually pleasant results. This is because the visual artifacts introduced in L^2 reconstruction are a consequence of over-penalization when noisy difference estimates are present. Therefore, we adapt L^2 regularization for both L^1 and L^2 reconstruction in both vanilla G-PT and our methods. We set $\lambda = 0.1$ for L^2 reconstruction and $\lambda = \sqrt{0.1}$ for L^1. Using the same λ has demonstrated near-optimal performance in our test scenes for both our method and G-PT, consistent with our conclusions in Section 3.2.

Mixture with G-PT. To combine with a regular gradient-domain estimator using inverse variance weight, we divide samples into at least 2 batches and estimate the sample variance for $\langle I \rangle_{GB}$ and $\langle I \rangle_G$. In practice, we find that even 2 batches work effectively well for high sample counts, so we set the batch number to $\max(2, \lfloor \frac{N}{64} \rfloor)$, with N being the number of samples used for rendering, demonstrating good performance in practice.

Choice of Basis. To show that our method could easily adapt different basis functions, we test two different choices of bases: 2-degree spherical harmonics (SH) which has 9 bases and 9 independent box functions that evenly partition the spherical domain similar to the scaling function of the Haar wavelet. We used the Monte Carlo integration approach as in Equation (15) to solve for α_i^l with BSDF sampling as a precomputation. We set the number of samples $K = \max(150, 3N)$ with K defined in Equation (15) and N is the number of samples for rendering. This precomputation typically takes $2-3\%$ of our total rendering time. Once we have coefficients stored in buffers, one for each l, $b_i^l(\omega)$ can be efficiently calculated based on Equation (18).

5. Results

All our experiments run on a desktop with Intel(R) Core(TM) i7-5930K CPU @ 3.50GHZ with 12 threads and NVIDIA RTX 2080Ti GPU. GPU is used only for reconstruction as for reference implementation by [KMA*15].

As our proposed method is designed to address scenarios involving high-frequency material maps, we initiate the experimentation by modifying the simple Cornell-Box scene. This experiment acts as a proof of concept and evaluates the ability of our method to handle changes in albedo, normal, and roughness with our chosen global basis. Our initial step involves generating a high-frequency checkerboard for albedo, creating a normal map by treating the checkerboard with Gaussian blur as the height map, and adjusting roughness by remapping the black and white range between 0.3 and 1. We conduct tests with only one property toggled at a time. The results are visually demonstrated in Figure 5, and the numerical result is shown with convergence plots in Figure 6. Our experiments under both L^1 and L^2 reveal that our method exhibits superior performance in handling changes in albedo and normal. While we observe a decreased performance compared to G-PT for high-frequency roughness, it is noteworthy that the error in this case is already low with G-PT itself compared to other scenarios.

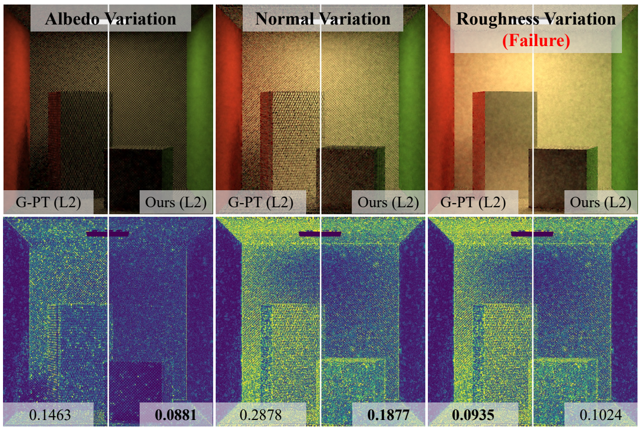


Figure 5: *Cornell-box test scenes with high-frequency albedo, normal, and roughness variation respectively. The top row shows the rendering result of G-PT and our method with spherical harmonics. The bottom row shows the corresponding MAPE error image. All scenes are rendered with 128 samples by our method and 147 samples by G-PT for same-time comparison. We see that with high-frequency albedo and normal map toggled, our method shows consistent improvement. When the roughness map is toggled our method performs comparable to G-PT, while the error is relatively low compared to the other two cases.*

