Specular Lobe-Aware Filtering and Upsampling for Interactive Indirect Illumination (Supplemental Material)

Y. Tokuyoshi
Square Enix Co., Ltd., Japan

Appendix A: Gaussian Based Weighting Function

This section proves that the Gaussian based weighting function (given by Eq. (2) in the paper) is the special case of our distribution-aware weighting function. In addition, the roles of two user-specified parameters are explained. When \( a_i(x) \propto g \left( x - x_i, \tau_i^2 \right) \) for \( x \in \mathbb{R}^m \) and \( \Omega = \mathbb{R}^m \), we can use \( b'(x', x) \propto g \left( x' - x, \tau^2 \right) \) where \( \tau^2 \) is the user-specified parameter. For this case, the smoothed distribution \( c_i(x) \) is also a Gaussian function as follows:

\[
c_i(x) \propto g \left( x - x_i, \tau_i^2 \right),
\]

where \( \tau_i^2 = \tau^2 + \nu^2 \). Thus, the normalized distribution function is given by

\[
p_i(x) = \frac{g \left( x - x_i, \tau_i^2 \right)}{\left( \pi \tau_i^2 \right)^{\frac{m}{2}}}.\]

Therefore, the similarity is derived as

\[
q_{i,j} = \left( \frac{2 \tau_i \tau_j}{\tau_i^2 + \tau_j^2} \right)^{\frac{m}{2}} \frac{1}{\left( \pi \tau_i^2 \right)^{\frac{m}{2}}} g \left( x_i - x_j, \frac{\tau_i^2 + \tau_j^2}{\beta} \right),
\]

and thus the weighting function is given by

\[
w(i, j) = \frac{\beta}{q_{i,j} \left( \frac{2 \tau_i \tau_j}{\tau_i^2 + \tau_j^2} \right)^{\frac{m}{2}} \frac{1}{\left( \pi \tau_i^2 \right)^{\frac{m}{2}}} g \left( x_i - x_j, \frac{\tau_i^2 + \tau_j^2}{\beta} \right)}.
\]

The term \( \frac{2 \tau_i \tau_j}{\tau_i^2 + \tau_j^2} \) detects the difference of \( \tau_i \) and \( \tau_j \) (i.e. difference of \( \tau_i \) and \( \tau_j \)). If \( \tau_i = \tau_j \), our weighting function is equivalent to the Gaussian based weighting function as follows:

\[
w(i, j) = g \left( x_i - x_j, \frac{2 \tau^2}{\beta} + \frac{2 \nu^2}{\beta} \right).
\]

This variance parameter is the linear transformation of input-dependent variance \( \tau_i^2 \) using user-specified parameters \( \nu^2 \) and \( \beta \). In order to control the kernel bandwidth using this linear transformation, these two parameters are introduced in this paper.

Appendix B: SG Approximation for Parametric BRDFs

Parametric BRDFs are fitted with a single SG by using Wang’s on-the-fly analytical approximation [WRG⁺09]. A BRDF is separated into two factors: the NDF \( D_i(h_i) \), and the rest of the factors \( M_i(\psi, \omega) \) as follows:

\[
\rho(y, \psi, \omega) = M_i(\psi, \omega) D_i(h_i),
\]

where \( h_i \) is the half-way vector of \( \psi_i \) and \( \omega_i \). Bell-shaped NDFs are approximated with an SG. For example, the Phong distribution is approximated as

\[
D_i(h_i) = \frac{n_i + 1}{2\pi} (\mathbf{n} \cdot h_i)^{n_i} \approx \mu_i G(h_i, \mathbf{n}, \lambda_i'),
\]

where \( n_i \) is the Phong exponent, \( \lambda_i' = n_i \) and \( \mu_i' = \frac{n_i + 1}{2\pi} \) for this model. Other bell-shaped parametric NDFs (e.g. Beckmann distribution) are also approximated with an SG analytically. Using spherical warping, the specular BRDF is approximated as

\[
\rho(y, \psi, \omega) = M_i(\psi, \omega) \mu_i' G(\alpha, \frac{y_i}{\mu(\psi, \omega)}, \lambda_i'),
\]

