Antialiasing Complex Global Illumination Effects in Path-space

Laurent Belcour, Ling-Qi Yan, Ravi Ramamoorthi, Derek Nowrouzezahrai

▶ **To cite this version:**

Laurent Belcour, Ling-Qi Yan, Ravi Ramamoorthi, Derek Nowrouzezahrai. Antialiasing Complex Global Illumination Effects in Path-space. [Research Report] 1375, Université de Montréal. 2015. hal-01200710v1

HAL Id: hal-01200710

https://hal.science/hal-01200710v1

Submitted on 17 Sep 2015 (v1), last revised 26 Apr 2016 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Antialiasing Complex Global Illumination Effects in Path-space

Laurent Belcour[1], Ling-Qi Yan[2], Ravi Ramamoorthi[3], and Derek Nowrouzezahrai[1]
[1]Université de Montréal, [2]UC Berkeley, [3]UC San Diego

We present the first method to efficiently and accurately predict antialiasing footprints to pre-filter color-, normal-, and displacement-mapped appearance in the context of multi-bounce global illumination. We derive Fourier spectra for radiance and importance functions that allow us to compute spatial-angular filtering footprints at path vertices, for both uni- and bi-directional path construction. We then use these footprints to antialias reflectance modulated by high-resolution color, normal, and displacement maps encountered along a path. In doing so, we also unify the traditional path-space formulation of light-transport with our frequency-space interpretation of global illumination pre-filtering. Our method is fully compatible with all existing single bounce pre-filtering appearance models, not restricted by path length, and easy to implement atop existing path-space renderers. We illustrate its effectiveness on several radiometrically complex scenarios where previous approaches either completely fail or require orders of magnitude more time to arrive at similarly high-quality results.

1. INTRODUCTION

Texturing with color, normal and displacement maps is a common approach to modelling fine details, increasing a scene's apparent complexity. Whenever such textures are used, however, accurate and efficient antialiasing approaches are necessary to avoid objectionable aliasing artifacts. Local prefiltering of color textures [Heckbert 1986] and, most recently, normal and displacement maps [Han et al. 2007; Dupuy et al. 2013; Yan et al. 2014], is a well understood problem; however, all existing prefilters works only treat *directly visible* surface appearance, projecting the *pixel footprint* onto the geometry. Very little work has investigated the implications of high-resolution appearance aliasing in the presence of complex light transport with global illumination (GI).

Prefiltering appearance in the presence of GI poses several challenges: since final pixel color results from a multi-dimensional integration of light paths incident on the pixel's footprint (and reflected towards a viewer), we need to express and propagate the *pixel footprint* across multiple bounces (e.g., at each light path vertex; see Figure 5). Intuitively, filtering should be applied at each vertex along a light path, where the final appearance is correctly antialiased by band-limiting its content by the frequency content of the sensor/pixel, emitters/lights, and reflectance along the path.

In previous work, ray differentials [Igehy 1999] (Figure 1, left) were devised to predict the pixel footprint from specular interactions, whereas path differentials [Suykens and Willems 2001] are the only technique available to estimate indirect pixel footprints from non-specular interactions (Figure 1, center). Unfortunately, both these approaches only treat eye-paths, leading to incorrect filtering footprints in the common scenario when the light's frequency content is non-negligible. We show how to theoretically account for both the pixel and light's frequency content at a vertex, resulting in the correct antialiasing footprints.

We extend covariance tracing [Belcour et al. 2013] to compute prefiltering bandlimits at path vertices (Figure 1, right), taking into account the sensor *and* emitter spectra, formulating approximate surface footprints from this bandlimit[1]. We also merge two independent unidirectional frequency analyses at path vertex connections, where pixel and light footprints are propagated independently across multiple scene interactions, in order to devise the first accurate bidirectional antialiasing approach. We apply our method to complex GI effects from surfaces with high-resolution normal, color, and displacement maps (Figures 6 to 11).

Our implementation is straightforward to integrate into modern renderers and we compare our filtered transport algorithms to path-sampling approaches with ray differentials [Igehy 1999] (when available), additionally employing different local appearance prefiltering methods (i.e., Heckbert's diffuse color filtering [1986] and Yan et al.'s specular normal map filtering [2014]). Both traditional uni- and bidirectional algorithms require orders of magnitude longer to converge to the same quality as our approach.

In summary, we present the following technical contributions:

- a frequency domain formulation of *GI prefiltering* (Section 4), including a unidirectional prefiltering analysis,
- an analysis relating traditional path-space transport formulations to their Fourier-domain counterparts, correctly treating spatial-angular sensor *and* emitter spectra in order to devise *bidirectional* vertex antialiasing footprints (Section 5), and
- two efficient techniques for predicting antialiasing footprints and for prefiltering high resolution appearance in GI effects, supporting uni- and bidirectional path tracing (Section 6).

2. PREVIOUS WORK

Prefiltering Local Appearance. Texture prefiltering is a foundational topic in computer graphics, and we refer readers to comprehensive surveys on color texture filtering [Heckbert 1986] and more general local appearance filtering [Bruneton and Neyret 2012]. Several works address *normal map prefiltering* [Fournier 1992; Toksvig 2005] by representing local normal distribution functions (NDFs) with bases amenable to fast filtering, often on the GPU [Han et al. 2007; Olano and Baker 2010; Dupuy et al. 2013]. While these interactive techniques are suitable for e.g., video games, none of them correctly treat GI.

Several approaches address more complex micro-structure prefiltering, such as on surfaces with scratches and grooves [Bosch et al. 2004; 2008], and two recent techniques accelerate sampling-based filtering solutions for surfaces with high-resolution NDFs using hierarchical spatial-angular pruning [Yan et al. 2014] and discrete microfacet importance sampling [Jakob et al. 2014]. All of these techniques only antialias the directly-visible appearance of a surface, resorting to an oversmoothed approximate BRDF model

[1]One would expect that filtering the appearance according to the sampling bandwidth would suffice; however in rendering, appearance is integrated over a surface footprint that can still introduce superfluous high frequencies.

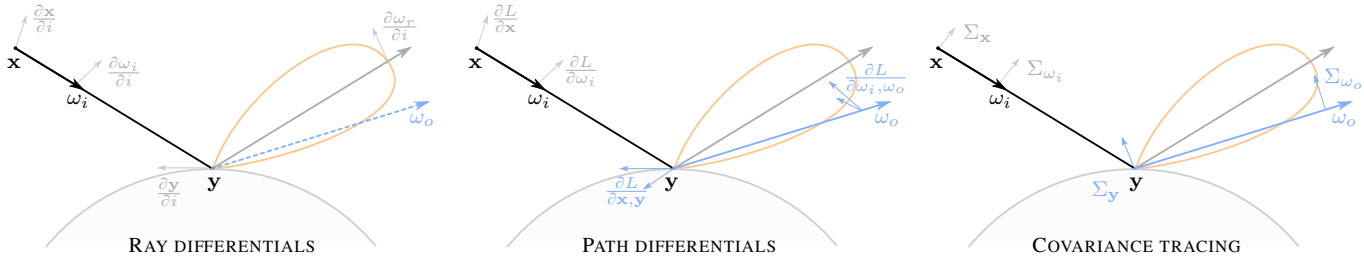


Fig. 1. RAY DIFFERENTIALS (left) propagate the differential of the specular ray interaction with respect to the pixel coordinates. This forbids the study of indirect footprints due to glossy interactions, where a ray spreads across non-mirror directions. PATH DIFFERENTIALS (center) track the gradient of the radiance (or importance), but they require the costly evaluation of partial path derivatives with respect to each position-direction path vertex pair. Our COVARIANCE TRACING approach (right) is simpler, more flexible, and it needs only track the variation of radiance in the local tangent plane of the ray.

as a stopgap solution for indirect bounces.

Prefiltering Indirect Effects. Some works consider special cases of the GI prefiltering problem, often extending the idea of a pixel footprint through to a specific secondary shading effect. Cone tracing [Heckbert and Hanrahan 1984; Shinya et al. 1987] can prefilter geometry and 3D color data [Neyret 1998; Crassin et al. 2009], approximating soft shadows and smooth indirect illumination. Ray differentials [Igehy 1999; Chen and Arvo 2000; Schjøth et al. 2007; Elek et al. 2014] propagate local differentials from sensors (or lights) through only specular reflection and refraction events, but they cannot correctly handle non-specular interactions.

Suykens and Willems' path differentials [2001] most closely resemble our work, albeit with several important differences: while they also identify the importance of devising prefiltering footprints at path vertices, path differentials only account for transport and filtering from eye subpaths and neglect the effects of the light source (and transport from light subpaths), resulting in over-blurring. Moreover, path differentials track and progressively update full path-space partial derivatives, with a compute cost that scales *quadratically* with the *path length* (Figure 1). Our solution avoids this costly computation and scalability issue, maintaining only a single 4×4 symmetric matrix along a path.

Frequency Analysis. We employ a frequency analysis of local light fields [Durand et al. 2005] where, instead of propagating and updating the full 4D spatial-angular light field spectra, we leverage a more compact 2^{nd}-order covariance representation [Belcour et al. 2013; Belcour et al. 2014]. Unlike traditional covariance tracing, we track and update the spectra of radiance *and* importance in order to derive prefiltering footprints along a path. We also derive a simpler BRDF reflection operator for the covariance matrix (Appendix A), and relate our Fourier analysis of path-space filtering to traditional path-space formulations of light transport. This allows us to design uni- and bidirectional path filtering algorithms that are easy to integrate atop existing path-space renderers.