As our current lossless bases focus on maintaining information in low frequency, the variance of the difference is dominated by the higher frequency "noise" defined in Section 3.1. Consequently, the "failure" case in Figure 5 where using constant albedo and normal with varying roughness only may not be conclusive, as high-frequency signals remain "similar" across different pixels. Therefore, we further explore how high-frequency noise affects the performance of our method. In the ablation study, we separate roughness from the choice of the checkerboard-patterned surface map, toggling both the albedo and normal map and gradually setting roughness as a global surface variable. The results are demonstrated in Figure 7. We observe that with decreasing roughness, the performance of our method decreases as expected due to high approximation error in our current choices of bases.

To test our method in a more realistic setting, we modify scenes from the rendering resource [Bit16] by applying different materi-

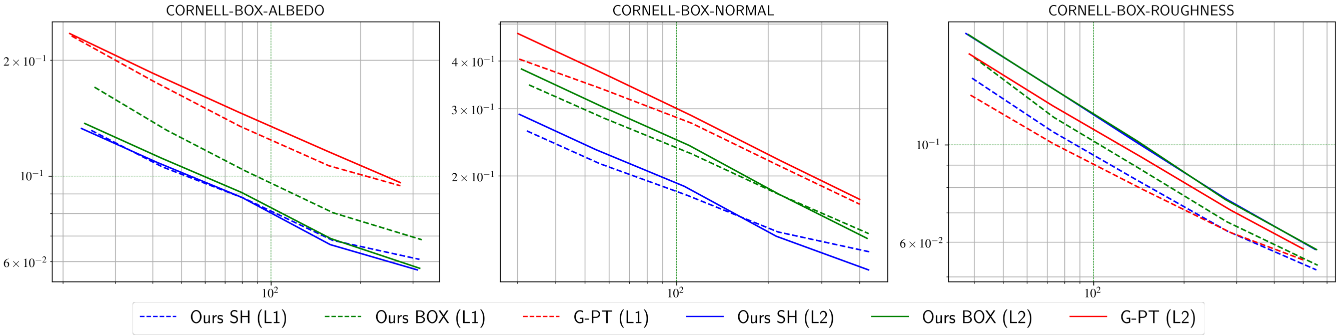


Figure 6: *Convergence plot (MAPE vs. Time) for high-frequency albedo, normal, and roughness toggled respectively with the Cornell box scene. The convergence plot with red represents G-PT, green for our method with box function, and blue for our method with spherical harmonics. Notice the error in the roughness toggled plot while G-PT performs better, the error is relatively low compared to the other cases.*

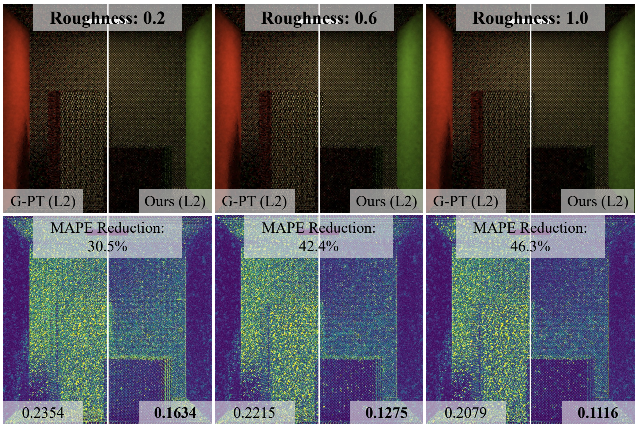


Figure 7: *Visualization of error images with G-PT and our reconstruction with gradually changing roughness from 0.2 to 1.0. Our method still shows variance reduction even in low roughness settings with high-frequency albedo and normal maps. While higher roughness value allows more informative basis expansion will result in better variance reduction overall.*

