Eurographics Symposium on Rendering (2025)
B. Wang and A. Wilkie (Editors)

Supplemental Material to "Selective Caching in Procedural Texture Graphs for Path Tracing"

Vincent Schüßler[1], Johannes Hanika[1], Basile Sauvage[2], Jean-Michel Dischler[2], and Carsten Dachsbacher[1]

[1]Karlsruhe Institute of Technology, Germany [2]ICube, Université de Strasbourg, France

1. Introduction

The footprint of a path for a vertex describes the distribution of its position, considering all random decisions along the way when assuming adjoint analog sampling. In practice, we will approximate this using a Gaussian distribution and use the footprint in texture space as input to our system.

In this supplemental document, we describe our approximate model for footprints (section 2), which we base on the half vector space formulation of light transport [KHD14]. Using the slope space parameterization for half vectors allows to easily incorporate a distribution of normals (NDF) [HKD15] based on a surface roughness estimate, which makes for a convenient and general BSDF approximation. Based on this model, we derive the equations for the footprint at a given path vertex (section 3). The resulting recurrence can be computed incrementally in linear time, starting from a Gaussian distribution in screen space. We summarize our results in algorithm 1, which shows how to use our derivations to compute footprints for a path.

2. Model

We base our derivation on a local first order approximation of light transport around a base path. The footprint follows from statistical properties of nearby paths, when we interpret this model stochastically. Specifically, we analyze the distribution of paths that are sampled analogously to the approximation, starting from the camera. Such a random path depends only on the pairwise independently sampled screen space position and half vectors.

2.1. Preliminaries: half vector space.

Given a base path of length $k+1$, we can parameterize it both as sequence of vertices $x_0,\dots,x_k$ or as a sequence of half vectors $h_1,\dots,h_{k-1}$ and endpoints x_0,x_k [KHD14]. As we want to relate changes in half vector to changes in vertex position, we make use of the Jacobian matrix of the mapping from one parameterization to the other, which is known as the constraint derivative matrix [Jak13]. This has a block tridiagonal structure and can be defined by the submatrices

$$A_i = \frac{\partial h_i}{\partial x_{i-1}}, \quad B_i = \frac{\partial h_i}{\partial x_i}, \quad C_i = \frac{\partial h_i}{\partial x_{i+1}}. \tag{1}$$

Algorithm 1 Footprint computation

Inputs: $\operatorname{var}(\Delta\mathbf{x}_1), \operatorname{var}(\Delta\mathbf{h}_1), \dots, \operatorname{var}(\Delta\mathbf{h}_{k-1}), (A_i, B_i, C_i)_{1\le i\le k-1}$
Outputs: $\operatorname{var}(\Delta\mathbf{x}_2), \dots, \operatorname{var}(\Delta\mathbf{x}_k)$

$\operatorname{var}(\Delta\mathbf{x}_2) \leftarrow C_1^{-1}\left(\operatorname{var}(\Delta\mathbf{h}_1) + B_1 \operatorname{var}(\Delta\mathbf{x}_1) B_1^T\right) C_1^{-T}$
$\operatorname{cov}_{p,pp} \leftarrow -C_1^{-1} B_1 \operatorname{var}(\Delta\mathbf{x}_1)$ ▷ eq. (15)
$\operatorname{var}_{pp} \leftarrow \operatorname{var}(\Delta\mathbf{x}_1)$
$\operatorname{var}_p \leftarrow \operatorname{var}(\Delta\mathbf{x}_2)$
for $i \in \{2,\dots,k-1\}$ **do**
  $\text{cross\_cov} \leftarrow B_i \operatorname{cov}_{p,pp} A_i^T$
  $\operatorname{var}(\Delta\mathbf{g}_i) \leftarrow A_i \operatorname{var}_{pp} A_i^T + B_i \operatorname{var}_p B_i^T$
    $+ \text{cross\_cov} + \text{cross\_cov}^T$ ▷ eq. (14)
  $\operatorname{var}(\Delta\mathbf{x}_{i+1}) \leftarrow C_i^{-1}\left(\operatorname{var}(\Delta\mathbf{h}_i) + \operatorname{var}(\Delta\mathbf{g}_i)\right) C_i^{-T}$
  $\operatorname{cov}_{p,pp} \leftarrow -C_i^{-1}\left(A_i \operatorname{cov}_{p,pp}^T + B_i \operatorname{var}_p\right)$ ▷ eq. (16)
  $\operatorname{var}_{pp} \leftarrow \operatorname{var}_p$
  $\operatorname{var}_p \leftarrow \operatorname{var}(\Delta\mathbf{x}_{i+1})$
end for

