Stylized Rendering as a Function of Expectation

REX WEST, Aoyama Gakuin University, Japan
SAYAN MUKHERJEE, The University of Tokyo, Japan and Blueqat Research, Japan

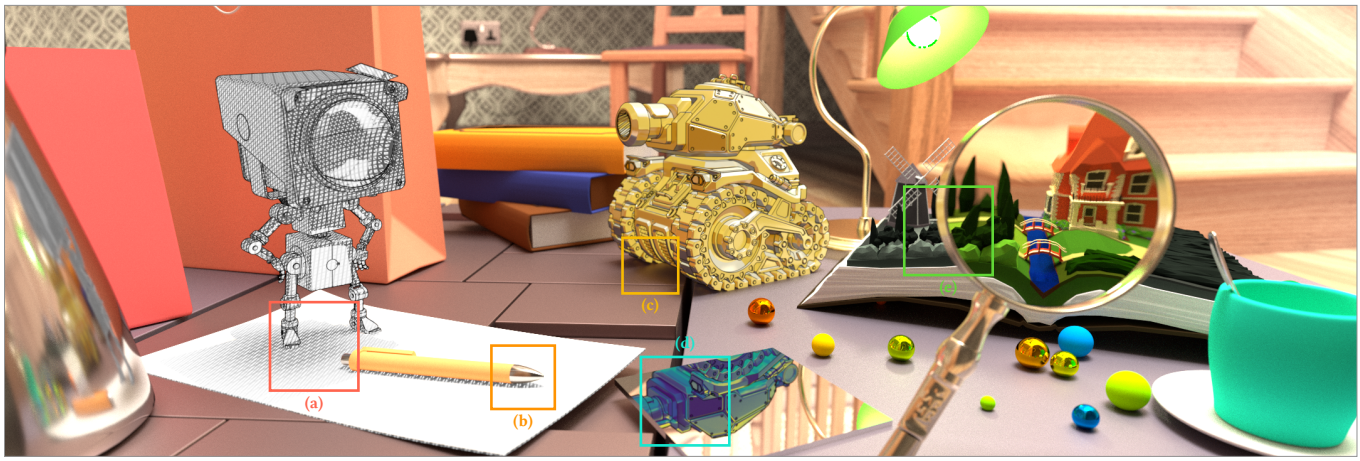

Fig. 1. The proposed generalization of the rendering equation fuses the photo-realistic visuals of physically-based rendering with a subset of the artistic visuals of stylized rendering. This gives us a great degree of freedom when designing the look-and-feel of rendered content. For example, in inset (a) we stylize the sheet of paper to produce a cross-hatched shadow to blend the hand-drawn feel of the robot with the environment around it. In reverse, the stylized shadow under the pen provides interesting visual contrast in inset (b). In inset (c) the bright gold cel shading brightens the shadowed floor, naturally providing visual cohesion. In inset (d) we stylize the tank differently depending on whether it is viewed directly or through the mirror on the table. In inset (e) the dull monotone colors of the story book come to life when viewed through the looking glass.

We propose a generalization of the rendering equation that captures both the realistic light transport of physically-based rendering (PBR) and a subset of non-photorealistic rendering (NPR) stylizations in a principled manner. The proposed formulation is based on the key observation that both classical transport and certain NPR stylizations can be modeled as a *function of expectation*. Given this observation, we generalize the recursive integrals of the rendering equation to *recursive functions of expectation*. As estimating functions of expectation can be challenging, especially recursive ones, we provide a toolkit for unbiased and biased estimation comprising prior work, general strategies, and a novel build-your-own strategy for constructing more complex unbiased estimators from simpler unbiased estimators. We then use this toolkit to construct a complete estimator for the proposed recursive formulation, and implement a sampling algorithm that is both conceptually simple and leverages many of the components of an ordinary path tracer. To demonstrate the practicality of the proposed method we showcase how it captures several existing stylizations like color mapping, cel shading, and cross-hatching, fuses NPR and PBR visuals, and allows us to explore visuals that were previously challenging under existing formulations.

CCS Concepts: • **Computing methodologies** → **Rendering**; **Ray tracing**.

Authors' addresses: Rex West, Aoyama Gakuin University, Japan, rexwest@gmail.com; Sayan Mukherjee, The University of Tokyo, Japan, Blueqat Research, Japan, sayan@phys.s.u-tokyo.ac.jp.

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM. This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in *ACM Transactions on Graphics*, https://doi.org/10.1145/3658161.

Additional Key Words and Phrases: rendering, non-photorealistic rendering

ACM Reference Format:
Rex West and Sayan Mukherjee. 2024. Stylized Rendering as a Function of Expectation. *ACM Trans. Graph.* 43, 4, Article 96 (July 2024), 19 pages. https://doi.org/10.1145/3658161

1 INTRODUCTION

Modern rendering systems based on physically-based rendering (PBR) form images by solving a central, physically-plausible integral formulation called the rendering equation. This integral formulation is compelling in both its ability to capture a wide-range of physical phenomena and ease in numerical estimation, and has received extensive attention in both research and industry. Unfortunately, it restricts rendered visuals to those that fit within its physically-plausible formulation.

In contrast, non-photorealistic rendering (NPR) aims to replicate stylized visuals, like those seen in comic books or animated films, in a 3D rendering system. These visuals do not necessarily follow from a physically-based formulation, but instead are the result of the artist's aesthetic goals, creativity, or even technical limitations in the process. Such stylized visuals can be used to express the mood of a scene, emphasize the motion of a character, or replicate a long-since-deprecated image forming process, and they have come to be an essential part of modern creative works.

To the best of our knowledge, there is currently no formulation that fully captures the artistic visuals of NPR and the physically-plausible visuals of PBR in a unified manner. Such a formulation is compelling as it would open up the possibility to seamlessly combine physical and artistic visuals in a principled manner, allow us to explore the application of research advances in PBR to NPR, and vice versa, and even potentially explore new types of visualizations.

As a step in that direction, we propose a generalization of the rendering equation that captures both realistic light transport and a subset of NPR stylizations under a single formulation. Our key observation is that, under a PBR formulation, a certain subset of NPR stylizations can be interpreted as modifying the light leaving the surface of objects. Stylizations such as cel shading and cross-hatching, as well as those classically used in image editing, such as contrast and saturation adjustment, can all be distilled down to the same operation: locally modifying the *color* of objects. In a PBR formulation, the color of an object is determined by the light leaving the surface (i.e. exitant radiance). Putting the two together, we can interpret that such stylizations effectively apply a style (a function) to the integral that solves for exitant radiance (an expectation) — a *function of expectation.*

Given this observation, we *generalize* the recursive integrals of the rendering equation to recursive functions of expectation. This generalization allows PBR effects like glossy reflections and color bleed to affect NPR stylizations, and vice versa, and extends the formulation of Doi et al. [2021] to multiple bounces.

As one might expect, such a reformulation introduces new challenges in unbiased and biased estimation that need to be addressed before it can be implemented in practice. We provide a toolkit for unbiased and biased estimation, comprising prior work, several general strategies, and a novel build-your-own strategy for constructing more complex unbiased estimators from simpler unbiased estimators. We then use this toolkit to construct a complete estimator for the proposed recursive formulation. To implement the complete estimator in practice we propose a conceptually simple tree-based sampling algorithm that comes with tunable performance, varying from linear to exponential overhead depending on stylization parameters, permits controllable variance and bias, and can be implemented using many of the components of an ordinary path tracer.

To demonstrate the practicality of the proposed formulation we implement several existing NPR stylizations like cel shading and cross-hatching, incorporate stylized and unstylized content in the same scene, and provide a few examples of stylizations that were previously challenging under existing formulations.

Concretely, our contributions are:

- A generalization of the rendering equation that captures both classical transport and a subset of NPR stylizations in a principled manner
- A practical estimator and sampling algorithm for the proposed formulation
- Advances in estimators for functions of expectation: proof(s) of unbiasedness for MC estimation of recursive functions of expectation, and a recipe (and related proofs) for combining estimators under addition, component-wise multiplication and composition operations.

2 BACKGROUND AND RELATED WORK

Physically-based rendering (PBR) forms images by solving a physically-plausible integral formulation called the rendering equation [Kajiya 1986], a simplified form of the radiative transfer equation [Chandrasekhar 1960]. This integral formulation is compelling for its ability to capture a wide-range of physical phenomena and its ease in numerical estimation.

The rendering equation is formulated as a recursive integral,

$$L(\mathbf{x},\mathbf{y}) = L_e(\mathbf{x},\mathbf{y}) + \int_{\mathcal{V}} f_r(\mathbf{x},\mathbf{y},\mathbf{z}) G(\mathbf{y},\mathbf{z}) L(\mathbf{y},\mathbf{z})\, \mathrm{d}\mathbf{z}\,, \tag{1}$$

that computes the exitant radiance $L(\mathbf{x},\mathbf{y})$ from the surface point $\mathbf{y}$ towards $\mathbf{x}$. Here $L_e(\mathbf{x},\mathbf{y})$ is radiance emitted from $\mathbf{y}$ towards $\mathbf{x}$, $f_r(\mathbf{x},\mathbf{y},\mathbf{z})$ is a bidirectional scattering distribution function (BSDF), $G(\mathbf{x},\mathbf{y})$ is the geometry term, $L(\mathbf{y},\mathbf{z})$ is the recursive incoming radiance, and $\mathcal{V}$ is the set of surface points.

To form an image, we must solve the rendering equation (1) for each pixel ρ,

$$I_\rho = \int_{\mathcal{V}^2} W_\rho(\mathbf{x},\mathbf{y}) G(\mathbf{x},\mathbf{y}) L(\mathbf{x},\mathbf{y})\, \mathrm{d}\mathbf{x}\, \mathrm{d}\mathbf{y}\,, \tag{2}$$

where $W_\rho(\mathbf{x},\mathbf{y})$ is the sensor responsivity term that determines the light contribution to the pixel ρ. However, there is often no closed-form solution to the pixel forming equation (2) in practice, so we numerically approximate it using Monte Carlo (MC) integration,

$$\langle L(\mathbf{x},\mathbf{y})\rangle = L_e(\mathbf{x},\mathbf{y}) + \frac{f_r(\mathbf{x},\mathbf{y},\mathbf{z}) G(\mathbf{y},\mathbf{z}) \langle L(\mathbf{y},\mathbf{z})\rangle}{p(\mathbf{z}|\mathbf{x},\mathbf{y})}\,, \tag{3}$$

$$\langle I_\rho\rangle = \frac{W_\rho(\mathbf{x},\mathbf{y}) G(\mathbf{x},\mathbf{y}) \langle L(\mathbf{x},\mathbf{y})\rangle}{p(\mathbf{x},\mathbf{y})}\,, \tag{4}$$

where $\langle\cdot\rangle$ denotes an estimate of a value, and, due to the linearity of the terms of the estimator, it follows that $I_\rho = \mathrm{E}\big[\langle I_\rho\rangle\big]$.

Readers interested in a more thorough introduction to physically-based rendering can refer to the textbook by Pharr et al. [2023].

2.1 Non-photorealistic rendering

In contrast to PBR, non-photorealistic rendering (NPR) attempts to replicate artistically stylized visuals in a 3D rendering system. This includes stylizations that replicate an artistic process for a physical medium, such as hatching and stippling, stylizations that imitate a technical limitation, such as cel-shading and half-tone, as well as stylizations that modify color properties, such as brightness, contrast, and saturation.

Several existing works in NPR stylization attempt to replicate an artistic process for a physical medium, like that of hand drawn sketch styles [Winkenbach and Salesin 1994; Meier 1996; Lake et al. 2000; Isenberg et al. 2006; Lawonn et al. 2018; Todo et al. 2022] used in artistic and scientific illustrations [Sousa et al. 2005]. Hatching methods [Hertzmann and Zorin 2000; Praun et al. 2001] use various techniques to place progressively denser lines to represent variations in object brightness. In particular, Deussen et al. [1999] and Medeiros et al. [2009] generate hatching lines by intersecting scene geometry with sets of planes (see Fig. 2 bottom right). Similar to hatching, stippling methods [Deussen and Isenberg 2013; Martin et al. 2017] use small randomized dots to represent variations in brightness.

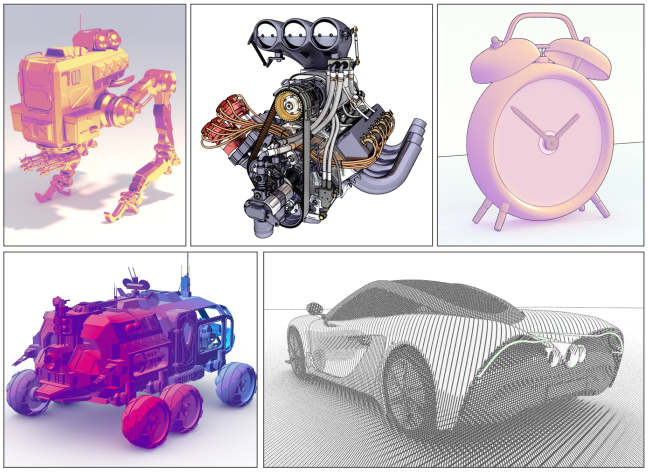

Fig. 2. Existing work in NPR captures a range of compelling artistic styles. Objects can be explicitly colorized (top, left) [Doi et al. 2021], drawn with feature lines to show silhouettes and creases (top, mid) [West 2021], replicate the detail-preserving style of technical illustrations (top, right) [Gooch et al. 1998], rendered using cel shading (bottom, left) [Barla et al. 2006], or a drawing style like cross-hatching (bottom, right) [Deussen et al. 1999].

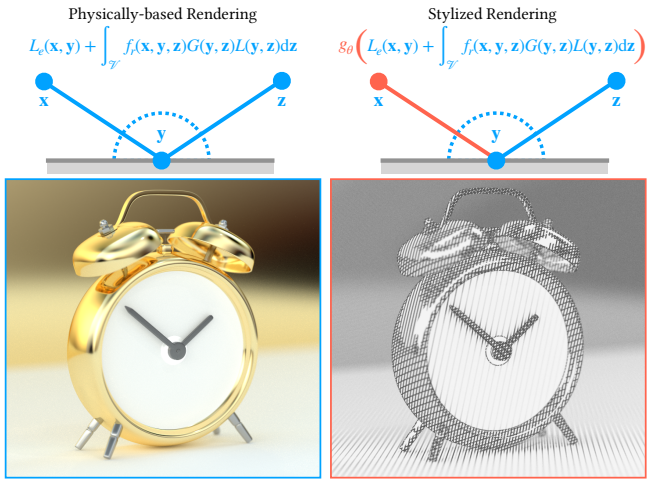

Fig. 3. Physically-based rendering (left) computes the pixel colors of a rendered image by solving for the amount of radiance (light) that bounces off objects towards a camera. Stylized rendering (right), as it modifies the perceived color of objects, effectively modifies this radiance. We can represent this modification as a *style function* g_θ, and applying it to the integral that solves for exitant radiance gives us what is known as a *function of expectation*.

Several other works attempt to replicate visual styles that are the by-product of a process limitation. Decaudin [1996] and Lake et al. [2000] demonstrate a cel shading look that replicates the discrete bands of color often seem in cartoon animations. Barla et al. [2006] introduce flexible parametrization to cel shading, allowing, for example, closer objects to be shaded differently than farther objects. Spindler et al. [2006] show examples of how to fine-tune cel shading to different target illustration styles. Work has also been done to replicate half-tone printing methods [Hall 1999; Deussen and Isenberg 2013] that represent gradual changes in tone or brightness with only a limited set of colors.