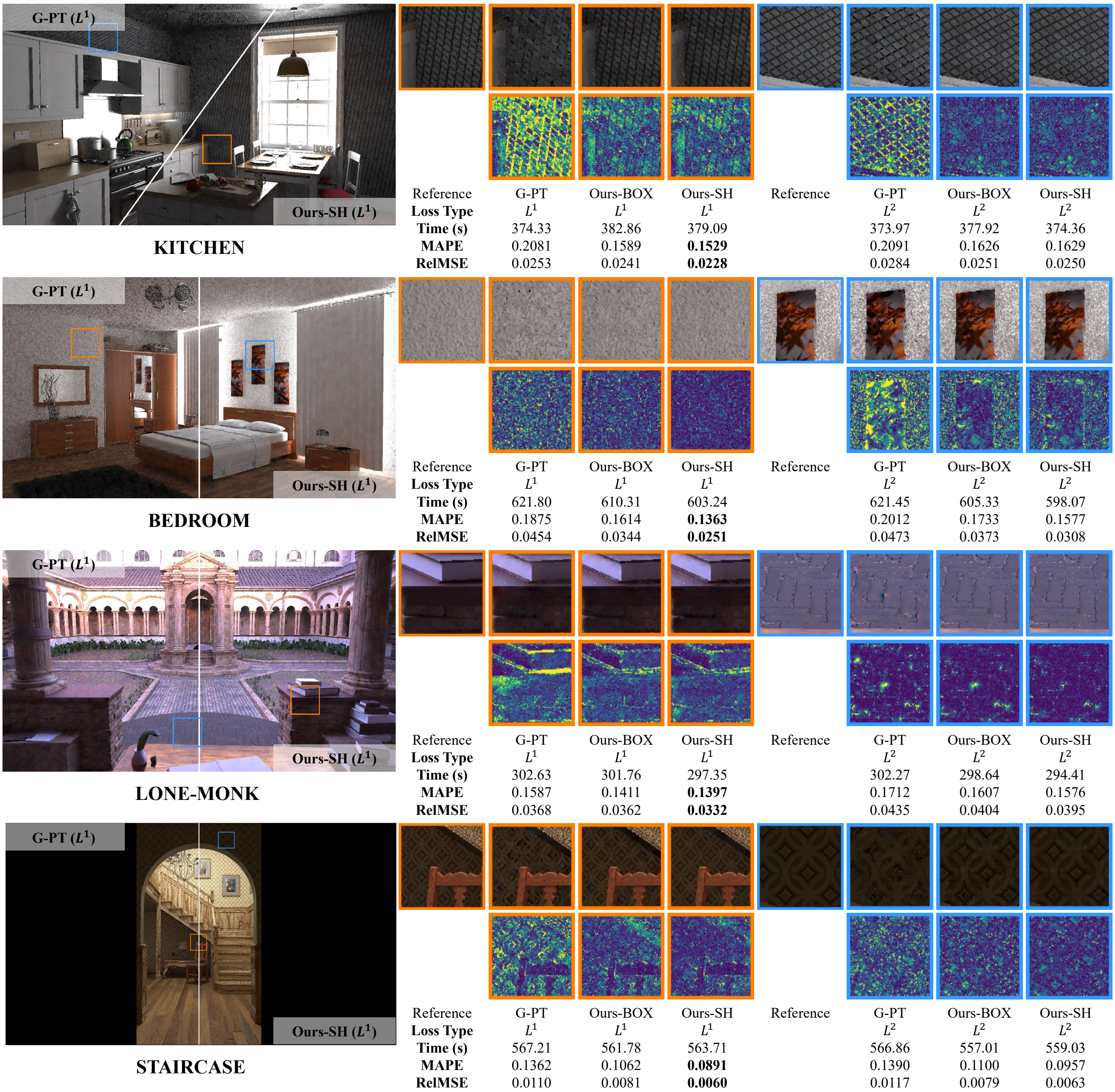
als with varying albedo, normal, and roughness on six test scenes: Staircase, Bedroom, Kitchen, Living-Room, Bathroom, and additionally Lone-Monk. We demonstrate our results in Figure 8 with the same-time convergence plot in Figure 10. For Staircase, Bedroom, Kitchen, and Lone-Monk, our method is both numerically and visually better than G-PT for both L^1 and L^2, respectively. Upon closer inspection of the Staircase scene, areas under low light conditions, which originally suffered from over-blurring in L^1 reconstruction and displayed noisy artifacts in L^2, are effectively captured by our method. In the Kitchen scene, where we apply a varying grid pattern texture map with varying albedo and roughness, vanilla G-PT struggles with noisy gradients around feature discontinuities. Our method decouples these discontinuities and results in an overall much cleaner result. In the Bedroom scene with a high-frequency normal map, as anticipated, we achieve further variance reduction, particularly when high-frequency features dominate the screen. The improvement in the lone-monk scene is somewhat limited due to a lower frequency material map compared to previous scenes. Our method still preserves more details under L^1 and effectively removes artifacts around sharp features under L^2 (commonly termed as dipole artifacts).