We use the slope space parameterization for $\mathbf{h} \in \mathbb{R}^2$ and a local tangent frame for $\mathbf{x} \in \mathbb{R}^2$. As a first order approximation, we use the constraint derivative matrix to map from changes in vertex position $\Delta\mathbf{x}$ to changes in half vector $\Delta\mathbf{h}$:

$$\begin{pmatrix} B_1 & C_1 & & & & 0 \\ \dots & \dots & \dots & & & \\ & A_i & B_i & C_i & & \\ & & \dots & \dots & \dots & \\ 0 & & & A_{k-1} & B_{k-1} & C_{k-1} \end{pmatrix} \begin{pmatrix} \Delta\mathbf{x}_1 \\ \dots \\ \Delta\mathbf{x}_i \\ \dots \\ \Delta\mathbf{x}_k \end{pmatrix} = \begin{pmatrix} \Delta\mathbf{h}_1 \\ \dots \\ \Delta\mathbf{h}_i \\ \dots \\ \Delta\mathbf{h}_{k-1} \end{pmatrix} \tag{2}$$

Please note that this matrix has one extra block column compared to the more common $(k-2)\times(k-2)$ block structure, since we assume only the beginning of the path to be fixed [Jak13, section 4.3].

2.2. Stochastic model

Given the random variables $\Delta\mathbf{x}_1$ and $\Delta\mathbf{h}_1,\dots,\Delta\mathbf{h}_{k-1}$, we use eq. (2) to compute all other $\Delta\mathbf{x}_2,\dots,\Delta\mathbf{x}_k$, provided that all $C_1,\dots,C_{k-1}$ are invertible. We proceed row-wise. The first row

$$B_1\Delta\mathbf{x}_1 + C_1\Delta\mathbf{x}_2 = \Delta\mathbf{h}_1 \tag{3}$$

yields a solution for $\Delta\mathbf{x}_2$:

$$\Delta\mathbf{x}_2 = C_1^{-1}\left(\Delta\mathbf{h}_1 - B_1\Delta\mathbf{x}_1\right). \tag{4}$$

Similarly, we obtain a solution for $\Delta\mathbf{x}_3, \ldots, \Delta\mathbf{x}_{k-1}$ using

$$\Delta\mathbf{x}_{i+1} = C_i^{-1}\left(\Delta\mathbf{h}_i - A_i\Delta\mathbf{x}_{i-1} - B_i\Delta\mathbf{x}_i\right). \tag{5}$$

3. Derivation

Since we approximate the footprint as a Gaussian distribution, it is sufficient to derive a covariance matrix of each distribution of vertex position. In the following, we will use our model to derive these covariances based on known $\mathrm{var}(\Delta\mathbf{x}_1)$ and $\mathrm{var}(\Delta\mathbf{h}_1), \ldots, \mathrm{var}(\Delta\mathbf{h}_{k-1})$. We derive $\mathrm{var}(\Delta\mathbf{x}_1)$ by projecting the pixel filter, for which we use a standard deviation of half a pixel, to the first vertex using ray differentials. For $\mathrm{var}(\Delta\mathbf{h}_i)$, we simply convert the Beckmann-equivalent roughness α to a standard deviation in slope space $\sigma = \frac{\alpha}{\sqrt{2}}$.

