Supplementary Material: Imperfect Image-Space Control Variates for Monte Carlo Rendering

CHANU YANG, Gwangju Institute of Science and Technology, South Korea
BOCHANG MOON, Gwangju Institute of Science and Technology, South Korea

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

ACM Reference Format:
Chanu Yang and Bochang Moon. 2025. Supplementary Material: Imperfect Image-Space Control Variates for Monte Carlo Rendering. *ACM Trans. Graph.* 44, 6, Article 205 (December 2025), 3 pages. https://doi.org/10.1145/3763335

1 Derivation of the Variance of a Control Variate Estimator with Known Expectations

We derive the variance of a control variate estimator $\hat{F}$ (Eq. 2 in the main paper), assuming that the expectation G_i of the i-th control variate $\bar{g}_i$ is known. To simplify the derivation, we treat the control variate coefficients β_i as fixed. Thus, this derivation corresponds to the conditional variance of $\hat{F}$ given the coefficients. This leads to the following expression:

$$\begin{aligned}\sigma^2(\hat{F}) &= \sigma^2\left(\bar{f} - \sum_{i=1}^{k}\beta_i(\bar{g}_i - G_i)\right) \\ &= \sigma^2(\bar{f}) + \sigma^2\left(\sum_{i=1}^{k}\beta_i\bar{g}_i\right) - 2\sigma^2\left(\bar{f}, \sum_{i=1}^{k}\beta_i\bar{g}_i\right),\end{aligned} \tag{1}$$

where $\sigma^2(\bar{f}) = \sigma^2(f(x))/n$, and $\sigma^2(\cdot,\cdot)$ denotes the covariance between two random variables. The second term in Eq. 1 is simplified as follows:

$$\begin{aligned}\sigma^2\left(\sum_{i=1}^{k}\beta_i\bar{g}_i\right) &= \sum_{i=1}^{k}\sum_{i'=1}^{k}\beta_i\beta_{i'}\sigma^2\left(\bar{g}_i, \bar{g}_{i'}\right) \\ &= \sum_{i=1}^{k}\sum_{i'=1}^{k}\beta_i\beta_{i'}\sigma^2\left(\frac{1}{n}\sum_{s=1}^{n}g_i(x_s), \frac{1}{n}\sum_{s'=1}^{n}g_{i'}(x_{s'})\right) \\ &= \sum_{i=1}^{k}\sum_{i'=1}^{k}\frac{\beta_i\beta_{i'}}{n^2}\left(\sum_{s=1}^{n}\sum_{s'=1}^{n}\sigma^2\left(g_i(x_s), g_{i'}(x_{s'})\right)\right) \\ &= \sum_{i=1}^{k}\sum_{i'=1}^{k}\frac{\beta_i\beta_{i'}}{n^2}\left(\sum_{s=1}^{n}\sigma^2\left(g_i(x_s), g_{i'}(x_s)\right)\right) \\ &= \frac{1}{n}\sum_{i=1}^{k}\sum_{i'=1}^{k}\beta_i\beta_{i'}\sigma^2\left(g_i(x), g_{i'}(x)\right) \\ &= \frac{1}{n}\boldsymbol{\beta}^\top V_g\boldsymbol{\beta}.\end{aligned} \tag{2}$$

Also, the covariance term in Eq. 1 is expressed as:

$$\begin{aligned}\sigma^2\left(\bar{f}, \sum_{i=1}^{k}\beta_i\bar{g}_i\right) &= \sum_{i=1}^{k}\beta_i\sigma^2\left(\frac{1}{n}\sum_{s=1}^{n}f(x_s), \frac{1}{n}\sum_{s'=1}^{n}g_i(x_{s'})\right) \\ &= \sum_{i=1}^{k}\beta_i\frac{1}{n^2}\sum_{s=1}^{n}\sum_{s'=1}^{n}\sigma^2\left(f(x_s), g_i(x_{s'})\right) \\ &= \sum_{i=1}^{k}\beta_i\frac{1}{n^2}\sum_{s=1}^{n}\sigma^2\left(f(x_s), g_i(x_s)\right) \\ &= \frac{1}{n}\sum_{i=1}^{k}\beta_i\sigma^2\left(f(x), g_i(x)\right) \\ &= \frac{1}{n}\boldsymbol{\beta}^\top V_{fg}.\end{aligned} \tag{3}$$