We also include some failure cases. In Figure 9, the living-room scene, while qualitatively comparable to G-PT due to darkness in the texture and lack of image features, shows in the error image and the convergence plot that our method with both BOX and SH still outperforms G-PT. The bathroom scene in Figure 9 demonstrates a "failure" case, where varying features are far apart from each other in general. However, our method still performs visually competitively (notice the wooden pattern around the bathtub) and quantitatively better convergence than G-PT under the same-time comparison.

6. Limitation and Future Work

Similar to related work in gradient-domain rendering [ZSWL21, MVZ16, KHL19], our method currently does not support depth of field or motion blur effect, as well as direct Monte Carlo integration within each pixel to have antialiasing. This issue is common to any technique that uses pixel-based information, such as per-pixel shading normal or albedos, so it is not unique to our method. For cases where we would like multiple subpixel samples, the computation for coefficients must be done for each sample, which introduces extra computation and memory costs. Using a common set of coefficients by varying residual distribution does not help due to our assumption that coefficients could vary significantly across pixels. One potential solution is to precompute the coefficients for arbitrary normal directions, but we leave it as future work. We want to mention that for antialiasing, rendering at a high resolution and downsampling is often the preferable approach because tone mapping should be done before antialiasing to properly deal with aliasing in HDR images. Our current method is compatible with this common practice.

Our method does not consider stochastic BSDF evaluation [HHdD16]. It could still help in variation reduction when conditional BSDF evaluation on the incoming direction has sufficiently low variance. Our current implementation does not handle specular reflection towards secondary hits and beyond. When encountered,



KITCHEN

Reference	G-PT	Ours-BOX	Ours-SH	Reference	G-PT	Ours-BOX	Ours-SH
Loss Type	L^1	L^1	L^1		L^2	L^2	L^2
Time (s)	374.33	382.86	379.09		373.97	377.92	374.36
MAPE	0.2081	0.1589	**0.1529**		0.2091	0.1626	0.1629
RelMSE	0.0253	0.0241	**0.0228**		0.0284	0.0251	0.0250

BEDROOM

Reference	G-PT	Ours-BOX	Ours-SH	Reference	G-PT	Ours-BOX	Ours-SH
Loss Type	L^1	L^1	L^1		L^2	L^2	L^2
Time (s)	621.80	610.31	603.24		621.45	605.33	598.07
MAPE	0.1875	0.1614	**0.1363**		0.2012	0.1733	0.1577
RelMSE	0.0454	0.0344	**0.0251**		0.0473	0.0373	0.0308

