# Supplemental: Stylized Rendering as a Function of Expectation

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

**ACM Reference Format:**
Rex West and Sayan Mukherjee. 2024. Supplemental: Stylized Rendering as a Function of Expectation. *ACM Trans. Graph.* 43, 4, Article 96 (July 2024), 6 pages. https://doi.org/10.1145/3658161

## S1 ORGANIZATION

The goal of this document is to provide further discussion on topics that supplement our primary exposition in the main paper. More precisely, we look into the following topics in more detail:

- An exposition on prior work on unbiased estimation of functions of expectations in Section S2.
- A primer on approximating functions using polynomial fits. While most prior work focuses on Taylor series expansion, we provide a concrete toolkit for calculating other polynomial fits for (potentially non-analytic) functions in Section S3.
- A proof of unbiasedness of a standard polynomial estimator where the individual samples do not necessarily need to be independent in Section S4.
- Experimental setup for renders shown in the results section of our main paper in Section S5.

## S2 PRIOR WORK ON UNBIASED ESTIMATION OF FUNCTIONS OF EXPECTATION

Several prior works in statistics and computer graphics have proposed unbiased estimators for several classes of functions of expectation, namely affine, polynomial, exponential, and reciprocal functions. Let us briefly review the concepts and estimators introduced by these prior works.

### S2.1 Affine

$$g_\theta(I) = a_1 I + a_0 \tag{S1}$$

For affine style functions $g_\theta$, constructing a group-unbiased estimator is straight-forward,

$$a_1 I + a_0 = a_1 \mathrm{E}[\langle I\rangle] + a_0 = \mathrm{E}[a_1\langle I\rangle + a_0] \ , \tag{S2}$$

where any unbiased estimator $\langle I\rangle$ will suffice,

$$\langle I\rangle = \frac{f(\mathbf{x})}{p(\mathbf{x})} \, , \tag{S3}$$

giving us the freedom to choose a single-sample or multi-sample MC estimator, or even more advanced estimators like multiple importance sampling (MIS) [Veach 1997]. Here $f$ is the integrand of the integral $I$. As was noted in Section 3, physically-based rendering can be represented by an affine identity style function $g_\theta$ where $a_0 = 0$ and $a_1 = 1$.

### S2.2 Polynomial

$$g_\theta(I) = \sum_{k=0}^{K} a_k I^k \tag{S4}$$

We can construct an unbiased estimator for a polynomial of expectation (S4) by making an unbiased estimate of each term $a_k I^k$ and taking their sum,

$$\sum_{k=0}^{K} a_k I^k = \sum_{k=0}^{K} a_k \mathrm{E}\left[\langle I^k\rangle\right] = \mathrm{E}\left[\sum_{k=0}^{K} a_k \langle I^k\rangle\right] \, , \tag{S5}$$

where each term $\langle I^k\rangle$ can be estimated in an unbiased manner by drawing $k$ samples $\mathbf{x}_i$ and taking their product,

$$\langle I^k\rangle_{\text{naive}} = \frac{f(\mathbf{x}_1)\cdots f(\mathbf{x}_k)}{p(\mathbf{x}_1,\ldots,\mathbf{x}_k)} \, , \tag{S6}$$

which we will refer to as the *naive* estimator. The samples $\mathbf{x}_i$ need not be independent so long as their joint probability density $p(\mathbf{x}_1,\ldots,\mathbf{x}_k)$ is *covering* (i.e. it has a non-zero density for all combinations of $\mathbf{x}_1,\ldots,\mathbf{x}_k$ with a non-zero product, see Section S4 for proof).

