Pacific Graphics 2018
H. Fu, A. Ghosh, and J. Kopf
(Guest Editors)

Volume 37 (2018), Number 7

Feature Generation for Adaptive Gradient-Domain Path Tracing

Jonghee Back[1], Sung-Eui Yoon[2] and Bochang Moon[1]

[1]GIST, South Korea
[2]KAIST, South Korea

Figure 1: *Given gradient-domain path tracing (G-PT), we visualize the noisy input color (a) and gradients, (b) and (c), and our feature image (d). Our feature (d) is fed into a state-of-the-art denoising method, adaptive polynomial rendering (APR), and this denoiser greatly reduces a numerical error, i.e., relative mean squared error (relMSE), thanks to our high-quality feature image.*

Abstract

In this paper, we propose a new technique to incorporate recent adaptive rendering approaches built upon local regression theory into a gradient-domain path tracing framework, in order to achieve high-quality rendering results. Our method aims to reduce random artifacts introduced by random sampling on image colors and gradients. Our high-level approach is to identify a feature image from noisy gradients, and pass the image to an existing local regression based adaptive method so that adaptive sampling and reconstruction using our feature can boost the performance of gradient-domain rendering. To fulfill our idea, we derive an ideal feature in the form of image gradients and propose an estimation process for the ideal feature in the presence of noise in image gradients. We demonstrate that our integrated adaptive solution leads to performance improvement for a gradient-domain path tracer, by seamlessly incorporating recent adaptive sampling and reconstruction strategies through our estimated feature.

CCS Concepts

•***Computing methodologies*** → *Ray tracing;*

1. Introduction

Monte Carlo (MC) ray tracing [Kaj86] has been a widely used rendering solution thanks to its generality in terms of synthesizing a variety of photorealistic rendering effects. The fundamental drawback of MC ray tracing is, however, that it requires a huge amount of ray samples to get rid of random noise stemming from MC integration.

Recently, gradient-domain rendering methods [LKL*13, KMA*15] were proposed as a novel, yet effective solution to employ image gradients in addition to radiance values. These methods robustly estimate the gradients by building offset light paths, which have a high correlation to original light paths. Especially, the gradient-domain path tracing (G-PT) [KMA*15] has demonstrated that a high-quality image can be generated by a Poisson reconstruction method, even with a simple modification to an existing rendering solution, i.e., path tracing.

The gradient-domain rendering approaches show promising results that can lead to an alternative of existing rendering solutions by significantly reducing noise (i.e., variance), but these methods still suffer from the random artifacts as estimated gradients are also random variables that can have high variances given a small number of samples.

As an active research direction with regard to reducing MC noise, image-space adaptive rendering methods (e.g., [SD12,

RMZ13]) have been successfully applied to existing MC ray tracing methods, while exploiting simplicity introduced by its image-space nature. The key technical components of those techniques are an adaptive reconstruction that fully utilizes rendering-specific features (e.g., normals, textures, and depths) and an adaptive sampling that distributes additional samples non-uniformly to minimize errors of reconstructed images.

As a recent attempt in this direction, local regression based adaptive methods [MCY14, MMMG16] demonstrated that high-quality reconstruction output can be produced when a high correlation between the features and ground truth images exists. However, it is still unclear how those can be employed for a gradient-domain rendering framework while fully employing the additional information, i.e., image gradients, in their adaptive rendering methods. In this paper, we present a new technical solution to allow for seamlessly integrating recent local regression based adaptive methods into a gradient-domain method, e.g., G-PT.

To this end, our key idea is to create a new feature using image gradients and feed the feature into adaptive methods so that both techniques (gradient-domain path tracing and adaptive approach) become mutually complementary for generating high-quality rendering output. Implementing the idea, however, requires to address a fundamental question, what is the ideal feature for the local regression based adaptive techniques. This question leads us to provide a solution that integrates the gradient-domain path tracing with a local regression based adaptive method. Specifically, we propose the following technical contributions:

- We derive an ideal feature for the state-of-the-art reconstruction based on local regression and transform the estimation problem for the ideal feature into the standard Poisson reconstruction problem, where image gradients can be employed.
- A weighted least squares is introduced to deal with heterogeneous variances of estimated gradients. This comes with a minor modification on the standard Poisson reconstruction, but with a high accuracy improvement.
- We adapt a bootstrap aggregating (bagging) to reduce the variances of estimated ideal features, and this allows for using an existing pre-filtering of recent adaptive methods by providing reconstruction variances of our feature.

We have demonstrated that our technical idea outperforms the state-of-the-art method [MVZ16], which extends a screened Poisson reconstruction to employ rendering-specific features for the gradient-domain rendering. Our result shows a performance improvement (e.g., up to $2\times$ lower error than the previous method) for a gradient-domain path tracer, by feeding our estimated ideal feature to a state-of-the-art adaptive method.

2. Related Work

In this section, we review prior methods directly related to our work.

Gradient-Domain Rendering. Gradient has been widely used as an additional information in the area of photorealistic rendering. One of the typical examples is the irradiance caching [WH92] and radiance caching techniques [KGPB05, JDZJ08] where gradients are employed to improve interpolation accuracy of irradiance or radiance values.

Recently, gradient-domain rendering has been studied as an alternative to Monte Carlo ray tracing methods. The gradient-domain rendering method [LKL*13] extended the existing Metropolis light transport to estimate image gradients, i.e., finite differences between adjacent pixels, followed by solving a screened Poisson equation for generating a final image. Kettunen et al. [KMA*15] demonstrated that a simple modification (e.g., offset light paths with a pixel distance) to path tracing allows for estimating image gradients efficiently, and reconstructed a final image using a screened Poisson reconstruction that takes the gradients as well as image colors as input. In addition, the Poisson reconstruction was recently extended from 2D to 3D so that the temporal coherence between frames can be utilized [MKD*16]. These methods demonstrated that reconstructed images using estimated gradients can have less variances than ordinary rendering output (e.g., path traced images), thanks to a high correlation between base and offset light paths.

These gradient-domain techniques utilized the sparsity of gradients of natural images [PGB03], but a challenge in rendering is that the gradients are random variables analogous to colors. For example, Kettunen et al. [KMA*15] computed image gradients with a sample mean of gradient samples, and those gradients are also affected by random noise unless an infinite number of samples are used. Furthermore, those errors (i.e., variances of gradients) have the heterogeneous property that the variances of estimated gradients can vary significantly across image regions. This challenge introduced by the heterogeneous noise motivates the usage of adaptive sampling and reconstruction, and we propose an integrated solution between the gradient-domain rendering and adaptive methods via a novel feature generation.

Adaptive Rendering with Features. Adaptive sampling and reconstruction has a long history [Kaj86]. A notable research direction for the adaptive rendering was to analyze a local frequency of light transport [DHS*05, BBS14], and a comprehensive survey on this topic is available [ZJL*15]. We focus on the image-space adaptive approaches that employ rendering-specific features (e.g., normal, texture, depth and visibility buffers), as those are directly related to our work.