LONE-MONK

Reference	G-PT	Ours-BOX	Ours-SH	Reference	G-PT	Ours-BOX	Ours-SH
Loss Type	L^1	L^1	L^1		L^2	L^2	L^2
Time (s)	302.63	301.76	297.35		302.27	298.64	294.41
MAPE	0.1587	0.1411	**0.1397**		0.1712	0.1607	0.1576
RelMSE	0.0368	0.0362	**0.0332**		0.0435	0.0404	0.0395

STAIRCASE

Reference	G-PT	Ours-BOX	Ours-SH	Reference	G-PT	Ours-BOX	Ours-SH
Loss Type	L^1	L^1	L^1		L^2	L^2	L^2
Time (s)	567.21	561.78	563.71		566.86	557.01	559.03
MAPE	0.1362	0.1062	**0.0891**		0.1390	0.1100	0.0957
RelMSE	0.0110	0.0081	**0.0060**		0.0117	0.0079	0.0063

Figure 8: *Equal time visual comparison reconstruction result for G-PT, our method with spherical harmonics and box basis. We include rendered result and error image visualization with MAPE metric for both L^1 (red box) and L^2 (green box) reconstruction. corresponding MAPE and RelMSE for the full image are provided as numerical reference. In almost all cases our method with spherical harmonics performs best both visually and numerically. Note that in most of the cases box function still outperforms regular G-PT.*

it automatically falls back to regular gradient-domain rendering for such pixels. Although we could apply our method at the first hit with non-perfect specular BSDF, in general, such a scenario can only be handled properly with world space techniques, which we leave as future work.

Our numerical experiments show that spherical harmonics consistently outperform the box function. However, we have not identified any theoretical reason behind it. By designing a better choice of global bases under the same lossless basis construction, as discussed in Section 3.1, we could ideally achieve even further im-