*Variance-minimal estimator.* Lee et al. [2019] and Kettunen et al. [2021] propose a variance-minimal estimator based on U-statistics [Lee 1990]. The key concept is to allocate the sampling budget of multiple high-variance naive estimates (S6) to a single low-variance estimate. Drawing a larger number $n$ of samples $\mathbf{x}_i$ than the minimal number $k$ required for a single estimate, we can construct a combinatorial $\binom{n}{k} = \frac{n!}{(n-k)!k!}$ number of symmetric estimates. Their resulting mean is both order-invariant and variance minimal [Halmos 1946]. Exploiting the symmetry of the estimates, Kettunen et al. [2021] propose a quadratic time recurrence relation based on Girard-Newton formulas, for which we present a simplified form[1],

$$s_{k,n} = s_{k,n-1} + s_{k-1,n-1} \cdot \frac{f(\mathbf{x}_k)}{p(\mathbf{x}_k)} \, , \tag{S7}$$

Authors' addresses: Rex West, Aoyama Gakuin University, Japan; Sayan Mukherjee, The University of Tokyo, Japan, Blueqat Research, Japan.

[1] The original form presented by Kettunen et al. [2021] is more numerically stable and we strongly recommend using their form in practice.

where $s_{k,0} = 0$ and $s_{0,n} = 1$. The resulting *variance minimal* estimator for a single term is then,

$$\langle I^k \rangle_{\min} = \binom{n}{k}^{-1} \sum_{\substack{J \subset \{1,\dots,n\} \\ |J| = k}} \prod_{j \in J} \frac{f(\mathbf{x}_j)}{p(\mathbf{x}_j)} = \binom{n}{k}^{-1} s_{k,n}\,. \tag{S8}$$

Kettunen et al. [2021] additionally propose a strategy of replacing the single-sample MC estimates $f(x)/p(x)$ with multi-sample MC estimates $\langle I \rangle_{\mathrm{m}} = \frac{1}{m}\sum_{j=1}^{m} \frac{f(x_j)}{p(x_j)}$, and use stratified sampling (in particular, a discrete comb) to further improve convergence rate.

*Multi-term polynomials.* When computing a polynomial of multiple terms we can reuse the same set of samples to estimate each term. For the variance-minimal estimator (S8), the cost of computing the lower order terms is fully amortized by computing the highest order term. In other words, the estimates of lower degree terms are computed as a by-product of estimating the highest degree term.

## S2.3 Power series

$$g_\theta(I) \leftrightarrow h_\theta(I) = \sum_{k=0}^{\infty} a_k (I - b)^k \tag{S9}$$

For an analytic style function $g_\theta$ we can construct a group-unbiased estimator from its Taylor series expansion $h_\theta$ around an expansion point $b$,

$$h_\theta(I) = \sum_{k=0}^{\infty} \underbrace{\frac{g_\theta^{(k)}(b)}{k!}}_{=a_k} (I - b)^k\,, \tag{S10}$$

where $g_\theta^{(k)}(b)$ is the $k^{\text{th}}$ derivative of $g_\theta$ evaluated at the expansion point $b$. One reasonable choice for the expansion point $b$ is an estimate $\langle I \rangle$ of the integral $I$.

To construct a practical estimator for a power series like in Eq. (S10) 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 three strategies to do so: single-term, iterative prefix-sum, and recursive prefix-sum.

*Single-term.* The single-term estimator selects a single, $k^{\text{th}}$ term of the series with some probability $P(k)$,

$$\langle h_\theta(I) \rangle_{\text{single}} = \frac{g_\theta^{(k)}(b)}{k!} \frac{\langle (I - b)^k \rangle}{P(k)}\,. \tag{S11}$$

*Iterative prefix-sum.* In contrast, the prefix-sum estimators select the first $n$ terms of the series. The iterative prefix-sum estimator 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 $n$,

$$\langle h_\theta(I) \rangle_{\text{iter}} = \sum_{k=0}^{n-1} \frac{g_\theta^{(k)}(b)}{k!} \frac{\langle (I - b)^k \rangle}{P(k < n)}\,. \tag{S12}$$

*Recursive prefix-sum.* The recursive prefix-sum estimator performs a Russian roulette-like process where each next term is accepted with some probability $P(k)$, else the series is terminated. The resulting probability of selecting the $k^{\text{th}}$ term is the product of accepting all lower index terms, $\prod_{k'=0}^{k} P(k')$, and the highest degree term $n$ is not chosen explicitly, but is a by-product of the process,

$$\langle h_\theta(I) \rangle_{\text{recur}} = \sum_{k=0}^{n-1} \frac{g_\theta^{(k)}(b)}{k!} \frac{\langle (I - b)^k \rangle}{\prod\limits_{k'=0}^{k} P(k')}\,. \tag{S13}$$