Yet other works address stylizations for technical illustrations and visualizations [Sousa et al. 2005] with an emphasis on detail enhancement [Gooch et al. 1998; Rusinkiewicz et al. 2006]

Adjacent to NPR stylizations is artistic appearance editing, where scene parameters are modified to achieve a target look. This includes artistically editing shadows and highlights [Anjyo and Hiramitsu 2003; Anjyo et al. 2006; Todo et al. 2007; Petikam et al. 2021], goal-based lighting [Costa et al. 1999], or material editing [Ben-Artzi et al. 2006]. For a thorough overview of artistic appearance editing we refer the reader to the survey by Schmidt et al. [2016].

Recently there have been a growing number of works leveraging machine learning to achieve stylization. Style transfer methods have been shown to successfully transfer artistic style from a style source image to a content image [Gatys et al. 2016; Zhang et al. 2017, 2018]. More recently there have been advances in learning and replicating painting techniques [Liu et al. 2023], as well as stylizing volumetric data [Tojo and Umetani 2022; Wang et al. 2023].

Readers interested in a more thorough overview of NPR stylizations can refer to the review by Lawonn et al. [2018].

2.2 Integrating NPR stylizations into PBR

Physically-based rendering is a compelling framework for a rendering system. It follows from a well-studied formulation, produces high-quality visuals, and is well-suited to the highly parallel architectures of modern computing platforms.

Modelling NPR stylizations within a PBR-like formulation would provide us with several desirable benefits. It would allow us to combine different stylizations into the same scene in a principled manner, seamlessly combine physical phenomena and artistic visuals, and incorporate stylizations into modern PBR-based renderers [Pharr et al. 2018].

To that end, several prior works demonstrate how to integrate certain stylizations into a PBR formulation. Doi et al. [2021] present two such methods, *Color Remapping that Affects the Color on a Path Trace* (ACP), and *Fidelity for Texture Values* (FTV). ACP maps single sample estimates of exitant radiance in Eq. (1) to a texture value, effectively colorizing an object based on brightness (see Fig. 2 top left). FTV, similar in concept to ACP, leverages progressive photon mapping [Hachisuka et al. 2008; Hachisuka and Jensen 2009] to produce significantly lower variance estimates of exitant radiance before mapping them to a texture value. Both methods extend the rendering equation (1) by using a texture to modify the radiance reflected from stylized surfaces.

West [2021] presents a method for feature line rendering in a PBR framework. They take a different approach than Doi et al. [2021] and reformulate feature lines as intersectable light sources. This allows them to treat feature lines as an ordinary component of a PBR formulation, avoiding the need for a new formulation entirely. Their approach is compelling, but it requires finding a physical, or near-physical, representation of a desired stylization, which can be exceptionally difficult.

While these existing methods produce visually pleasing results, and hint at how we might incorporate NPR into PBR, they only cover a limited set of stylizations.

However, let us observe that a subset of stylizations in NPR have a key commonality — they locally modify the *color* of objects. In a PBR context the perceived color of an object is determined by the light leaving the surface, or exitant radiance, which is formulated as an *integral*. These stylization can then be interpreted as a *function* that modifies exitant radiance. Applying a function g_θ that represents a stylization to the exitant radiance $L(\mathbf{x}, \mathbf{y})$, an integral, results in a *function of integration* (see Fig. 3).

2.3 Functions of expectation

Functions of integration are better known in their stochastic-form as *functions of expectation*. Many interesting integrals I do not have a closed-form solution, making the direct evaluation of the function of integration $g(I)$ impossible. By writing the integral I as an expectation $\mathrm{E}[\langle I \rangle]$ of a random process,

$$g(I) = g\left(\mathrm{E}[\langle I \rangle]\right) , \tag{5}$$

we can formulate numerical approximations.

Outside of computer graphics, Blanchet and Glynn [2015] and Dauchet et al. [2018] explore the general strategy of representing an analytic function of expectation as a power series expansion and unbiasedly estimating the polynomial terms, Lee et al. [2019] propose an estimator for products of expectation (i.e. polynomials) that is provably variance-minimal, and Wang and Wang [2022] discuss methods for unbiased estimation of functions of expectation when simulation of the underlying distribution is intractable.

In rendering, Georgiev et al. [2019] and Kettunen et al. [2021] encounter functions of expectation in the formulation for volumetric transmittance (an exponential of expectation) and construct unbiased estimators using power series expansion. In particular, Georgiev et al. [2019] present several strategies for finite term selection from a power series to construct practical estimators. Kettunen et al. [2021] in their unbiased ray marching formulation, present a fast $O(n^2)$ recurrence relation to compute the variance-minimal estimator of Lee et al. [2019]. Misso et al. [2022] discuss functions of expectation in the context of debiasing methods, where unbiased estimation of the tail of an infinite series (e.g. power or telescoping) compensates for a biased, finite term estimator. Several other topics in rendering encounter functions of expectation, including unbiased photon gathering [Qin et al. 2015], differentiable rendering [Bangaru et al. 2020], specular manifold sampling [Zeltner et al. 2020], and path connections for refractive media [Pediredla et al. 2020].

3 STYLIZED RENDERING EQUATION

Our key observation is that a subset of NPR stylizations can be represented in PBR as a style function applied to exitant radiance (i.e. a function of expectation).

Given this observation, we propose to generalize the rendering equation (1) from a *recursive integral* to *recursive functions of expectation*.

We refer to this generalization as the *stylized rendering equation*,

$$L(\mathbf{x}, \mathbf{y}) = g_\theta\left(L_e(\mathbf{x}, \mathbf{y}) + \int_{\mathcal{V}} f_r(\mathbf{x}, \mathbf{y}, \mathbf{z}) G(\mathbf{y}, \mathbf{z}) L(\mathbf{y}, \mathbf{z}) \mathrm{d}\mathbf{z}\right) \tag{6}$$

where the style function $g_\theta : \mathbb{R}^k \to \mathbb{R}^k$ is a vector function of k dimensions parameterized by θ. In general, the parametrization θ can depend on the vertices $\mathbf{x}, \mathbf{y}$. Note that g_θ being a vector function implies that L, L_e, f_r, and G are similarly vector functions.

When the style function g_θ is the identity function, i.e. $g_\theta(I) = I$ for all parametrizations θ, the stylized rendering equation (6) reduces to the familiar rendering equation (1). Stylization is then any non-trivial function g_θ.

Motivation. The generalization (6) has several desirable aspects, where it:

- allows us to seamlessly integrate certain NPR stylization and PBR. Under this formulation PBR is now simply an instance of Eq. (6) with an affine style function.
- has flexible parametrization, providing control over when and how radiance is stylized.
- opens up visualizations that were previously challenging under existing formulations, like stylizing an object differently depending on the path taken up to the point of stylization.

Throughout the rest of this paper we will cover the necessary tool set to implement the stylized rendering equation and explore how these aspects come to fruition in practice.

3.1 Properties

Existence and uniqueness. In general, there is no guarantee that an iterative algorithm will reach a solution of (6) for arbitrary style functions g_θ. However, in most practical situations for rendering, we would apply stylization only on finitely many bounces. Therefore, after a fixed number of bounces, the iterative procedure would reduce to the usual rendering equation, implying convergence to a unique solution.

We make a note here that the proposed formulation in Eq. (6) closely resembles the non-linear Fredholm integral equation of the second kind in the special case that all the applied style functions are identical to a fixed function g. More precisely, one can consider the auxiliary non-linear Fredholm equation given by

$$L'(\mathbf{x}, \mathbf{y}) = L_e(\mathbf{x}, \mathbf{y}) + \int_{\mathcal{V}} f_r(\mathbf{x}, \mathbf{y}, \mathbf{z}) G(\mathbf{y}, \mathbf{z}) g(L'(\mathbf{y}, \mathbf{z})) \mathrm{d}\mathbf{z} . \tag{7}$$

When the integrand and L_e are bounded and g is Lipschitz, L' has a unique solution (see, e.g., Wazwaz [2011]). Further, if $L'_0(\mathbf{x}, \mathbf{y})$ is a solution to (7), then $g \circ L'_0(\mathbf{x}, \mathbf{y})$ is a solution to (6).

The practical implications of having a bounce-dependent style function g_θ is immense, as we demonstrate in Section 5. We defer a rigorous theoretical study of the integral equation (6) with bounce-dependent style functions g_θ to future work.

Reciprocity of Light. Since (6) contains a non-linearity in g_θ, the integral is not separable into paths as in the case of the usual rendering equation. Therefore, it is difficult to formulate the notion of reciprocity in the stylized context, and investigating reciprocity and the possibility of a path integral formulation is compelling future work.

Divergence. Divergence can occur if the total stylized outgoing radiance is greater than the total incoming radiance,

$$\int_{\mathcal{V}} L(\mathbf{x},\mathbf{y})\mathrm{d}\mathbf{x} > \int_{\mathcal{V}} L(\mathbf{y},\mathbf{z})\mathrm{d}\mathbf{z}\,. \tag{8}$$

This is similar to the conservation of energy observed in the rendering equation.

Pixel forming. The pixel forming equation (2) is unchanged. We simply replace the definition of $L(\mathbf{x},\mathbf{y})$ in Eq. (1) with that of Eq. (6).

Parametrization. The style function g_θ can be parameterized, allowing for stylizations that vary over the parameter(s) θ. For example, a spatially varying style function can be achieved by parameterizing g_θ with the vertex position $\mathbf{x}$, $\theta = \mathbf{x}$, or object-specific stylization can be achieved by parameterizing g_θ with the object ID. We can also control which bounces to perform stylization at, for example, only the first bounce, which captures many existing screen-space stylizations, or at only the second bounce so that an object looks different whether it is viewed directly or through a reflection. Put another way, even though the formulation is recursive, stylization does not need to be, and we have precise control of how and when stylization is performed. The parametrization of g_θ is flexible and is a partial implementation of the mapping matrix presented by Gooch [2010].

Composition. Two style functions, $g_{\theta 1}$ and $g_{\theta 2}$, can be composed to form yet another style function, $g_\theta(I) = (g_{\theta 1} \circ g_{\theta 2})(I)$. One compelling composition is that of map-reduce, where we map $g^{\text{map}}_\theta(I)$ to a different dimensionality, apply a style function $g'_\theta(I)$, then reduce $g^{\text{rdc}}_\theta(I)$ back to the original dimensionality, such that $g_\theta(I) = (g^{\text{rdc}}_\theta \circ g'_\theta \circ g^{\text{map}}_\theta)(I)$. This sort of map-reduce can be useful for styles like cel-shading, where we need to map a vector sample to a scalar brightness, apply a function that converts the brightness to (potentially discrete) bands, and then reduce the modified brightness back to a vector sample. We demonstrate further examples of composition in Section 5.3 and Fig. 10.

Tristimulus rendering. For tristimulus rendering, the k in $g_\theta : \mathbb{R}^k \to \mathbb{R}^k$ is 3 and the color channels are the components of the vector function (e.g. red, green, and blue in RGB rendering). While we can estimate the rendering equation (1) independently for each color channel, this is not necessarily the case with the stylized rendering equation (6), as some style functions g_θ may introduce component-dependence (e.g. the cel shading example in the previous paragraph). More explicitly, the stylized rendering equation (6) is component-separable only if the style function g_θ can be decomposed into k component-specific scalar style functions $g_{\theta i} : \mathbb{R} \to \mathbb{R}$ such that $g_\theta(\mathbf{x}) = \langle g_{\theta 1}(x_1), \ldots, g_{\theta k}(x_k)\rangle$ for $x_1, \ldots, x_k \in \mathbf{x}$. Appendix B provides further details on component dependence and separability.

Compatibility with non-transport stylizations. The stylized rendering equation (6) expresses how light is transported, but places no restriction on, and makes no modifications to, the underlying geometry or materials of the scene. This allows us to combine NPR stylizations captured by Eq. (6) out-of-the-box with methods that modify geometric and material properties, such as mesh deformation or a wide variety of appearance editing methods [Schmidt et al. 2016] that modify material properties, lighting, and object properties like vertex positions and surface normals. We demonstrate one such example in Fig. 12.

4 MONTE CARLO ESTIMATION OF THE STYLIZED RENDERING EQUATION

Given the generalized formulation we would then like to derive a practical Monte Carlo estimator. To do so, we will first look at unbiased and biased estimators for individual style functions (i.e. functions of expectation) $g_\theta(I)$. We will then explore how to compose these local estimators into a complete estimator for recursive functions of expectation, and how to implement this complete estimator in practice.

For some style functions g_θ it is possible to produce an unbiased estimate of $g_\theta(I)$ from a group of unbiased estimates $\langle I\rangle_1, \langle I\rangle_2, \ldots$ of I. We refer to such estimators as *group-unbiased* estimators, or more compactly, *gu-estimators*.

Group-unbiased estimators. For a given style function $g_\theta : \mathbb{R}^k \to \mathbb{R}^k$, a gu-estimator is an explicitly defined function $\hat{g}_\theta : \mathbb{R}^{n\times k} \to \mathbb{R}^k$ that takes n unbiased estimates $\langle I\rangle_1, \ldots, \langle I\rangle_n$ and outputs an unbiased estimate of $g_\theta(I)$, i.e.

$$g_\theta(I) = \mathrm{E}[\hat{g}_\theta(\langle I\rangle_1, \ldots, \langle I\rangle_n)]\,. \tag{9}$$

We will refer to the family $\mathcal{G}$ of style functions for which there is a tractable, known gu-estimator as *group-unbiased style functions*,

$$\mathcal{G} := \left\{ \begin{array}{c} g_\theta : \mathbb{R}^k \to \mathbb{R}^k : \exists \text{ gu-estimator } \hat{g}_\theta : \mathbb{R}^{n\times k} \to \mathbb{R}^k \text{ s.t.} \\ \forall \text{ integral } I \in \mathbb{R}^k \text{ with unbiased estimates } \langle I\rangle_i, \\ g_\theta(I) = \mathrm{E}[\hat{g}_\theta(\langle I\rangle_1, \ldots, \langle I\rangle_n)]\,. \end{array} \right\} \tag{10}$$

Biased estimators. For style functions for which there is no known gu-estimator (e.g. the step function), or for which unbiased estimation is prohibitively expensive or slow to converge, we can use biased estimation. Biased estimators have the nice property that they tend to work for (almost) any function and have intuitive, controllable properties. We will look at two different biased estimation strategies: finite-term polynomial approximation of the style function g_θ (for which we can construct a gu-estimator), and directly applying the style function g_θ to an estimate of I (for which we can construct a telescoping series).

4.1 Prior work in unbiased estimation of functions of expectation

Several prior works in statistics and computer graphics have proposed group-unbiased estimators for functions of expectation. Let us briefly review the concepts and estimators introduced by these prior works, as they will provide the foundational components of a more general class of group-unbiased estimators in Section 4.2. For an in-depth exposition, we refer the reader to the supplemental material.

4.1.1 *Power series.*

$$g_\theta(I) \leftrightarrow h_\theta(I) = \sum_{k=0}^{\infty} a_k (I-b)^k \tag{11}$$

For an analytic style function g_θ we can construct a group-unbiased estimator from its Taylor series expansion h_θ around an expansion point b,

$$h_\theta(I) = \sum_{k=0}^{\infty} \underbrace{\frac{g_\theta^{(k)}(b)}{k!}}_{=a_k} (I-b)^k, \tag{12}$$

where $g_\theta^{(k)}(b)$ is the k^{th} derivative of g_θ evaluated at the expansion point b.

To construct a practical estimator for a power series like in (12) we will need to select a finite number of terms to evaluate while still maintaining the unbiasedness of the estimator. Georgiev et al. [2019] discuss several strategies to do so, including the iterative prefix-sum strategy, which selects n with some probability $P(n)$ and weights each of the first n terms by the probability $P(k < n)$ that their index k is less than n,

$$\langle h_\theta(I) \rangle_{\text{iter}} = \sum_{k=0}^{n-1} \frac{g_\theta^{(k)}(b)}{k!} \frac{\langle (I-b)^k \rangle}{P(k<n)}. \tag{13}$$

Each of the polynomial terms $\langle (I-b)^k \rangle$ can then be unbiasedly estimated using a product of k samples of $\langle I \rangle - b$, or the variance-minimal recurrence relation of Kettunen et al. [2021].