3.1. Preliminaries: properties of cross-covariance

We make use of the following basic properties of cross-covariance:

$$\mathrm{var}(\mathbf{X}) = \mathrm{cov}(\mathbf{X}, \mathbf{X}), \tag{6}$$
$$\mathrm{var}(\mathbf{X}+\mathbf{Y}) = \mathrm{var}(\mathbf{X}) + \mathrm{var}(\mathbf{Y}) + \mathrm{cov}(\mathbf{X}, \mathbf{Y}) + \mathrm{cov}(\mathbf{Y}, \mathbf{X}), \tag{7}$$
$$\mathrm{cov}(\mathbf{X}, \mathbf{Y}) = \mathrm{cov}(\mathbf{Y}, \mathbf{X})^T, \tag{8}$$
$$\mathrm{cov}(\mathbf{X}+\mathbf{Y}, \mathbf{Z}) = \mathrm{cov}(\mathbf{X}, \mathbf{Z}) + \mathrm{cov}(\mathbf{Y}, \mathbf{Z}), \tag{9}$$
$$\mathrm{cov}(A\mathbf{X}, B\mathbf{Y}) = A\,\mathrm{cov}(\mathbf{X}, \mathbf{Y})B^T, \tag{10}$$

where $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ are random variables and A, B are matrices.

3.2. Propagating covariance

Using these properties, we derive covariances of any $\mathrm{var}(\Delta\mathbf{x}_i)$ with $2 \le i \le k$. Please note that the following equations hold even without assuming Gaussianity. We first proof the following lemma, to simplify our derivation later on.

Lemma. Any change in half vector $\Delta\mathbf{h}_j$ is uncorrelated with any change in vertex position that does not occur after it on the path:

$$\forall i \in \{1, \ldots, j\}\colon\ \mathrm{cov}(\Delta\mathbf{x}_i, \Delta\mathbf{h}_j) = 0. \tag{11}$$

Proof By induction. The base case $i = 1$ follows by definition, since $\Delta\mathbf{x}_1$ and $\Delta\mathbf{h}_1, \ldots, \Delta\mathbf{h}_{k-1}$ are pairwise independent:

$$\mathrm{cov}(\Delta\mathbf{x}_1, \Delta\mathbf{h}_j) = 0.$$

For the inductive step ($i \to i+1$), we plug in eqs. (4) and (5).
Case $i+1 = 2$.

$$\begin{aligned}\mathrm{cov}(\Delta\mathbf{x}_2, \Delta\mathbf{h}_j) &= \mathrm{cov}(C_1^{-1}\left(\Delta\mathbf{h}_1 - B_1\Delta\mathbf{x}_1\right), \Delta\mathbf{h}_j)\\ &= C_1^{-1}\,\mathrm{cov}(\Delta\mathbf{h}_1 - B_1\Delta\mathbf{x}_1, \Delta\mathbf{h}_j)\\ &= C_1^{-1}(\underbrace{\mathrm{cov}(\Delta\mathbf{h}_1, \Delta\mathbf{h}_j)}_{=0} - B_1\underbrace{\mathrm{cov}(\Delta\mathbf{x}_1, \Delta\mathbf{h}_j)}_{=0}) = 0\\ &\qquad \text{by definition} \qquad \text{induct. hypothesis}\end{aligned}$$

Case $i+1 > 2$.