*Estimating the selected terms.* Each of the selected estimate terms $\langle (I - b)^k \rangle$ of the finite-term polynomial estimators in Eqs. (S11) to (S13) can then be robustly estimated using the naive (S6) or variance-minimal (S8) estimators discussed in Section S2.2.

## S2.4 Exponential

$$g_\theta(I) = a^I \tag{S14}$$

For exponentials of expectation we can apply the strategy of Section S2.3 and construct a group-unbiased estimator from its Taylor series expansion (S10),

$$\langle a^I \rangle = \sum_{k=0}^{\infty} \frac{\ln(a)^k a^b}{k!} \langle (I - b)^k \rangle\,, \tag{S15}$$

for which we can select a finite number of terms using some strategy (e.g. as in Eqs. (S11) to (S13)), and estimate the terms using a known polynomial term estimator (e.g. as in Eqs. (S6) and (S8)).

## S2.5 Reciprocal

$$g_\theta(I) = \frac{a}{I} \tag{S16}$$

Similarly to exponential style functions (S14), we can expand a reciprocal function into its Taylor series and construct a group-unbiased estimator for each of its terms,

$$\left\langle \frac{a}{I} \right\rangle = \sum_{k=0}^{\infty} a b^{-k-1} \langle (b - I)^k \rangle\,, \tag{S17}$$

which is valid when the integral $I \neq 0$, and for which we can, once again, select a finite number of terms using one of the strategies in Eqs. (S11) to (S13), and estimate them using Eqs. (S6) and (S8).

## S2.6 Other power series

$$g_\theta(I) \leftrightarrow h_\theta(I) = \sum_{k=0}^{\infty} a_k (I - b)^k \tag{S18}$$

One notices that the majority of the current work on power series estimators for functions of expectations focus on Taylor expansion. However, Taylor expansion is fundamentally a local procedure, and only accurately predicts function behavior around the point of expansion $b$. Hence, in cases where the style function $g_\theta$ under consideration has discontinuities, or otherwise fundamentally changes shape in different parts of the domain, Taylor expansion is unable to provide an accurate representation. Other popular power series methods such as Lagrange interpolation and Hermite interpolation suffer from similar limitations of being unstable, or intractable for functions that are not sufficiently smooth. This fundamental limitation of these power series leads us to investigate other ways of representing analytic functions using power series.

For brevity, we limit discussion to real-valued style functions $g_\theta : \mathbb{R} \to \mathbb{R}$, since extension to vector-valued $g_\theta$ under component-separability is straightforward, and denote the vector space of all analytic functions as $\mathcal{F}$, with usual notions of function addition and scalar multiplication.

*Polynomial bases.* Technically, the Taylor expansion of a style function $g_\theta$ around a point $x = b$ is an infinite expansion of $g_\theta$ in the function space $\mathcal{F}$ with basis $\{1, (x-b), (x-b)^2, \ldots\}$, with coefficients $a_k = g_\theta^{(k)}(b)/k!$. As an alternative to Taylor expansion of a style function $g_\theta \in \mathcal{F}$, we can explore different polynomial bases $\mathcal{T} = \{T_0(x), T_1(x), T_2(x), \ldots\}$. Obtaining a power series representation of $g_\theta$ from its basis representation, $g_\theta(x) = \sum_{k=0}^{\infty} a_k T_k(x)$, is a straightforward change-of-basis transformation from $\mathcal{T}$ to $\{1, (x-b), (x-b)^2, \ldots\}$, such that $g_\theta(x) = \sum_{k=0}^{\infty} a'_k (x-b)^k$.