Sen and Darabi [SD12] introduced a mutual information to statistically measure a functional relationship between random parameters and features, and optimized denoising parameters. Rousselle et al. [RMZ13] proposed an adaptive sampling and reconstruction technique that utilizes a variety of features such as G-buffers (normal, texture, and depth buffers) and caustics. Recently, Moon et al. [MCY14] applied a local regression theory to this adaptive rendering problem, and estimated optimal denoising parameters and adjusted sampling density across the image. This local regression based method was improved in terms of efficiency via sparse reconstruction [MIGYM15], or robustness through high-order polynomials [MMMG16] and non-local means weighting [BRM*16]. The main advantage guided by these adaptive methods is that these approaches are able to handle the heterogeneous noise appropriately by adjusting denoising parameters and sampling density across the

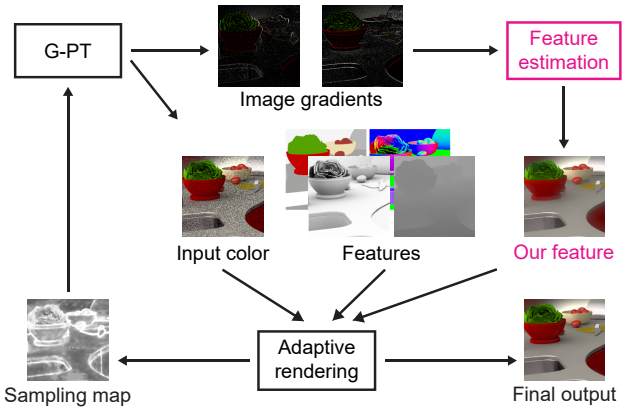

Figure 2: *This figure illustrates the overall flow of our approach utilizing ideal features within the gradient-domain path tracer (G-PT). Our method also allows for recent adaptive reconstruction and sampling, which performs error analysis and generates sampling maps for G-PT. Our main contribution lies on the feature generation module (shown in the magenta color) utilizing image gradients generated by G-PT. Since our technique of generating and utilizing features is well modularized, an adaptive rendering method can be integrated into the G-PT framework without major changes in the existing approaches.*

image. These methods, however, were mainly integrated into the ordinary Monte Carlo rendering framework.

Recently, Manzi et al. [MVZ16] proposed a regularized reconstruction that employs rendering-specific features (and its gradients) in the gradient-domain rendering framework [KMA*15]. It enabled to improve the original reconstruction, while preserving high-frequency details contained in feature buffers. However, producing high-quality results is still challenging in regions where features fail to predict edges, as the high-frequency information should be reconstructed by noisy gradients in this case. Our method also aims at keeping high-frequency edges, while reducing noise for the gradient-domain rendering. Departing from this approach, we take a totally different approach, and propose a new integrated solution for performing adaptive rendering (sampling and reconstruction) given a gradient-domain renderer, by passing a novel feature computed from image gradients into a local regression based adaptive method.

3. Ideal Feature for Reconstruction

In this section, we describe local regression based reconstruction methods that utilize features as edge-stopping functions, and then derive an ideal feature that minimizes reconstruction errors.

3.1. Local Regression based Reconstruction

The local regression methods assume a general statistical model:

$$\tilde{g}(i) \equiv \tilde{g}(f_i) = g(f_i) + \varepsilon_i, \tag{1}$$

where $\tilde{g}(i)$ is the noisy intensity at a pixel $i = (x,y)$ generated by Monte Carlo ray tracing. The image function $\tilde{g}(i)$ can be defined on a high-dimensional feature space f_i that includes the pixel position and rendering-specific information such as normal, texture, and depth values [MCY14]. The input intensity $\tilde{g}(f_i)$ is assumed to be corrupted by a random error ε_i, which is an independent and identically distributed value with the zero mean $E(\varepsilon_i) = 0$ and also has a finite variance. The ultimate goal of the reconstruction is to reconstruct the unknown term $g(f_i)$ from the observed input $\tilde{g}(f_i)$.

Based on the statistical model, the regression methods locally approximate the ground truth, $g(f_i)$, with a k-order function using:

$$g(f_i) \approx g(f_c) + \sum_{j=1}^{k} g^{(j)}(f_c) \frac{(f_i - f_c)^j}{j!}, \tag{2}$$

where f_i and f_c are feature values at pixel i and c and the superscript (j) denotes j-times differentiation. We use the Taylor approximation (Eq. 2) that assumes 1D feature for simplicity, but this can be easily extended to a multi-dimensional version for a high-dimensional feature. When an enough order k is provided, the approximation error becomes zero as this approximation function goes to a function that interpolates each value $g(f_i)$, as discussed in a recent regression based denoising paper [MMMG16].

Technically, the approximation error (i.e., remainder terms of Taylor series) is directly related to the reconstruction bias $E(\hat{g}(f_i) - g(f_i))$, where $\hat{g}(f_i)$ is the final output by a local regression based reconstruction. A mathematical derivation for the bias expression is available in a local regression literature [RW94]. As a special case, if the approximation (Eq. 2) by a local model is exact, the approximation error (also the bias) becomes zero. Unfortunately, the variance tends to be increased as the order, k, of a local model increases. This trade-off between the variance and bias with regard to the order k has been widely studied in local regression literatures [CL96].

Based on this regression theory, we present a technique to generate a new feature f_i, in order to reduce the approximation error even with a low order k while maintaining a low variance for a gradient-domain framework. Our approach may be considered similar to the prior reconstruction method [MVZ16] for a gradient-domain rendering system, as both methods are reducing random artifacts caused by noise of estimated gradients. However, the main difference is that our technique constitutes a new feature and feeds the feature into existing adaptive rendering methods as an additional one so that both adaptive sampling and reconstruction functionalities can be utilized for a gradient-domain rendering framework.

3.2. Ideal Feature for Local Regression

While local regression methods for reconstruction have been widely studied, choosing better or ideal features has been less studied. In this section, we discuss properties of ideal features for the local regression methods.

Ideally, a feature should provide a low approximation error (also a low squared bias) so that a local model with a low order (e.g., linear function) can be selected. In this sense, an ideal feature f_i can be defined as a feature that has a perfect linear correlation with the ground truth:

$$g(f_i) = C_1 f_i + C_2 \ (C_1, C_2 \in \mathbb{R}, C_1 \neq 0). \tag{3}$$

In this case, higher order derivatives (from second derivatives) with respect to the feature f_i become zero, and thus the error by the approximation (Eq. 2) also becomes zero even though the local model is defined as the linear function.