While a majority of the current work on unbiased estimation of analytic functions of expectations focus on Taylor expansion, we note here that there are other ways to expand analytic functions into a power series. For example, Bernstein polynomials, Chebyshev polynomials, and Legendre polynomials are popular alternatives used in physics, engineering and applied mathematics literature. Applying these polynomial bases to rendering is an interesting direction of future work.

4.1.2 *Telescoping series.*

$$g_\theta(I) \leftrightarrow J(\infty) = J(k) + \sum_{j=k}^{\infty} \underbrace{J(j+1) - J(j)}_{\Delta_j} \tag{14}$$

Similarly to power series, telescoping series estimators [McLeish 2011; Rhee and Glynn 2012; Misso et al. 2022] are based on a convergent infinite series representation (14). In the context of stylized rendering, to form a telescoping series we can reformulate $g_\theta(I)$ as $g_\theta(I) = \lim_{k\to\infty} J(k)$, where $J(k)$ is a biased approximation of $g_\theta(I)$ and the bias monotonically vanishes as the parameter k approaches infinity.

One possible way to formulate $J(k)$ is as $J(k) = g_\theta(I(k))$ for some sequence $I(k)$ that converges to I, the exitant radiance we wish to stylize. When the function g_θ is continuous, one can then write $g_\theta(I) = \lim_{k\to\infty} g_\theta(I(k))$. Setting $I(k) = \frac{1}{k}\sum_{i=1}^{k} \frac{f(x_i)}{p(x_i)}$, a k-sample Monte-Carlo estimator of I is a natural choice, and, although realizations of $I(k)$ are random variables, the continuous mapping theorem of Mann and Wald [1943] guarantees convergence of $g_\theta(I(k))$ to $g_\theta(I)$.

To construct a group-unbiased telescoping series estimator, we can set $J(k) = g_\theta(I(k))$, and then use a term selection strategy such as iterative prefix-sum to stochastically select the first n terms,

$$\langle g_\theta(I) \rangle_{\text{tele}} = g_\theta(I(k)) + \sum_{j=k}^{k+n-1} \frac{g_\theta(I(j+1)) - g_\theta(I(j))}{P(j-k<n)}. \tag{15}$$

4.2 Combining group-unbiased style functions under certain operations

$$g_\theta = g_{\theta 1} \oplus g_{\theta 2} \tag{16}$$

Group-unbiased style functions g_θ have the interesting property that they can be combined under certain operations $\oplus$ to produce yet another group-unbiased style function for which we can construct a gu-estimator $\hat{g}_\theta$. This allows us to construct unbiased estimators for more sophisticated style functions by using simpler style functions as building blocks.

Theorem 4.1. *The family of group-unbiased style functions $\mathcal{G}$ defined in Eq. (10) is closed under the following operations (see Appendix D for proof):*

- *Addition:* $g_{\theta 1}, g_{\theta 2} \in \mathcal{G} \Rightarrow g_{\theta 1} + g_{\theta 2} \in \mathcal{G}$.
- *Component-wise multiplication:* $g_{\theta 1}, g_{\theta 2} \in \mathcal{G} \Rightarrow g_{\theta 1} \cdot g_{\theta 2} \in \mathcal{G}$.
- *Function composition:* $g_{\theta 1}, g_{\theta 2} \in \mathcal{G} \Rightarrow g_{\theta 1} \circ g_{\theta 2} \in \mathcal{G}$.

In mathematical terms, $\mathcal{G}$ forms a function ring *(more specifically, an integral domain) under addition and component-wise multiplication that is also closed under function composition.*

To use this in practice, for some operation on the operand style functions $g_{\theta 1}$ and $g_{\theta 2}$, we only need to perform a corresponding operation on their respective gu-estimators $\hat{g}_{\theta 1}$ and $\hat{g}_{\theta 2}$:

- Addition: $\hat{g}_\theta = \hat{g}_{\theta 1} + \hat{g}_{\theta 2}$ is a gu-estimator for $g_{\theta 1} + g_{\theta 2}$ by linearity of expectation, requiring only $\max\{n_1, n_2\}$ samples.
- Component-wise multiplication: $\hat{g}_\theta = \hat{g}_{\theta 1} \cdot \hat{g}_{\theta 2}$ is a gu-estimator for $g_{\theta 1} g_{\theta 2}$, requiring $(n_1 + n_2)$ mutually-independent estimates of the integral I.
- Composition: $\hat{g}_\theta(I) = \hat{g}_{\theta 1}(\{\hat{g}_{\theta 2}(\{\langle I \rangle_{1,\ldots,n_2}\})_{1,\ldots,n_1}\})$ is a gu-estimator for $g_{\theta 1} \circ g_{\theta 2}$, requiring n_1 independent estimates of $g_{\theta 2}(I)$ for each evaluation of $\hat{g}_{\theta 1}$, and n_2 independent estimates of I for each evaluation of $\hat{g}_{\theta 2}$, for a total of $n_1 n_2$ independent estimates of I.

We provide a concrete example of combining group-unbiased estimators in Appendix D.

Up to this point we have considered style functions g_θ that were group-unbiased. For style functions for which there is no known gu-estimator (e.g. those with discontinuities), or for which unbiased estimation is prohibitively expensive or slow to converge, we can use biased estimation.

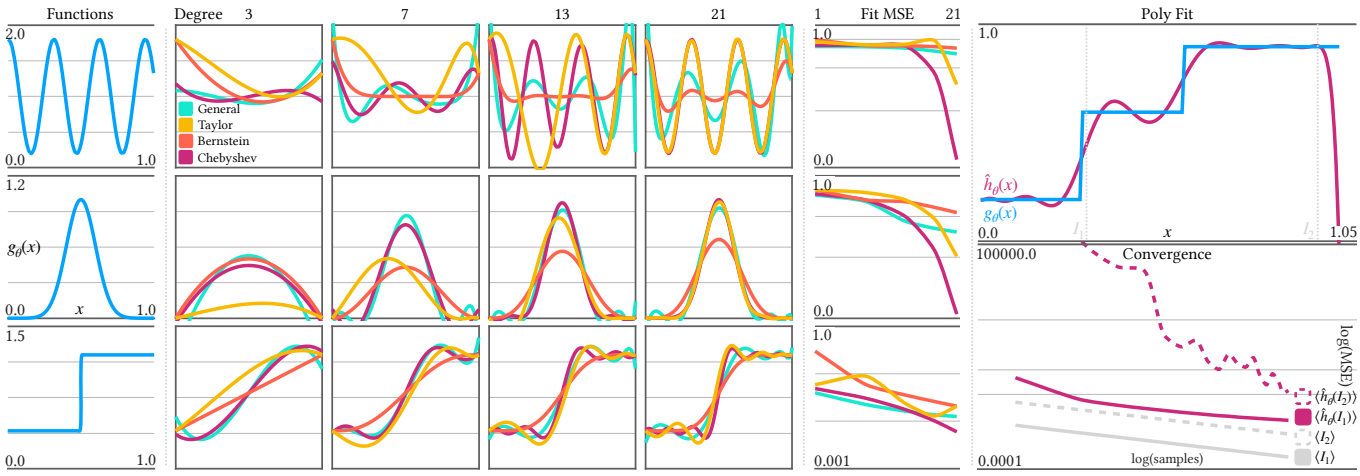

Fig. 4. Left: Different polynomial bases will provide more accurate approximations depending on the function being approximated and the desired maximum degree. In general, if nothing is known about the function being approximated, the Chebyshev basis is a good choice, showing superior convergence as the maximum degree increases. Right: Discontinuities in the original function g_θ can be challenging to approximate with a finite degree polynomial, resulting in potentially undesirable fluctuations in otherwise flat regions. Furthermore, many polynomial bases are only guaranteed to converge to the function over a fixed interval, with no guarantees on function shape outside of this interval. We show convergence for two integrals (right): I_1 where the integral and all samples fall within the fit interval, and I_2 where the integral falls within the interval, but the integrand samples may fall outside. Here polynomial approximation $\tilde{h}_\theta$ grows exceptionally negative for values greater than 1, and the gu-estimator $\langle \tilde{h}_\theta(I_2) \rangle$ suffers from extreme variance and slow convergence.

4.3 Polynomial approximation

$$g_\theta(I) \approx \tilde{h}_\theta(I) = \sum_{k=0}^{n} a_k I^k \tag{17}$$

One strategy for biased estimation of a style function g_θ is polynomial approximation. We can use a finite-degree polynomial $\tilde{h}_\theta$ to approximate the shape of the style function g_θ and construct a group-unbiased estimator for the terms. If the shape of the polynomial is sufficiently close to the shape of the style function the resulting bias from approximation may be acceptably low. We provide further discussion of polynomial approximations in the supplemental material.

4.3.1 Canonical experiments. In Fig. 4 (left) we explore the fit accuracy of a general polynomial approximator that attempts to find the best fit coefficients directly, and three bases: Taylor, Bernstein, and Chebyshev, for an ablation of different style functions g_θ and degree. Bernstein polynomials show poor fit and slow convergence for all three style functions. As a general trend, all other methods are comparable for lower degree fits. For higher degrees, Chebyshev (and Legendre) basis shows superior convergence than the commonly used Taylor approximation which computes local fits.

In Fig. 4 (right) we demonstrate some issues inherent to polynomial approximations. Most polynomial fits are computed over a fixed interval, and show erratic behavior even slightly outside this interval. This is even more prominent for functions that drastically change behavior over a small domain, for example near discontinuities due to large coefficients for higher-order terms in the approximation. Further, estimating higher order terms require more samples, potentially leading to more variance and numerical precision issues. As such, numerical stability is only guaranteed when all estimates $\langle I \rangle_i$, as well as the final integral value I, are entirely contained in the fit interval. In practice, one can choose a fit interval that sufficiently covers the sample distribution and clamp outliers to the interval. While clamping is biased, the bias introduced is usually smaller than the variance of outliers (see the render in Fig. 11).

These drawbacks of polynomial approximation, especially for discontinuous functions, leads us to investigate alternative strategies that introduce controllable bias but admit numerical stability.

4.4 Direct application

$$g_\theta(I) \approx g_\theta\left(\langle I \rangle\right) \tag{18}$$

One of the most straight-forward strategies for biased estimation of $g_\theta(I)$ is to directly apply the style function g_θ to an estimate $\langle I \rangle$ of the integral I. While direct application is generally biased, it provides a reliable, fixed budget fallback option for when there are no tractable options for unbiased estimation of g_θ, or when a polynomial approximation $\tilde{h}_\theta$ produces an unstable, high-variance estimator.

One such inner-estimator $\langle I \rangle$ is an ordinary n-sample MC estimator,

$$g_\theta\left(\langle I \rangle\right) = g_\theta\left(\frac{1}{n}\sum_{i=1}^{n}\frac{f(x_i)}{p(x_i)}\right). \tag{19}$$

Note that this is the same as the initial term $I(n)$ of the telescoping series example discussed in Section 4.1.2.

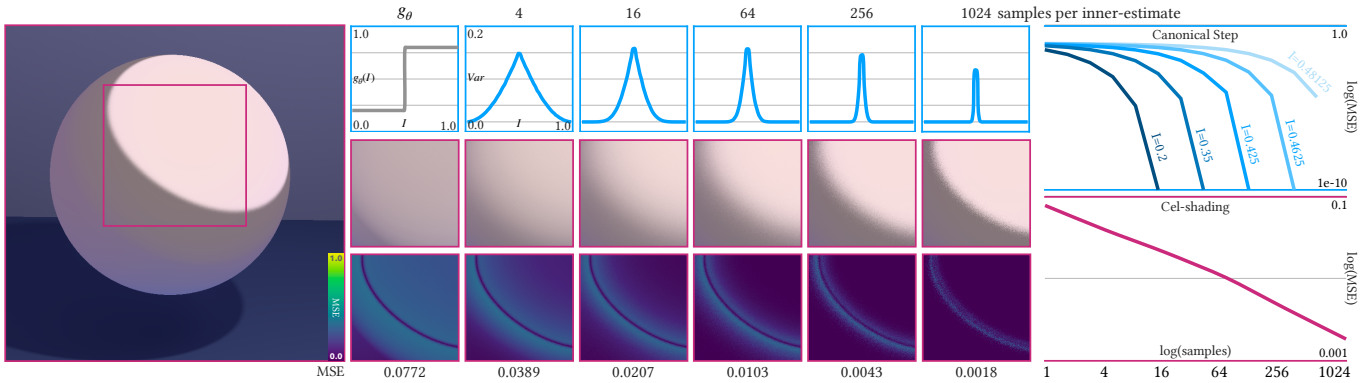

Fig. 5. For a biased direct application estimator (19) of a step function of expectation, and its rendering counterpart, cel shading, the variance of the inner-estimator $\langle I \rangle$ results in a probability of estimates $g_\theta(\langle I \rangle)$ falling on either side of the step. Averaging many such estimates produces a mollification of the style function around the step boundary. We note that variance is high for integrals closer to the step boundary, as a small change in value of the inner-estimate $\langle I \rangle$ can result in a large change in value at the discontinuous step boundary. However, for a given integral I, as the number of samples (column labels) of the inner-estimate $\langle I \rangle$ increases, the direct application estimate $g_\theta(\langle I \rangle)$ of the step function of expectation converges super-linearly (right). We explore the phenomenon further in Appendix C.

Due to the nature of the Monte-Carlo estimation of I, increasing the sample count n of the inner-estimator in Eq. (19) leads to a linear decrease in the variance of $\langle I \rangle$. The bias in this estimator can be quantified via a Taylor expansion of g_θ around I as follows (assuming that g_θ has well-behaved derivatives at I),

$$g_\theta(\langle I \rangle) - g_\theta(I) \approx g'_\theta(I)(\langle I \rangle - I) + \frac{g''_\theta(I)}{2}(\langle I \rangle - I)^2 + O((\langle I \rangle - I)^3)\,. \quad (20)$$

By taking expectation of both sides of Eq. (20) and ignoring the higher order error terms,

$$\mathrm{E}[g_\theta(\langle I \rangle) - g_\theta(I)] \approx \frac{g''_\theta(I)}{2} \cdot \mathrm{Var}(\langle I \rangle)\,. \quad (21)$$

Hence, in practice, the expected bias in Eq. (19) is approximately proportional to the Variance of $\langle I \rangle$, resulting in a mollification of the style function. A closer analysis of the bias in Eq. (19) for non-analytic style functions g_θ is much more challenging, but would lead to interesting applications in debiasing. We leave this avenue of investigation as future work.

Canonical experiments. In Fig. 5 we demonstrate the effect of biased estimation on a challenging case for polynomial bases, the step function. When g_θ is a step function, (20) suggests that locally, when I falls far from the point of discontinuity, the bias in the estimator is directly proportional to the error $\hat{I} - I$. Further, in the case of a step function we observe a super-linear convergence of the bias with respect to the sample count n (i.e. variance) of the inner-estimator. We delve into the details of this fast convergence in Appendix C.

Analogous to the step function, cel shading in rendering maps continuous brightness values to discrete bands. While we observe similar super-linear convergence to the step function for individual pixels (insets, bottom rows), the overall convergence of the entire region is near-linear, with error concentrating around the discontinuity (right, bottom). Here the bias caused by variance of the inner-estimate produces a blurring-like effect on the step boundary that fades as sample count of the inner-estimate increases.