*Orthogonality.* Further, one can equip the function space $\mathcal{F}$ with an inner-product $\odot$, and if the polynomials $T_n(x)$ are chosen so that they satisfy orthogonality,

$$T_m(x) \odot T_n(x) = C(m)\delta_{mn}\,, \tag{S19}$$

for some $C(m) \neq 0$, and $\delta_{mn}$ denotes the Dirac Delta, where $\delta_{mn} = 0$ if $m \neq n$ and 1 otherwise, then we can calculate the coefficients $a_n$,

$$a_n = T_n(x) \odot g_\theta(x)\,. \tag{S20}$$

Many such orthogonal bases $T$ are used in the literature, and below we discuss a few popular choices.

*Chebyshev Polynomials.* This basis is given by $T_0(x) = 1$, $T_1(x) = x$, and

$$T_{n+1}(x) = 2nT_n(x) - T_{n-1}(x)\,. \tag{S21}$$

It is known that under the inner product $f_1 \odot f_2 = \int_{-1}^{1} \frac{f_1(x) f_2(x)}{\sqrt{1-x^2}} \mathrm{d}x$, $T = \{T_0(x), T_1(x), T_2(x) \ldots\}$ forms an orthogonal basis. In particular, $\int_{-1}^{1} T_m(x)T_n(x)\mathrm{d}x = C(m)\delta_{mn}$, where $C(0) = \pi$ and $C(i) = \frac{\pi}{2}$ for every $i \geq 1$. This relation lets us compute the coefficients for the power series representation of any analytic $f : [-1, 1] \to \mathbb{R}$ by,

$$a_n = \int_{-1}^{1} \frac{g_\theta(x) T_n(x)}{C(n)\sqrt{1-x^2}} \mathrm{d}x. \tag{S22}$$

*Legendre Polynomials.* This basis is defined by $T_0(x) = 1$, $T_1(x) = x$, and the recursion formula

$$(n+1)T_{n+1}(x) = (2n+1)xT_n(x) - nT_{n-1}(x)\,. \tag{S23}$$

Then, it is known that $\int_{-1}^{1} T_m(x)T_n(x)\mathrm{d}x = \frac{1}{2m+1}\delta_{mn}$, so $T_m$ and $T_n$ are perpendicular with respect to the inner product $f_1 \odot f_2 = \int_{-1}^{1} f_1(x) f_2(x)\mathrm{d}x$. Therefore, in this case, we can compute, for a $g_\theta \in \mathcal{F}$ that has domain contained in $[-1, 1]$,

$$a_n = (2n+1)\int_{-1}^{1} g_\theta(x) T_n(x)\mathrm{d}x. \tag{S24}$$

*Bernstein Polynomials.* In general, Bernstein polynomials are not orthogonal, and show very slow convergence. While there has been effort on converting Bernstein polynomials into an orthogonal basis via the Gram-Schmidt procedure [Bellucci 2014], expressing a generic function in terms of Bernstein polynomials first requires us to fix a finite degree $n$, and we do not get an infinite series representation for analytic functions.

Note that due to the nature of the orthogonality condition, we need the analytic function $g_\theta$ to have a domain of $[-1, 1]$. This can be effectively achieved for any $g_\theta$ that is defined on a finite domain, by simple shifting and scaling of the function arguments.

*Estimation of coefficients.* For some analytic functions $f(x)$, the exact evaluation of the integrals in (S24) and (S22) can be challenging. Hence in practice, one estimates these coefficients. When the points that are taken on the real axis to sample these integrals are a fixed distance apart from each other, we obtain a Riemann sum. In general, if we take points $x_1, \ldots, x_n \in [-1, 1]$ to sample the integrals, then the estimation of the coefficients $a_n$ effectively calculates an approximate polynomial fit through the points $(x_1, g_\theta(x_1)), \ldots, (x_n, g_\theta(x_n))$. Therefore, this gives an effective way of calculating interpolating polynomials passing through a fixed number of points. We further discuss the idea of polynomial fitting in Section S3.