$$\begin{aligned}&\mathrm{cov}(\Delta\mathbf{x}_{i+1}, \Delta\mathbf{h}_j) = \mathrm{cov}(C_i^{-1}\left(\Delta\mathbf{h}_i - A_i\Delta\mathbf{x}_{i-1} - B_i\Delta\mathbf{x}_i\right), \Delta\mathbf{h}_j)\\ &= C_i^{-1}\,\mathrm{cov}(\Delta\mathbf{h}_i - A_i\Delta\mathbf{x}_{i-1} - B_i\Delta\mathbf{x}_i, \Delta\mathbf{h}_j)\\ &= C_i^{-1}(\underbrace{\mathrm{cov}(\Delta\mathbf{h}_i, \Delta\mathbf{h}_j)}_{=0} - A_i\underbrace{\mathrm{cov}(\Delta\mathbf{x}_{i-1}, \Delta\mathbf{h}_j)}_{=0} - B_i\underbrace{\mathrm{cov}(\Delta\mathbf{x}_i, \Delta\mathbf{h}_j)}_{=0})\\ &\qquad \text{by definition} \qquad \text{induct. hypothesis} \qquad \text{induct. hypothesis}\\ &= 0\end{aligned}$$

□

Footprint for $i = 2$.

$$\begin{aligned}\mathrm{var}(\Delta\mathbf{x}_2) &= \mathrm{var}\left(C_1^{-1}\left(\Delta\mathbf{h}_1 - B_1\Delta\mathbf{x}_1\right)\right) && \text{eq. (4)}\\ &= C_1^{-1}\,\mathrm{var}\left(\Delta\mathbf{h}_1 - B_1\Delta\mathbf{x}_1\right)C_1^{-T}\\ &= C_1^{-1}\left(\mathrm{var}(\Delta\mathbf{h}_1) + \mathrm{var}(-B_1\Delta\mathbf{x}_1)\right)C_1^{-T} && \text{eq. (11)}\\ &= C_1^{-1}\left(\mathrm{var}(\Delta\mathbf{h}_1) + B_1\,\mathrm{var}(\Delta\mathbf{x}_1)B_1^T\right)C_1^{-T}\end{aligned} \tag{12}$$

Footprint for $2 < i \le k$. Since $\mathrm{cov}(\Delta\mathbf{x}_i, \Delta\mathbf{x}_{i-1}) \neq 0$, the case for any following vertex x_{i+1} is slightly more involved:

$$\begin{aligned}\mathrm{var}(\Delta\mathbf{x}_{i+1}) &= \mathrm{var}\left(C_i^{-1}\left(\Delta\mathbf{h}_i - A_i\Delta\mathbf{x}_{i-1} - B_i\Delta\mathbf{x}_i\right)\right) && \text{eq. (5)}\\ &= C_i^{-1}\left(\mathrm{var}(\Delta\mathbf{h}_i) + \mathrm{var}(-\Delta\mathbf{g}_i)\right)C_i^{-T} && \text{eq. (11)}\\ &= C_i^{-1}\left(\mathrm{var}(\Delta\mathbf{h}_i) + \mathrm{var}(\Delta\mathbf{g}_i)\right)C_i^{-T},\end{aligned} \tag{13}$$

where we defined $\Delta\mathbf{g}_i := A_i\Delta\mathbf{x}_{i-1} + B_i\Delta\mathbf{x}_i$.

To compute $\mathrm{var}(\Delta\mathbf{g}_i)$, we expand terms:

$$\begin{aligned}\mathrm{var}(\Delta\mathbf{g}_i) &= \mathrm{var}\left(A_i\Delta\mathbf{x}_{i-1} + B_i\Delta\mathbf{x}_i\right)\\ &= A_i\,\mathrm{var}(\Delta\mathbf{x}_{i-1})A_i^T + B_i\,\mathrm{var}(\Delta\mathbf{x}_i)B_i^T\\ &\quad + A_i\,\mathrm{cov}(\Delta\mathbf{x}_i, \Delta\mathbf{x}_{i-1})^T B_i^T + B_i\,\mathrm{cov}(\Delta\mathbf{x}_i, \Delta\mathbf{x}_{i-1})A_i^T\end{aligned} \tag{14}$$

of which the remaining unknown term is the cross-covariance $\mathrm{cov}(\Delta\mathbf{x}_i, \Delta\mathbf{x}_{i-1})$. We again treat the cases $i = 2$ and $i > 2$ separately and expand using eqs. (4) and (5).