4.5 A complete estimator

We can construct a complete estimator for the pixel forming equation (2) for the stylized rendering equation (6) by replacing each function of expectation $g_\theta(I)$ of the recursive formulation with a group-unbiased estimator thereof, such that,

$$\langle I_\rho \rangle = \frac{W_\rho(\mathbf{x},\mathbf{y})G(\mathbf{x},\mathbf{y})\langle L(\mathbf{x},\mathbf{y})\rangle}{p(\mathbf{x},\mathbf{y})}\,, \quad (22)$$

where $\langle L(\mathbf{x},\mathbf{y})\rangle$ is,

$$\langle L(\mathbf{x},\mathbf{y})\rangle = \hat{g}_\theta\left(\left\{L_e(\mathbf{x},\mathbf{y}) + \frac{f_r(\mathbf{x},\mathbf{y},\mathbf{z}_i)G(\mathbf{y},\mathbf{z}_i)\langle L(\mathbf{y},\mathbf{z}_i)\rangle}{p(\mathbf{z}_i|\mathbf{x},\mathbf{y})}\right\}_{1\cdots k}\right), \quad (23)$$

and where $\hat{g}_\theta$ is a group-unbiased estimator for the style function g_θ, $i \in 1, \cdots, k$, and the number of estimates k required to produce a single estimate $\langle L(\mathbf{x},\mathbf{y})\rangle$ is $\hat{g}_\theta$-dependent (e.g. for a group-unbiased polynomial estimator $\hat{g}_\theta = \langle I^n \rangle$ the required number of estimates k should be at least n).

The recursive MC integration estimator for the rendering equation (3) requires only one sample per expansion. In contrast, each expansion of the complete estimator (23) requires a minimum of k samples. Recursively satisfying the sampling requirement k of the estimator $\hat{g}_\theta$ at each next surface point $\mathbf{z}$ will produce a tree of samples (see Fig. 6). Let us emphasize that, while a tree is the minimal unit of sampling for the complete estimator (23), the sampling cost is *not necessarily exponential*. We discuss this point in Section 4.6.

Unbiasedness. The complete estimator is then unbiased if, for the given parametrizations θ, and the given integrals I, the stylized rendering equation (6) is convergent and each estimator $\hat{g}_\theta$ is group-unbiased (see Appendix A for proof). Conversely, for some style function g_θ, if we choose to use a non-group-unbiased estimator, for example the biased direct application estimator of Section 4.4, $\hat{g}_\theta = g_\theta(\langle I \rangle_1, \cdots, \langle I \rangle_k)$, there is no guarantee on unbiasedness.

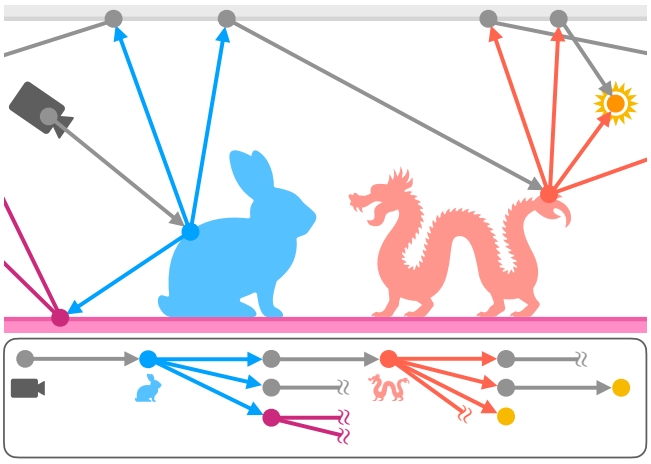

Fig. 6. Evaluating a group-unbiased estimator at a surface point may require more than one integrand sample (e.g. the blue vertex on the bunny requires 3 samples). Starting from the camera and recursively expanding outwards, we observe that satisfying the sample requirements of each next gu-estimator forms a tree (bottom), similar to that of distribution ray tracing [Cook et al. 1984].

4.6 Implementation

One way to satisfy the sampling requirement of the complete estimator (23) is depth-first recursion. For a given pixel ρ, we start from the camera at $\mathbf{x}$ and trace a ray towards a first hit-point $\mathbf{y}$. At $\mathbf{y}$ we determine the sampling requirement k for the local estimator $\hat{g}_\theta$ and draw a first sample $\mathbf{z}$. This process repeats at each $\mathbf{z}$ and we proceed to recursively expand the complete estimator until a termination condition is met, for example, hitting the sky box or a fully absorbing surface. This sampling algorithm is similar in spirit to distribution ray tracing [Cook et al. 1984], and we provide a more concrete form in Algorithm 1.

Sampling cost. The sampling cost of Algorithm 1 is highly tunable. As discussed in Section 3.1, we can use parametrization to control when and how an object is stylized, and this concept extends to sampling cost. By stylizing at only specific levels of recursion, or alternatively, applying a stylization a specific number of times along a path of recursion, we can keep the sampling cost bounded. For example, by performing stylization only the first time an object is encountered along a path of expansion, we achieve a natural visual cohesion between the object and its environment for a fixed, linear sampling cost. Further, we can use techniques like Russian roulette [Rath et al. 2022] to early-terminate the expansion of tree branches without affecting the unbiasedness of the estimator.

Divergence. As discussed in Section 3.1, divergence can occur if, for a given style function, the total stylized outgoing radiance is greater than the total incoming radiance. One solution is to apply energy-increasing style functions at only a finite number of depths of recursion. There may still be an increase in total system energy, but the system won't diverge. Another option is to clamp excess energy, similar to iterative path filtering [Deng et al. 2021]. Clamping, however, can significantly affect the rendered result.

Algorithm 1: Tree sampling for stylized rendering

```
for pixel ∈ image do
    ray = camera.sampleRay(pixel)
    x = ray.origin
    y = scene.intersects(ray)
    sample = W_ρ(x, y)G(x, y)stylize(x, y)/p(x, y)
    image[pixel].accumulate(sample)
Input: x: the previous vertex, y: the current vertex, θ: the stylization
       parameters (e.g. path prefix, vertex depth, position, time, etc.)
Output: L_s: the stylized radiance
Function stylize (x, y):
    style = determineStyle(θ, y)
    sampleCount = style.requiredSamples(θ, y)
    samples = []
    for i ∈ {0, ..., sampleCount − 1} do
        if isTerminal(y) then
            L_o = L_e(x, y)
        else
            ray = sampleNextRay(x, y)
            z_i = scene.intersects(ray)
            L_i = stylize(y, z_i)
            L_o = L_e(x, y) + f_r(x, y, z_i)G(y, z_i)L_i/p(z_i|x, y)
        samples.add(L_o)
    L_s = style.apply(θ, samples)
    return L_s
```

Consistency when using biased estimators. By itself a biased estimator like the direct application estimator (19) is *consistent*, i.e. as n tends to infinity over a single realization it will converge on the correct solution. However, in the complete estimator (23), we have nested local estimators, and for the complete estimator to produce a sample with bounded memory, each biased local estimator must produce an estimate in a bounded number of samples. Exploration of methods to make the complete estimator consistent even with biased local estimators would be an interesting direction of future work.

Discontinuous style functions. For a discontinuous style function we are unable to apply the unbiased estimation strategies of Sections 4.1 and 4.2 due to its discontinuous nature, and the biased estimation strategies of Sections 4.3 and 4.4 can produce undesirable fluctuations and mollification. For results in this paper we use a biased direct application estimator (19) with high sampling rates for sufficiently low mollification. Such high sampling rates are permissible when demonstrating the theory, but can be too costly for general use. In practice, clean edges can be worth the cost of a slight divergence from theory and a modest amount of additional bias. If information from deeper recursion levels is not required (i.e. an estimate of global illumination is not necessary for stylization) we recommend using a fast, low error integrator, like linearly transformed cosines for direct illumination [Heitz et al. 2016, 2018].

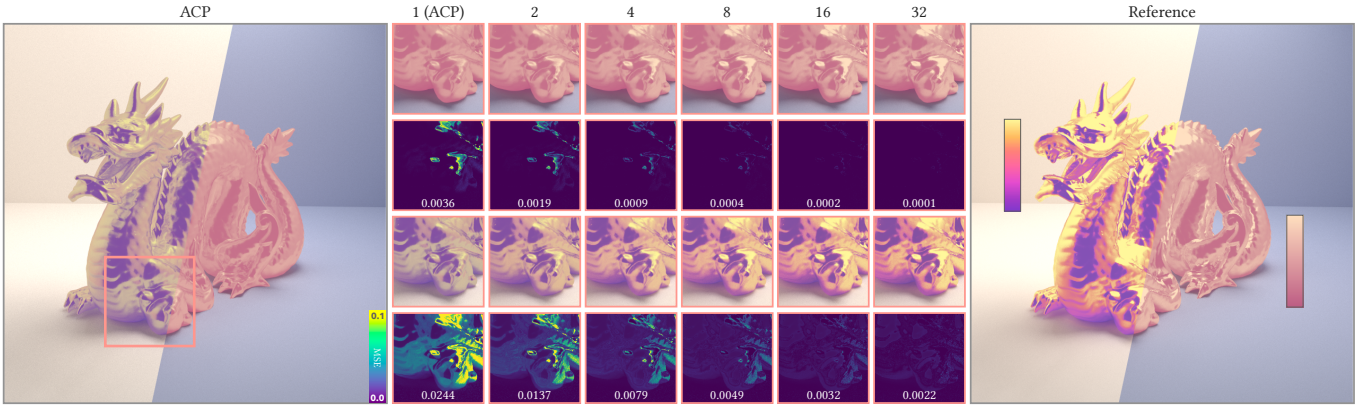

Fig. 7. A style function that maps exitant radiance to a color gradient allows us to specify exact colors for dark and bright regions. ACP [Doi et al. 2021] approximates this mapping by applying the style function directly to a single sample estimate of the exitant radiance. For near-linear gradients (right split) the bias introduced by using a single sample estimate is minimal. For more complex gradients (left split) the variance of the single sample estimate can result in color distortion. Their method, however, is an instance of the direct application estimator in Section 4.4. As we increase the sample count of the inner-estimate (middle insets) the variance is reduced and the bias of direct application quickly fades, with the 32 sample results almost visually indistinguishable from the reference.

5 RESULTS

Now that we have a complete estimator and a sampling algorithm, let us explore how we can use it to render compelling visuals. We will first look at several examples of basic stylization, namely: color mapping and cel shading, and compare our rendered results to those generated by Doi et al. [2021]. We will then look at properties of the proposed method, including recursive stylization and parametrization, and a few examples of unbiased estimation via power series and combining gu-estimators. We will wrap up with rendering examples of polynomial approximation estimators, several existing NPR styles [Deussen et al. 1999; Hall 1999; Gooch et al. 1998; Barla et al. 2006; West 2021], and a discussion of more advanced stylization scenarios like those shown in the teaser Fig. 1 and Fig. 13.

All results were rendered on a CPU-based RGB renderer using the sampling algorithm detailed in Section 4.5, using a single workstation with a 24-core (8 performance, 16 efficiency) Intel i9-13900KF and 64GB of 4800 Mhz DDR5 memory. For our main results and comparisons we use a standardized test scene. This allows us to focus on specific stylizations and how they affect transport, making the issues they face, effects they produce, and comparisons clear and easy to parse. We include specific experimental setup and implementation details in the supplemental material.

All results shown in this paper, including Figs. 2, 3 and 5, were rendered using the proposed formulation, estimator, and sampling algorithm. The only exception is Fig. 8 where we implemented the FTV method of Doi et al. [2021] as a separate integrator.

5.1 Color mapping and cel-shading

One common stylization is color mapping, where we take some value (e.g. exitant radiance) and map it to a color. For example, colorization of monotone images, heat maps, error maps like those seen in Fig. 7, and even cel shading. Cel shading, an instance of color mapping, converts the value to discrete bands of color. This can replicate a technical limitation in some early image forming systems and processes where only a finite number of colors were available. Some modern uses of cel shading limit, not the color palette, but the brightness of colors, providing for a similar visual style with greater freedom in color expression.

Doi et al. [2021] present two methods for integrating specific instances of color mapping into a PBR framework: *Affects the Color on Path trace* (ACP) and *Fidelity for Texture Values* (FTV). We will look at the results of integrating those same methods into the proposed formulation, and compare both the visual and convergence rate differences. At their core, both methods of Doi et al. [2021] are specific instances of the biased direct application estimator from Section 4.4 with limited parameterizations θ. This lets us leverage our understanding of the convergence properties (21) of the direct application estimator (19) to overcome several challenges in these previous methods.

5.1.1 Color mapping and ACP. ACP maps the range of exitant radiance *estimates* of individual path tracing samples to a gradient, and many of such mapped estimates are then averaged to form an image. This has the interesting quirk that the resulting average might not be a color on the gradient. ACP is an instance of the direct application estimator from (19) where the sample count n equals 1, and from the convergence properties (21) we know that the color inaccuracy produced by the method is proportional to the variance of the radiance estimates.

In Fig. 7 we demonstrate the output visuals of ACP and the more general direct application estimator on two different gradients: one near-linear and one non-linear. For near-linear gradients (right split) the bias introduced by using a single sample estimate is minimal. Radiance estimates that exceed the maximum mapped value of the gradient are clamped to the maximum value, producing slight underestimation. As we increase the sample count n of the inner-estimate the underestimation bias quickly vanishes.

For non-linear gradients (left split) the variance of the single sample estimate can result in potentially strong color inaccuracy. If sample variance is high, samples may be mapped to a wide range of gradient colors, and the average of these colors is not guaranteed

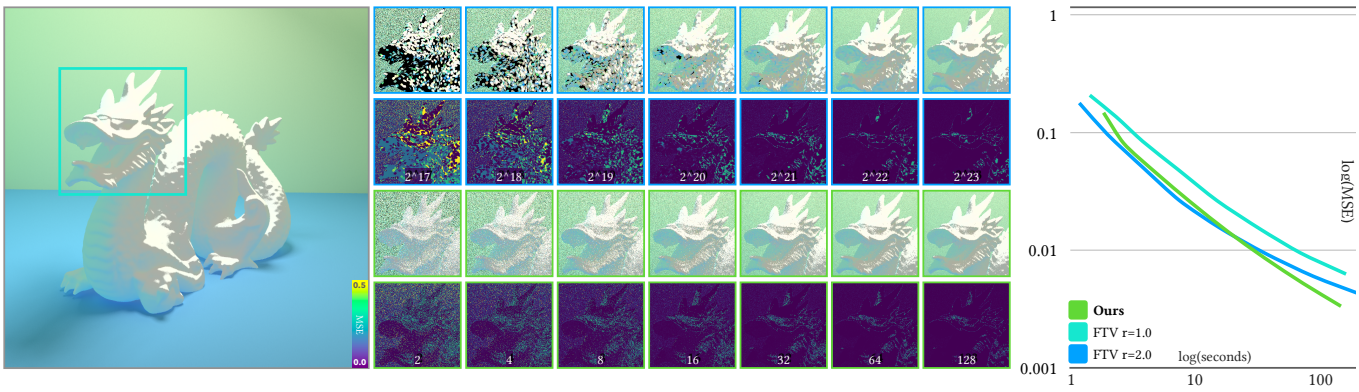

Fig. 8. Here we demonstrate a global illumination variant of cel shading that produces natural looking color bleed (see the dragon's chest) while still preserving the trademark discrete bands of cel shading. When using a direct application estimator (19) for cel shading, small fluctuations due to variance of the inner-estimate can result in large fluctuations in the cel shading estimate. Top rows: FTV [Doi et al. 2021] uses photon density estimation to mitigate the variance of the inner-estimate, but leads to splotchy artifacts for early renders and gradual accumulation of bias as rendering progresses. Bottom rows: The proposed brute force estimator has controllable bias, leading to better estimates given a sufficient sampling budget.

to exist in the gradient. Increasing the sample count n of the inner-estimate can reduce this color inaccuracy, and at 32 samples per estimate we observe almost no visual difference from the reference.