## S3 BACKGROUND ON POLYNOMIAL APPROXIMATION OF A FUNCTION

Approximating a style function $g_\theta$ by an $n$-degree polynomial $\hat{h}_\theta$ boils down to finding the best $a_0, a_1, \ldots, a_n$, such that, $g_\theta(x) \approx \hat{h}_\theta(x) = a_0 + a_1 x + \cdots + a_n x^n$, for every $x$ in the domain of $g_\theta$. This is equivalent to solving an optimization problem to minimize some measure $\mathcal{L}$, e.g. $\mathcal{L}^2(g_\theta, \hat{h}_\theta) = \int_D (g_\theta(x) - \hat{h}_\theta(x))^2 \mathrm{d}x$, the $L^2$- distance. Even if this optimization problem has a theoretical solution, solving it directly using gradient-based or simplex-based algorithms can be computationally expensive and has no guarantee on convergence.

Let us now explore some polynomial fitting techniques that are popular in scientific areas other than rendering.

*Taylor Fit.* A degree $k$ Taylor approximation around a single fit point $b = x_1$ of a $k$ times-differentiable $g_\theta$ is given by

$$\hat{h}_\theta(x) = \sum_{i=0}^{k} \frac{g_\theta^{(i)}(b)}{i!}(x-b)^i\,. \tag{S25}$$

*Bernstein Fit.* For a fixed degree $k$ and $g_\theta : [0, 1] \to \mathbb{R}$, the Bernstein fit requires $(k+1)$ equally spaced points on the interval $[0, 1]$ as input. So, in this case, we need $x_i = i/k$, and the Bernstein fit for such a $g_\theta$ is given by

$$\hat{h}_\theta(x) = \sum_{i=0}^{k} g_\theta\,(i/k) \cdot \binom{k}{i} x^i (1-x)^{k-i}\,. \tag{S26}$$

Although $\hat{h}_\theta$ converges to $g_\theta$ uniformly on $[0,1]$, this convergence is slow for functions with complicated shapes, making it difficult to use in practice.

*Chebyshev Fit.* As before, let $T_0(x) = 1$, $T_1(x) = x$ and $T_{n+1}(x) = 2nT_n(x) - T_{n-1}(x)$. For any number of fit points $(x_1, g_\theta(x_1)), \ldots, (x_n, g_\theta(x_n))$, we can calculate a Chebyshev fit by

$$\hat{h}_\theta(x) = \sum_{i=0}^{k} \hat{a}_i T_i(x)\,,\ \ \hat{a}_i = \sum_{j=1}^{n} \frac{g_\theta(x_j) T_i(x_j)}{C(i)\sqrt{1-x_j^2}} \cdot (x_j - x_{j-1})\,, \tag{S27}$$

where $C(0) = \pi$ and $C(i) = \frac{\pi}{2}$ for every $i \geq 1$, and we set $x_0 = -1$ by convention. In practice, one can sample the fit points $x_i$ as evenly spaced on the interval $[-1, 1]$, or use the so-called Chebyshev nodes $x_j = \cos\left(\frac{2j-1}{2n}\pi\right)$, but we have observed no perceivable difference in our experiments. Furthermore, Chebyshev fits have the useful property that for real-valued analytic functions, the error in the fit decays exponentially with $n$ [Tadmor 1986], hence making it a very good general-purpose fit for a wide variety of functions.

*Legendre Fit.* Recall that the Legendre polynomials satisfy $T_0(x) = 1$, $T_1(x) = x$, and the recursion formula $(n+1)T_{n+1}(x) = (2n+1)xT_n(x) - nT_{n-1}(x)$. We can exactly retrace the steps for computing a Chebyshev fit for these polynomials to compute a Legendre fit for a function $g_\theta$ with domain $[-1, 1]$ as,

$$\hat{h}_\theta(x) = \sum_{i=0}^{k} \hat{a}_i T_i(x)\,,\ \ \hat{a}_i = (2i+1) \sum_{j=1}^{i} g_\theta(x_j) T_i(x_j) \cdot (x_j - x_{j-1})\,. \tag{S28}$$

where we let $x_0 = -1$. In our experiments, the error bounds and the fit polynomials calculated by the Legendre fit matches very closely with Chebyshev, making them almost indistinguishable on the fit intervals.