3. OVERVIEW

We will introduce two extended path tracing approaches, each suitable for prefiltering the effects of complex appearance models on the final global illumination. Our **uni-** and **bidirectional antialiasing** approaches (Sections 4 and 5) enable *filtered evaluation* of BSDFs along each path vertex, as illustrated in Figure 2. In Section 6 we detail how our approaches can be readily integrated atop any ray tracer with ray differential support. We present and discuss results in Section 7 before concluding in Section 8.

Unidirectional antialiasing. Starting from the sensor, we propagate a pixel's filter (Figure 2, red) along an eye-path using covariance tracing [Belcour et al. 2013], without considering occlusion. At each scattering interaction, the covariance matrix is first used to compute a spatial footprint which, in turn, is used to perform local filtering of the scattering model at the surface. Here, we employ existing texture filtering [Heckbert 1986] and normal map filtering [Yan et al. 2014] approaches; our technique is insensitive to the choice of approach used to perform the local filtering. After each locally filtered event, the covariance matrix is updated according to an appropriate light transport operation, resulting in an indirect "pixel" filter leaving the surface. Standard path tracing continues, repeating these filtering and evolution steps at each vertex, until path termination.

Bidirectional antialiasing. To antialias appearance in a bidirectional light transport simulation, we simultaneously propagate the covariance matrix representing the pixel's filter for an eye subpath (Figure 2, red) *as well as* the covariance matrix representing the light's importance for the light subpath (Figure 2, blue). As with the unidirectional setting, reflectance profiles at each vertex along both of the subpaths are prefiltered using existing local appearance filtering techniques. The key remaining issue to address in order to correctly filter light transport in a bidirectional setting is the manner in which the eye- and light-subpaths' covariance matrices are combined when the two subpaths are connected: when forming a connection between an eye- and light-subpath, we merge the covariance matrices from each subpath into a new **6D covariance matrix** that correctly captures the combined frequency content of the propagated light and sensor light-fields (Section 5.2). We can then use this matrix to compute a filter footprint to antialias the reflectance at the connection without over-blurring the result.

4. PREFILTERING THE RENDERING EQUATION

Our goal is to antialias complex lighting effects, specifically, transport effects that arise from (potentially many) indirect interactions from scenes with realistic reflectance profiles that are modulated by any combination of high-resolution color, normal and displacement maps. Starting with the rendering equation, we derive an expression for the propagation of a spatial-angular footprint (e.g., a pixel footprint) through a scene (Section 4.1). We use frequency analysis (Section 4.2) to derive an easy-to-compute expression for the propagated footprint that we then use to prefilter texture-modulated reflectance models at vertices of the light transport path (Section 4.3).

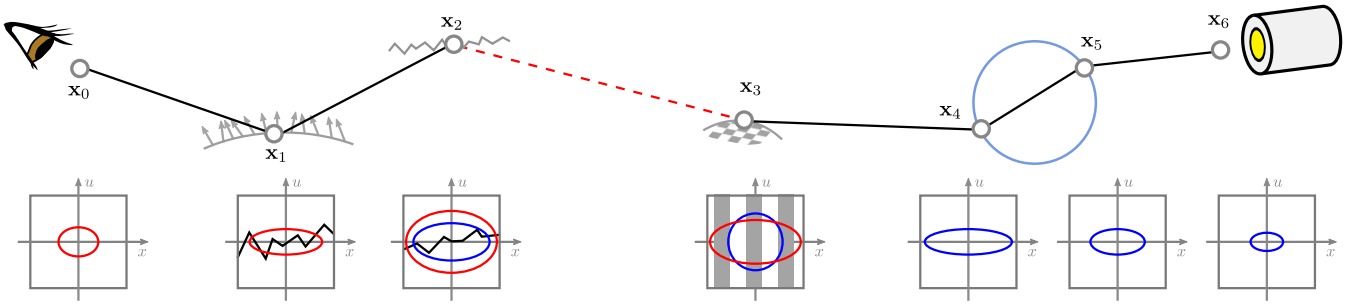


Fig. 2. To compute appropriate filtering footprints at path vertices (visualized as 2D spatial-angular "slices" at each path vertex, bottom row), we propagate covariance from the eye, the light, or both (Sections 4 and 5). At each surface interaction, we update the covariance and compute equivalent ray differentials for filtering (Section 6). Our path vertex filtering takes the indirect effects of detailed normal maps ($\mathbf{x}_1$), displacement maps ($\mathbf{x}_2$), and color maps ($\mathbf{x}_3$) on the final global illumination into account. We also treat bidirectional connections (dashed red line; Section 5).

4.1 Indirect Pixel Footprint

The rendering equation [Kajiya 1986] models global illumination at a shade point as an integral over incident light directions Ω:

$$L(\mathbf{x},\omega_o) = L^0(\mathbf{x},\omega_o) + \int_\Omega L(\mathbf{y}(\mathbf{x}),-\omega)\,\widetilde{\rho}(\mathbf{x},\omega_o,\omega)\,\mathrm{d}\omega, \qquad (1)$$

where $L(\mathbf{x},\omega)$ is the radiance at surface point $\mathbf{x}$ (with normal $\mathbf{n}$) in direction ω, $\widetilde{\rho}(\mathbf{x},\omega_o,\omega) = \rho(\mathbf{x},\omega_o,\omega)\,(\mathbf{n}\cdot\omega)$ is the spatially-varying cosine-weighted BRDF, $L^0(\mathbf{x},\omega_o)$ is the emitted radiance at $\mathbf{x}$ towards ω_o, and $\mathbf{y}$ is the closest surface to $\mathbf{x}$ in direction ω.

Equation 1 does not model a pixel's response on the image sensor. This response is defined by a point spread function or, equivalently, a pixel filter function. Given the sensor filter $s^0_{\mathcal{I}}(\mathbf{x},\omega)$ for pixel $\mathcal{I}$, we can extend the rendering equation to one that additionally models the final pixel sensor's response as:

$$L_{\mathcal{I}} = \int_{\Omega_I\times\mathcal{P}_I} s^0_{\mathcal{I}}(\mathbf{x},\omega_o)\,L(\mathbf{x},\omega_o)\,\mathrm{d}\omega_o\,\mathrm{d}\mathbf{x}\,, \qquad (2)$$

where $\Omega_I \times \mathcal{P}_I$ is the spatial-angular integration domain of the pixel filter, comprising all directions through the pixel and all image locations over the pixel area. We choose to parameterize the pixel filter using surface points directly visible through the pixel, and with directions pointing towards the pixel at these points, as opposed to points on the sensor and directions leaving the sensor; this choice will simplify our exposition later on.

Substituting the outgoing radiance $L(\mathbf{x},\omega_o)$ in Equation 1 into Equation 2, and having $\mathbf{y}$ depend implicitly on $\mathbf{x}$ for brevity, gives:

$$L_{\mathcal{I}} = L^0_{\mathcal{I}} + \iint s^0_{\mathcal{I}}(\mathbf{x},\omega_o)L(\mathbf{y},-\omega)\,\widetilde{\rho}(\mathbf{x},\omega_o,\omega)\,\mathrm{d}\mathbf{x}\,\mathrm{d}\omega\,\mathrm{d}\omega_o,$$

where $L^0_{\mathcal{I}} = \langle s^0_{\mathcal{I}}, L^0_{\mathcal{I}}\rangle$ is the observed emission at pixel $\mathcal{I}$.

Prefiltering is a commonly used in rendering. When computing the *directly-visible* appearance of a surface with e.g., high-frequency variation, many filtering solutions have been proposed: most notably, the product of a pixel filter (projected onto a directly-visible surface) and the spatially-varying BRDF can be directly filtered, reducing aliasing when shading at low sampling rates. If the incident radiance $L(\mathbf{y},-\omega)$ is constant for all surface points directly visible to a pixel, we can replace the product of the pixel filter and cosine-weighted BRDF with its mean $\overline{\rho}_{\mathcal{I}} = \int s^0_{\mathcal{I}}(\mathbf{x},\omega_o)\widetilde{\rho}(\mathbf{x},\omega_o,\omega)\mathrm{d}\mathbf{x}\,\mathrm{d}\omega_o$.

Then, the *filtered* observed shading for pixel $\mathcal{I}$ is:

$$L_{\mathcal{I}} = L^0_{\mathcal{I}} + \int \overline{\rho}_{\mathcal{I}}(\omega)\,L(\mathbf{y},-\omega)\,\mathrm{d}\mathbf{x}\,\mathrm{d}\omega\,.$$

Recall that $\mathbf{y}$ depends on $\mathbf{x}$, and we restrict the spatial-angular integration domain to the extent of pixel $\mathcal{I}$ (i.e., the domain of $s^0_{\mathcal{I}}$).

Departing from the *local* filtering scenario of previous work to the *global* transport setting, we observe that the integrated product of the pixel filter and the cosine-weighted BRDF at a directly visible surface point $\mathbf{x}$ can also serve as an *indirect "pixel" filter*, which we can propagate and apply to the incoming radiance at $\mathbf{x}$ (or, equivalently, the outgoing radiance at $\mathbf{y}$; Figure 3). This recursively defines the k-bounce propagated indirect filter from the eye as:

$$s^k_{\mathcal{I}}(\mathbf{y},\omega) = \int s^{k-1}_{\mathcal{I}}(\mathbf{x},\omega_o)\,\widetilde{\rho}(\mathbf{x},\omega_o,\omega)\,\mathrm{d}\omega_o\,, \qquad (3)$$

where $\mathbf{y}$ is the closest point visible from $\mathbf{x}$ in direction ω. This implicitly defines the propagation of a "filtered" path into the scene.