5.1.2 Cel shading and FTV. Similarly to ACP, FTV also maps exitant radiance to a color, but leverages photon density estimation to produce highly converged estimates before applying the color mapping. This allows for the highly accurate, low variance estimates required by some mappings, for example cel shading, where high variance estimates result in mollification of the otherwise discrete color bands.

Discussion. In Fig. 8 we demonstrate the convergence of the proposed sampling algorithm (1) and FTV [Doi et al. 2021] for a single style evaluation with increasing sampling budget. When using a direct application estimator (19) for cel shading, small fluctuations due to variance of the inner-estimate can result in large fluctuations in the cel shading estimate. FTV [Doi et al. 2021] (top rows) attempts to overcome this by producing a low variance inner-estimate using photon density estimation, showing lower error for early estimates but suffering from accumulated bias as rendering progresses. Here we note that the correlated sample use of photon density estimation results in splotchy artifacts that fade as the number of photons increases from 2^{17} to 2^{23}. The proposed brute-force algorithm (1) (bottom rows) is less efficient, but has controllable bias allowing for lower error estimates with a sufficient sampling budget. Here we see strong blurring of the discrete bands at 2 samples per inner-estimate that disappears as the sampling budget increases.

While the concept of reusing samples to efficiently reduce inner-estimate variance is inspiring, the use of photon density estimation in stylized rendering poses some challenges. As photons are randomly cast throughout the scene, we can not directly control the number of samples available at estimator evaluation time. Variable sample count is a minor issue for a biased estimator like Eq. (19), which can function with as few as 1 sample, but can potentially fail to produce an estimate for power series estimators (Section 4.1.1), where there is a minimum required number of samples. Moreover, as the underlying method of Doi et al. [2021] is progressive photon mapping, stylization can only be applied at one vertex along a path, which places a limitation on the types of visualizations achievable.

Despite these drawbacks, the concept of reuse is inspiring and would make for an interesting direction for future work.

5.2 Parametrically-varying style functions

The style function g_θ of the proposed formulation is parameterized by θ, allowing us to control which objects are stylized, how that stylization changes over the parameters θ, and even how styles are blended.

In Fig. 9 (left), we demonstrate recursively applying a style function that increases color saturation. Here we see a visible increase in color intensity around regions of strong inner-reflections as the levels of recursion increases. However, performance suffers greatly as the number of recursion levels increases from 0 to 3, with 3 levels of recursion incurring a 22 times larger overhead compared to the unstylized render. Each evaluation of the style function requires 8 samples, with 3 levels of recursion effectively requiring $8^3 = 512$ samples for a single pixel sample. Unless recursive stylization (i.e. applying the same stylization on the same object recursively) is essential to a desired look, we strongly recommend using parametrization to apply a style at only a limited number of recursions.

In Fig. 9 (right), we demonstrate several permutations of stylizing an object at different vertex depths, and how that stylization is further affected by other stylizations in its path prefix. In the first row we can see clear visual discontinuity when stylizing at specific vertex depths, with the fourth column showing stylization only when the dragon is viewed through both the glass pane and the mirror. In the fourth row we see the effect of all three stylizations interacting, where stylizations earlier in the path modify the look of stylizations deeper in the path. The fifth column uses a "first hit only" parametrization that performs stylization of an object only the first time it is encountered along a path through the tree sample. This results in visual coherency without the excessive cost of self-recursion.

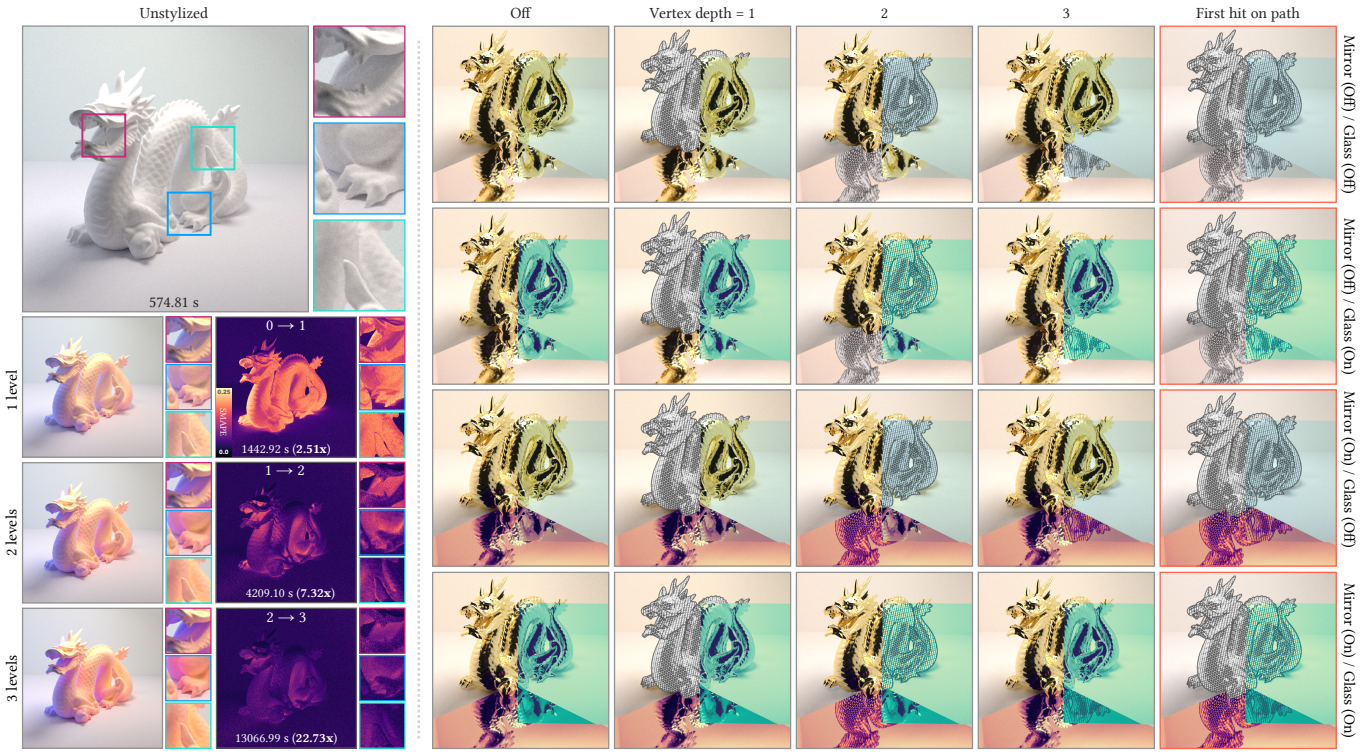

Fig. 9. Left: The stylized rendering equation (6) has the unique aspect that we can apply styles *recursively*. Here we show 4 configurations of a saturation-increasing style function performing stylization 0, 1, 2, and 3 times along a given path. We can observe a clear difference in rendered output as the number of levels of recursion increases, with the bottom image showing strong saturation in regions of inner-reflection. Recursive stylization, however, comes at a significant cost (see time measurements) and we recommend only recursively styling an object when it is essential to the visual design. Right: Similarly, the parametrization of g_θ allows us to control exactly when stylization occurs. Here we demonstrate a how stylizing a dragon at specific vertex depths affects the rendered image (columns). The mirror on the floor and the glass pane on the right side can also be stylized to produce complex chains of stylization (rows). Note that stylizing at specific vertex depths can result in the dragon looking different depending on whether it was viewed directly or indirectly. In the red insets (rightmost column) we demonstrate stylizing that occurs at the first occurrence of the dragon along an expansion path, resulting in consistent stylization regardless of what path we took to reach the dragon.

5.3 Unbiased stylization using power series and certain operations

For analytic style functions, e.g. gamma correction, we can construct a group-unbiased estimator from their power series expansion (see Section 4.1.1). Gamma correction is formulated as a non-integer power of the input, $g_\theta(I) = I^{1/G}$, with a power series of,

$$g_\theta(I) = \sum_{k=0}^{\infty} b^{1/G-k} \cdot \frac{(1/G)(1/G-1)\cdots(1/G-k+1)}{k!} \cdot (I-b)^k . \quad (24)$$

for which we can construct a group-unbiased estimator.

In Fig. 10 (left) we demonstrate unbiased estimation of gamma correction on the surface of the dragon. In the insets (a)-(f) we show the same-time rendering result for 6 different estimation strategies of the power series (24). In inset (a) we use the single-term selection strategy [Georgiev et al. 2019] with a fixed expansion point $b = 0.5$ and a naive polynomial estimator (i.e. product of independent samples). The resulting noise from estimation is severe. In inset (b) we replace the single-term selection strategy with the iterative prefix-sum selection strategy [Georgiev et al. 2019], significantly reducing variance for minimal additional overhead. In inset (c) we further replace the naive term estimator with a recurrence relation term estimator [Kettunen et al. 2021]. In inset (d) we replace the fixed expansion point $b = 0.5$ with an independent, low sample estimation $b = \langle I \rangle$ of the integral I, for a tangible reduction in variance for only a modest increase in sampling cost. For numerical stability purposes we clamp the expansion point b such that $b > 0.1$, which does not affect the unbiasedness of the estimator. In inset (e), in an attempt to maximize the benefit of the recurrence relation term estimator, we introduce oversampling, drawing 2x as many samples as the highest degree of the prefix of each realization. In inset (f) we reach close to the maximum amount of variance reduction we could (empirically) with an 8x oversampling rate. These results suggest that when using the recurrence relation term estimator it can be beneficial to spend a significant portion of the sampling budget on a few, high quality estimates rather than many, lower quality estimates. In the convergence plot (right, purple line) we show the convergence of the high performance estimator used for inset (f).

In Fig. 10 (middle) we show examples of combining two group-unbiased style functions, one for color mapping to a sepia tone (A), and one for increasing contrast (B), to produce yet another group-unbiased style function. Both compositions, $A \circ B$ and $B \circ A$, result in

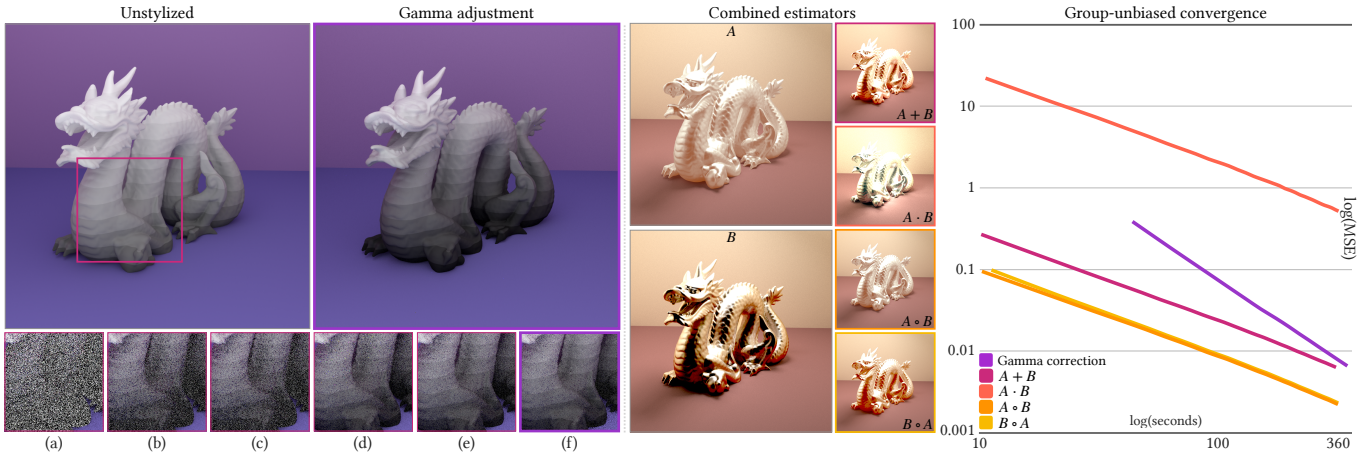

Fig. 10. Analytic style functions can be unbiasedly estimated using their power series expansion (see Section 4.1.1). Here we demonstrate unbiased estimation of gamma correction on the surface of the dragon (left) and its convergence (right plot, purple line). In the insets (a)-(f) we show the same-time rendering result for 6 different estimation strategies of the power series and detail each in Section 5.3. Given 2 group-unbiased style functions A and B we can combine their gu-estimators under addition, multiplication, and composition (see Section 4.2). The resulting combined estimator is yet again group-unbiased and we show their convergence in the plot on the right. Note that the variance of $A \cdot B$ is disproportionately high compared to the other operations as the product of radiance values above 1 produce even larger radiance values.

stylizations with a similar brightness to the operand style functions A and B, and show similar convergence (right, yellow and orange lines). Addition and multiplication, on the other hand, both cause an increase in energy showing higher variance, with multiplication (red line) suffering from extreme variance.

5.4 Biased stylization using polynomial approximation

The direct application estimator (19) is convenient, but its bias comes in the form of mollification of the style function g_θ (see (20)). For strongly non-linear style functions this can produce significant color distortion. Polynomial approximation is also biased, but that bias comes from, not the estimation method, which is unbiased, but the difference in shape of the polynomial approximation and the original style function. For a sufficiently tight fit, and a sample distribution that reasonably falls within the fit interval, the bias introduced will be reasonably small. While an unbiased estimator for the polynomial approximation may have higher variance than a direct application estimator, this variance will vanish as rendering progresses, whereas the bias accumulated by direct application will not.

In Fig. 11 we demonstrate one such example, where we stylize a glossy dragon with a tie-dye-like effect. The tie-dye effect is implemented as cosine waves of different frequency and phase operating on each of the RGB channels independently. We estimate the tie-dye stylization using two estimation strategies: direct application (19) and polynomial approximation (17). For the polynomial approximation, we use a Chebyshev basis up through degree 20 fit on the interval of $[-1, 4]$. We choose 4 as the maximum to allow for exitant radiance samples that exceed 1, and then further clamp any samples that exceed the interval maximum of 4. The direct application estimator (top rows) produces low variance estimates but at the cost of mollifying the style function, producing a desaturation-like color distortion. Polynomial approximation, on the other hand, due to a tight fit, preserves the bright colors of the style but initially shows higher error in the form of variance. As rendering progresses, the direct application estimator continues to accumulate bias, while the polynomial approximation converges out much of the variance, producing lower error and visibly better results given a sufficient computational budget.

5.5 Representing existing stylizations

The proposed formulation captures a subset of existing NPR stylizations, including classical shading methods like the technical drawings of Gooch et al. [1998] and toon shading of Barla et al. [2006], drawing methods like cross-hatching [Deussen et al. 1999], printing methods like half-tone [Hall 1999], and can also be combined with PBR-based NPR methods like the feature line rendering (PBFLR) of West [2021].

In Fig. 12 (left) we replicate the respective Fig.7 of the paper by Gooch et al. [1998] for technical drawing stylization and the paper by Barla et al. [2006] for toon shading stylization. The proposed formulation allows stylizations like these to naturally affect their surroundings. In the respective middle insets we can observe how each stylization affects the shadowed region under the dragon's belly. Interestingly, both methods assume a single incident radiance direction, giving us a closed-form solution independent of the exitant radiance. This allows us to construct a *zero* sample exact solution, making surfaces shaded with these styles terminal conditions in Algorithm 1.