where \( \lambda_i' = 2(\mathbf{n} \cdot \psi_i) - \psi_i \), and \( \alpha = \lambda_i' \frac{\lambda_i}{\nu(\psi, \omega)} \). Finally, we obtain the following SG approximation:

\[
w_i(\omega, \alpha, 0) \approx \mu_i G(\alpha, \frac{y_i}{\mu(\psi, \omega)}, \lambda_i'),
\]

where \( \mu_i = M_i(y_i', \alpha, \nu(\psi, \omega), 0) \). To compute this approximation, the view direction \( \psi_i \), surface normal \( \mathbf{n}_i \), reflectance \( w_i \), and BRDF parameters (e.g. \( n_i \)) are necessary. For deferred shading pipelines, \( \mathbf{n}_i, R_i \) and BRDF parameters are given by the G-buffer. The view direction \( \psi_i \) is computed with the camera position and the position \( y_i \) which can be given by the position buffer in the G-buffer. Instead of the position buffer, \( y_i \) can be calculated using the camera projection matrix and the depth buffer for memory reduction. Therefore, the parametric specular lobe at each pixel is inexpensively approximated with an SG on-the-fly for real-time
applications with dynamic scenes. ASGs are also usable in the same manner [XSD*13].

Appendix C: Product Integrals of SGs

Spherical Gaussians. The product integral of two SGs is derived in [TS06] as

\[ \int_{S^2} G(\omega, \xi_1, \lambda_1)G(\omega, \xi_2, \lambda_2)d\omega = \frac{4\pi \sinh(r)}{\exp(\lambda_1 + \lambda_2)r}, \]

where \( r = ||\lambda_1 \xi_1 + \lambda_2 \xi_2||. \) Since this is not closed in SG basis, Iwasaki et al. [IDN12] introduced the following approximation:

\[ \int_{S^2} G(\omega, \xi_1, \lambda_1)G(\omega, \xi_2, \lambda_2)d\omega \approx \frac{2\pi G(\xi_1, \lambda_1, 0)}{\lambda_1 + \lambda_2}. \]

This approximation error is small if \( \lambda_1 \) or \( \lambda_2 \) are large.

Anisotropic spherical Gaussians. The approximate product integral of an ASG and SG is closed in ASG basis as follows:

\[ \int_{S^2} \tilde{G}(\omega, \xi_1, \xi_2, \lambda_1, \lambda_2)G(\omega, \xi_3, \lambda_3)d\omega \approx \frac{2\pi \tilde{G}(\xi_1, \xi_2, \xi_3, \lambda_1, \lambda_2, \lambda_3)}{\sqrt{(2\lambda_1 + \lambda_2)(2\lambda_2 + \lambda_3)}}. \]

The approximate integral of an ASG is given by

\[ \int_{S^2} \tilde{G}(\omega, \xi_1, \xi_2, \lambda_1, \lambda_2)d\omega \approx \frac{\pi}{\lambda_3} \exp\left(-\frac{\lambda_1}{\lambda_3}\right) \frac{F(t)}{2\lambda_3} \left(F\left(t + \frac{t}{\lambda_3}\right) \right), \]

where \( \lambda_3 \geq \lambda_1, \lambda_2, \) \( t = \lambda_3 - \lambda_1, \) and \( F(t) = \int_0^{\infty} \exp\left(-\frac{\lambda_1}{\lambda_3}\right) \phi \) which is approximately obtained with a precomputed 1D texture or an analytical rational approximation. This paper employs the analytical approximation. ASG definition can be equivalently written in algebraic form:

\[ \tilde{G}(\omega, \lambda) = \exp\left(\omega^T A \omega\right), \]

where \( A \) is a \( 3 \times 3 \) symmetric matrix, and \( \xi_1 \) is the eigenvector with the smallest eigenvalue. If ASG lobes are not low frequency, the product of two ASGs is approximated with an ASG as the following equation:

\[ \tilde{G}(\omega, \lambda_1, \lambda_2) \approx C(\xi_3, 3, \xi_3, 1, \xi_3, 2) \tilde{G}(\omega, \lambda_3), \]

where \( \lambda_3 = \lambda_1 + \lambda_2, \xi_3, 3 \) is the eigenvector with the smallest eigenvalue of \( \lambda_3, \) and \( C(\xi_3, 3, \xi_3, 1, \xi_3, 2) = \max(\xi_3, 3 \cdot \xi_3, 1, 0) \max(\xi_3, 2 \cdot \xi_3, 0). \) Thus, the product integral of two ASGs is approximately obtained as follows:

\[ \int_{S^2} \tilde{G}(\omega, \lambda_1)\tilde{G}(\omega, \lambda_2)d\omega \approx \frac{\pi}{\lambda_3} \exp\left(-\frac{\lambda_1}{\lambda_3}\right) \frac{F(t)}{2\lambda_3} \left(F\left(t + \frac{t}{\lambda_3}\right) \right), \]

where \( \lambda_1, \lambda_2, \lambda_3 \) are eigenvalues of \( \lambda_3, \) and \( \lambda_1 \geq \lambda_2 \geq \lambda_3. \)

Appendix D: Diffuse Lobe Similarity

This section reformulates the normal-aware weighting function based on the diffuse lobe similarity. For Lambertian surfaces, the diffuse lobe is given as \( a_i(\omega) \propto \max(\mathbf{n} \cdot \omega, 0). \) As described in Eq. (9) in the paper, the smoothing kernel is an SG with lobe sharpness \( \kappa. \) If \( \kappa = \infty, c_i(\omega) = a_i(\omega) \) is obtained. Thus the similarity of two diffuse lobes is given by

\[ \kappa_{ij} = \frac{1}{2\pi} \int_{S^2} \max(\mathbf{n} \cdot \omega, 0) \max(\mathbf{n} \cdot \omega, 0)d\omega = \frac{\sin \phi_{ij} + (\pi - \phi_{ij}) \cos \phi_{ij}}{\pi}, \]

where \( \phi_{ij} = \arccos(\mathbf{n} \cdot \mathbf{n}). \) As shown in Fig. D.1, this diffuse lobe similarity is bell-shaped and analogous to an ASG as \( \kappa_{ij} \approx G(\mathbf{n}_i, \mathbf{n}_j, \lambda_c), \) where the lobe sharpness \( \lambda_c = 0.9426 \) is obtained by using the least squares method (the integrated squared fitting error: 0.00245). Using Eq. (C.1), this paper extends the above similarity for arbitrary \( \kappa \) as follows:

\[ \kappa_{ij} \approx G(\mathbf{n}_i, \mathbf{n}_j, \frac{\lambda_c \kappa}{2\lambda_c + \kappa}). \]

Therefore, an SG can be used for the weighting function as

\[ w_\omega(i, j) = \psi_{ij} \approx G(\mathbf{n}_i, \mathbf{n}_j, \frac{1}{\sigma^2}) = w_n(i, j). \]
where $\sigma^2_n = \frac{2\lambda c + \kappa}{\beta \lambda c}$. Hence, the normal-aware weighting function is the special case of our lobe-aware weighting function for diffuse surfaces. This reformulation provides unified parameters $\kappa$ and $\beta$ between diffuse and specular surfaces, and reduces material-dependent parameter tuning. Fig. D.2 shows the spatio-temporal filtering using $\beta = 20$ and $\kappa = 100$ for diffuse surfaces.

Appendix E: Code Optimization

Our approximated lobe-aware weighting function (i.e. Eq. (12) in the paper) can be rewritten as follows:

$$
\omega_{\omega}(i, j) \approx \left( \frac{2\sqrt{\lambda_i \lambda_j}}{\lambda_i + \lambda_j} \right)^\beta G\left( \xi_i, \xi_j; \frac{\beta \lambda_i \lambda_j}{\lambda_i + \lambda_j} \right) = \exp \left( \frac{\beta}{2} \log \frac{4u}{v} \right) + \frac{\beta u}{v} \left( \frac{\xi_i}{\xi_j} - 1 \right),
$$

where $u = \lambda_i \lambda_j$ and $v = \lambda_i + \lambda_j$. This is used in our implementation for code optimization, because it has only an additional mathematical function (i.e. log) compared to the conventional normal-aware weighting function. In addition, exp and log functions can be compiled using faster intrinsics.

References


[TS06] TSAI Y.-T., SHIH Z.-C.: All-frequency precomputed radiance transfer using spherical radial basis functions and clustered tensor approximation. ACM Trans. Graph. 25, 3 (2006), 967–976. 2