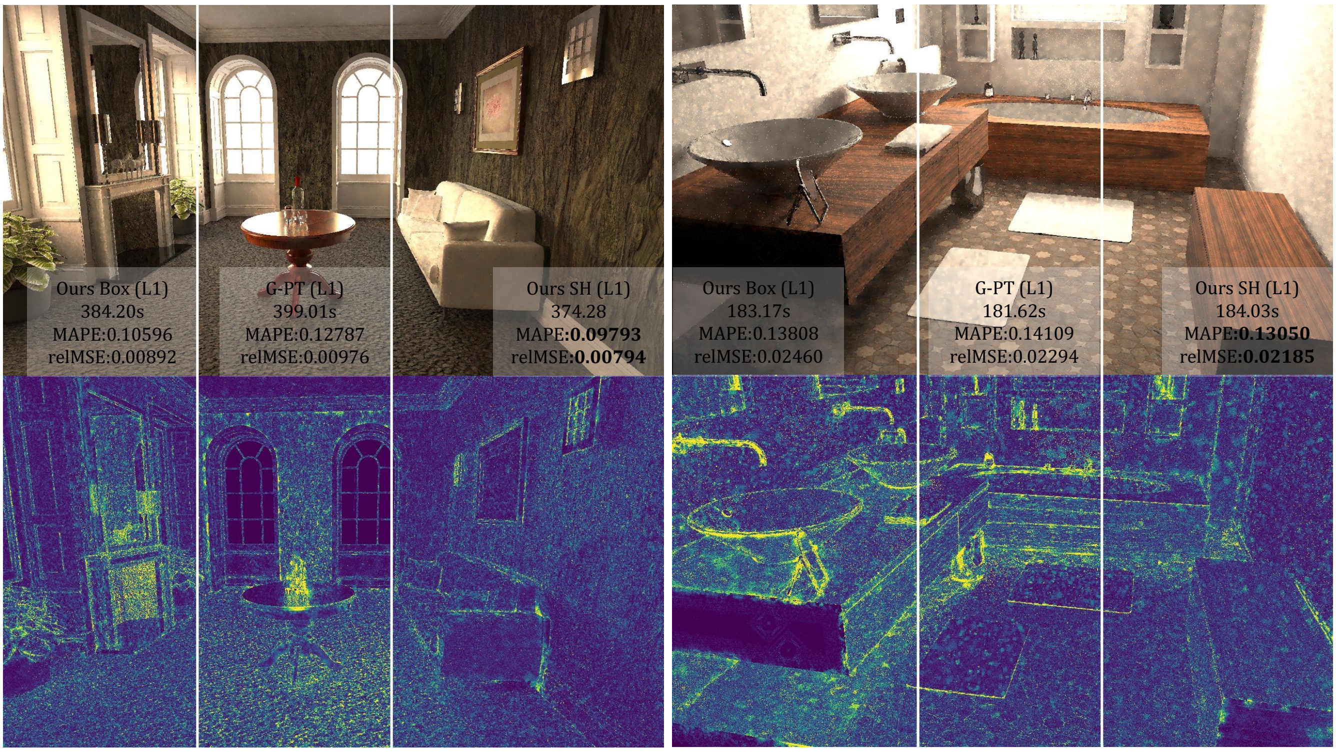


Figure 9: *The scene on the left (LIVING-ROOM) indeed contains high-frequency information. While the lack of visible features and dark texture limits the qualitative improvement, we can still visualize the quantitative improvement from the error image. The scene on the right (BATHROOM) lacks high-frequency features. Although the tiles on the floor are patterned and have similar colors, they dominate many pixels, restricting the benefit of our algorithm. It is worth noting that our method with spherical harmonics (SH) still reduces error for the wooden pattern on the side of the bathtub, and overall, it still performs numerically better than the baseline G-PT.*

provement. Additionally, while our results experimentally show performance improvements, we currently lack theoretical understanding compared to G-PT. We ideally would like to have a provable improvement over G-PT since G-PT is a special case of our method. Although our combination with G-PT helps get rid of artifacts introduced by our method, a better reconstruction technique or a better design of pixel-wise basis could yield a more stable reconstruction step.

Inspired by prior work on advanced reconstruction in gradient-domain rendering [MVZ16, KHL19, GLL*19], we believe that incorporating more advanced reconstruction techniques into our basis integral formulation could lead to immediate improvements. One can interpret our decomposition of gradient-domain rendering as a more effective path reuse among pixels, which may have applications in some recent work on spatial-temporal path reuse [LKB*22, OLK*21, BWP*20]. While our current method is designed for offline use, we envision that a more specialized lossless basis expansion focused on cosine-weighted BSDFs evaluation could potentially facilitate real-time applications in the future.

Another future work is in the realm of generalized denoising. Modern denoising methods rely on auxiliary image space data [Áfr23, KHL19, ZJL*15] to capture pixel-dependent information and to avoid over-blurring of sharp features. However, high-frequency variation, even if present in those auxiliary data does not necessarily indicate entirely different pixel intensities. For example, it is often the case that incident illumination is smooth even when albedo changes rapidly [WRC88, KGPB05]. Our basis expansion approach decouples this high-frequency per-pixel variation, and this concept can be useful to apply to denoising. Our framework itself should also be applicable to any problems where we need to solve multiple integrals that are correlated to each other.

Acknowledgments

We thank Xiaochun Tong for providing implementation advice on LuisaRenderer, and Xin Peng for helping to make our teaser figure. We also appreciate the constructive feedback from the reviewers, which significantly improved the clarity of our paper. This project was funded by the Natural Sciences and Engineering Research Council of Canada (NSERC) under Grant No. RGPIN-2020-03918.