In Fig. 12 (middle) we demonstrate three different ways to handle stylization of lines generated using PBFLR: unstylized, inheriting the style of the surface a line correspond to, and custom user-specified stylization. Here the scene has a grayscale stylization applied to all objects that gives it a similar dynamic range to a pencil sketch. In the unstylized inset we use the default line colors as specified in the scene file; in this case, solid black. In the inherited style inset

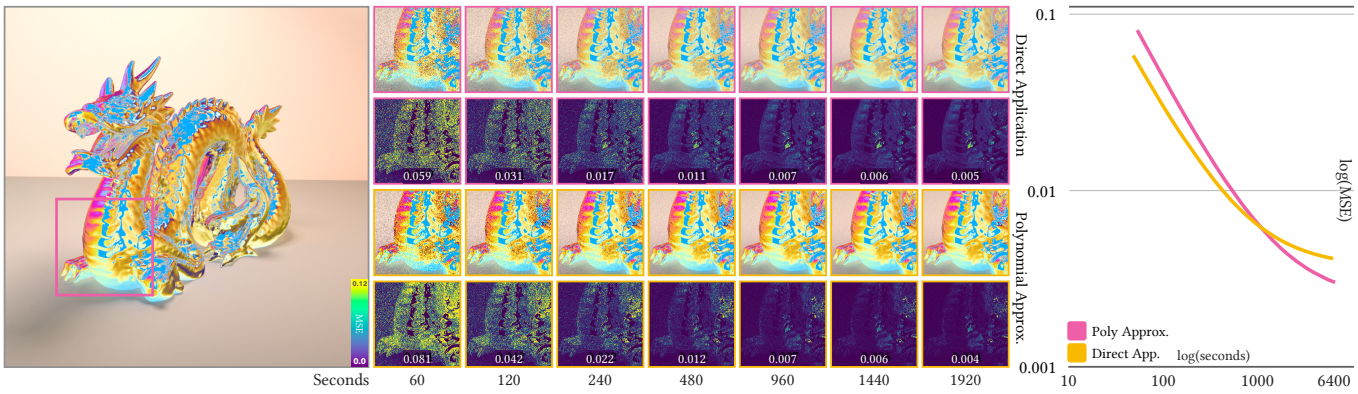

Fig. 11. For some style functions g_θ a polynomial approximation can provide an accurate representation. Here we use a polynomial approximation of degree 20 for a style function that produces a tie-dye effect by applying a cosine wave of different frequency and phase to each RGB channel. For the same sampling budget of 32 samples per evaluation, the polynomial approximation estimator (bottom rows) shows more accurate colors than a direct application estimator (top rows). Though the polynomial approximation starts out with higher error in the form of variance, it accumulates less bias than the direct application estimator, preserving the rich colors of the reference.

we stylize the lines using the stylization of the object the lines correspond to, causing the default black color to be mapped to a softer dark gray. In the custom inset we treat lines as their own object and can stylize them independently.

In Fig. 12 (right) we demonstrate rendering results for cross-hatching [Deussen et al. 1999] and half-tone [Hall 1999]. Both stylizations were implemented using a direct application estimator (19) and each have an interesting bias profile. In the hatching results the hard edges of the hatch lines are preserved in the direction normal to the slice planes, but produce a variable transparency like effect in directions tangent to the planes. In the half-tone result, we use a regular grid of spheres that change size based on exitant radiance similar in concept to Q-maps [Hall 1999] (i.e. parametrically varying 3D textures). As a result, each tone sphere will have softer or harder edges proportional to the variance of the inner-estimate of the direct application estimator (see (21)). In the middle inset of each method we demonstrate simple colorization using the global illumination information.

5.6 Advanced stylizations

In Fig. 1 we highlight the freedom of expression and level of visual coherency the proposed formulation can achieve in a single, coherent scene.

In inset (a), the robot has a cross-hatched shadow that helps blend its hand-drawn aesthetic with the physically shaded surrounding environment. To achieve this effect we stylized the sheet of paper using a variant of cross-hatching that, instead of using a "background color", lets exitant radiance pass-through, and controlled the region where this stylization is applied by only stylizing when visibility to a virtual point light was occluded. We use a similar stylization for the pen in inset (b), demonstrating a stylized shadow for a physically rendered object.

In inset (c), the tank combines a cel shaded look with subtle gradients in the highlights and around rounded edges. We achieved this effect by using a linear blend between the fast, closed-form cel shading of Barla et al. [2006] and the underlying reflectance from a Trowbridge-Reitz microfacet model. The closed-form cel shading ignores occlusion, causing the tank to produce light in otherwise dark regions. The proposed formulation is robust to even these non-physical effects, and the floor surrounding the tank brightens naturally, providing visual cohesion.

The stylization of the tank, however, goes beyond just cel shading. In inset (d) the tank is stylized using a viridis color map when viewed through the glossy mirror. Note, however, that this alternate stylization is only applied when viewed through exactly that mirror, and the reflection of the tank in the other objects (e.g. the side of the looking glass) is the original gold cel shading.

In inset (e), the contents of the story book come to life more physically than figuratively, where the otherwise monotone landscape is colorized when viewed through the looking glass. We achieved this effect by parameterizing a saturation style function with the path prefix and adjusting how saturated the exitant radiance is depending on the path taken.

In Fig. 3, though not the intent of the figure, we showcase depth-of-field from a thin lens camera on an alarm clock cross-hatched using the method of Deussen et al. [1999] and the feature line method of West [2021]. The cross-hatching captures the essence of the physical light transport and the depth-of-field gives the clock a clear presence in the 3D environment.

In Fig. 13 we showcase pattern matching. The pirate statue in the center is stylized with a style function that pattern matches the path prefix against a set of registered patterns, and if there is a match, stylizes using the style function associated with the matched pattern. We can stylize the pirate statue differently depending on if it is viewed directly, or through each of the 11 mirrors. Parametrization is one of the interesting aspects of the proposed formulation and we believe it provides a means to explore new visual expressions that were not easily captured by existing formulations.

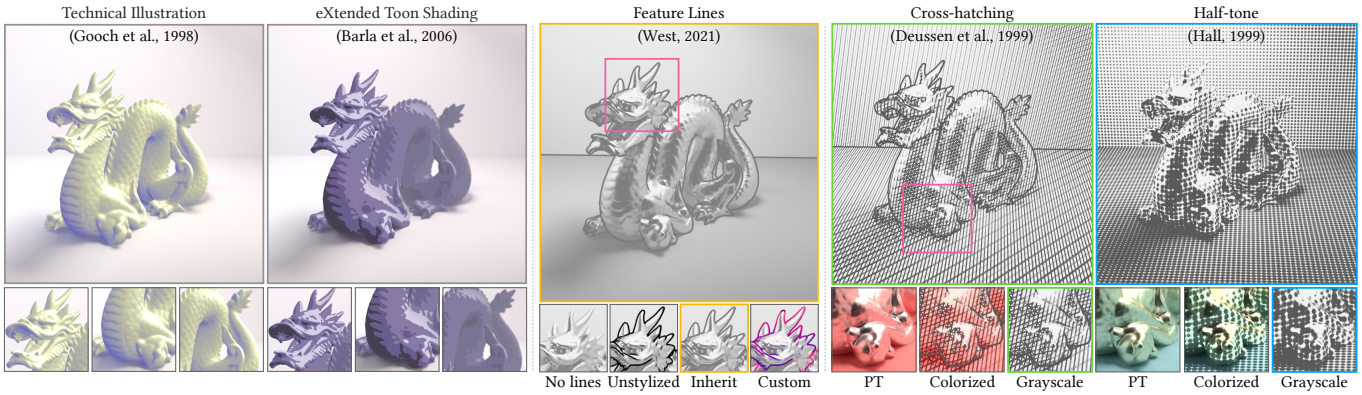

Fig. 12. Left: The technical illustration style of Gooch et al. [1998] and the toon shading style of Barla et al. [2006] can both be implemented under the proposed formulation. Here we recreate the stylization of the respective Fig. 7 of both works, using the parameterization θ to capture information about user-specified light direction. Middle: The feature line rendering method of West [2021] can be seamlessly combined with the proposed formulation. In their method lines are just another (light emitting) 3D primitive, allowing us the freedom to preserve the lines' unstylized color, inherit the stylization of the line emitting surface, or use a custom stylization altogether. Right: The parameterization θ of style functions g_θ can also be used to store global information, like stroke placement, needed to implement stylizations like cross-hatching [Deussen et al. 1999] and half-tone printing [Hall 1999].

6 LIMITATIONS AND FUTURE WORK

Functions of expectation. Unbiased estimation for functions of expectation is currently limited to continuous functions. For discontinuous functions of expectation, existing unbiased estimation strategies may not converge. Biased alternatives like polynomial approximation or direct application are similarly not ideal as they result in accumulated bias that affects the final image. To that end, exploring convergent unbiased estimators for discontinuous functions of expectation would be an interesting avenue of future work.

Stylized rendering. One limitation of the stylized rendering equation (6) is that it only locally modifies exitant radiance. As a result, some stylizations are not well-captured by the presented formulation, such as those based on convolutions and non-linear diffusion processes, as well as more abstract stylizations such as oil painting. Convolutions [Winnemöller et al. 2006, 2012] are particularly interesting as they have a clear mathematical formulation. A stylized rendering formulation that can additionally capture such stylizations would open up PBR to an even wider range of NPR styles.

Each pixel sample in the proposed method requires sampling a tree through the scene. This can be expensive, especially for heavily branching trees. The FTV method of Doi et al. [2021] successfully amortizes sampling cost using photon density estimation, but it lacks controllable bias and limits the possible stylizations. Methods like path filtering [Keller et al. 2014; Deng et al. 2021; West et al. 2022] also greatly amortize sampling costs by reusing ray samples, but have controllable bias. Deriving an efficient sample reuse algorithm for stylized rendering is one fruitful avenue of future work.

The path integral formulation [Veach 1997] has played a significant role in the advancement of PBR, allowing us to explore new, low variance estimation methods and combine existing methods in new ways. It stems from a reformulation of the rendering equation (1) that relies on the linearity of the terms. The stylized rendering equation (6) violates this assumption, making the path integral formulation inaccessible. It would be an interesting direction of future work to explore the concept of a stylized path integral formulation.

7 CONCLUSION

In this paper we propose the *stylized rendering equation*, a generalization of the rendering equation that captures both realistic light transport and a subset of stylized visuals under a single formulation. This new formulation brings many new challenges, so to make it immediately accessible we introduce an initial tool set of: local estimation strategies for style functions, a complete estimator that combines local estimators, and a simple, tree-based sampling algorithm. Further, we produce several compelling visuals, and believe that the proposed proposed formulation has strong potential to spark further research in combining NPR and PBR techniques.

ACKNOWLEDGMENTS

We express our gratitude to the following people for their constructive feedback and suggestions that greatly helped refine this manuscript: the anonymous reviewers, Prof. Yonghao Yue, Prof. Shinichiro Akiyama, Sabyasachi Mukherjee, and Jamorn Sriwasansak.

We also express our gratitude to the following Blend Swap users: *JulioIsmar* for the thermos in Fig. 1, *MahmoodDZ* for the shopping bags in Fig. 1, *ruwo* for the camboto in Fig. 1, *arktosk* for the sheet of paper in Fig. 1, *prodestroyer* for the pencil in Fig. 1, *dukedrop* for the stack of books in Fig. 1, *dergoldstein* for the chibi tank in Fig. 1, *Kilt2007* for the lamp in Fig. 1, *artturi* for the story book in Fig. 1, *luismanet* for the magnifying glass in Fig. 1, *kananav* for the coffee cup in Fig. 1, *Chemsy* for the coffee table in Fig. 1, *wig42* for the staircase scene in Fig. 1, *wundersound* for the battle sprinter mech in Fig. 2, *vajrablue* for the manned sci-fi rover in Fig. 2, *Jorgeferrer* for the astreo type 3 in Fig. 2, *ChrisKuhn* for the dragster engine in Fig. 2, *chams* for the alarm clock in Figs. 2 and 3, *Stanford Computer Graphics Laboratory* for the stanford dragon in Figs. 7 to 12, and *BenDansie* for the plunderbuss pete in Fig. 13.

This work was supported in part by a grant from JST FOREST Program, JPMJFR206R, Japan, and also by the Center of Innovations for Sustainable Quantum AI (JST Grant Number JPMJPF2221).

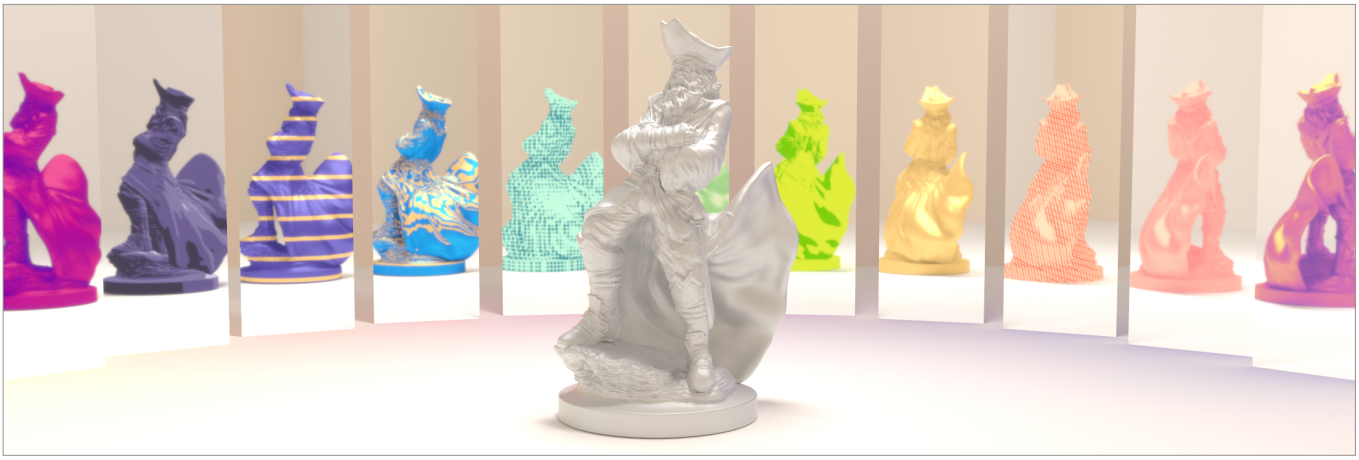

Fig. 13. What you see isn't always what you get (in the reflection). Here the pirate statue is stylized differently depending on which mirror it is viewed through. We parameterize the style function with the path prefix (list of vertices up to the one we are stylizing at) and choose a stylization based on which mirror was most recently visited. We can even control subtleties of the transport, like the color of light reflected by each mirror at different bounces to create a subtle color splash at the bottom of each.

REFERENCES

Ken-ichi Anjyo and Katsuaki Hiramitsu. 2003. Stylized Highlights for Cartoon Rendering and Animation. *IEEE Comput. Graph. Appl.* 23, 4 (jul 2003), 54–61. https://doi.org/10.1109/MCG.2003.1210865

Ken-ichi Anjyo, Shuhei Wemler, and William Baxter. 2006. Tweakable Light and Shade for Cartoon Animation. In *Proceedings of the 4th International Symposium on Non-Photorealistic Animation and Rendering* (Annecy, France) *(NPAR '06)*. Association for Computing Machinery, New York, NY, USA, 133–139. https://doi.org/10.1145/1124728.1124750

Sai Praveen Bangaru, Tzu-Mao Li, and Frédo Durand. 2020. Unbiased Warped-Area Sampling for Differentiable Rendering. *ACM Trans. Graph.* 39, 6, Article 245 (nov 2020), 18 pages. https://doi.org/10.1145/3414685.3417833

Pascal Barla, Joëlle Thollot, and Lee Markosian. 2006. X-Toon: An Extended Toon Shader. In *Proceedings of the 4th International Symposium on Non-Photorealistic Animation and Rendering* (Annecy, France) *(NPAR '06)*. Association for Computing Machinery, New York, NY, USA, 127–132. https://doi.org/10.1145/1124728.1124749

Aner Ben-Artzi, Ryan Overbeck, and Ravi Ramamoorthi. 2006. Real-Time BRDF Editing in Complex Lighting. *ACM Trans. Graph.* 25, 3 (jul 2006), 945–954. https://doi.org/10.1145/1141911.1141979

Jose H. Blanchet and Peter W. Glynn. 2015. Unbiased Monte Carlo for optimization and functions of expectations via multi-level randomization. In *2015 Winter Simulation Conference (WSC)*. 3656–3667. https://doi.org/10.1109/WSC.2015.7408524

Subrahmanyan Chandrasekhar. *Radiative Transfer*. NY.