In practice, this linear relationship is not useful for our problem of generating an ideal feature f_i, as it requires the ground truth function $g(f_i)$ that is just the ultimate goal of reconstruction. Hence we further expand this relationship by differentiating $g(f_i)$ with respect to image coordinates i:

$$\begin{aligned} \frac{dg(f_i)}{di} = \frac{dg(f_i)}{df_i}\frac{df_i}{di} \Rightarrow \\ \frac{dg(i)}{di} = C_1 \frac{df_i}{di}. \end{aligned} \tag{4}$$

Note that the feature f_i is also an image function that gives a value at the pixel position i, and $g(f_i) \equiv g(i)$ thanks to Eq. 1. The constant term C_1 can be arbitrary chosen except for $C_1 \neq 0$, and we thus set this constant 1. Therefore, the final form has a simple relationship $\frac{dg(i)}{di} = \frac{df_i}{di}$.

This simple relationship implies that an ideal feature is an image function that has equivalent derivatives to those of the ground truth image function $g(i)$. Furthermore, according to this relationship, computing the ideal feature under the linear model is translated into solving a Poisson equation, which has been extensively studied. According to the best of our knowledge, the relationship between an ideal feature and Poisson equation has not been discussed in the related field.

Poisson Reconstruction. Solving the Poisson equation has been studied well in gradient-domain image processing such as image composition [Aga07]. Nonetheless, these techniques aim to compute final colors, not the features, when the image gradients $\frac{dg(i)}{di}$ are given. Unfortunately, the ground truth image gradients are unknown in rendering, and thus we should estimate the gradients to compute the ideal features and it typically introduces an estimation error (e.g., noise). In a way, this is also a chicken-and-egg problem, since computing the ideal features is transformed into an estimation problem for image gradients. In the next section, we propose a feature generation process to tackle this technical challenge. Fig. 2 shows our overall process to generate and utilize an ideal feature within a gradient-domain path tracing framework.

4. Ideal Feature Generation

In this section, we propose an estimation process to generate a feature function f_i given the ideal relationship $\frac{dg(i)}{di} = \frac{df_i}{di}$. This ideal relationship leads to the standard Poisson reconstruction problem for 2D images [BCCZ08] that minimizes the following:

$$\underset{f}{\operatorname{argmin}} \left\| \begin{pmatrix} \nabla^{dx} f \\ \nabla^{dy} f \end{pmatrix} - \begin{pmatrix} v^{dx} \\ v^{dy} \end{pmatrix} \right\|^2, \tag{5}$$

where ∇^{dx} and ∇^{dy} are the image gradient operators along x and y directions (e.g., finite differences) applied to f (a vectorized feature image from f_i), and v^{dx} and v^{dy} are given gradients that estimate $\nabla^{dx} g$ and $\nabla^{dy} g$, respectively. The gradient-domain path tracing [KMA*15] proposed a simple, yet effective way to estimate the image gradients (v^{dx} and v^{dy}) by utilizing the offset paths that have a high correlation with an original path. We adopt the previous method that estimates the gradients for our problem.

The previous work [KMA*15] employed the screened Poisson reconstruction that adds a soft-constraint, $\alpha||\tilde{g}(i) - f||^2$, into the minimization goal (Eq. 5) with a user parameter α, but our problem of feature generation does not require this additional term. This is mainly because the ideal feature image is not necessary to become the ground truth image, i.e., C_2 can be an arbitrary number in Eq. 3.

The solution of minimizing the objective function (Eq. 5) has been extensively studied in gradient-domain composition problems [Aga07], and it can be represented by a sparse linear system:

$$L^T L f = L^T v, \tag{6}$$

where L is the sparse matrix consisting of the Laplacian operator. This equation is essentially the normal equation of an ordinary least squares (OLS). In addition, this is a sparse linear system, and thus can be efficiently solved by an iterative solver (e.g., conjugate gradient). As we convert the ideal feature problem into the standard Poisson equation widely used in gradient-domain problems, this reconstruction part can be easily interpreted.

The main challenge in rendering, however, is that the computed gradient value at each pixel is a random variable containing noise, which typically has a non-uniform distribution across image space. For example, the gradient-domain rendering [KMA*15] estimated image gradients as sample means of the finite differences between radiances of offset paths and original random paths. When the number of samples is small, the variance of the gradient values can be high, leading to noisy output, i.e., noisy feature image f. Fig. 3 (d) shows a feature image generated by solving the linear system without modification (Eq. 6).

This noisy feature can lead to a non-trivial problem in reconstruction, since the statistical model of the local regression (Eq. 1) assumes that the feature image f is noise-free. Hence we propose two main technical contributions in subsequent sections to robustly estimate the feature image f given the noisy gradients.

4.1. Weighted Least Squares

We statistically model the image gradient v_i (i.e., the sample mean of estimated gradient samples), at pixel i as the following:

$$v_i = \nabla g_i + e_i, \tag{7}$$

where e_i is a random error that has the zero mean, i.e., $E(e_i) = 0$, and a finite variance $Var(e_i)$.

The solution of OLS (Eq. 6) has known as the best linear unbiased estimator (BLUE), if the errors e_i are uncorrelated with the zero mean and have equal variances (i.e., $e_i = e_j$ when $i \neq j$) under the Gauss-Markov theorem [Sta09]. We are able to suppose that each sample mean v_i is independent each other as the sample mean is constructed by offsets of random samples from unbiased MC ray tracing (e.g., path tracing), but in rendering the homogeneity assumption typically breaks down. In other words, the MC error e_i of

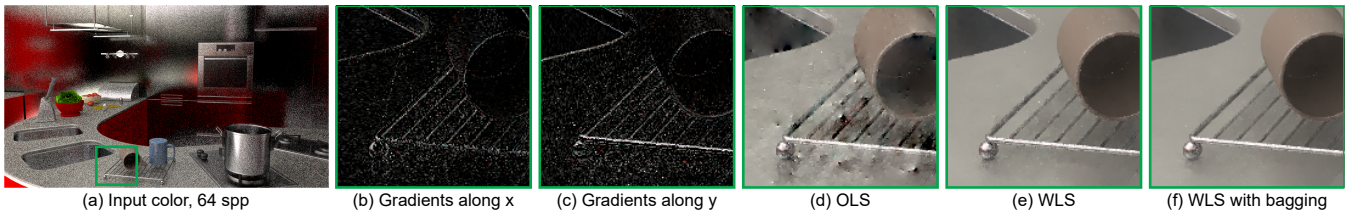

Figure 3: *Features generated by different approaches, given input gradients (b) and (c) from G-PT. The feature image (d) computed by ordinary least squares (OLS) exhibits image artifacts due to heterogeneous errors of estimated image gradients. Weighted least squares (WLS) guided by the variances of sampled gradients mitigates this problem, but leaves some spike noise caused by overfitting (e). Bootstrap aggregating (bagging), which averages the multiple results of WLS, reduces the noise and thus produces an improved feature image (f).*

the observed gradients v_i can significantly vary locally across image regions. As a result, the output of OLS is still unbiased, but not the best in terms of realizing the minimum variance.