References

[Áfr23] Áfra A. T.: Intel® Open Image Denoise, 2023. https://www.openimagedenoise.org. 10

[Bit16] Bitterli B.: Rendering resources, 2016. https://benedikt-bitterli.me/resources/. 7

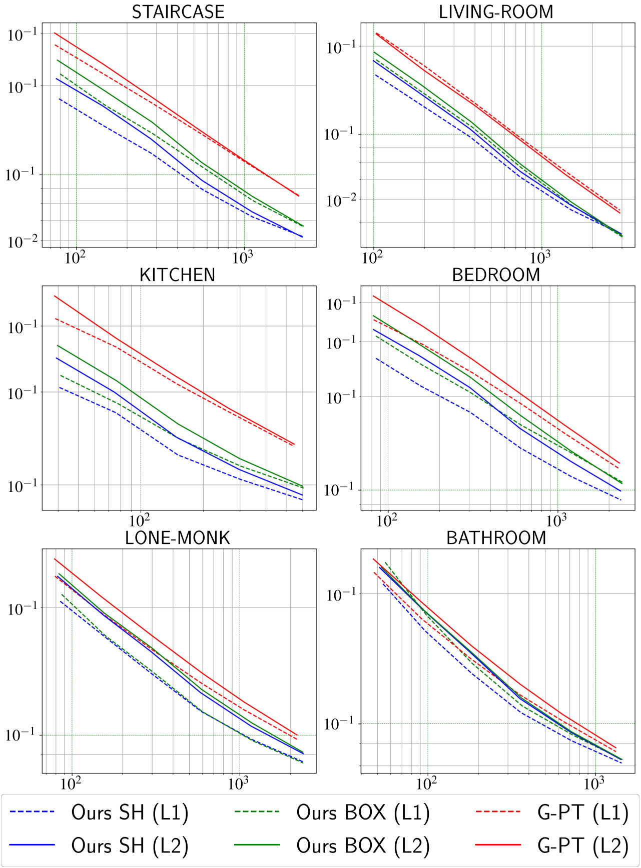


Figure 10: *The convergence plot for both L^1 and L^2 reconstructions is presented. Dashed lines represent L^1, and solid lines represent L^2. It is evident that under the same reconstruction loss, our method with both choices of basis consistently outperforms G-PT.*

[BPE17] Bauszat P., Petitjean V., Eisemann E.: Gradient-domain path reusing. *ACM Transactions on Graphics (TOG) 36*, 6 (2017), 1–9. 2

[BWP*20] Bitterli B., Wyman C., Pharr M., Shirley P., Lefohn A., Jarosz W.: Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting. *ACM Transactions on Graphics (TOG) 39*, 4 (2020), 148–1. 10

[GHV*18] Gruson A., Hua B.-S., Vibert N., Nowrouzezahrai D., Hachisuka T.: Gradient-domain volumetric photon density estimation. *ACM Transactions on Graphics (TOG) 37*, 4 (2018), 1–13. 2

[GLL*19] Guo J., Li M., Li Q., Qiang Y., Hu B., Guo Y., Yan L.-Q.: GradNet: unsupervised deep screened poisson reconstruction for gradient-domain rendering. *ACM Transactions on Graphics (TOG) 38*, 6 (2019), 1–13. 10

[HGNH17] Hua B.-S., Gruson A., Nowrouzezahrai D., Hachisuka T.: Gradient-domain photon density estimation. In *Computer Graphics Forum* (2017), vol. 36, Wiley Online Library, pp. 31–38. 2

[HGP*19] Hua B.-S., Gruson A., Petitjean V., Zwicker M., Nowrouzezahrai D., Eisemann E., Hachisuka T.: A Survey on Gradient-Domain Rendering. In *Computer Graphics Forum* (2019), vol. 38, Wiley Online Library, pp. 455–472. 2, 3, 6

[HHdD16] Heitz E., Hanika J., d'Eon E., Dachsbacher C.: Multiple-scattering microfacet BSDFs with the Smith model. *ACM Trans. Graph. 35*, 4 (jul 2016). 8

[Kaj86] Kajiya J. T.: The Rendering Equation. *SIGGRAPH Comput. Graph. 20*, 4 (aug 1986), 143–150. 2, 6

[KGPB05] Křivánek J., Gautron P., Pattanaik S., Bouatouch K.: Radiance caching for efficient global illumination computation. *IEEE Transactions on Visualization and Computer Graphics 11*, 5 (2005), 550–561. 10