Robert L. Cook, Thomas Porter, and Loren Carpenter. 1984. Distributed Ray Tracing. 18, 3 (July 1984), 137–145. https://doi.org/10/c9thc3

António Cardoso Costa, António Augusto Sousa, and Fernando Nunes Ferreira. 1999. Lighting Design: A Goal Based Approach Using Optimisation. In *Eurographics Workshop on Rendering*, Dani Lischinski and Greg Ward Larson (Eds.). The Eurographics Association. https://doi.org/10.2312/EGWR/EGWR99/317-328

Jérémi Dauchet, Jean-Jacques Bezian, Stéphane Blanco, Cyril Caliot, Julien Charon, Christophe Coustet, Mouna El Hafi, Vincent Eymet, Olivier Farges, Vincent Forest, Richard Fournier, Mathieu Galtier, Jacques Gautrais, Anaïs Khuong, Lionel Pelissier, Benjamin Piaud, Maxime Roger, Guillaume Terrée, and Sebastian Weitz. 2018. Addressing nonlinearities in Monte Carlo. *Scientific Reports* 8, 1 (05 Sep 2018), 13302. https://doi.org/10.1038/s41598-018-31574-4

Philippe Decaudin. 1996. *Cartoon-Looking Rendering of 3D-Scenes*. Technical Report 2919. INRIA Rocquencourt. http://www.antisphere.com/Research/Cartoon.php

Xi Deng, Miloš Hašan, Nathan Carr, Zexiang Xu, and Steve Marschner. 2021. Path Graphs: Iterative Path Space Filtering. *ACM Trans. Graph.* 40, 6, Article 276 (dec 2021), 15 pages. https://doi.org/10.1145/3478513.3480547

Oliver Deussen, Jörg Hamel, Andreas Raab, Stefan Schlechtweg, and Thomas Strothotte. 1999. An Illustration Technique Using Hardware-Based Intersections and Skeletons. In *Proceedings of the 1999 Conference on Graphics Interface '99* (Kingston, Ontario, Canada). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 175–182.

Oliver Deussen and Tobias Isenberg. 2013. *Halftoning and Stippling*. Springer London, London, 45–61. https://doi.org/10.1007/978-1-4471-4519-6_3

K. Doi, Yuki Morimoto, and R. Tsuruno. 2021. Global Illumination-Aware Stylised Shading. *Computer Graphics Forum* 40 (10 2021), 11–20. https://doi.org/10.1111/cgf.14397

Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. 2016. Image Style Transfer Using Convolutional Neural Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Iliyan Georgiev, Zackary Misso, Toshiya Hachisuka, Derek Nowrouzezahrai, Jaroslav Křivánek, and Wojciech Jarosz. 2019. Integral Formulations of Volumetric Transmittance. *ACM Trans. Graph.* 38, 6, Article 154 (nov 2019), 17 pages. https://doi.org/10.1145/3355089.3356559

Amy Gooch, Bruce Gooch, Peter Shirley, and Elaine Cohen. 1998. A Non-Photorealistic Lighting Model for Automatic Technical Illustration. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '98)*. Association for Computing Machinery, New York, NY, USA, 447–452. https://doi.org/10.1145/280814.280950

Amy A. Gooch. 2010. Towards Mapping the Field of Non-Photorealistic Rendering. In *Proceedings of the 8th International Symposium on Non-Photorealistic Animation and Rendering* (Annecy, France) *(NPAR '10)*. Association for Computing Machinery, New York, NY, USA, 159–164. https://doi.org/10.1145/1809939.1809958

Toshiya Hachisuka and Henrik Wann Jensen. 2009. Stochastic Progressive Photon Mapping. 28, 5 (Dec. 2009), 130:1–130:8. https://doi.org/10/d8xxn3

Toshiya Hachisuka, Shinji Ogaki, and Henrik Wann Jensen. 2008. Progressive Photon Mapping. 27, 5 (Dec. 2008), 130:1–130:8. https://doi.org/10/cn8h39

P. Hall. 1999. Nonphotorealistic Rendering by Q-mapping. *Computer Graphics Forum* (1999). https://doi.org/10.1111/1467-8659.00300

Eric Heitz, Jonathan Dupuy, Stephen Hill, and David Neubelt. 2016. Real-Time Polygonal-Light Shading with Linearly Transformed Cosines. 35, 4 (July 2016), 41:1–41:8. https://doi.org/10/f89ksz

Eric Heitz, Stephen Hill, and Morgan McGuire. 2018. Combining Analytic Direct Illumination and Stochastic Shadows. New York, NY, USA, 2:1–2:11. https://doi.org/10/gfznb7

Aaron Hertzmann and Denis Zorin. 2000. Illustrating Smooth Surfaces. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '00)*. ACM Press/Addison-Wesley Publishing Co., USA, 517–526. https://doi.org/10.1145/344779.345074

Tobias Isenberg, Petra Neumann, Sheelagh Carpendale, Mario Costa Sousa, and Joaquim A. Jorge. 2006. Non-Photorealistic Rendering in Context: An Observational Study. In *Proceedings of the Fourth International Symposium on Non-Photorealistic Animation and Rendering (NPAR 2006, June 5–7, 2006, Annecy, France)*, Doug DeCarlo and Lee Markosian (Eds.). ACM Press, New York, 115–126. https://doi.org/10.1145/1124728.1124747

James T. Kajiya. 1986. The Rendering Equation. 20, 4 (Aug. 1986), 143–150. https://doi.org/10/cvf53j

Alexander Keller, Ken Dahm, and Nikolaus Binder. 2014. Path Space Filtering *(SIGGRAPH '14)*. ACM, 68:1–68:1. https://doi.org/10/gfz6mr

Markus Kettunen, Eugene D'Eon, Jacopo Pantaleoni, and Jan Novák. 2021. An Unbiased Ray-Marching Transmittance Estimator. *ACM Trans. Graph.* 40, 4, Article 137 (jul 2021), 20 pages. https://doi.org/10.1145/3450626.3459937

Adam Lake, Carl Marshall, Mark Harris, and Marc Blackstein. 2000. Stylized Rendering Techniques for Scalable Real-Time 3D Animation. In *Proceedings of the 1st*

International Symposium on Non-Photorealistic Animation and Rendering (Annecy, France) *(NPAR '00)*. Association for Computing Machinery, New York, NY, USA, 13–20. https://doi.org/10.1145/340916.340918

Kai Lawonn, Ivan Viola, Bernhard Preim, and Tobias Isenberg. 2018. A Survey of Surface-Based Illustrative Rendering for Visualization. *Computer Graphics Forum* 37, 6 (2018), 205–234. https://doi.org/10.1111/cgf.13322 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.13322

A Lee, S Tiberi, and G Zanella. 2019. Unbiased approximations of products of expectations. *Biometrika* 106, 3 (04 2019), 708–715. https://doi.org/10.1093/biomet/asz008 arXiv:https://academic.oup.com/biomet/article-pdf/106/3/708/29133126/asz008.pdf

X. Liu, Y. Wu, and P. Hall. 2023. Painterly Style Transfer With Learned Brush Strokes. *IEEE Transactions on Visualization and Computer Graphics* 01 (nov 2023), 1–12. https://doi.org/10.1109/TVCG.2023.3332950

H. B. Mann and A. Wald. 1943. On Stochastic Limit and Order Relationships. *The Annals of Mathematical Statistics* 14, 3 (1943), 217 – 226. https://doi.org/10.1214/aoms/1177731415

Domingo Martin, Germn Arroyo, Alejandro Rodrguez, and Tobias Isenberg. 2017. A Survey of Digital Stippling. *Comput. Graph.* 67, C (oct 2017), 24–44. https://doi.org/10.1016/j.cag.2017.05.001

Don McLeish. 2011. A general method for debiasing a Monte Carlo estimator. *Monte Carlo Methods and Applications* 17, 4 (2011), 301–315. https://doi.org/doi:10.1515/mcma.2011.013

Jonatas Medeiros, Mario Sousa, Luiz Velho, and Carla Freitas. 2009. Perspective Contouring in Illustrative Visualization. In *Proceedings of the 2009 XXII Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI '09)*. IEEE Computer Society, USA, 48–55. https://doi.org/10.1109/SIBGRAPI.2009.49

Barbara J. Meier. 1996. Painterly Rendering for Animation. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '96)*. Association for Computing Machinery, 477—-484. https://doi.org/10.1145/237170.237288

Zackary Misso, Benedikt Bitterli, Iliyan Georgiev, and Wojciech Jarosz. 2022. Unbiased and consistent rendering using biased estimators. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 41, 4 (2022). https://doi.org/10.1145/3528223.3530160

Adithya Pediredla, Yasin Karimi Chalmiani, Matteo Giuseppe Scopelliti, Maysamreza Chamanzar, Srinivasa Narasimhan, and Ioannis Gkioulekas. 2020. Path tracing estimators for refractive radiative transfer. *ACM Trans. Graph.* 39, 6, Article 241 (nov 2020), 15 pages. https://doi.org/10.1145/3414685.3417793

Lohit Petikam, Ken Anjyo, and Taehyun Rhee. 2021. Shading Rig: Dynamic Art-Directable Stylised Shading for 3D Characters. *ACM Trans. Graph.* 40, 5, Article 189 (sep 2021), 14 pages. https://doi.org/10.1145/3461696

Matt Pharr, Brent Burley, Per Christensen, Marcos Fajardo, Luca Fascione, and Christopher Kulla. 2018. Design and Implementation of Modern Production Renderers. In *ACM SIGGRAPH 2018 Panels* (Vancouver, British Columbia, Canada) *(SIGGRAPH '18)*. Association for Computing Machinery, New York, NY, USA, Article 4, 2 pages. https://doi.org/10.1145/3209621.3214901

Matt Pharr, Wenzel Jakob, and Greg Humphreys. *Physically based rendering: From theory to implementation*. MIT Press.

Emil Praun, Hugues Hoppe, Matthew Webb, and Adam Finkelstein. 2001. Real-Time Hatching. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '01)*. Association for Computing Machinery, New York, NY, USA, 581. https://doi.org/10.1145/383259.383328

Hao Qin, Xin Sun, Qiming Hou, Baining Guo, and Kun Zhou. 2015. Unbiased Photon Gathering for Light Transport Simulation. *ACM Trans. Graph.* 34, 6, Article 208 (Oct. 2015), 14 pages. https://doi.org/10.1145/2816795.2818119

Alexander Rath, Pascal Grittmann, Sebastian Herholz, Philippe Weier, and Philipp Slusallek. 2022. EARS: efficiency-aware russian roulette and splitting. *ACM Trans. Graph.* 41, 4, Article 81 (jul 2022), 14 pages. https://doi.org/10.1145/3528223.3530168

Chang-Han Rhee and Peter Glynn. 2012. A new approach to unbiased estimation for SDE's. *Proceedings - Winter Simulation Conference* (07 2012). https://doi.org/10.1109/WSC.2012.6465150

Szymon Rusinkiewicz, Michael Burns, and Doug DeCarlo. 2006. Exaggerated Shading for Depicting Shape and Detail. *ACM Trans. Graph.* 25, 3 (jul 2006), 1199–1205. https://doi.org/10.1145/1141911.1142015

Thorsten-Walther Schmidt, Fabio Pellacini, Derek Nowrouzezahrai, Wojciech Jarosz, and Carsten Dachsbacher. 2016. State of the Art in Artistic Editing of Appearance, Lighting and Material. *Computer Graphics Forum* 35, 1 (2016), 216–233. https://doi.org/10.1111/cgf.12721 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.12721

Mario Costa Sousa, Amy Ashurst Gooch, and Bruce Gooch. 2005. Illustrative Scientific Visualization Framework. In *Computational Aesthetics in Graphics, Visualization and Imaging*, Laszlo Neumann, Mateu Sbert, Bruce Gooch, and Werner Purgathofer (Eds.). The Eurographics Association. https://doi.org/10.2312/COMPAESTH/COMPAESTH05/057-065

Martin Spindler, Niklas Röber, Robert Döhring, and Maic Masuch. 2006. Enhanced Cartoon and Comic Rendering. In *EG Short Papers*, Dieter Fellner and Charles Hansen (Eds.). The Eurographics Association. https://doi.org/10.2312/egs.20061047

Hideki Todo, Ken-ichi Anjyo, William Baxter, and Takeo Igarashi. 2007. Locally Controllable Stylized Shading. In *ACM SIGGRAPH 2007 Papers* (San Diego, California) *(SIGGRAPH '07)*. Association for Computing Machinery, New York, NY, USA, 17–es. https://doi.org/10.1145/1275808.1276399

Hideki Todo, Kunihiko Kobayashi, Jin Katsuragi, Haruna Shimotahira, Shizuo Kaji, and Yonghao Yue. 2022. Stroke Transfer: Example-Based Synthesis of Animatable Stroke Styles. In *ACM SIGGRAPH 2022 Conference Proceedings* (Vancouver, BC, Canada) *(SIGGRAPH '22)*. Association for Computing Machinery, New York, NY, USA, Article 54, 10 pages. https://doi.org/10.1145/3528233.3530703

Kenji Tojo and Nobuyuki Umetani. 2022. Recolorable Posterization of Volumetric Radiance Fields Using Visibility-Weighted Palette Extraction. *Computer Graphics Forum* 41, 4 (2022), 149–160. https://doi.org/10.1111/cgf.14594 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.14594

Eric Veach. 1997. *Robust Monte Carlo Methods for Light Transport Simulation*. Ph.D. Thesis. Stanford University, United States – California.

Can Wang, Ruixiang Jiang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. 2023. NeRF-Art: Text-Driven Neural Radiance Fields Stylization. *IEEE Transactions on Visualization and Computer Graphics* (2023), 1–15. https://doi.org/10.1109/TVCG.2023.3283400

Guanyang Wang and Tianze Wang. 2022. Unbiased Multilevel Monte Carlo methods for intractable distributions: MLMC meets MCMC.

Abdul-Majid Wazwaz. *Linear and nonlinear integral equations*. Vol. 639. Springer.

Rex West. 2021. Physically-Based Feature Line Rendering. *ACM Trans. Graph.* 40, 6, Article 246 (dec 2021), 11 pages. https://doi.org/10.1145/3478513.3480550

Rex West, Iliyan Georgiev, and Toshiya Hachisuka. 2022. Marginal Multiple Importance Sampling. In *SIGGRAPH Asia 2022 Conference Papers* (Daegu, Republic of Korea) *(SA '22)*. Association for Computing Machinery, New York, NY, USA, Article 42, 8 pages. https://doi.org/10.1145/3550469.3555388

Georges Winkenbach and David H. Salesin. 1994. Computer-Generated Pen-and-Ink Illustration. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '94)*. Association for Computing Machinery, New York, NY, USA, 91–100. https://doi.org/10.1145/192161.192184

Holger Winnemöller, Jan Kyprianidis, and Sven Olsen. 2012. XDoG: An eXtended difference-of-Gaussians compendium including advanced image stylization. *Computers & Graphics* 36 (10 2012), 740–753. https://doi.org/10.1016/j.cag.2012.03.004

Holger Winnemöller, Sven C Olsen, and Bruce Gooch. 2006. Real-Time Video Abstraction. 25, 3 (2006), 1221–1226. https://doi.org/10/fpfbck

Tizian Zeltner, Iliyan Georgiev, and Wenzel Jakob. 2020. Specular manifold sampling for rendering high-frequency caustics and glints. *ACM Trans. Graph.* 39, 4, Article 149 (aug 2020), 15 pages. https://doi.org/10.1145/3386569.3392408

Lvmin Zhang, Yi Ji, Xin Lin, and Chunping Liu. 2017. Style transfer for anime sketches with enhanced residual u-net and auxiliary classifier gan. In *2017 4th IAPR Asian conference on pattern recognition (ACPR)*. IEEE, 506–511.