Our idea is to employ a statistical adaption, weighted least squares (WLS) [Sta09], to restore the BLUE property by taking account of unequal variances, i.e., heterogeneity:

$$(L^T W L) f = L^T W v, \tag{8}$$

where W is a diagonal matrix that has the element $W_{ii} = \frac{1}{Var(e_i)}$.

In the gradient-domain path tracing, we can estimate $Var(e_i)$ as the variance of sample mean v_i at a pixel i, since the estimated gradients are an average of gradient samples. In the case that the errors e_i are correlated, the weighted matrix W needs to be the inverse of the variance-covariance matrix, but in our case we can simply set W to be the diagonal matrix.

This is a simple modification to the standard Poisson reconstruction, but the quality improvement in terms of noise (i.e., variance) of feature images can be significant, especially when the distribution of the errors e_i is skewed. In Fig. 3, OLS generates artifacts (e.g., dipole artifacts) especially near edges where the variances of estimated gradients are much higher than other parts. Our simple alteration in the reconstruction produces an improved feature f by taking account of unequal variances.

Weighting each row in the linear system (Eq. 8) itself was previously investigated in the L1 reconstruction [KMA*15], which performs an iteratively reweighted least squares where the diagonal matrix W is updated by a function of residuals from the previous iteration. Also, a conceptually similar weighting was designed from a control variates standpoint [RJN16]. Similar to these previous approaches, we weight each gradient to mitigate the heterogeneity in estimated gradients. Nevertheless, our approach adopts a very simple weighting scheme that relies on a different perspective, i.e., Gauss-Markov theorem [Sta09].

Our method drastically reduces the variance, but the estimated feature from WLS still shows some spike noise because of noisy gradients. Technically, this is an overfitting problem occurred in a regression context including least squares fitting. As aforementioned, the noise in feature images can be propagated into the final result of reconstruction methods. We thus propose a general solution to mitigate this overfitting issue in the next section.

Algorithm 1 Robust Feature Generation

Input: Gradient samples $s_i = (s_i^1, ..., s_i^{n_i})$ per each pixel i
Output: Features f and variances $Var(f)$
  for Iteration $b = 1$ to B **do**
    for pixel i **do**
      Resample $|s_i|$ gradient samples, $s_i(b)$, from the set s_i
      Compute the sample mean $v_i(b) = \frac{\sum_{j=1}^{n_i} s_i^j}{n_i}$
      Calculate the variance of the sample mean $Var(v_i(b))$
    end for
    Solve $(L^T W L) f(b) = L^T W v(b)$ with $W_{ii} = \frac{1}{Var(v_i(b))}$ (Sec. 4.1)
  end for
  for pixel i **do**
    Compute the sample mean $f_i = \frac{1}{B}\sum_{b=1}^{B} f_i(b)$
    Compute the variance of the feature $Var(f_i)$
  end for
  return Features f and variances $Var(f)$

4.2. Bagging with Adaptive Weights

Given the WLS solution, we introduce a bootstrap aggregating, also called bagging, which is commonly used to avoid overfitting (i.e., reduce variance) in machine learning techniques (e.g., decision tree methods [Alp04]). Bagging is a simple but general process, which averages multiple outputs of regression (e.g., WLS) by computing new training sets from an input set (e.g., gradient samples). The sets are randomly generated with replacement by a uniform random function from the input set.

Our bagging process is built upon the original bagging method [Alp04] that changes sample data by resampling in each run, but we make adaptation tailored to our problem. In particular, we measure reliability of each bootstrapped sample set by the variance. In other words, we use the uniform random function that selects problematic samples (i.e., outliers) equally compared to other normal samples, but the problematic case tends to have a higher variance than the case where no outliers are chosen. The variances as well as the resampled gradients are passed to our regression, WLS, to perform a least squares that down-weights unreliable samples by the variances. Algorithm 1 shows the pseudocode of our bagging process for generating a robust feature image f.

In Fig. 3, our proposed bagging alleviates the overfitting issue.

In addition, an inherent advantage of the bagging process is that it allows us to compute the variance of our estimated feature f, which can be utilized in pre-filtering of recent reconstruction methods. We employ a recent non-local means filter [RMZ13] to further reduce the remaining noise (i.e., $Var(f)$).

As the final step of our method, we pass the constructed feature f into denoising methods (e.g., adaptive polynomial rendering [MMMG16]), to generate a final output or a sampling map to adaptively redistribute ray samples over the image space. Note that adaptive sampling that guides more samples on high reconstruction errors is a technical advantage of recent adaptive methods, and our feature generation allows to integrate this adaptive sampling into the gradient-domain path tracing.

5. Implementation Details

We have implemented our feature generation on top of the gradient-domain path tracing (G-PT) [KMA*15]. Specifically, we have employed the source code provided by the authors. To implement our bagging procedure, we have stored gradient samples per pixel to compute the sample means and variances of bootstrap samples (i.e., gradients). While this approach is conceptually simple, it may require a non-trivial computational and memory overhead when a large number of samples are used. To reduce the overhead, we have sub-averaged multiple samples into a pre-defined set of bins (e.g., eight bins per pixel) instead of storing all samples individually. As a result, our bagging process is performed using the sub-averaged samples. Technically, it is equivalent for our bagging process to select all the samples within a bin at a time. Given the small number of bins, we found that a small iteration count (e.g., ten times) used in the bagging compromises the balance between quality of our estimated feature and computational overhead properly.

For the final reconstruction, we use a recent denoising method, adaptive polynomial rendering (APR) [MMMG16] that approximates an image region with 2D image polynomials. This method also employs rendering-specific features (i.e., normal, texture, and depth buffers) in a linear form. As this method takes an arbitrary set of rendering-specific features as input, we additionally include an ambient occlusion buffer [MVZ16] for improving their denoising quality. Since most recent adaptive methods including APR (e.g., [RMZ13], [MCY14], [MIGYM15], [BRM*16]) are able to work with other features (including our feature) without changing their main algorithms, we simply feed our estimated ideal feature to APR so that the method produces either a sampling map for intermediate stages or a final image for the last step.

As the pre-filtering that takes our feature and its variance, we adopt the non-local means filter [RMZ13] that adjusts an amount of blurring by variances. We use a 7×7 patch and 11×11 denoising window for the pre-filtering. In addition, for APR we use a 43×43 filtering window and three iterations for its adaptive sampling where additional samples are distributed across images.

Our approach works well in most cases, but we found that it can have difficulty given an extreme scenario where the underlying gradient estimation (i.e., G-PT) fails to capture meaningful gradient structures. In rendering, this is common when a light path can be computed only on a low probability, e.g., scenes where indirect illumination is dominant. When we fail to find light paths, most radiance samples can end up zero radiances, and thus there is only a low chance for capturing proper gradient samples.

Our simple idea for such non-visible paths is to add an ambient light with a very small intensity (e.g., 0.01) during estimating image gradients. The ambient lighting is a well-known heuristic in rendering [RPG99], and it illuminates all parts in a scene equally. Computing the input gradients of G-PT in a principled manner even for the extreme case is out of the scope of this work, but we found that our simple heuristic alleviates the difficulty of estimating image gradients, when non-visible paths are generated.