$$\begin{aligned}\mathrm{cov}(\Delta\mathbf{x}_2, \Delta\mathbf{x}_1) &= \mathrm{cov}\left(C_1^{-1}\left(\Delta\mathbf{h}_1 - B_1\Delta\mathbf{x}_1\right), \Delta\mathbf{x}_1\right)\\ &= C_1^{-1}\,\mathrm{cov}\left(\Delta\mathbf{h}_1 - B_1\Delta\mathbf{x}_1, \Delta\mathbf{x}_1\right)\\ &= C_1^{-1}\left(\mathrm{cov}\left(\Delta\mathbf{h}_1, \Delta\mathbf{x}_1\right) + \mathrm{cov}\left(-B_1\Delta\mathbf{x}_1, \Delta\mathbf{x}_1\right)\right)\\ &= C_1^{-1}\,\mathrm{cov}\left(-B_1\Delta\mathbf{x}_1, \Delta\mathbf{x}_1\right) = -C_1^{-1}B_1\,\mathrm{var}\left(\Delta\mathbf{x}_1\right).\end{aligned} \tag{15}$$

$$\begin{aligned}&\forall i > 2\colon\ \mathrm{cov}(\Delta\mathbf{x}_i, \Delta\mathbf{x}_{i-1})\\ &= \mathrm{cov}\left(C_{i-1}^{-1}\left(\Delta\mathbf{h}_{i-1} - A_{i-1}\Delta\mathbf{x}_{i-2} - B_{i-1}\Delta\mathbf{x}_{i-1}\right), \Delta\mathbf{x}_{i-1}\right)\\ &= C_{i-1}^{-1}\left(\mathrm{cov}\left(-A_{i-1}\Delta\mathbf{x}_{i-2}, \Delta\mathbf{x}_{i-1}\right) + \mathrm{cov}\left(-B_{i-1}\Delta\mathbf{x}_{i-1}, \Delta\mathbf{x}_{i-1}\right)\right)\\ &= C_{i-1}^{-1}\left(-A_{i-1}\,\mathrm{cov}\left(\Delta\mathbf{x}_{i-2}, \Delta\mathbf{x}_{i-1}\right) - B_{i-1}\,\mathrm{var}\left(\Delta\mathbf{x}_{i-1}\right)\right)\\ &= -C_{i-1}^{-1}\left(A_{i-1}\,\mathrm{cov}\left(\Delta\mathbf{x}_{i-1}, \Delta\mathbf{x}_{i-2}\right)^T + B_{i-1}\,\mathrm{var}\left(\Delta\mathbf{x}_{i-1}\right)\right).\end{aligned} \tag{16}$$

We apply this equation recursively to arrive at a solution. We show how to iteratively compute footprints for a path in algorithm 1.

References

[HKD15] HANIKA, JOHANNES, KAPLANYAN, ANTON, and DACHSBACHER, CARSTEN. "Improved Half Vector Space Light Transport". *Computer Graphics Forum (Proceedings of the Eurographics Symposium on Rendering)* 34.4 (July 1, 2015), 65–74. DOI: 10/gfzv83 1.

[Jak13] JAKOB, WENZEL ALBAN. "Light Transport on Path-Space Manifolds". PhD thesis. Cornell University, 2013 1.

[KHD14] KAPLANYAN, ANTON S., HANIKA, JOHANNES, and DACHSBACHER, CARSTEN. "The Natural-Constraint Representation of the Path Space for Efficient Light Transport Simulation". *ACM Transactions on Graphics (Proceedings of SIGGRAPH)* 33.4 (July 2014), 102:1–102:13. DOI: 10/f6cz85 1.