## S4 UNBIASED ESTIMATION OF POLYNOMIALS OF EXPECTATION

Let us formalize the conditions required for an unbiased estimator of a polynomial of expectation. We will focus on the single-term, n-degree polynomial case, as an unbiased multi-term estimator can be easily constructed by taking the sum of unbiased single-term estimators.

In integral form we have,

$$g(I) = I^n = \left(\int_{\mathcal{X}} f(x)\mathrm{d}x\right)^n$$

Let us reorganize the terms such that, instead of computing the integral and raising it to the power $n$, we are taking the product of $n$ instances of the integral,

$$\begin{aligned} I^n &= \left(\int_{\mathcal{X}} f(x)\mathrm{d}x\right)^n \\ &= \left(\int_{\mathcal{X}} f(x_1)\mathrm{d}x_1\right) \cdots \left(\int_{\mathcal{X}} f(x_n)\mathrm{d}x_n\right) \\ &= \int_{\mathcal{X}^n} f(x_1) \cdots f(x_n)\mathrm{d}x_1 \cdots \mathrm{d}x_n \\ &= \int_{\mathcal{X}^n} \prod_{i=1}^{n} f(x_i)\mathrm{d}x_1 \cdots \mathrm{d}x_n \end{aligned}$$

Given this form, let us then define a more compact notation,

$$\int_{\mathcal{X}^n} \prod_{i=1}^{n} f(x_i)\mathrm{d}x_1 \cdots \mathrm{d}x_n = \int_{\mathcal{X}^n} \bar{f}(\overline{\mathbf{x}})\mathrm{d}\overline{\mathbf{x}} \tag{S29}$$

where $\overline{\mathbf{x}} = \{x_1, \cdots, x_n\}, \overline{\mathbf{x}} \in \mathcal{X}^n$, the integrand $\bar{f}(\overline{\mathbf{x}}) = \prod_{i=1}^{n} f(x_i)$, and the measure is $\mu(\overline{\mathbf{x}}) = \mu(x_1) \cdots \mu(x_n)$.

An single-sample Monte Carlo estimator for the right-hand side integral of Eq. (S29) is then,

$$\langle I^n \rangle = \frac{\bar{f}(\overline{\mathbf{x}})}{p(\overline{\mathbf{x}})}$$

where the expectation of the single-sample estimator,

$$\mathrm{E}\left[\frac{\bar{f}(\overline{\mathbf{x}})}{p(\overline{\mathbf{x}})}\right] = \int_{\mathcal{X}^n} \frac{\bar{f}(\overline{\mathbf{x}})}{p(\overline{\mathbf{x}})} p(\overline{\mathbf{x}})\mathrm{d}\overline{\mathbf{x}} = \int_{\mathcal{X}^n} \bar{f}(\overline{\mathbf{x}})\mathrm{d}\overline{\mathbf{x}} = I^n \tag{S30}$$

holds when the pdf $p(\overline{\mathbf{x}}) > 0$ where-ever $\bar{f}(\overline{\mathbf{x}}) \neq 0$, i.e. the sampling process of $p(\overline{\mathbf{x}})$ is *covering*.

As such, each of the samples $\mathbf{x}_1, \cdots, \mathbf{x}_n$ *need not be sampled independently*, so long as the joint pdf $p(\overline{\mathbf{x}})$ satisfies the above condition.

## S5 EXPERIMENTAL SETUP FOR RESULTS IN THE MAIN PAPER

All results were rendered on a CPU-based RGB renderer using the sampling algorithm detailed in main document, using a single workstation with a 24-core (8 performance, 16 efficiency) Intel i9-13900KF and 64GB of 4800 Mhz DDR5 memory.