Given the recursive nature of Equation 3, we can rewrite the observed (and filtered) pixel intensity as an infinite sum:

$$L_{\mathcal{I}} = L^0_{\mathcal{I}} + \sum_{k=0}^{N}\int s^k_{\mathcal{I}}(\mathbf{x},\omega_o)\,\widetilde{\rho}(\mathbf{x},\omega_o,\omega)\,L^0(\mathbf{y},-\omega)\,\mathrm{d}\mathbf{x}\,\mathrm{d}\omega\,. \qquad (4)$$

We use Equation 4 to antialias complex GI effects (after any number of indirect bounces): at each bounce k, we will antialias the

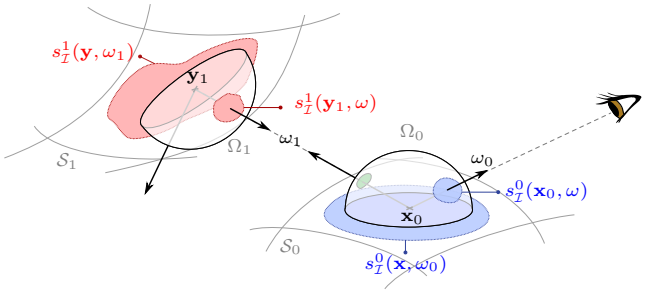


Fig. 3. We recursively construct indirect pixel filters by propagating the interaction of the pixel filter $s^0_{\mathcal{I}}$ with each path vertex's BSDF (in green). Since the indirect pixel filter for bounce-k, $s^k_{\mathcal{I}}$, defines a 4D field, we only visualize 2D slices above: these correspond to angular-distributions (e.g., $s^0_{\mathcal{I}}(\mathbf{x}_0,\omega)$) when fixing a position (e.g., $\mathbf{x}_0 \in \mathcal{S}_0$), or to spatial-distributions (e.g., $s^1_{\mathcal{I}}(\mathbf{y},\omega_1)$) when fixing a direction (e.g., $\omega_1 \in \Omega_1$).

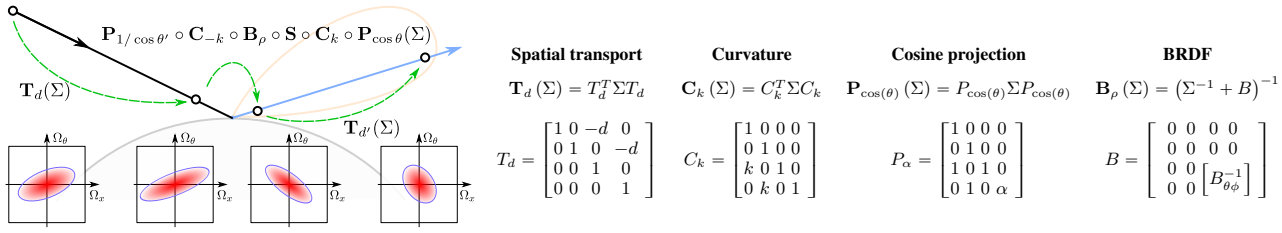


Fig. 4. Illustrating the effect of fundamental light field operators on the covariance matrix of the light field's spatial-angular spectra, for various stages of light transport (left; see [Belcour 2012] for full derivations). The middle row (right) shows the covariance-space operators applied to the covariance matrix Σ for these stages, and the bottom row illustrates their mathematical form. Free-space transport and object curvature operators are shears in covariance-space, while the BRDF reflection operator acts as an angular bandlimiting kernel. We introduce a more efficient evaluation of the BRDF matrix operator in Appendix A.

cosine-weighted BRDF according to the indirect filter's bandlimit. An alternative interpretation is to imagine the projected "image" of the indirect filter, at a surface along a path (i.e., at a path vertex), as an integration footprint over which we evaluate the mean cosine-weighted BRDF (Figure 5). This view resembles ray differentials, however it is not limited to purely specular interactions. Our rendering method computes the appropriate prefiltering bandlimit, and we will also show how to approximate this footprint in order to support renderers that already use ray differentials (Section 6). Two scenarios arise when antialiasing multi-bounce shading effects.

If we assume that the incident radiance variation at each indirect surface interaction (i.e., each bounce) is negligible compared to the propagated indirect filter's variation, we can simply antialias the local reflectance model at each path vertex's surface according to only the bandlimit of the indirect pixel footprint. This corresponds to a straightforward extension of traditional filtering methods for directly-visible reflectance [Olano and Baker 2010; Dupuy et al. 2013; Yan et al. 2014; Jakob et al. 2014] to multi-bounce effects. We detail this *unidirectional* filtering approach in Sections 4.2 and 4.3, outlining its integration into a unidirectional path tracer.

Whenever the incident radiance's variation cannot be ignored, i.e., when it has a higher frequency content than the (indirect) pixel footprint (e.g., with caustics), we can no longer decouple the filtering integral, and the aforementioned unidirectional scheme will fail to properly antialias the multi-bounce shading effect. To solve this problem, we will show how to evaluate the correct bandlimit, taking

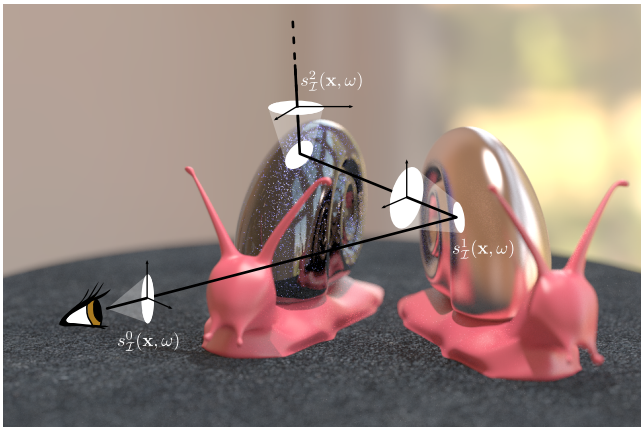


Fig. 5. To antialias complex appearance with global illumination, we propagate pixel footprints along a path through the scene. Expressing the indirect pixel filters $s_\mathcal{I}^k$ compactly and accounting for the light interaction is key to correctly and efficiently perform appearance prefiltering.

the bandlimits of both the propagated indirect pixel footprint *and* the propagated light footprints into account during filtering (Section 5). We will similarly detail the integration of such a filtering scheme atop bidirectional path tracing.

4.2 Unidirectional Connection Covariance Matrix

To filter appearance at arbitrary vertices along a path connecting a pixel to an emitter, we need to be able to estimate the indirect pixel bandwidth at the vertex. We use frequency analysis [Durand et al. 2005] to accurately predict the Fourier spectra of local light fields along a path. Conceptually, this analysis predicts the spatial and angular variation of a light field around any vertex on a light path. We use this theory to consider the evolution of the light field, starting from the sensor's filtering kernel $s_\mathcal{I}^0(\mathbf{x},\omega_o)$.

First, we define the local Fourier Transform (FT) of the indirect pixel filter $s_\mathcal{I}^k(\mathbf{x},\omega)$ (after k bounces), centered around $(\mathbf{x},\omega)$, as:

$$\hat{s}_\mathcal{I}^k(\boldsymbol{\xi}) = \int_\mathcal{D} s_\mathcal{I}^k(\mathbf{x}+\mathbf{u},\omega+\mathbf{v})\, e^{i[\mathbf{u},\mathbf{v}]^T\boldsymbol{\xi}}\, \mathrm{d}\mathbf{u}\, \mathrm{d}\mathbf{v}, \tag{5}$$

where $\boldsymbol{\xi} = [\hat{\mathbf{x}},\hat{\mathbf{v}}]$ is a 1×4 vector of the spatial- and angular-frequency coordinates, $\mathcal{D} = \mathcal{H}^2 \times \mathcal{P}_x$ is the local light field's domain (tangent to $(\mathbf{x},\omega)$), and $\widehat{\mathcal{D}} = \widehat{\mathcal{H}^2} \times \widehat{\mathcal{P}_x}$ is its frequency space equivalent. We denote by $\hat{g}$ the FT of an arbitrary function g.

Frequency analysis of light transport [Durand et al. 2005] allows us to determine the FT of a spatial-angular distribution propagated along a light path (i.e., $s_\mathcal{I}^k(\mathbf{x},\omega)$), using only the FT of the original distribution (i.e., $s_\mathcal{I}^0(\mathbf{x},\omega)$) and the FT of the global transport operator $\mathcal{T}$. To do so, we decompose FT of $\mathcal{T}$ into individual transport operations and treat each bounce of transport sequentially:

$$\begin{aligned} \hat{s}_\mathcal{I}^{k+1}(\boldsymbol{\xi}) &= \mathcal{T}\left[\hat{s}_\mathcal{I}^k(\boldsymbol{\xi})\right] \\ &= \mathbf{P}_{1/\cos\theta'} \circ \mathbf{C}_{-\kappa} \circ \mathbf{B}_\rho \circ \mathbf{S} \circ \mathbf{C}_\kappa \circ \mathbf{P}_{\cos\theta} \circ \mathbf{T}_d\left[\hat{s}_\mathcal{I}^k(\boldsymbol{\xi})\right], \end{aligned}$$

where $\mathbf{P}_{\{\mathrm{X}\}}, \mathbf{C}_\kappa, \mathbf{B}, \mathbf{S}$, and $\mathbf{T}_d$ are atomic transport operators used to decompose the global transport operator $\mathcal{T}$ (see Figure 4) and, e.g., applying this compound operation once with $k = 0$ would propagate the sensor filter by one bounce, corresponding conceptually to previous directly-visible appearance filtering work (albeit using our formulation). We refer readers to Chapter 3 of Belcour's thesis [2012] for full derivations of these operators.