[KHL19] Kettunen M., Härkönen E., Lehtinen J.: Deep convolutional reconstruction for gradient-domain rendering. *ACM Transactions on Graphics (TOG) 38*, 4 (2019), 1–12. 8, 10

[KMA*15] Kettunen M., Manzi M., Aittala M., Lehtinen J., Durand F., Zwicker M.: Gradient-Domain Path Tracing. *ACM Trans. Graph. 34*, 4 (2015). 2, 3, 6, 7

[LKB*22] Lin D., Kettunen M., Bitterli B., Pantaleoni J., Yuksel C., Wyman C.: Generalized resampled importance sampling: foundations of ReSTIR. *ACM Transactions on Graphics (TOG) 41*, 4 (2022), 1–23. 10

[LKL*13] Lehtinen J., Karras T., Laine S., Aittala M., Durand F., Aila T.: Gradient-Domain Metropolis Light Transport. *ACM Trans. Graph. 32*, 4 (2013). 2, 3

[MKA*15] Manzi M., Kettunen M., Aittala M., Lehtinen J., Durand F., Zwicker M.: Gradient-Domain Bidirectional Path Tracing. In *EGSR (EI&I)* (2015), pp. 65–74. 2, 5

[MKD*16] Manzi M., Kettunen M., Durand F., Zwicker M., Lehtinen J.: Temporal gradient-domain path tracing. *ACM Transactions on Graphics (TOG) 35*, 6 (2016), 1–9. 2

[MVZ16] Manzi M., Vicini D., Zwicker M.: Regularizing image reconstruction for gradient-domain rendering with feature patches. In *Computer graphics forum* (2016), vol. 35, Wiley Online Library, pp. 263–273. 4, 8, 10

[NRH04] Ng R., Ramamoorthi R., Hanrahan P.: Triple product wavelet integrals for all-frequency relighting. In *ACM SIGGRAPH 2004 Papers*. 2004, pp. 477–487. 4

[OLK*21] Ouyang Y., Liu S., Kettunen M., Pharr M., Pantaleoni J.: ReSTIR GI: Path resampling for real-time path tracing. In *Computer Graphics Forum* (2021), vol. 40, Wiley Online Library, pp. 17–29. 10

[SKS23] Sloan P.-P., Kautz J., Snyder J.: Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*. 2023, pp. 339–348. 4

[SSC*17] Sun W., Sun X., Carr N. A., Nowrouzezahrai D., Ramamoorthi R.: Gradient-Domain Vertex Connection and Merging. In *EGSR (EI&I)* (2017), pp. 83–92. 2

[WRC88] Ward G. J., Rubinstein F. M., Clear R. D.: A ray tracing solution for diffuse interreflection. In *Proceedings of the 15th annual conference on Computer graphics and interactive techniques* (1988), pp. 85–92. 10

[XZW*22] Xu Z., Zeng Z., Wu L., Wang L., Yan L.-Q.: Lightweight Neural Basis Functions for All-Frequency Shading. In *SIGGRAPH Asia (Conference Track)* (November 2022). 4

[ZJL*15] Zwicker M., Jarosz W., Lehtinen J., Moon B., Ramamoorthi R., Rousselle F., Sen P., Soler C., Yoon S.-E.: Recent advances in adaptive sampling and reconstruction for Monte Carlo rendering. In *Computer Graphics Forum* (2015), vol. 34, Wiley Online Library, pp. 667–681. 10

[ZSWL21] Zhuang T., Shen P., Wang B., Liu L.: Real-time Denoising Using BRDF Pre-integration Factorization. In *Computer Graphics Forum* (2021), vol. 40, Wiley Online Library, pp. 173–180. 2, 5, 8

[ZZC*22] Zheng S., Zhou Z., Chen X., Yan D., Zhang C., Geng Y., Gu Y., Xu K.: LuisaRender: A High-Performance Rendering Framework with Layered and Unified Interfaces on Stream Architectures. *ACM Trans. Graph. 41*, 6 (nov 2022). 6