By substituting Eqs. 2 and 3 into Eq. 1, we arrive at the final expression for the variance:

$$\sigma^2(\hat{F}) = \frac{1}{n}\left(\sigma^2\left(f(x)\right) + \boldsymbol{\beta}^\top V_g\boldsymbol{\beta} - 2\boldsymbol{\beta}^\top V_{fg}\right). \tag{4}$$

2 Derivation of the Variance of Our Control Variate Estimator

We relax the assumption that the expectations G_i of the control variates $\bar{g}_i$ are known, and instead replace them with their unbiased estimates $\bar{h}_i$, which are assumed to be independent across pixels. Under this change, the variance expression with known expectations (Eq. 1) transforms into:

$$\begin{aligned}\sigma^2(\hat{F}) &= \sigma^2\left(\bar{f} - \sum_{i=1}^{k}\beta_i(\bar{g}_i - \bar{h}_i)\right) \\ &= \sigma^2(\bar{f}) + \sigma^2\left(\sum_{i=1}^{k}\beta_i\bar{g}_i\right) - 2\sigma^2\left(\bar{f}, \sum_{i=1}^{k}\beta_i\bar{g}_i\right) + \sigma^2\left(\sum_{i=1}^{k}\beta_i\bar{h}_i\right).\end{aligned} \tag{5}$$

This modification introduces an additional error term, i.e., the last term in Eq. 5. Since $\bar{h}_i$ is independent across pixels, this additional term can be simplified as follows:

$$\sigma^2\left(\sum_{i=1}^{k}\beta_i\bar{h}_i\right) = \sum_{i=1}^{k}\beta_i^2\sigma^2(\bar{h}_i) = \frac{1}{n}\sum_{i=1}^{k}\beta_i^2\sigma^2(h_i(x)) = \frac{1}{n}\boldsymbol{\beta}^\top V_h\boldsymbol{\beta}. \tag{6}$$

Since the variance of our control variate estimator (Eq. 5) shares all terms with the known-expectation case (Eq. 1), except for the additional error term, we obtain the final expression for the variance by adding the error term from Eq. 6 to Eq. 4:

$$\sigma^2(\hat{F}) = \frac{1}{n}\left(\sigma^2\left(f(x)\right) + \boldsymbol{\beta}^\top V_g\boldsymbol{\beta} - 2\boldsymbol{\beta}^\top V_{fg} + \boldsymbol{\beta}^\top V_h\boldsymbol{\beta}\right). \tag{7}$$

Authors' Contact Information: Chanu Yang, yangchanu@gm.gist.ac.kr, Gwangju Institute of Science and Technology, South Korea; Bochang Moon, bmoon@gist.ac.kr, Gwangju Institute of Science and Technology, South Korea.

ACM 1557-7368/2025/12-ART205
https://doi.org/10.1145/3763335

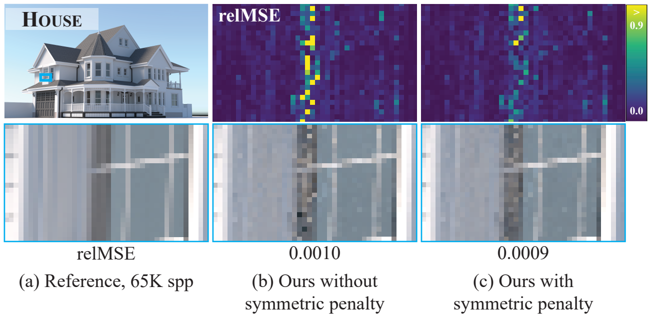

Fig. 1. Our results with and without the symmetric penalty. Adopting the penalty allows our method (c) to reduce high-frequency noise compared to the case without it (b), due to improved error estimation of independent pixel estimates. We use 192 spp for the test. 3D model courtesy of MrChimp2313.