To characterize the bandlimit of $\hat{s}_\mathcal{I}^k(\boldsymbol{\xi})$, we use the compact 2nd-order *covariance* representation of Belcour et al. [2013], without considering time nor occlusion. This corresponds to the 4×4 co-

variance matrix of the 4D FT of the propagated sensor footprint:

$$\Sigma\left(\hat{s}_{\mathcal{I}}^{k}\right) = \int_{\hat{\mathcal{D}}} \left(\boldsymbol{\xi}\cdot\boldsymbol{\xi}^{T}\right) \hat{s}_{\mathcal{I}}^{k}(\boldsymbol{\xi})\,\mathrm{d}\boldsymbol{\xi}\,. \tag{6}$$

The atomic transport operators applied to the 4D light field spectra can all be expressed as matrix operators (applied to the covariance matrix; see Figure 4, right); e.g., the free-space travel operator is:

$$\mathbf{T}_d\left[\,\hat{s}_{\mathcal{I}}^{0}(\boldsymbol{\xi})\,\right] \simeq \mathbf{T}_d\left[\Sigma\right] = T_d^{T}\,\Sigma\,T_d\,.$$

To estimate the covariance matrix after k bounces, Σ_k, we first start at the sensor with covariance Σ_0 and propagate for k bounces using the covariance transport operators, a process illustrated in Figure 4. At each vertex, we can estimate the prefiltering footprint from the covariance matrix to perform antialiasing (Section 4.3).

4.3 Unidirectional Path Filtering with Covariance

We first treat forward path tracing, where paths are constructed from the sensor through the scene and towards emitters. When evaluating transport contributions at path vertices, we can reduce aliasing artifacts from i.e., high-frequency texture, normal, and geometry variation by replacing pointwise evaluations of vertex contribution with *prefiltered values*. This reduces spatial, angular, and temporal noise in the final renderings (see Figure 6).

We filter the spatially-varying BRDF $\rho(\mathbf{x}, \omega_o, \omega)$ at a path vertex $\mathbf{x}_k$, where the effects of any color, normal, and/or displacement map within the indirect pixel filter $s_{\mathcal{I}}^{k}$ is correctly incorporated into a prefiltered $\widetilde{\rho}$. For example, in the case of high-frequency normal maps, the *effective BRDF* is obtained by integrating the NDF over the indirect pixel filter's footprint. The filtered appearance is:

$$\widetilde{\rho}(\mathbf{x}_k, \omega_o, \omega) = \int_{\mathcal{F}} \rho(\mathbf{x}_k{+}\mathbf{u}, \omega_o{+}\omega, \omega)\; h(\mathbf{u}, \omega)\,\mathrm{d}\mathbf{u}\,\mathrm{d}\omega \;=\; \rho * h\,, \tag{7}$$

where $\mathcal{F} = \mathcal{P}_u \times \Omega$ is the 4-dimensional filtering *footprint* (over local positions and directions about $\mathbf{x}_k$) and h is the filtering *kernel*. Once we determine the filtering kernel, we can use existing local filtering methods (see below) to compute Equation 7.

To derive the filtering kernel at $\mathbf{x}_k$, from the propagated sensor covariance Σ_k, we use a Mahalanobis distance of the light field at $\mathbf{x}_k$ to a 0-mean distribution with covariance Σ_k^{-1} as:

$$h(\mathbf{u}, \omega) = g\left(\left[\mathbf{u}, \omega\right]^{T}\,\Sigma_k\,\left[\mathbf{u}, \omega\right]\right) \Big/ K_g\,, \tag{8}$$

where $g(x)$ is any standard 1D filter and K_g is a normalization term that takes the footprint dimensionality (i.e., 4) into account. In practice, we use a Gaussian filter (and so $K_g = (2\pi)^2|\Sigma|$); however, any normalized 1D filter is compatible with our formulation.

We note, however, that most filtered BRDF models are not defined with respect to a filter g, but instead to a surface footprint $\mathcal{F}$. In these cases, we extract an equivalent footprint from the covariance (Section 6). Note that by using the surface footprint we lose the ability to incorporate the pixel filter's profile during antialiasing.

Figures 6 and 7 illustrate the benefits of our covariance path tracing (CPT) over traditional path tracing with (carefully hand-tuned) ray differentials (PT). Since we account for any BRDF interaction, we are able to prefilter appearance even at indirect bounces. This allows us to efficiently render highly detailed textures using EWA filtering [Heckbert 1986] (Figure 6) and normal maps using the BRDF filtering model of Yan et al. [2014] (Figure 7).

Next, we extend our prefiltering formulation to bidirectional path tracing [Veach and Guibas 1994; Lafortune and Willems 1993].

5. COVARIANCE FILTERING IN PATH SPACE

Naïvely extending unidirectional prefiltering to bidirectional path tracing (BDPT) fails to correctly antialias all vertices along a path. Depending on the vertex used to form a bidirectional connection, unidirectional prefiltering can overblur the connection if, e.g.,

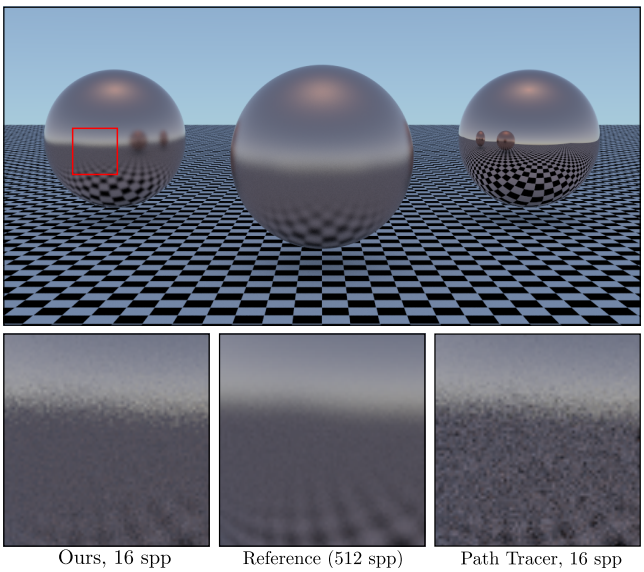


Fig. 6. INDIRECT TEXTURE FILTERING. Our method (bottom left) filters the textured floor after glossy (left & middle spheres) and diffuse (floor) indirect bounces. Even at prohibitively low sampling rates, we outperform path tracing with carefully hand-tuned ray differentials (bottom right).

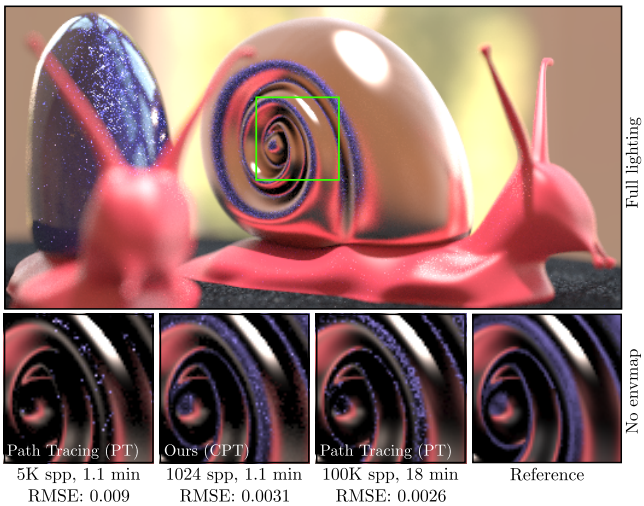


Fig. 7. SNAILS. Equal time & equal error comparison. PT suffers from high-frequency noise (and temporal incoherence; see video) due to aliasing in the indirect reflectance sampling caused by a detailed normal map and environment map. CPT's results visually correspond much more closely to ground truth and remain temporally coherent in animation (see video) despite having roughly equal error (due to filtering bias; see Section 7).

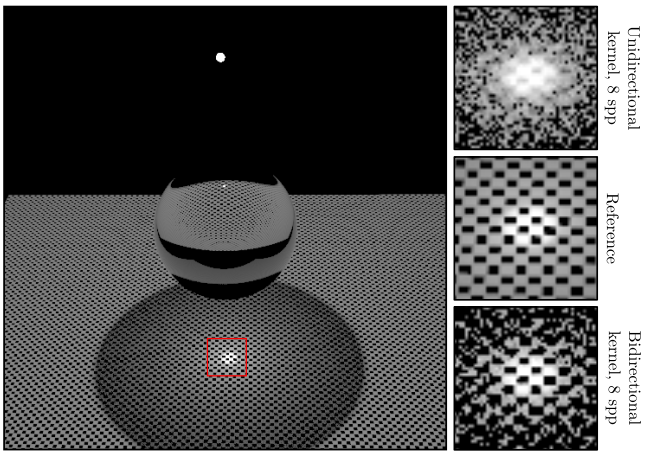


Fig. 8. In BDPT, unidirectional filtering with carefully tuned ray differentials, path differentials, or our unidirectional method (top right) will not correctly account for the frequency content of both eye and light subpaths, resulting in texture over-blurring. Our bidirectional prefiltering (bottom right) treats the throughput's full frequency content, avoiding the over-blurring.

there is a substantial mismatch between the light and eye subpaths' frequency content (Figure 8). Since BDPT can form connections at each vertex from both directions, and then combine them using MIS, we need prefiltering kernels that are sensitive to the way bidirectional connections are formed. To do so, we first introduce a frequency analysis of the path space formulation of light transport that will simplify our bidirectional prefiltering kernel derivations.