6. Results and Discussion

We have tested our method on a Windows machine with a Xeon 3.0 GHz CPU and NVidia GTX 1080 Ti graphics card. Given the recent gradient-domain path tracing (G-PT) framework [KMA*15], we have compared our method with the $L1$ reconstruction that directly computes the final image without utilizing adaptive rendering methods [KMA*15]. We have also tested a regularized $L1$ reconstruction [MVZ16] that adapts the original reconstruction using a constraint term formed by G-buffers. Specifically, the additional features of the method include normal, texture, world coordinates and ambient occlusion buffers.

We have also tested APR by varying its features. For example, we have run APR with G-buffers (normal, texture, and depth and ambient occlusion buffers), which do not utilize image gradients generated by G-PT. Also, this method is set to employ only our feature instead of the G-buffers. Additionally, we run APR by utilizing both features (G-buffers and our feature) and other potential combinations (e.g., image gradients or $L1$ image).

For clear visual comparisons, we additionally include noisy color buffers (input color) of G-PT without any post-reconstruction. For comparing numerical accuracy of each method, we have used the relative mean squared error (relMSE) [RKZ11].

Scenes. For our tests, we have used the following scenes: 1) DOF-Kitchen, 2) Kitchen, 3) Bathroom, and 4) Door. We have used 1280×720 image resolutions for the tested scenes. For the DOF-Kitchen scene (first row in Fig. 5), we simulate a depth-of-field effect to verify robustness of our feature against the distributed effect. The Kitchen scene (second row in Fig. 5) is a glossy-dominant scene where most high-frequency edges are generated by glossy reflections, and these scenarios introduce a technical challenge for recent denoising methods depending on G-buffers that can capture only a limited set of edges (e.g., discontinuities of normals, texture and depth). On the other hand, the Bathroom and Door scenes (third and fourth rows in Fig. 5) are cases that work well with G-buffers based denoising methods, since most edges can be predictable by the geometric information. However, the estimated gradients tend to be very noisy due to the scene configuration where indirect lighting is dominant.

Comparisons with Other Features. Fig. 4 shows results of the state-of-the-art adaptive rendering, APR, with different feature sets, as recent denoising methods are able to include arbitrary features within the denoising process. We disable the adaptive sampling of

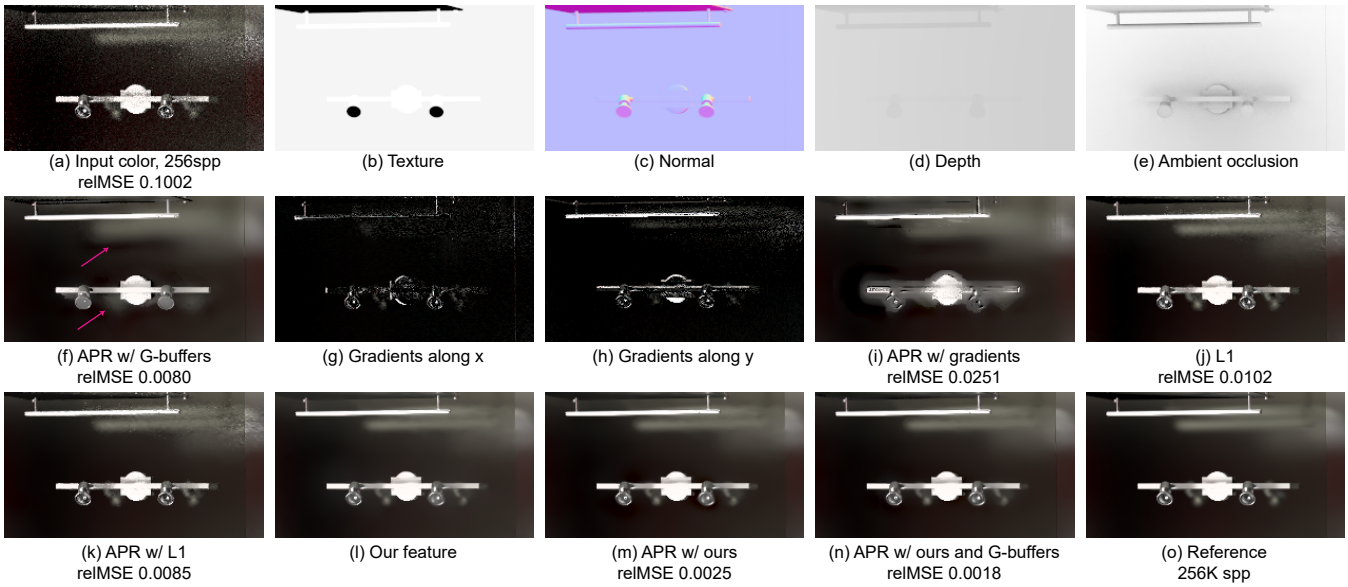

Figure 4: *Denoising results with different features for Kitchen scene. Given the input image (a) generated by a uniform sampling, a recent denoising method (i.e., APR) is tested with G-buffers ((b), (c), (d) and (e)), image gradients ((g) and (h)), L1 result (j), and our feature (l). The G-buffers ((b), (c), (d) and (e)) capture only geometric edges, and thus APR with the G-buffers (f) fails to preserve the complex edges (pointed by arrows). The gradients ((g) and (h)) contain a noisy but complete set of edges, but directly employing the gradients as features leads to a completely over-blurred result (i). We also test the L1 image (j) as a feature of APR, but this shows a sub-optimal result (k) that has a similar pattern of noise in the L1 (j). Our feature (l) estimates an ideal feature using the gradients ((g) and (h)), and makes APR to produce more accurate results ((m) and (n)), by preserving the complex edges.*

the APR for this test, in order to properly compare reconstruction quality of APR given different features. APR with G-buffers preserves the geometric edges that G-buffers have, but can fail to reconstruct more complex edges (e.g., glossy reflections). APR has a functionality of increasing the order of local functions to better approximate the complex areas, but it is still challenging to properly reconstruct those regions when input buffers are very noisy. Technically, this is mainly because high order functions have higher reconstruction variances, and APR tends to select low order functions that minimize MSE (squared bias plus variance) for those areas.

In addition, we can directly utilize image gradients from G-PT as features, and thus we pass the pre-filtered gradients into APR. To pre-filter the noisy gradient, we utilize the non-local means filter [RMZ13]. APR with image gradients shows a completely over-blurred result, since correlation between the gradients and the ground truth image can be low. Note that the ideal feature is not the gradients of the ground truth image, but it is an image function that has gradients equal to the gradient of the ground truth (described in Sec. 3).