### S5.1 Color mapping and ACP

For results in Fig. 7 we use the biased direct application estimator of (19) to match the ACP method of Doi et al. [2021]. The references were rendered with a brute-force 1024 samples per style function evaluation. The style function uses the "first hit only" parametrization, and is implemented as follows: First compute the inner-estimate $\langle I \rangle$ by averaging $n$ samples, then compute the brightness $u$ of the estimate $\langle I \rangle$, clamp $u$ to the minimum and maximum range of the gradient, and finally map the clamped brightness to a gradient value. We used a simple average of channels for the brightness, $u = (\langle I \rangle_r + \langle I \rangle_g + \langle I \rangle_b)/3$.

### S5.2 Cel shading and FTV

For results in Fig. 8 we use the biased direct application estimator (19). The references were rendered with a brute-force 4096 samples per style function evaluation. The style function uses the "first hit only" parametrization. The style function is a global illumination (GI) variant of cel shading that naturally preserves the color information of GI while still producing clear cel steps. The style is implemented as follows: First compute the inner-estimate $\langle I \rangle$ by averaging $n$ samples, then compute the brightness $u = (\langle I \rangle_r + \langle I \rangle_g + \langle I \rangle_b)/3$ of the estimate $\langle I \rangle$, determine the target brightness $u'$ using a user-defined mapping $m$, $u' = m(u)$, scale the estimate $\langle I \rangle$ to the target brightness, $\langle I \rangle' = \langle I \rangle u'/u$. The user-defined mapping $m$ used for Fig. 8 is $m(u) = 0.4$ for $u < 0.75$ and $m(u) = 0.95$ for $u \geq 0.75$.

## S5.3 Recursive application of style functions

For the results in Fig. 9 we use the biased direct application estimator (19). The images were rendered with 8 samples per style function evaluation. The style functions use a "first $N$ hits only" parametrization, which is a variation of the "first hit only" parametrization for recursive stylization. The style function for saturation adjustment is implemented as follows: First compute the inner-estimate $\langle I \rangle$ by averaging $n$ samples, compute the brightness $u = (\langle I \rangle_r + \langle I \rangle_g + \langle I \rangle_b)/3$ of the estimate $\langle I \rangle$, compute the minimum saturation estimate $\langle I \rangle_{\text{min}} = \langle u, u, u \rangle$, determine the minimum channel value $c_{min} = \min(\langle I \rangle_r, \langle I \rangle_g, \langle I \rangle_b)$ and the maximum channel value $c_{max} = \max(\langle I \rangle_r, \langle I \rangle_g, \langle I \rangle_b)$, compute the pre-normalized maximum saturation estimate $\langle I \rangle_{\text{max}} = (\langle I \rangle - c_{min})/c_{max}$, compute the brightness of the pre-max estimate $u' = (\langle I \rangle_{\text{max}r} + \langle I \rangle_{\text{max}g} + \langle I \rangle_{\text{max}b})/3$, compute the final color by using a user-defined weight $w$ to linearly interpolate the brightness normalized minimum and maximum saturation values, $\langle I \rangle' = w\langle I \rangle_{\text{min}} + (1 - w)\langle I \rangle_{\text{max}} u/u'$. We then linearly blend the exitant radiance with the maximally saturated color using a weight of $w = 0.5$.

## S5.4 Unbiased stylization using power series and certain operations

For the gamma correction results in Fig. 10 we use several variants of the power series estimator in Section S2.3. The images for these results were rendered with a variable number of samples due to stochastic prefix length. For the combined gu-estimator results in Fig. 10 we use the affine (S3) and naive polynomial term (S6) estimators. The images for the combined estimator results were rendered with 1 sample per operand style function evaluation. All style functions in Fig. 10 use the "first hit only" parametrization. The sepia tone is implemented as color mapping (see Section S5.1) with an extrapolated gradient that covers the range of $[0, \infty)$. The style function for contrast adjustment is implemented as follows: First compute the inner-estimate $\langle I \rangle$ by averaging $n$ samples, compute the final color using a user-specified contrast $c$ factor and a pivot value $p$, $\langle I \rangle' = c(\langle I \rangle - p) + p$.