We will show that prefiltering kernels for bidirectional vertex connections can be estimated from the sum of covariance matrices propagated unidirectionally from eye and light (Equation 14), permitting simple BRDF antialiasing during subpath connection.

We first briefly introduce the traditional path space formulation of light transport, and a formulation of local path space FTs, before deriving the light field covariance matrix for an *isolated* path vertex (Section 5.1). We relate this local covariance matrix to its unidirectional counterpart from Section 4.3 in order to devise an expression for a *bidirectional covariance matrix* (Section 5.2) that we can use to filter bidirectional light transport connections (Section 5.3). Section 6 will use our theory below to formulate a novel bidirectional global illumination filtering algorithm atop standard BDPT.

5.1 Frequency Analysis of Path Space

Let $\overline{\mathbf{x}} = [\mathbf{x}_0, \dots, \mathbf{x}_{n-1}]$ be a length-n path, Ω_n the set of all length-n paths (Figure 9a/b), and Ω the set of all possible paths. The throughput of a path connecting a sensor pixel to an emitter point is:

$$f(\overline{\mathbf{x}}) = s^0_{\mathcal{I}}(\mathbf{x}_0)\, G(\mathbf{x}_0 \rightarrow \mathbf{x}_1)\, L^0(\mathbf{x}_n) \prod_{k=1}^{n-2} \rho(\mathbf{x}_k)\, G(\mathbf{x}_k \rightarrow \mathbf{x}_{k+1}), \quad (9)$$

where $G(\mathbf{x}_k \rightarrow \mathbf{x}_{k+1})$ accounts for the form factor and visibility between two path vertices, and the remaining terms are path space equivalents of the pixel filter, BRDF, and light source emission.

We define the local FT of f around path $\overline{\mathbf{x}}$ as (see Figure 9c):

$$\widehat{f}_{\overline{\mathbf{x}}}(\boldsymbol{\xi}) = \int_{\Omega_n} w([\overline{\mathbf{x}} - \overline{\mathbf{y}}]^T \Sigma_{\max} [\overline{\mathbf{x}} - \overline{\mathbf{y}}])\, f(\overline{\mathbf{y}})\, e^{i \overline{\mathbf{y}}^T \boldsymbol{\xi}}\, \mathrm{d}\overline{\mathbf{y}}\,, \quad (10)$$

where $w([\overline{\mathbf{x}} - \overline{\mathbf{y}}]^T \Sigma_{\max} [\overline{\mathbf{x}} - \overline{\mathbf{y}}])$ is a windowing function around $\overline{\mathbf{x}}$ that (conceptually) restricts the analysis to the local subspace where the FT is well defined (note that Ω_n is a cartesian product of differential surface manifolds). Here, $\Sigma_{\max}$ controls the extent of the filter and the frequency content of the window propagated into $\widehat{f}_{\overline{\mathbf{x}}}(\boldsymbol{\xi})$, and $\boldsymbol{\xi} = [\widehat{\mathbf{y}}_0, \dots, \widehat{\mathbf{y}}_{n-1}]$ is a $1 \times 2n$ frequency coordinate vector for a length-n path's Fourier domain dual (n dimensions with 2 coordinates per dimension; similarly to Equation 5).

In practice, $\Sigma_{\max}$ will be determined by the renderer's pixel sampling rate, as well as any additional stratification schemes applied during pixel and light sampling (see Section 6 for details).

Constructing the covariance matrix, we obtain (see Figure 9d):

$$\Sigma_{\overline{\mathbf{x}}} = \int_{\widehat{\Omega}_n} (\boldsymbol{\xi} \cdot \boldsymbol{\xi}^T)\, \widehat{f}_{\overline{\mathbf{x}}}(\boldsymbol{\xi})\, \mathrm{d}\boldsymbol{\xi}\,. \quad (11)$$

Conceptually, $\Sigma_{\overline{\mathbf{x}}}$ encodes the correlation of path pairs (in length-n path subspace Ω_n): entry (i, j) of this matrix corresponds to the covariance obtained by varying vertices $\mathbf{x}_i$ and $\mathbf{x}_j$ while fixing the all remaining vertices (integration of the FT is equivalent to evaluation at 0 in the primal path space; similarly to Equation 6).

While conducting an analysis on $\Sigma_{\overline{\mathbf{x}}}$ is an interesting avenue of research, we are instead interested in the path space covariance about a *single* vertex $\mathbf{x}_k$ of path $\overline{\mathbf{x}}$, averaging out the effects of all remaining vertices. We define the *vertex covariance* $\Sigma_{\mathbf{x}_k}$ below in a manner that only considers variations and correlations of the two *directly neighboring* vertices (i.e., $\mathbf{x}_{k-1}$ and $\mathbf{x}_{k+1}$), averaging out the contribution to the covariance of the remaining vertices. Concretely, $\mathbf{x}_k$'s spatial-angular path vertex throughput covariance is:

$$\Sigma_{\mathbf{x}_k} = \int_{\widehat{D}} (\boldsymbol{\xi}_k \cdot \boldsymbol{\xi}_k^T)\, \widehat{f}_{\overline{\mathbf{x}}}(\boldsymbol{\xi}_k)\, \mathrm{d}\boldsymbol{\xi}_k, \quad (12)$$

where the domain $\widehat{D} = \widehat{\mathbf{x}}_k \perp \widehat{\Omega}_n$ is the projection of path vertex Fourier domain onto the entire Fourier domain of length-n path space[2], and $\boldsymbol{\xi}_k = [0, ..., 0, \widehat{\mathbf{x}}_{k-1}, \widehat{\mathbf{x}}_k, \widehat{\mathbf{x}}_{k+1}, 0, ..., 0]$ is a $1 \times 2n$ frequency coordinate vector for a *3-point subset* of a length-n path's Fourier domain dual (i.e., non-zero elements for three vertices, each with two coordinates): instead of capturing light field covariance over an *entire* path (i.e., $\Sigma_{\overline{\mathbf{x}}}$), $\Sigma_{\mathbf{x}_k}$ represents light field covariance about the 3-point configuration centered at $\widehat{\mathbf{x}}_k$.

Next, we will show how to estimate $\Sigma_{\mathbf{x}_k}$, a covariance matrix suitable for use in filtering a bidirectionally connected path vertex with our unidirectional analysis (Sections 4.2 and 4.3). Specifically, we will see that the unidirectional covariance matrices at the eye- and light-subpaths will combine to form the proper covariance matrix at the bidirectionally-connected vertex k (see Figure 9d).

5.2 Bidirectional Connection Covariance Matrix

Similar to Section 4, we can begin by expressing the contribution of length-n paths to the observed/integrated radiance for a pixel $\mathcal{I}$ using the propagated indirect pixel filter and the integrated path

[2]We could equivalently interpret $\widehat{D}$ as the (frequency space) restriction of length-n path space to the 3-point vertex configuration centered at $\mathbf{x}_k$.

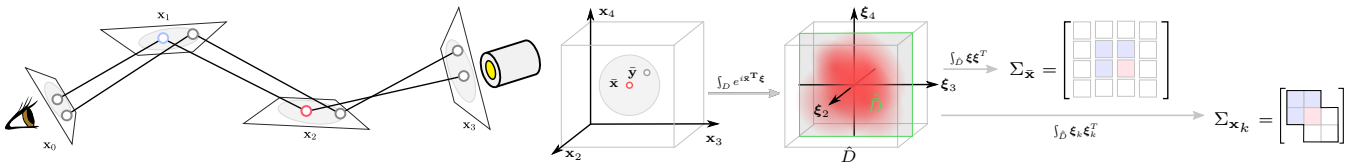


Fig. 9. Consider paths of fixed length (a), where we window the throughput (b) in order to perform a local Fourier analysis (c) of the path throughput. The throughput's (local) frequency spectrum can be compactly represented by its covariance (d): we use the *vertex covariance matrix* (far right) to compute antialiasing filters in path space. Here, we are interested in the covariance at vertex k (red), constructed from its neighboring vertices (blue & gray).

throughput at the 3-point configuration centered at vertex $\mathbf{x}_k$:

$$L_{\mathcal{I}}^{n} = \int s_{\mathcal{I}}^{k}(\mathbf{x}_{k-1}, \mathbf{x}_k)\, \rho(\mathbf{x}_k)\, L^{n-k}(\mathbf{x}_k, \mathbf{x}_{k+1})\, \mathrm{d}\mathbf{x}_{k-1}\, \mathrm{d}\mathbf{x}_k\, \mathrm{d}\mathbf{x}_{k+1}.$$

Here, $s_{\mathcal{I}}^{k}(\mathbf{x}_{k-1}, \mathbf{x}_k)$ is the propagated pixel filter and $L^{n-k}(\mathbf{x}_k, \mathbf{x}_{k+1})$ is the incident radiance from the light subpath to the vertex.

In order to properly filter bidirectional path connections, we have to take care to segment the reflectance $\rho(\mathbf{x}_k)$ at the connection point $\mathbf{x}_k$, which is the function we will be filtering, from our local path vertex light field covariance $\Sigma_{\mathbf{x}_k}$. Mathematically, if we define the Fourier spectrum of the local light field at $\mathbf{x}_k$ as $\hat{f}_k(\overline{\mathbf{x}}) = \mathcal{F}\left[s_{\mathcal{I}}^{k}(\mathbf{x}_{k-1}, \mathbf{x}_k)\, \rho(\mathbf{x}_k)\, L^{n-k}(\mathbf{x}_k, \mathbf{x}_{k+1})\right]$ (where $\mathcal{F}[\cdot]$ is the FT operator) and we understand that this local FT spectrum is effectively represented by the local vertex light field covariance $\Sigma_{\mathbf{x}_k}$ (from Equation 12), then we can start to reason about how to perform a segmentation of $\rho(\mathbf{x}_k)$ from this spectrum.