One may consider to use the $L1$ image computed by the screened Poisson reconstruction as a feature of APR. We also test this image as a feature within APR, but this shows sub-optimal results (e.g., a similar pattern of noise in the $L1$ image), since the feature can have a high correlation with the random error (i.e., ε_i in Eq. 1) of an input image. Technically, removing the correlation completely is not straightforward, since the $L1$ image is generated by the screened Poisson reconstruction that utilizes the input image as a regularization term for the reconstruction. On the other hand, our feature allows for capturing the complex edges, and it leads to an improved denoising result. Note that recent denoisers including APR are not limited to a specific feature, and thus both our feature and G-buffers can be naturally utilized in an adaptive framework and this combination leads to a further improvement.

Equal-time Comparisons. Fig. 5 shows same-time comparisons of the tested methods. The original APR with G-buffers can be performed without image gradients, so it does not require generating offset light paths. As a result, for a fair comparison, we generate the results of APR with G-buffers using a path tracing so that it can use much more samples compared to the other methods including our approach.

All the other methods including our method rely on image gradients, and thus these approaches use the gradient-domain path tracing [KMA*15], which has higher overhead due to the additional processing for offset paths. The main goal of our approach is to eliminate random noise for the gradient-domain renderer, and thus we can limit the baseline renderer to the gradient-domain one for all the tested methods. In this case, APR with G-buffers cannot use the demonstrated sample count, as sampling time is bounded by a given renderer. Nevertheless, we force the APR with G-buffers to use a standard path tracer for fairness.

The $L1$ reconstruction preserves high-frequency edges well, but leaves some high-frequency noise due to noisy image gradients. APR with our feature image (APR with ours) shows a smoother

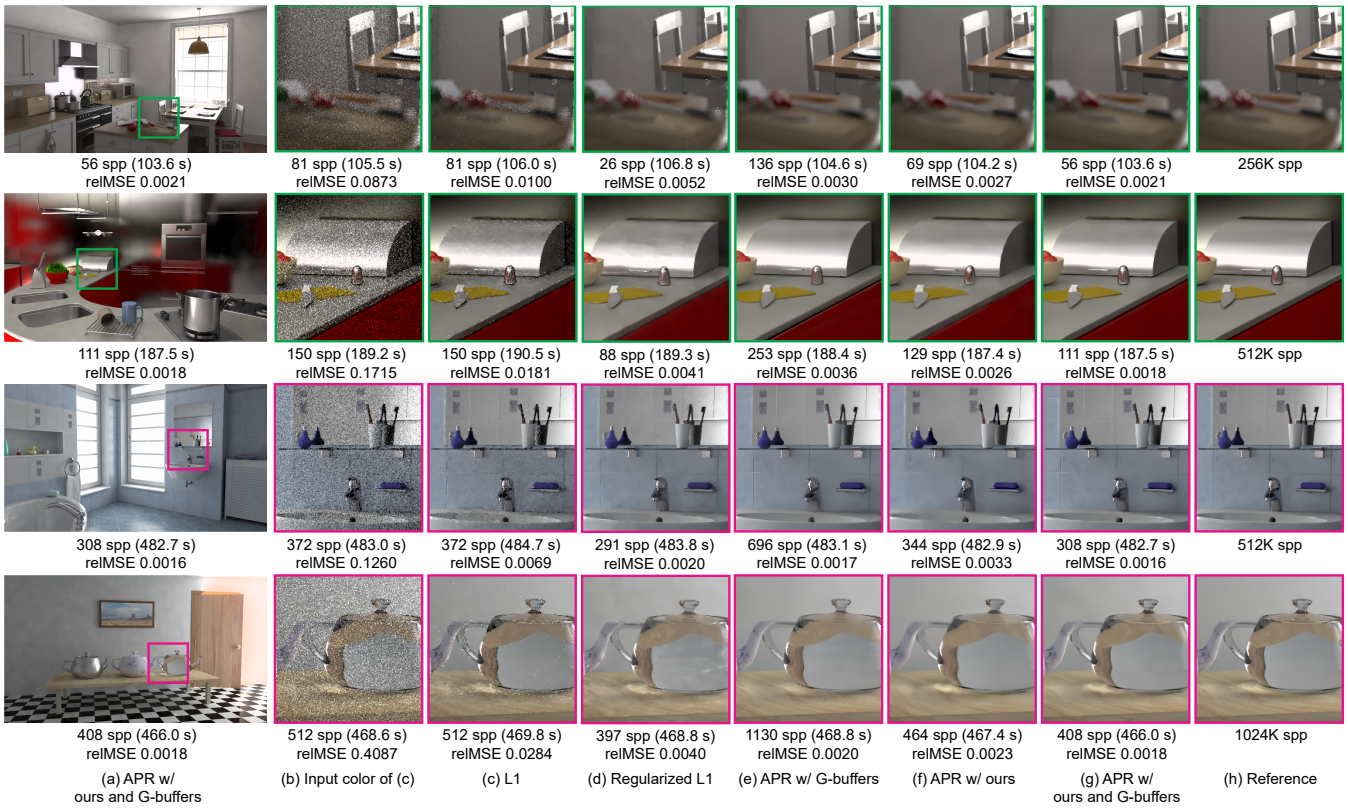

Figure 5: *Equal-time comparisons. Given a gradient-domain path tracing (G-PT), our method generates a new feature image that is passed into an adaptive rendering method (APR). Within the denoising framework, our feature can be used instead of using the G-buffers or together with the G-buffers. The denoising results (g) of APR using both G-buffers and our feature show better numerical accuracy and visual quality, compared to the previous L1 reconstruction (c), regularized L1 (d), and APR with G-buffers (e) that does not employ our feature.*

image compared to $L1$, since we take account of noise (i.e., Eq. 7) of the image gradients generated by G-PT. Note that both methods do not utilize any geometric information, but our approach enables high-quality image denoising by simply passing a feature image into a recent adaptive method (e.g., APR).

In rendering, it is easy to capture G-buffers, which have been a common characteristic of recent denoising methods. We thus additionally test APR that uses either G-buffers or combined features (our feature and G-buffers). APR with only G-buffers preserves the geometric edges that the buffers can contain well, but fails to properly denoise the non-geometric edges that the conventional features cannot capture. However, when APR employs both features (G-buffers and our feature), its results are improved by utilizing the correlation between our features and ground truth images in places (e.g., glossy reflections) where G-buffers fail to capture the correlation.

Numerical Convergence. Fig. 6 shows relative MSE convergence of tested methods across scenes. The errors of APR with our feature are significantly lower than ones of $L1$ results, and this result indicates that our integrated solution of G-PT and an adaptive rendering method (i.e., APR) is able to boost the performance of the gradient-domain rendering framework even when G-buffers are not exploited.