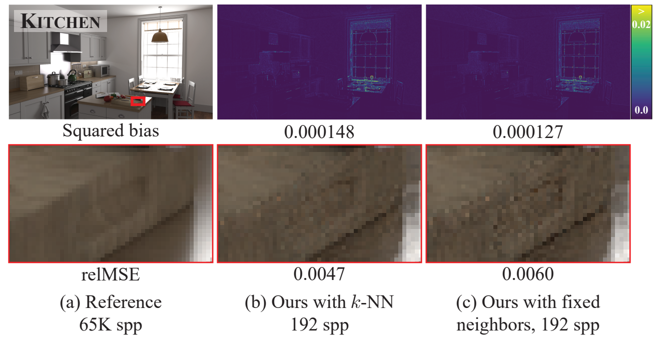

Fig. 2. Results of our technique when selecting k control variates using k-nearest neighbor (k-NN) selection (b) based on pixel estimates in the CRN image, and when using fixed neighboring pixels (i.e., all pixels within an image window) (c). In both cases, we set $k = 25$. While the k-NN selection introduces higher bias than the alternative, it suppresses noise more effectively, thereby reducing the relMSE. 3D model courtesy of Jay-Artist.

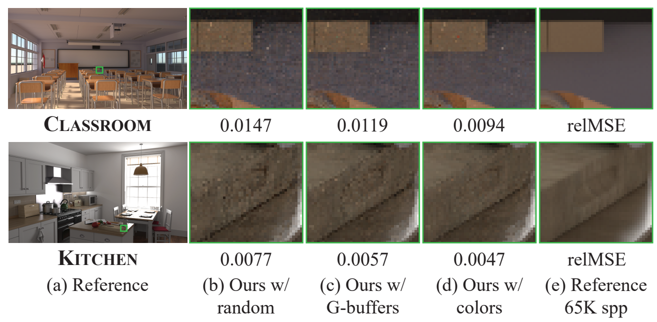

Fig. 3. Results of our method with different strategies for selecting k-nearest neighbors: (b) random selection, (c) G-buffer-based selection, and (d) our chosen approach using pixel colors in the CRN image. While G-buffer-based selection (c) reduces noise near geometric edges compared to random selection (b), it introduces more noise in flat regions than our chosen design (d). We use 192 spp for the evaluation. 3D model courtesy of NovaAshbell (Classroom) and Jay-Artist (Kitchen).

3 Relation to IDUW [Back et al. 2023]

Let us express the estimator of IDUW applied to each pixel c, which uses the same input (CRN and PT) as our method and adopts our notations, as follows:

$$\hat{F}_c = \beta_c \bar{h}_c + \sum_{i \in \Omega'_c} \beta_i \bar{h}_i + \sum_{i \in \Omega'_c} \beta_i (\bar{f}_c - \bar{g}_i), \tag{8}$$

where the neighboring pixel set Ω'_c includes all pixels in an image window, excluding pixel c itself. If we instead define a set Ω_c that includes the center pixel c, then Eq. 8 can be rewritten as:

$$\begin{aligned} \hat{F}_c &= \sum_{i \in \Omega_c} \beta_i \bar{h}_i + \sum_{i \in \Omega_c} \beta_i (\bar{f}_c - \bar{g}_i) \\ &= \bar{f}_c - \sum_{i \in \Omega_c} \beta_i (\bar{g}_i - \bar{h}_i). \end{aligned} \tag{9}$$

Note that $\sum_{i \in \Omega'_c} \beta_i (\bar{f}_c - \bar{g}_i) = \sum_{i \in \Omega_c} \beta_i (\bar{f}_c - \bar{g}_i)$ because of $\bar{f}_c = \bar{g}_c$ and $\sum_{i \in \Omega_c} \beta_i \bar{f}_c = \bar{f}_c$ due to the normalization constraint $\sum_{i \in \Omega_c} \beta_i = 1$ used in their method.