Specifically, when evaluating $\Sigma_{\mathbf{x}_k}$ at $\mathbf{x}_k$, we are still incorporating the frequency content of $\rho(\mathbf{x}_k)$ (the function we actually want to filter) in the local light field's frequency space covariance. Instead, we need to determine the frequency content of the *reflectance-restricted throughput* (RRT) at $\mathbf{x}_k$, which does not incorporate the frequency content of $\rho(\mathbf{x}_k)$, which we denote as $f_k(\overline{\mathbf{x}})/\rho(\mathbf{x}_k)$.

The RRT can be defined as the product of the propagated indirect pixel filter and the incident radiance at the vertex:

$$f_k(\overline{\mathbf{x}})/\rho(\mathbf{x}_k) = s_{\mathcal{I}}^{k}(\mathbf{x}_{k-1}, \mathbf{x}_k) \times L^{n-k}(\mathbf{x}_k, \mathbf{x}_{k+1}).$$

The FT of $f_k(\overline{\mathbf{x}})/\rho(\mathbf{x}_k)$ is then the convolution of the FT of each term along $\mathbf{x}_k$. We can perform this composition entirely in the covariance domain [Belcour et al. 2013] to obtain the full RRT covariance matrix at $\mathbf{x}_k$ as:

$$\Sigma_{\mathbf{x}_k}^{\mathrm{RRT}} = \Sigma\left(\hat{s}_{\mathcal{I}}^{k} \ast_k \hat{L}^{n-k}\right), \tag{13}$$

where $\ast_k$ corresponds to a restricted convolution at $\mathbf{x}_k$ and $\Sigma_{\mathbf{x}_k}^{\mathrm{RRT}}$ is a 6×6 covariance matrix that captures the frequency content of the 3-point configuration centered at $\mathbf{x}_k$ (and connected to $\mathbf{x}_{k-1}$ and $\mathbf{x}_{k+1}$; see Figure 9d, left). The matrix dimensionality is consistent with the dimensionality of light transport at $\mathbf{x}_k$: a six-dimensional function, comprising three points with two spatial dimensions each.

We apply the covariance convolution property to evaluate $\Sigma_{\mathbf{x}_k}^{\mathrm{RRT}}$ as the sum of the covariances of the spectra:

$$\Sigma_{\mathbf{x}_k}^{\mathrm{RRT}} = \Sigma\left(\hat{s}_{\mathcal{I}}^{k} \ast_k \hat{L}^{n-k}\right) = \Sigma\left(\hat{s}_{\mathcal{I}}^{k}\right) \oplus_k \Sigma\left(\hat{L}^{n-k}\right), \tag{14}$$

where $\Sigma(\hat{s}_{\mathcal{I}}^{k})$ and $\Sigma(\hat{L}^{n-k})$ are 4×4 propagated eye- and light-subpath covariance matrices (Section 4.2), and $\oplus_k$ is a restricted matrix sum that only sums the overlapping 2×2 central region of these two covariances. Note, the two empty 2×2 submatrix blocks in the top-right and bottom-left corners of $\Sigma_{\mathbf{x}_k}^{\mathrm{RRT}}$ indicate that correlations between $\mathbf{x}_{k-1}$ and $\mathbf{x}_{k+1}$ are not modelled with our formulation. The resulting covariance, $\Sigma_{\mathbf{x}_k}^{\mathrm{RRT}}$, does however model the *joint impact* of the spatial-angular emitter *and* sensor spectra at $\mathbf{x}_k$.

5.3 Using Covariance for Bidirectional Path Filtering

To filter the reflectance $\rho(\mathbf{x}_k)$ at a bidirectional connection vertex $\mathbf{x}_k$, we extend the method in Section 4.3: we compute a filtering footprint from the bidirectional light field at $\mathbf{x}_k$, based on our analysis above, before constructing an *equivalent ray differential* (see Section 6) from the covariance to use when filtering the reflectance.

Standard ray differentials, however, can only be constructed from a 4×4 covariance matrix (where spatial-angular covariance can be converted to spatial-angular differential offsets), and so the question of which 4×4 submatrix $\widetilde{\Sigma}_{\mathbf{x}_k}$ we select from the 6×6 covariance $\Sigma_{\mathbf{x}_k}$ is fundamental to the construction of our *equivalent ray differentials*: we can either use the top-left $\Sigma_{\mathbf{x}_{k-1}\to\mathbf{x}_k}$ or bottom-right $\Sigma_{\mathbf{x}_k\leftarrow\mathbf{x}_{k+1}}$ 4×4 submatrix of $\Sigma_{\mathbf{x}_k}^{\mathrm{RRT}}$, corresponding to either the eye or light connection at $\mathbf{x}_k$ (see Figure 9d, far right). These options correspond to which "direction" we want to filter from, and we will discuss how to choose between them in Section 6.

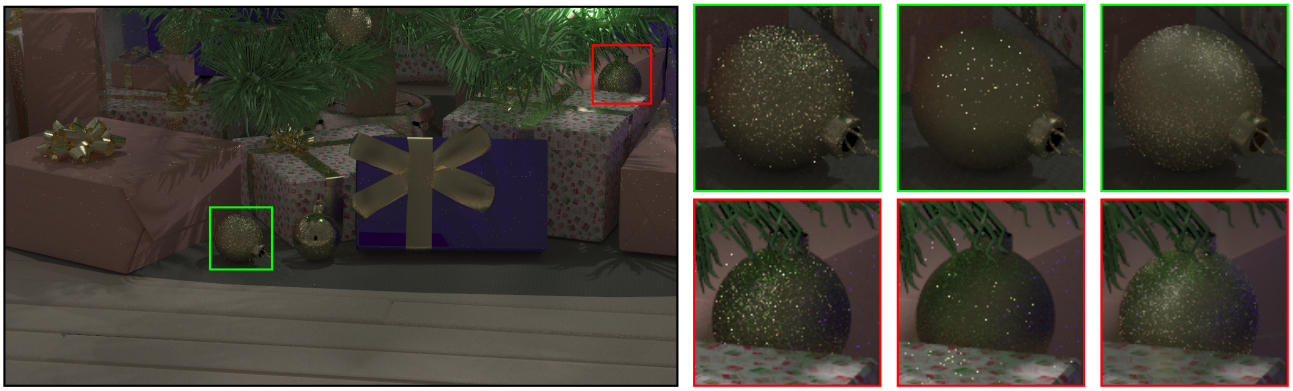
Given our choice of the path vertex covariance submatrix $\Sigma_{\mathbf{x}_k}^{\mathrm{RRT}}$ we will use for filtering, we derive the filtering kernel h and footprint $\mathcal{F}$ exactly as in Section 4.3. The filtered contribution of the k^{th} vertex (to the total path throughput f) can be expressed as the convolution of the kernel with the (unfiltered) throughput contribution: $\widetilde{\rho}_k \approx \rho \ast_k h$, where $\ast_k$ is a "local" convolution at $\mathbf{x}_k$ of the throughput, instead of over the entire space of length-n paths Ω_n.

6. IMPLEMENTATION DETAILS

We present two new rendering algorithms, covariance path tracing (CPT; based on Section 4) and covariance bidirectional path tracing (CBDPT; based on Section 5), that combine our covariance filtering formulations with covariance tracing to antialias complex global illumination effects. Given the ubiquitous support for ray differentials in modern renderers (i.e., PBRT [Pharr and Humphreys 2010] and Mitsuba [Jakob 2010]), we will derive expressions for our filtering footprints (Sections 4.3 and 5.3) based on *equivalent ray differentials* (see below), allowing us to leverage existing support for efficient texture filtering in modern renderers.

In practice, our algorithms require few changes to existing uni- and bidirectional integrators, described at a high-level as follows:

(1) We compute 4×4 covariance matrices from either the sensor or the light during path tracing: at each interaction, we update the covariance using standard covariance operators (see [Belcour



CBDPT (Ours) 3.78 hrs (2048 spp) | BDPT 3.78 hrs (3968 spp) | Reference 141 days (100M spp)

Fig. 10. CHRISTMAS scene. CBDPT renders 1024 spp in 3.78 hours while BDPT can generate 3968 spp in equal time in this complex scene, where emitters are modeled realistically and all visible illumination is from caustics. BDPT completely fails to render the indirect normal mapped microfacet reflections on the ornaments. The reference was run on a cluster of Intel Xeon E5 CPUs and took 1133 core days (**141 days on a 8 core machine**).

et al. 2013; Belcour et al. 2014]) and store covariance at each path vertex for potential bidirectional connections;

(2) When forming bidirectional connections, we combine the eye and light covariance submatrices to construct the radiance reduced throughput (RRT) covariance from the eye's direction;

(3) We compute equivalent ray differentials from these footprints (see below) to evaluate texture coordinate partial derivatives with respect to a virtual pixel footprint and to perform the appropriate local texture/appearance filtering.

We now discuss specific implementation details of our algorithms.

Covariance Stratification. Pixel sampling rate defines the maximum observable frequency content, corresponding to the diagonal entries of Σ_{max}. At sensor vertices $\mathbf{x}_0$, we scale the outgoing filter's covariance to conservatively account for any pixel stratification, ensuring that we remain consistent and avoid over-blurring.

Bi-directional Covariance Connections. We compute the RRT covariance $\Sigma^{\text{RRT}}_{\mathbf{x}_k}$ at a vertex (Section 5.2, Equation 14) using propagated eye and light subpath covariances, $\Sigma(\widehat{s}^k_I)$ and $\Sigma(\widehat{L}^{n-k})$.