Regularized L1 reconstruction [MVZ16] shows the best results (42 to 44% lower error than that of APR with ours and G-buffers) with a small number of samples (i.e., 16 spp) for DOF-Kitchen and Bathroom scenes, but produces the worst result (5× higher error than that of our approach) for Door scene where input gradients are extremely noisy. On the other hand, our solution, APR with ours and G-buffers, consistently produces good numerical results across scenes even for Door scene, as our process robustly generates a feature while reducing noise in estimated gradients. In addition, the computational overhead of the prior is fairly large (e.g., 1 minute), compared to that of APR with ours and G-buffers (8.2 secs). As a result, for the DOF-Kitchen and Bathroom scenes, our method produces more accurate numerical results (relMSE 0.0023 and 0.0072 with 51 spp) than those (relMSE 0.0080 and 0.0239 with 16 spp) of the regularized solution, given an equal-time.

APR with only G-buffers produces relatively good numerical results for Bathroom and Door scenes where the features can predict most of high-frequency information well. However, the performance gain guided by G-buffers diminishes for DOF-Kitchen and Kitchen scenes where non-geometric information (e.g., glossy

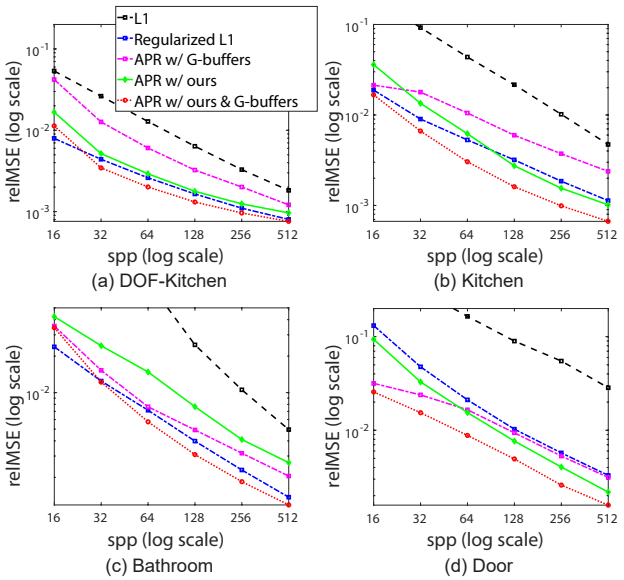

Figure 6: *Relative MSE convergence of the tested methods for DOF-Kitchen, Kitchen, Bathroom, and Door scenes. Except for the small number of samples (i.e., 16 spp), APR with ours and G-buffers shows better results than other methods numerically.*

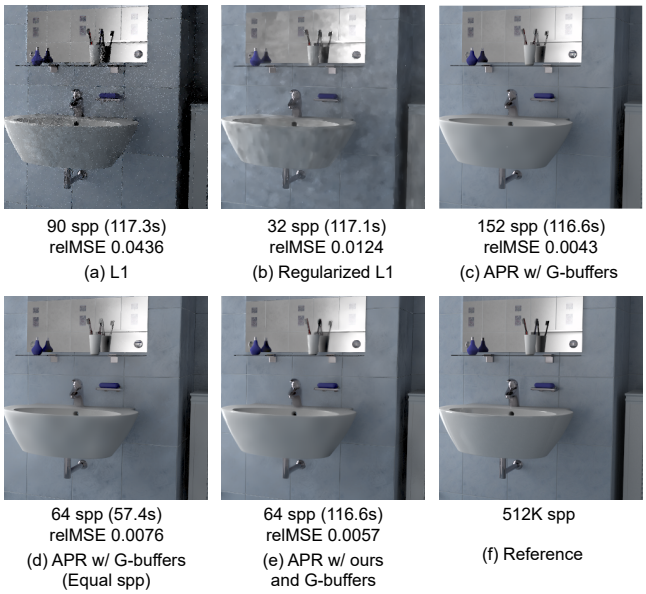

Figure 7: *Failure case for the Bathroom scene where most of edges can be captured by G-buffers. Given an equal-time, our method (e) shows better denoising results, compared to the other methods (L1 (a), regularized L1 (b)) that utilize a gradient-domain path tracer. However, APR with G-buffers produces higher reconstruction quality than ours, as this previous approach employs a standard path tracer that has much lower overhead than the gradient-domain rendering system. Nevertheless, our approach (APR with our feature and G-buffers) is able to show similar performance given a large number of samples (see Fig. 5) for the equal-time, as the rate of error convergence can be improved when our feature is included (see Fig. 6).*

reflections) is a dominant factor. APR with only our feature outperforms APR with G-buffers for the scenes from 32 spp, as our new feature can help to predict the edge information that G-buffers cannot contain. Overall, APR with both features (ours and G-buffers) shows better results compared to the others for all scenes except a very small number of samples (e.g., 16 spp). As a result, our feature makes a recent adaptive method to be successfully integrated into the gradient-domain rendering framework, by passing an estimated optimal feature into the local regression based adaptive method.

Animated Sequences. Temporal coherence between input frames is typically employed by a recent adaptive method, which performs a final reconstruction on 3D volumes by extending a 2D denoising window to a 3D one [MMMG16]. In addition, we extend the non-local means pre-filtering [RMZ13] into a 3D one (e.g., five adjacent frames) to further reduce the remaining noise in our feature by increasing the chance of finding similar patches. Accompanied videos show input colors, our features, and final reconstruction results by APR that utilizes both features (i.e., our feature and G-buffers). Flickering artifacts are still noticeable on final images, especially when light densities are low (e.g., shadows), but the final sequences have much reduced flickering compared to the input, while preserving detailed high-frequency information thanks to our feature.

Computational Overhead. The main computational bottleneck of our method is the bagging process that performs multiple iterations (e.g., ten times) of WLS, and the computational time for generating our feature image and its variance is 1.3 seconds given the tested image resolution. Also, the time of the non-local means pre-filtering is 0.2 seconds. These computational times are mostly constant in terms of the number of samples as we use a fixed number of bins (Sec. 5). These overheads are not ignorable, but are small portions of total rendering times (e.g., more than one minute in Fig. 5). As a result, given the equal-time comparisons, our method shows higher quality over the tested prior methods.

Limitations. The main limitation of our method is that our feature image may miss some high-frequency edges, which G-buffers can contain, especially when image gradients are very noisy. Fig. 7 shows this scenario where noisy gradients do not provide enough information on high-frequency edges. Our feature can still improve the denoising output of APR, when ours is included into their method as an additional feature. Since the convergence rate of our method (APR with both features) is faster than that of APR with G-buffers (shown in Fig. 6), our method shows similar or better results for an equal-time comparison (Fig. 5) given a large number of samples. However, with a relatively small number of samples, APR with G-buffers can outperform our approach where most of high-frequency information can be predicted by G-buffers (e.g., Bathroom scene). Nonetheless, our method shows the most robust performance across different scenes, as we aim at estimating the ideal feature that contains all high-frequency information (not just geometry edges).