The transformed expression (Eq. 9) corresponds to the control variate estimator (Eq. 7) in our paper. A key difference, however, is that the coefficients β_i in IDUW are determined heuristically using the variances of $\bar{f}_c - \bar{g}_i$, rather than being selected optimally as in our method.

4 Additional Analyses

Analysis of the symmetric penalty (Sec. 4 in the main paper). To estimate the per-pixel errors of independent pixel estimates in the PT image, we use a cross-bilateral filter (see Eq. 11 in the main paper). Because this error estimation is imperfect due to the simplicity of the filter, we additionally employ a symmetric penalty. Since the representation of the optimal coefficients of control variates (Eq. 10 in the main paper) does not include this symmetric term, adopting it can be regarded as a heuristic.

To assess the practical benefit of this heuristic, we analyze the relMSE values of our chosen design, which uses the cross-bilateral filter, on ten test scenes (listed in Table 1 of the main paper), both with and without the symmetric penalty. We use 192 spp for the evaluation. In addition, we perform the same analysis with oracle errors (i.e., the squared differences between independent pixel estimates and ground-truth values), instead of the errors estimated by the filter.

When the errors are estimated perfectly (i.e., with oracle errors), adopting this heuristic yields errors that are on average only 5% higher than in the case without the heuristic. This result indicates that introducing the symmetric penalty term does not strongly violate the theory (Eq. 10 in the main paper). In contrast, when the errors are estimated using the cross-bilateral filter, the heuristic provides an average 15% improvement in relMSE. Fig. 1 shows a qualitative comparison of our method with and without the symmetric penalty, demonstrating that the heuristic helps reduce high-frequency noise caused by inaccurate error estimation.

Analysis of control variate selection. We select k neighboring pixels for each center pixel within a search window of size 11×11, based on the squared difference between the center and neighboring pixel estimates in the correlated input (CRN). We set $k = 25$ as the default value. Since the bias of our control variate technique arises from the dependency between the control variate coefficients and the input pixel estimates, selecting neighboring pixels using their values becomes a source of bias. An alternative is to select these control variates in a data-independent manner to avoid such bias. As an example, we can select all pixels within a 5×5 window, resulting in the same number of control variates (i.e., 25).

Fig. 2 compares the squared bias and relMSE for the KITCHEN scene between data-dependent selection using the k-nearest neighbors and the data-independent alternative. Our chosen design increases the squared bias by 16.5% compared to the alternative, due to the data dependency. However, it achieves a 27.7% reduction in relMSE by lowering noise through the selection of more correlated neighboring estimates. This result shows a trade-off between bias and overall error (relMSE). In the main paper, we adopt the k-nearest neighbor selection because the additional squared bias introduced by data dependency is generally smaller than the resulting noise reduction.

We also evaluate two alternative strategies for selecting the k-nearest neighbors, instead of using pixel estimates from the CRN image: (1) random selection, where k neighbors are chosen randomly, and (2) G-buffer-based selection, where neighbors are chosen according to a geometric difference between the center pixel and its neighbors. Specifically, we compute the difference as $1 - \exp(-\|\bar{a} - \bar{a}_i\|^2)\exp(-\|\bar{n} - \bar{n}_i\|^2)$, where $\bar{a} - \bar{a}_i$ and $\bar{n} - \bar{n}_i$ denote the differences in albedo and normal values between the center pixel and neighboring pixel i, respectively.

As shown in Fig. 3, G-buffer-based selection reduces noise compared to random selection. However, it introduces more noise in flat regions than our chosen design, which relies on the pixel estimates in the CRN image.

References

Jonghee Back, Binh-Son Hua, Toshiya Hachisuka, and Bochang Moon. 2023. Input-dependent uncorrelated weighting for Monte Carlo denoising. In *SIGGRAPH Asia 2023 Conference Papers* (Sydney, NSW, Australia) *(SA '23)*. ACM, New York, NY, USA, Article 9, 10 pages.