## S5.5 Polynomial approximation

For the polynomial approximation results Fig. 11, the tie-dye effect is implemented as cosine waves of different frequency and phase operating on each of the RGB channels independently. 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 wave functions for each RGB channel are: $r' = \frac{-cos(2\pi(r+1))+1}{2}$, $g' = \frac{-cos(2.15\pi(g+1.13))+1}{2}$, and $b' = \frac{-cos(2.3\pi(b+1.29))+1}{2}$ respectively.

## S5.6 Representing existing stylizations

For results in Fig. 12 we use the biased direct application estimator (19). The images were rendered with a 64 samples per style function evaluation. The style function uses the "first hit only" parametrization.

*Experimental setup [Gooch et al. 1998; Barla et al. 2006].* Interestingly, both stylizations have completely closed-form solutions that we can compute with 0 samples, removing the need for the "first hit only" parametrization. We follow the respective formulations of each paper, with one exception: the "light direction" is not well-defined in a global illumination context, so we treat it as a user-specifiable parameter. For the results in Fig. 12 (left) we point the "light direction" towards the center of the area lights above the dragon.

*Experimental setup for feature lines [West 2021].* For the results in Fig. 12 (middle) we use the color mapping style function of Section S5.1. The images were rendered with a 8 samples per style function evaluation. The style function uses the "first hit only" parametrization. The *custom* feature line stylization maps the $y$ position of line vertices to a gradient, such that the lines vary in color vertically over $y$.

*Experimental setup for cross-hatching [Deussen et al. 1999].* The style function for cross-hatching [Deussen et al. 1999] is implemented as follows: define $n$ sets of evenly-spaced planes tangent to the camera's view direction, orient each set of planes at different angles around the view direction such that the cross product of the normals of any two plane sets is the view direction. Each set of planes represents one direction of hatching. At style evaluation, compute the inner-estimate $\langle I \rangle$ by averaging $n$ samples, followed by the brightness $u = (\langle I \rangle_r + \langle I \rangle_g + \langle I \rangle_b)/3$ of the estimate $\langle I \rangle$. Next, select $m \in \{0, \ldots, n\}$ sets of planes depending on the brightness, where $m = 0$ when $u$ is maximum, and $m = n$ when $u$ is minimum. For these $m$ selected sets of planes, and a user defined line thickness $d$, determine if the position $p$ of the current vertex lies within distance $d/2$ of any plane, $|(p - o) \cdot n| < d/2$, where $o$ is a point on a plane and $n$ is its unit normal. If yes, the style function returns a user-specified line color, and otherwise returns a user-specified background color.

*Experimental setup for half-tone [Hall 1999].* The style function for half-tone [Hall 1999] is implemented as follows: define a grid of evenly-spaced points over 3D space with an interval width of $d$. At style evaluation, compute the inner-estimate $\langle I \rangle$ by averaging $n$ samples, compute the brightness $u = (\langle I \rangle_r + \langle I \rangle_g + \langle I \rangle_b)/3$ of the estimate $\langle I \rangle$, then map $u$ to a radius $r$ for spheres centered at each grid point. If the position $p$ of the current vertex is less than a distance of $r$ from the nearest grid point $g$, i.e. $|p - g| < r$, then return a user-specified dark tone, and otherwise return a user-specified light tone.

## REFERENCES

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

Michael A. Bellucci. 2014. On the explicit representation of orthonormal Bernstein polynomials. arXiv:math.CA/1404.2293

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.

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

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

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

Paul R. Halmos. 1946. The Theory of Unbiased Estimation. *The Annals of Mathematical Statistics* 17, 1 (1946), 34–43. http://www.jstor.org/stable/2235902

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

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

A. J. Lee. *U-Statistics: Theory and Practice (1st ed.)*. Routledge. https://doi.org/10.1201/9780203734520

Eitan Tadmor. 1986. The Exponential Accuracy of Fourier and Chebyshev Differencing Methods. *SIAM J. Numer. Anal.* 23, 1 (1986), 1–10. https://doi.org/10.1137/0723001 arXiv:https://doi.org/10.1137/0723001

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

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