7. Conclusion

In this paper, we have presented an integrated solution that adaptively performs a feature-based reconstruction and sampling within the G-PT framework. This novel integration is achieved seamlessly by feeding a feature image into an existing adaptive rendering method. This high-level approach is technically implemented by deriving an ideal feature in the form of image gradients so that the feature can be reconstructed by a Poisson reconstruction. In addition, we have proposed a robust feature estimation process including a weighted least squares and bagging, in order to take account of heterogeneous noise in the input image gradients from G-PT. Our method has shown a drastic error reduction even without exploiting the most common feature (i.e., G-buffers), compared to existing reconstruction for G-PT (i.e., $L1$).

Interesting research directions lie ahead. A direction is to estimate the ideal feature more robustly when we use a small number of samples and the input gradients are extremely noisy. Practically, this should be addressed for a fast preview of pre-rendering results. Additionally, we have shown that our simple extension to the animation works reasonably well, but to further reduce flickering we would like to test our method within the temporal gradient-domain framework [MKD*16] and investigate features tailored for such approach.

Acknowledgements

We are appreciative of the anonymous reviewers for the constructive comments. We also thank Taeyoung Kim for a test scene and Wenzel Jakob for the Mitsuba renderer [Jak10]. The Door scene was modeled after Eric Veach by Miika Aittala, Samuli Laine, and Jaakko Lehtinen. The Kitchen and Bathroom scenes were ported to the Mistuba renderer by Anton Kaplanyan and Tiziano Portenier, and the DOF-Kitchen scene was modeled by blendswap.com user Jay-Artist. This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2017R1C1B2003893), and partly supported by the National Research Foundation of Korea (NRF) (No. 2017M3C4A7066317).

References

[Aga07] AGARWALA A.: Efficient gradient-domain compositing using quadtrees. *ACM Transactions on Graphics 26*, 3 (2007). 4

[Alp04] ALPAYDIN E.: *Introduction to Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2004. 5

[BBS14] BELCOUR L., BALA K., SOLER C.: A local frequency analysis of light scattering and absorption. *ACM Transactions on Graphics 33*, 5 (2014). 2

[BCCZ08] BHAT P., CURLESS B., COHEN M., ZITNICK C. L.: Fourier analysis of the 2d screened poisson equation for gradient domain problems. In *European Conference on Computer Vision* (2008), Springer, pp. 114–128. 4

[BRM*16] BITTERLI B., ROUSSELLE F., MOON B., IGLESIAS-GUITIÁN J. A., ADLER D., MITCHELL K., JAROSZ W., NOVÁK J.: Nonlinearly weighted first-order regression for denoising monte carlo renderings. *Computer Graphics Forum 35*, 4 (2016), 107–117. 2, 6

[CL96] CLEVELAND W. S., LOADER C. L.: Smoothing by Local Regression: Principles and Methods. In *Statistical Theory and Computational Aspects of Smoothing*. Springer, New York, 1996, pp. 10–49. 3

[DHS*05] DURAND F., HOLZSCHUCH N., SOLER C., CHAN E., SILLION F. X.: A frequency analysis of light transport. *ACM Transactions on Graphics 24*, 3 (2005), 1115–1126. 2

[Jak10] JAKOB W.: Mitsuba renderer, 2010. https://www.mitsuba-renderer.org. 10

[JDZJ08] JAROSZ W., DONNER C., ZWICKER M., JENSEN H. W.: Radiance caching for participating media. *ACM Transactions on Graphics 27*, 1 (2008), 7:1–7:11. 2

[Kaj86] KAJIYA J. T.: The rendering equation. In *ACM SIGGRAPH '86* (1986), pp. 143–150. 1, 2

[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. 2

[KMA*15] KETTUNEN M., MANZI M., AITTALA M., LEHTINEN J., DURAND F., ZWICKER M.: Gradient-domain path tracing. *ACM Transactions on Graphics 34*, 4 (2015), 123:1–123:13. 1, 2, 3, 4, 5, 6, 7

[LKL*13] LEHTINEN J., KARRAS T., LAINE S., AITTALA M., DURAND F., AILA T.: Gradient-domain metropolis light transport. *ACM Transactions on Graphics 32*, 4 (2013), 95:1–95:12. 1, 2

[MCY14] MOON B., CARR N., YOON S.-E.: Adaptive rendering based on weighted local regression. *ACM Transactions on Graphics 33*, 5 (2014), 170. 2, 3, 6

[MIGYM15] MOON B., IGLESIAS-GUITIAN J. A., YOON S.-E., MITCHELL K.: Adaptive rendering with linear predictions. *ACM Transactions on Graphics 34*, 4 (2015), 121:1–121:11. 2, 6

[MKD*16] MANZI M., KETTUNEN M., DURAND F., ZWICKER M., LEHTINEN J.: Temporal gradient-domain path tracing. *ACM Transactions on Graphics 35*, 6 (2016), 246:1–246:9. 2, 10

[MMMG16] MOON B., MCDONAGH S., MITCHELL K., GROSS M.: Adaptive polynomial rendering. *ACM Transactions on Graphics 35*, 4 (2016), 40:1–40:10. 2, 3, 6, 9

[MVZ16] MANZI M., VICINI D., ZWICKER M.: Regularizing image reconstruction for gradient-domain rendering with feature patches. *Computer Graphics Forum 35*, 2 (2016), 263–273. 2, 3, 6, 8

[PGB03] PÉREZ P., GANGNET M., BLAKE A.: Poisson image editing. *ACM Transactions on Graphics 22*, 3 (2003), 313–318. 2

[RJN16] ROUSSELLE F., JAROSZ W., NOVÁK J.: Image-space control variates for rendering. *ACM Transactions on Graphics 35*, 6 (2016), 169. 5

[RKZ11] ROUSSELLE F., KNAUS C., ZWICKER M.: Adaptive sampling and reconstruction using greedy error minimization. *ACM Transactions on Graphics 30*, 6 (2011), 159:1–159:12. 6

[RMZ13] ROUSSELLE F., MANZI M., ZWICKER M.: Robust denoising using feature and color information. *Computer Graphics Forum 32*, 7 (2013), 121–130. 1, 2, 6, 7, 9

[RPG99] RAMASUBRAMANIAN M., PATTANAIK S. N., GREENBERG D. P.: A perceptually based physical error metric for realistic image synthesis. In *ACM SIGGRAPH '99* (1999), pp. 73–82. 6

[RW94] RUPPERT D., WAND M.: Multivariate locally weighted least squares regression. *The annals of statistics 22*, 3 (1994), 1346–1370. 3

[SD12] SEN P., DARABI S.: On filtering the noise from the random parameters in Monte Carlo rendering. *ACM Transactions on Graphics 31*, 3 (2012), 18:1–18:15. 1, 2

[Sta09] STAPLETON J. H.: *Linear statistical models*, vol. 719. John Wiley & Sons, 2009. 4, 5

[WH92] WARD G. J., HECKBERT P. S.: Irradiance gradients. *Proceedings of the 3rd Eurographics Workshop on Rendering* (1992), 85–98. 2

[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. *Computer Graphics Forum 34*, 2 (2015), 667–681. 2