When forming a *filtered bi-directional connection*, we have two choices for the covariance matrix that we use to compute the filtering footprint (Section 5.3): either the eye path submatrix $\Sigma_{\mathbf{x}_{k-1}\to\mathbf{x}_k}$ or the light path submatrix $\Sigma_{\mathbf{x}_k\leftarrow\mathbf{x}_{k+1}}$. In our implementation, we choose $\Sigma_{\mathbf{x}_{k-1}\to\mathbf{x}_k}$ in order to remain compatible with existing appearance antialiasing techniques: all existing models are defined according to an incident 2D surface footprint. This footprint is encoded in the 2D submatrix of vertex k. The remaining components of the covariance matrix describe the angular footprint for antialiasing, which we could use to to more efficiently prefilter appearance in the presence of defocus, for example. We leave this application to future work.

Filtering with Equivalent Ray Differentials. To simplify our implementation, we build atop ray differentials [Igehy 1999] to evaluate prefiltered appearance models.

A ray differential is defined by a central ray $\boldsymbol{r} = \{\boldsymbol{o}, \vec{d}\}$ and its partial derivatives with respect to the pixel position: $\{\partial\boldsymbol{o}/\partial u, \partial\vec{d}/\partial u\}$, $\{\partial\boldsymbol{o}/\partial v, \partial\vec{d}/\partial v\}$. Partial derivatives are taken with respect to the pixel frame $(\vec{u}, \vec{v})$, and the ray differential is propagated into the scene and updated to account for all specular interactions from the sensor.

Assuming a Gaussian spectrum, we derive the 4D filtering footprint using the SVD decomposition of the covariance $\Sigma_k = U\,\Lambda\,U^{-1}$, where the eigenvectors $\{\mathbf{x}_0, \mathbf{x}_1, \boldsymbol{\omega}_0, \boldsymbol{\omega}_1\}$ define the local spatial-angular coordinate frame of the 4D footprint, and the eigenvalues $\{\Lambda_{\mathbf{x}_0}, \Lambda_{\mathbf{x}_1}, \Lambda_{\boldsymbol{\omega}_0}, \Lambda_{\boldsymbol{\omega}_1}\}$ correspond to the variance along the axes of this frame. The footprint size is inversely proportional to the standard deviation, in each axis. We use three times the standard deviation to capture most of the energy of the pixel filter (i.e., 99.6% for Gaussian spectra). The equivalent ray differential is then:

$$\begin{aligned} \partial\boldsymbol{o}/\partial u &= 3/\sqrt{\Lambda_{\mathbf{x}_0}}\mathbf{x}_0 \quad \partial\boldsymbol{o}/\partial v = 3/\sqrt{\Lambda_{\mathbf{x}_1}}\mathbf{x}_1 \\ \partial\vec{d}/\partial u &= 3/\sqrt{\Lambda_{\boldsymbol{\omega}_0}}\boldsymbol{\omega}_0 \quad \partial\vec{d}/\partial v = 3/\sqrt{\Lambda_{\boldsymbol{\omega}_0}}\boldsymbol{\omega}_1 \end{aligned}$$

7. RESULTS AND DISCUSSION

We implement CPT and CBDPT in Mitsuba [Jakob 2010], with minimal modifications, overloading the RAYDIFFERENTIAL class to perform covariance filter construction (Section 6). We compare our two algorithms to their unfiltered counterparts, forward unidirectional (PT) and bidirectional path tracing (BDPT), both using the reference Mitsuba implementations. Our implementation is not optimized and relies as much as possible on existing Mitsuba functionality, with the exception of an implementation of Yan et al.'s local filtering model [2014] (originally conceived to antialias directly visible high-resolution normal mapped specular surfaces).

We demonstrate the usefulness and efficiency of our methods on scenes with complex light transport (ASTRONAUT, TEXTURE, SNAILS, CHRISTMAS, and CAUSTIC). All results are computed with multi-threading on a 4-core Intel i7-2600K with 8GB of RAM.

TEXTURE (Figure 6) illustrates our method's ability to antialias textures viewed through indirect reflection. We use EWA for albedo filtering and compare our CPT to PT with ray differentials. We use a low sampling rate to highlight the form of our error versus that of PT. Note that, at this sampling rate, CPT still contains some noise due to integrand discontinuities (i.e., visibility).

SNAILS and ASTRONAUT (Figures 7 and 12) illustrates our ability to correctly predict filtering kernels for indirect bounces, even on

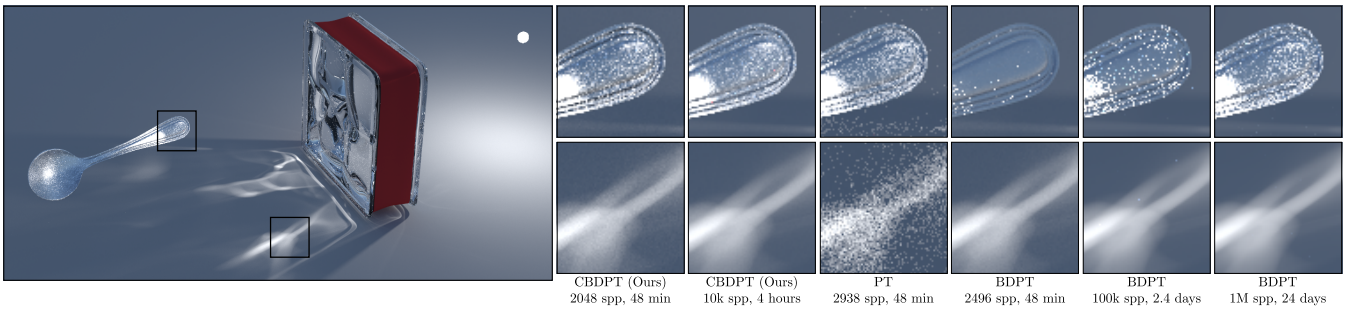


Fig. 11. CAUSTIC scene. We compare our bidirectional method (CBDPT) to PT and BDPT. After 48 minutes, CBDPT begins visually converging, much faster than both PT and BDPT. BDPT requires prohibitively long render times (here, **24 days**) before converging visually on e.g., the spoon's handle.

challenging cases like the curved snail shell. CPT uses 1.1 mins (1024 spp at 512×512; Figure 7, middle), the majority of which is spent filtering the normal-mapped reflectance [Yan et al. 2014], whereas PT reaches equal error in 18 mins (100K spp; right) and equal time with 5K spp (left). The equal time result has high variance and, despite a roughly equivalent RMSE/SNR, our result's errors are not visually apparent (due to filtering bias). Moreover, the smoothness of our result corresponds much more closely to ground truth and remains temporally coherent (see video). For ASTRONAUT, images are rendered in HD resolution (1280×720). We compare CPT and PT for a roughly equal render time (1024 spp). Despite the relatively simple geometry of the scene, radiometric complexity is high: it is impossible for PT to form a connection between the spoon and the distant star lights after the glossy bounce. We also use BDPT to generate results with high sampling rates (1M and 10M spp), requiring **1 and 10 days** to render *just the inset images*! Using BDPT at these rates to render the full image would have taken **13 and 131 days**. Even at this exagerated setting, the BDPT results have not converged, although they hint shape of the reflection CPT can generate in a negligible fraction of that time.

CHRISTMAS and CAUSTIC (Figures 10 and 11) both use CBDPT. CHRISTMAS is our most complex scene: spherical emitters are embedded in glass bulbs, and so *all emitted light* is due to indirect caustics, which is subsequently reflected indirectly off the glittery Christmas ornaments and shiny presents (both with high-resolution color and normal maps modulating microfacet reflectance models). CAUSTIC casts a complex caustic, resulting from curved reflections and refractions off the displacement mapped glass block, onto the ground and a glittery normal-mapped spoon. CHRISTMAS is rendered at 1280×720 and CAUSTIC at 1024×512. In both cases, BDPT requires a prohibitive amount of time to converge to the correct, temporally coherent result (**141 day** render time on 8 cores for 100M spp). Neither PT nor BDPT can resolve *both* the ground caustic and its indirect reflection off the spoon in CAUSTIC.

We list performance statistics for covariance tracing and path vertex filtering comparing timings of PT or BDPT with ray differentials ("Alt.") to only tracing covariance ("Cov. only") and to tracing *and* filtering appearance at vertices ("Ours"):

Scene	Alt.	Ours	Cov. only
ASTRONAUT (1024spp)	10.9 min	15.6 min	13.5 min
SNAIL (128spp)	56 s	4 min	1.15 min
CAUSTIC (1024spp)	14.5 min	48 min	19.8 min
CHRISTMAS (1024 spp)	48.35 min	3.8 hours	1.0 hour

Regardless of scene complexity, covariance tracing adds an overhead of 1.23-$1.36\times$ atop (BD)PT; the majority of our approach's additional cost is due to local appearance filtering (e.g., [Yan et al. 2014]). Note that, with these render times, no existing approach comes close to visual convergence, whereas all our results do.

Discussion and Future Work. Current local appearance antialiasing mostly treat spatial aliasing but largely ignore angular aliasing (some works consider the correlation between view elevation and spatial footprint, however). Filtering according to both spatial and angular footprints could lead to more robust local antialiasing, and our global approach could also readily leverage such extensions, as we already provide the angular bandwidth of indirect pixel/emitter filters as input to the local filtering methods we use.

We are also interested in exploring whether the *entire path covariance* $\Sigma_{\overline{\mathbf{x}}}$ can be computed and, if so, how it could be applied to, e.g., Markov Chain Monte Carlo methods. Note that this matrix formulation can be interpreted as an extension of Chen and Arvo's work [2000] to non-specular surface interactions.