LvMin Zhang, Chengze Li, Tien-Tsin Wong, Yi Ji, and ChunPing Liu. 2018. Two-stage Sketch Colorization. *ACM Transactions on Graphics* 37, 6 (Nov. 2018). https://doi.org/10.1145/3272127.3275090

A UNBIASED ESTIMATION OF RECURSIVE FUNCTIONS OF EXPECTATION

We present a proof by induction that shows that for multiple style functions $g_{\theta 1}, \ldots, g_{\theta r} : \mathbb{R}^k \to \mathbb{R}^k$ and a vector function f with range in $\mathbb{R}^k$, the complete estimator (23) provides an unbiased estimate for the pixel value,

$$I_\rho^{(r)} := g_{\theta r}\left(\int_{\mathcal{X}} g_{\theta(r-1)}\left(\int_{\mathcal{X}} \cdots g_{\theta 1}\left(\int_{\mathcal{X}} f(\overline{\mathbf{x}}_1)\mathrm{d}\overline{\mathbf{x}}_1\right)\cdots\right)\mathrm{d}\overline{\mathbf{x}}_r\right). \tag{25}$$

Since all of the estimators we consider in Section 4 use the idea of polynomial expansion or telescoping series, it suffices to prove the above for only style functions $g_{\theta i}$ that are all polynomials or continuous functions that are debiased using (14).

Therefore, suppose that for each $1 \le i \le r$, g_i is a style function with a sampling requirement of k_i, and has a corresponding gu-estimator given by $\hat{g}_{\theta i} : \mathbb{R}^{k\times k_i} \to \mathbb{R}^k$. In case of polynomial g_i, $k_i \ge \deg g_i$, and for continuous g_i, k_i depends on the aggressiveness of the stochastic termination of (14).

Then, Algorithm 1 recursively calculates the following estimators:

$$\begin{aligned} J_1 &= \int_{\mathcal{X}} f(\overline{\mathbf{x}}_1)\mathrm{d}\overline{\mathbf{x}}_1 \\ \langle I_\rho^{(1)}\rangle &= \hat{g}_{\theta 1}\left(\langle J_1\rangle_1, \ldots, \langle J_1\rangle_{k_1}\right) \\ &\vdots \\ J_i &= \int_{\mathcal{X}} I_\rho^{(i-1)}\mathrm{d}\overline{\mathbf{x}}_i \\ \langle I_\rho^{(i)}\rangle &= \hat{g}_{\theta i}\left(\langle J_i\rangle_1, \ldots, \langle J_i\rangle_{k_i}\right). \end{aligned}$$

Calculating $\langle I_\rho^{(i)}\rangle$ requires first estimating the integral $J_i = \int_{\mathcal{X}} I_\rho^{(i-1)}\mathrm{d}\overline{\mathbf{x}}_i$ k_i times. Therefore, to obtain a single estimate for the pixel value $I_\rho^{(r)}$ at depth r, we need $k_1 k_2 \cdots k_r$ different estimates of the integral J_1. Our goal in this section is to prove the following using induction on the recursion depth, r:

$$\mathrm{E}\left[\langle I_\rho^{(r)}\rangle\right] = g_{\theta r}\left(\int_{\mathcal{X}} g_{\theta(r-1)}\left(\int_{\mathcal{X}} \cdots g_{\theta 1}\left(\int_{\mathcal{X}} f(\overline{\mathbf{x}}_1)\mathrm{d}\overline{\mathbf{x}}_1\right)\cdots\right)\mathrm{d}\overline{\mathbf{x}}_r\right). \tag{26}$$

Clearly (26) holds for $r = 1$ by definition and the fact that $\hat{g}_{\theta 1}$ is a recipe for $g_{\theta 1}$, i.e. $\mathrm{E}\left[\hat{g}_{\theta 1}(\langle J_1\rangle_1, \ldots, \langle J_1\rangle_{k_1})\right] = g_{\theta 1}(J_1)$. Suppose (26) holds for $r = 1, \ldots, i-1$.

When $r = i$, we add another polynomial or continuous style function $g_{\theta i}$ and recipe $\hat{g}_{\theta i} : \mathbb{R}^{k\times k_i} \to \mathbb{R}^k$. First, the complete estimator (23) creates independent estimates $\langle J_i\rangle_1, \ldots, \langle J_i\rangle_{k_i}$ of J_i using $k_1 \cdots k_i$ mutually independent unbiased samples for J_1. Finally, as $\hat{g}_{\theta i}$ is a gu-estimator for $g_{\theta i}$ with sampling requirement of k_i, we have

$$\mathrm{E}\left[\langle I_\rho^{(i)}\rangle\right] = \mathrm{E}\left[\hat{g}_{\theta i}(\langle J_i\rangle_1, \ldots, \langle J_i\rangle_{k_i})\right] = g_{\theta i}(J_i) = I_\rho^{(i)},$$

as desired. Therefore, the final pixel value in (25) is unbiasedly estimated by the proposed recursive estimator $\langle I_\rho^{(r)}\rangle$.

B UNBIASED ESTIMATION OF LINEAR CROSS-INTEGRAL FUNCTIONS OF EXPECTATION

Let us now look into the situation where one of the function of expectations is *cross-integral* (i.e., for example, when it averages the three tristimulus channels to compute an estimate of the brightness, then uses this brightness estimate to compute a change to each channel). The main challenge in this scenario is that the inputs to the style function $g : \mathbb{R}^k \to \mathbb{R}^k$ may be correlated and may be estimates for different integrals. Such a situation can occur in specific stylizations such as cel-shading.

More rigorously, let us consider n integrals $I_i = \int_{\mathcal{X}} f_i(\overline{\mathbf{x}}_i)\mathrm{d}\overline{\mathbf{x}}_i$, $1 \le i \le n$. Suppose now that $g : \mathbb{R}^n \to \mathbb{R}^n$ is a style function, and we wish to estimate $g(I_1, \ldots, I_n)$. Consider a single component of g, say $g_1 : \mathbb{R}^n \to \mathbb{R}$. For now, assume that g_1 is a multivariate polynomial, and focus on a single term $t(x_1, \ldots, x_n) = \prod_{i=1}^n x_i^{k_i}$. Then following Section 2 of the supplemental material, we can create an unbiased estimator $\prod_{i=1}^n \prod_{j=1}^{k_i} \langle I_i\rangle_j$ for $t(I_1, \ldots, I_n)$ using k_i many different estimators for the integral I_i, as long as the sampling process satisfies $p(\overline{\mathbf{x}}_1, \ldots, \overline{\mathbf{x}}_n) > 0$ whenever $\prod_{i=1}^n f_i(\overline{\mathbf{x}}_i) \neq 0$. This implies that for style functions g whose individual components are representable as multivariate polynomials and for a covering sampling process, $g(I_1, \ldots, I_n)$ is unbiasedly estimable.

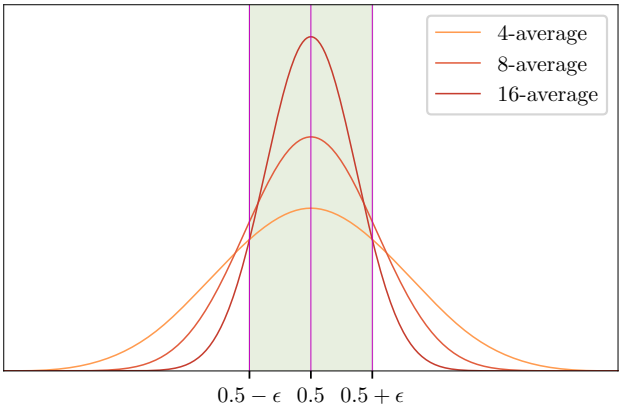

Fig. 14. Probability distribution of the average of n uniform $(0, 1)$-random variables for $n = 4, 8, 16$. As n increases and ε is fixed, more and more of the PDF is covered by $(0.5 - \varepsilon, 0.5 + \varepsilon)$.

Following our discussions in Section 4, we also reach a similar conclusion that any group-unbiased style function which is cross-component similarly admits a gu-estimator, as long as the above covering assumption holds.

C CONVERGENCE OF STEP STYLIZATION FUNCTIONS

Here we provide a mathematical explanation about why the step stylization function of Figure 5 exhibits super-linear convergence with an increase in the number of internal estimates.

Suppose we are estimating a function of expectation given by $g(I)$, where g is a step function and $I = \int_{\mathcal{X}} f(\overline{\mathbf{x}})\mathrm{d}\overline{\mathbf{x}}$. Then, the n^{th} Monte-Carlo estimator for I with uniform weights is given by $X_n = \frac{1}{n}(\langle I\rangle_1 + \cdots + \langle I\rangle_n)$, where $\langle I\rangle_i$ are independent estimates of I. Let $\mathrm{Var}(\langle I\rangle_i) = \sigma^2$ for every i. As g is a step function, there exists $\varepsilon > 0$ such that g is constant on the interval $(I - \varepsilon, I + \varepsilon)$. By the central limit theorem, X_n converges to a normal distribution centered at I and variance σ^2/n. So for large enough n, $|X_n - I|$ will almost surely be less than ε. More precisely, Chebyshev's inequality gives us

$$\Pr(|X_n - I| > \varepsilon) \le \frac{\mathrm{Var}(X_n)}{\varepsilon^2} = \frac{\sigma^2}{n\varepsilon^2}.$$

Hence, when $n \gg \frac{\sigma^2}{\varepsilon^2}$, this would imply that $|X_n - I| \le \varepsilon$ with probability $1 - o(1)$, thus leading to $g(X_n) = g(I)$. Figure 14 illustrates how, as the number of internal estimates n increases, the probability distribution of X_n has a higher peak at I, thus implying quickly decreasing tail probabilities of $|X_n - I| \ge \varepsilon$ for a fixed ε.

This convergence can also be super-linear in some special cases. For e.g., consider the integral estimation problem $I = \int_0^1 f(x)\mathrm{d}x = 0.5$, where $f(x)$ is given by the step function

$$f(x) = \begin{cases} 0, & 0 \le x \le 0.5, \\ 1, & 0.5 < x \le 1, \\ 0, & \text{otherwise.} \end{cases}$$

Then for X chosen uniformly on the interval $[0,1]$, $\langle I \rangle = f(X)$ behaves as a Bernoulli(0.5) random variable. Thus, $nX_n = \langle I \rangle_1 + \cdots + \langle I \rangle_n$ follows the Binomial distribution, which, by Hoeffding's inequality, satisfies

$$\begin{aligned}\Pr(|X_n - 1/2| > \varepsilon) = \Pr(|nX_n - n/2| > n\varepsilon) &\leq 2\exp(-2n^2\varepsilon^2/n) \\ &= 2\exp(-2\varepsilon^2 n).\end{aligned}$$

Hence, in this case, the n^{th} Monte-Carlo estimate converges exponentially to the mean, which explains the steep drop-off in the MSE of Figure 5.

D UNBIASEDNESS OF COMBINING GU-ESTIMATORS UNDER CERTAIN OPERATIONS

Here we show a proof of Theorem 4.1, which is our main tool for combining different group-unbiased style functions into new ones. Recall that we wish to show that for $\mathcal{G}$ as defined in (10), if $g_1, g_2 \in \mathcal{G}$, then $g_1 + g_2$, $g_1 \cdot g_2$ and $g_1 \circ g_2$ are all group-unbiased.

Proof of Theorem 4.1. Suppose g_1 and g_2 are style functions in $\mathcal{G}$ with gu-estimators $\hat{g}_{\theta 1}$ and $\hat{g}_{\theta 2}$, respectively. That is, for any I with unbiased estimators $\{\langle I \rangle_i : i \geq 1\}$, we have

$$\mathrm{E}\left[\hat{g}_{\theta j}(\langle I \rangle_1, \ldots, \langle I \rangle_{n_j})\right] = g_j(I) \text{ for } j = 1, 2.$$

- **Addition:** To provide an unbiased estimator of $(g_1 + g_2)(I)$, we can use $\max\{n_1, n_2\}$ estimates of I to form the estimator $\hat{g}_{\theta 1}(\langle I \rangle_1, \ldots, \langle I \rangle_{n_1}) + \hat{g}_{\theta 2}(\langle I \rangle_1, \ldots, \langle I \rangle_{n_2})$. Hence, $g_1 + g_2 \in \mathcal{G}$.
- **Multiplication:** Suppose $\langle I \rangle_1, \ldots, \langle I \rangle_{n_1+n_2}$ are independent estimates for I. Then, as $\mathrm{E}\left[\hat{g}_{\theta 1}(\langle I \rangle_1, \ldots, \langle I \rangle_{n_1}) \cdot \hat{g}_{\theta 2}(\langle I \rangle_{n_1+1}, \ldots, \langle I \rangle_{n_1+n_2})\right] = g_1(I) \cdot g_2(I)$, hence we have $g_1 \cdot g_2 \in \mathcal{G}$.
- **Composition:** To construct an unbiased estimate of $g_2(g_1(I))$, we can start with $n_1 n_2$ independent estimates $\langle I \rangle_1, \ldots, \langle I \rangle_{n_1 n_2}$ of I. For each $1 \leq i \leq n_2$, let $\hat{Z}_i = \hat{g}_{\theta 1}(\langle I \rangle_{(i-1)\cdot n_1+1}, \ldots, \langle I \rangle_{i \cdot n_1})$. Note that $\mathrm{E}\left[\hat{Z}_i\right] = g_1(I)$ for each i, and therefore the estimator $\hat{g}_{\theta 2}(\hat{Z}_1, \ldots, \hat{Z}_{n_2})$ is an unbiased estimator for $g_2(g_1(I))$.

This completes the proof of Theorem 4.1. □

An example. As an example, we demonstrate how to make a gu-estimator for a ratio of exponentials, a^I/b^I by combining exponential and reciprocal style functions using composition and multiplication. Let $g_{\theta 1}(x) = a^x$, $g_{\theta 2}(x) = 1/x$ and $g_{\theta 3}(x) = b^x$, where,

$$\frac{a^I}{b^I} = \left(g_{\theta 1} \cdot (g_{\theta 2} \circ g_{\theta 3})\right)(I). \tag{27}$$

We observe that the corresponding gu-estimator is of the same construction,

$$\left\langle \frac{a^I}{b^I} \right\rangle = \hat{g}_{\theta 1}(I) \cdot (\hat{g}_{\theta 2} \circ \hat{g}_{\theta 3})(I). \tag{28}$$

where for gu-estimators the composition operator $\circ$ is defined as $(\hat{g}_{\theta 2} \circ \hat{g}_{\theta 3})(I) = \hat{g}_{\theta 2}(\{\hat{g}_{\theta 3}(\{\langle I \rangle\}_{1,\ldots,n_3})\}_{1,\ldots,n_2})$, and n_2 and n_3 are the sampling requirements of $\hat{g}_{\theta 2}$ and $\hat{g}_{\theta 3}$.

In other words, if a style function g_θ can be constructed from a set of group-unbiased style function $g_{\theta i}$ operands using the operations $\oplus$ in Theorem 4.1, then a corresponding group-unbiased estimator $\hat{g}_\theta$ can be constructed by applying the same operations to the respective gu-estimators $\hat{g}_{\theta i}$ of the operand style functions.