Extensions of our formulation should be able to antialias motion blurred appearance, since covariance tracing inherently supports motion covariance estimation; similarly, since covariance tracing can also be applied to participating media [Belcour et al. 2014], one could generalize our theory to filter volumetric effects so long as local volumetric prefiltering solutions can be devised.

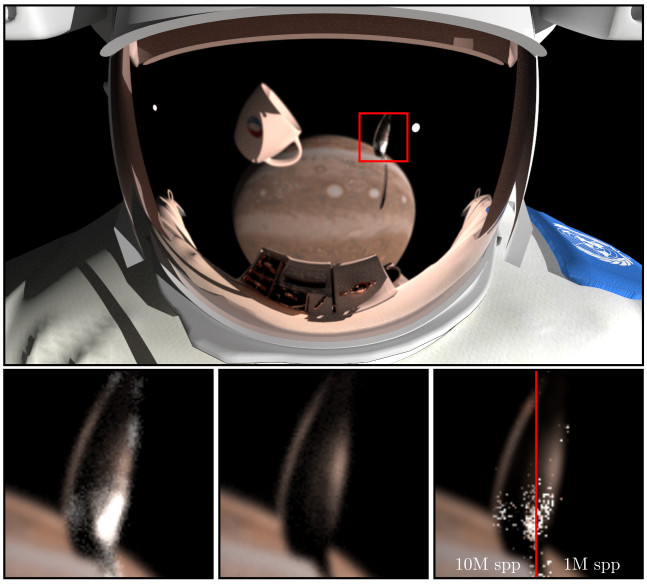
Lastly, our bidirectional path space covariance can be applied to problems beyond antialiasing: indeed, it can be used to predict kernel sizes or regularization filters for density estimation and vertex merging algorithms. Specifically, the 2$^{\text{nd}}$-order nature of covariance, and its relation to the Hessian, makes it practical for density estimation [Belcour and Soler 2011; Belcour et al. 2014].

8. CONCLUSION

We have presented an approach to filter complex global illumination appearance, taking the secondary effects of detailed color, normal, and displacement maps into account. We build two complementary theoretical views of the problem, including relating frequency analysis of light transport to path space formulations of the problem, in order to devise unidirectional and bidirectional filtering techniques. Our method is simple to implement in existing renderers and generates equal quality renders for complex images in orders of magnitude less time than the state-of-the-art.

REFERENCES

BELCOUR, L. 2012. Frequency analysis of light transport: from theory to implementation. Ph.D. thesis, Grenoble Université.



CPT (Ours)
15.6 min (1024 spp)
[**full frame** render time]

PT
15.6 min (1465 spp)
[**full frame** render time]

BDPT
1 & 10 **days** (1M & 10M spp)
[**inset** render time]

Fig. 12. ASTRONAUT scene. We compare our unidirectional CPT to PT and BDPT. PT cannot form the necessary connections to the small star-like luminaires, mandating a filtering approach that supports indirectly-visible objects and high-frequency appearance texturing. Bi-directional approaches are able to form these connections, but they do so with such low probability that multi-day renderings are still unable to visually converge.

BELCOUR, L., BALA, K., AND SOLER, C. 2014. A local frequency analysis of light scattering and absorption. ACM Transactions on Graphics 33, 5.

BELCOUR, L. AND SOLER, C. 2011. Frequency-Based Kernel Estimation for Progressive Photon Mapping. In ACM SIGGRAPH Asia Posters. No. 47.

BELCOUR, L., SOLER, C., SUBR, K., HOLZSCHUCH, N., AND DURAND, F. 2013. 5D Covariance tracing for efficient defocus and motion blur. ACM Transactions on Graphics 32, 3, 31:1–31:18.

BOSCH, C., PUEYO, X., MÉRILLOU, S., AND GHAZANFARPOUR, D. 2004. A physically-based model for rendering realistic scratches. In Computer Graphics Forum. Vol. 23. 361–370.

BOSCH, C., PUEYO, X., MÉRILLOU, S., AND GHAZANFARPOUR, D. 2008. A resolution independent approach for the accurate rendering of grooved surfaces. In Computer Graphics Forum. Vol. 27. 1937–1944.

BRUNETON, E. AND NEYRET, F. 2012. A Survey of Non-linear Prefiltering Methods for Efficient and Accurate Surface Shading. IEEE Transactions on Visualization and Computer Graphics 18, 2 (Feb.).

CHEN, M. AND ARVO, J. 2000. Theory and application of specular path perturbation. ACM Transactions on Graphics 19, 1 (Oct.), 246–278.

CRASSIN, C., NEYRET, F., LEFEBVRE, S., AND EISEMANN, E. 2009. GigaVoxels. In ACM Symposium on Interactive 3D Graphics and Games.

DUPUY, J., HEITZ, E., IEHL, J.-C., POULIN, P., NEYRET, F., AND OSTROMOUKHOV, V. 2013. Linear efficient antialiased displacement and reflectance mapping. ACM Transactions on Graphics 32, 6 (Nov.), 1–11.

DURAND, F., HOLZSCHUCH, N., SOLER, C., CHAN, E., AND SILLION, F. X. 2005. A frequency analysis of light transport. ACM Transactions on Graphics 24, 3, 1115–1126.

ELEK, O., BAUSZAT, P., RITSCHEL, T., MAGNOR, M., AND SEIDEL, H.-P. 2014. Progressive Spectral Ray Differentials.

FOURNIER, A. 1992. Filtering normal maps and creating multiple surfaces. Tech. Rep. TR-92-41, Department of Computer Science, University of British Columbia, Vancouver, BC, Canada.

HAN, C., SUN, B., RAMAMOORTHI, R., AND GRINSPUN, E. 2007. Frequency domain normal map filtering. ACM Transactions on Graphics 26, 3, 28.

HECKBERT, P. S. 1986. Survey of texture mapping. Computer Graphics and Applications 6, 11, 56–67.

HECKBERT, P. S. AND HANRAHAN, P. 1984. Beam tracing polygonal objects. ACM SIGGRAPH Computer Graphics 18, 3, 119–127.

IGEHY, H. 1999. Tracing ray differentials. In ACM SIGGRAPH 1999.

JAKOB, W. 2010. Mitsuba renderer. http://www.mitsuba-renderer.org.

JAKOB, W., HAŠAN, M., YAN, L.-Q., LAWRENCE, J., RAMAMOORTHI, R., AND MARSCHNER, S. 2014. Discrete Stochastic Microfacet Models. ACM Transactions on Graphics 33, 4.

KAJIYA, J. T. 1986. The rendering equation. In ACM SIGGRAPH. Vol. 20.

LAFORTUNE, E. P. AND WILLEMS, Y. D. 1993. Bi-directional path tracing. In Proceedings of Compugraphics. 145–153.

NEYRET, F. 1998. Modeling Animating and Rendering Complex Scenes using Volumetric Textures. IEEE Transactions on Visualization and Computer Graphics 4, 1, 55–70.

OLANO, M. AND BAKER, D. 2010. Lean mapping. In ACM I3D. 181–188.

PHARR, M. AND HUMPHREYS, G. 2010. Physically Based Rendering, 2nd Edition. Morgan Kaufmann.

SCHJØTH, L., FRISVAD, J. R., ERLEBEN, K., AND SPORRING, J. 2007. Photon differentials. In GRAPHITE. 179.

SHINYA, M., TAKAHASHI, T., AND NAITO, S. 1987. Principles and applications of pencil tracing. ACM Computer Graphics 21, 4 (Aug.), 45–54.

SUYKENS, F. AND WILLEMS, Y. 2001. Path differentials and applications. In Eurographics Workshop on Rendering. 257–268.

TOKSVIG, M. 2005. Mipmapping normal maps. Journal Graphics Tools 10, 3, 65–71.

VEACH, E. AND GUIBAS, L. 1994. Bidirectional Estimators for Light Transport. In Proceedings of Eurographics Rendering Workshop.

WOODBURY, M. 1950. Inverting modified matrices. Tech. Rep. 42, Princeton University.

YAN, L.-Q., HAŠAN, M., JAKOB, W., LAWRENCE, J., MARSCHNER, S., AND RAMAMOORTHI, R. 2014. Rendering glints on high-resolution normal-mapped specular surfaces. ACM Transactions on Graphics 33, 4.

APPENDIX

A. IMPROVING THE SCATTERING OPERATOR

We improve the performance of evaluating the band-limiting covariance reflectance operator. The reflectance operator is expressed as a double inversion of the covariance matrix [Belcour et al. 2013, Equation 18]:

$$\Sigma' = \left(\Sigma^{-1} + B^{-1}\right)^{-1},$$

where B is the is the 4×4 rank 2 covariance matrix of the BRDF:

$$B = [0\ 0\ 0\ 0; 0\ 0\ 0\ 0; 0\ 0\ b_u\ 0; 0\ 0\ 0\ b_v]$$

We can avoid computing the costly double matrix inversion with Woodbury's matrix identity [Woodbury 1950]:

$$\Sigma' = \Sigma - \Sigma U\left(B + V\Sigma U\right)^{-1} V\Sigma,$$

with $U^T = V = [0\ 0\ 1\ 0\ 0; 0\ 0\ 0\ 1\ 0]$, and $B = [b_u\ 0; 0\ b_v]$. Finally, $B + V\Sigma U$ can be inverted analytically as it is a 2×2 matrix:

$$B + V\Sigma U = \begin{bmatrix} b_u + \sigma_{uu} & \sigma_{uv} \\ \sigma_{vu} & b_v + \sigma_{vv} \end{bmatrix}.$$