Supplementary Material: Differentiable Rendering for Specular Materials via Projected Specular Manifolds

RUIZENG LI, University of Science and Technology of China, China and Laoshan Laboratory, China
PEIQI WANG, University of Science and Technology of China, China
BEIBEI WANG*, School of Intelligence Science and Technology, Nanjing University, China
LIGANG LIU*, University of Science and Technology of China, China and Laoshan Laboratory, China

ACM Reference Format:
Ruizeng Li, Peiqi Wang, Beibei Wang, and Ligang Liu. 2026. Supplementary Material: Differentiable Rendering for Specular Materials via Projected Specular Manifolds. *ACM Trans. Graph.* 45, 6, Article 198 (December 2026), 4 pages. https://doi.org/10.1145/3842514

1 Velocity Computation on the PSM

We consider a light path $\overline{\boldsymbol{x}} = \boldsymbol{x}_0\boldsymbol{x}_1\dots\boldsymbol{x}_{n+1}$, where $\boldsymbol{x}_0$ is on a surface that is not specular or glossy, $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$ are on specular or glossy objects, and $\boldsymbol{x}_{n+1}$ lies on an object that forms the PSM. To compute the velocity of $\boldsymbol{x}_1$ induced by the motion of $\boldsymbol{x}_{n+1}$, we need to consider the implicit mapping from $\boldsymbol{x}_{n+1}$ to $\boldsymbol{x}_1$:

$$\boldsymbol{h}_i(\boldsymbol{x}_{i-1}, \boldsymbol{x}_i, \boldsymbol{x}_{i+1}, \theta) = \boldsymbol{h}_i^0, \ i = 1, \dots, n, \tag{1}$$

where $\boldsymbol{h}_i^0$ is the static local half-vector. Since the half-vector constraint is two-dimensional, we use $\hat{\boldsymbol{h}}$ to denote the first two coordinates of $\boldsymbol{h}$. With the material-form mapping $\mathsf{X}(\cdot, \theta)$, we have

$$\hat{\boldsymbol{h}}_i(\mathsf{X}(\boldsymbol{p}_{i-1}, \theta), \mathsf{X}(\boldsymbol{p}_i, \theta), \mathsf{X}(\boldsymbol{p}_{i+1}, \theta), \theta) = \hat{\boldsymbol{h}}_i^0, \ i = 1, \dots, n. \tag{2}$$

The mapping $\mathsf{X}(\cdot, \theta)$ can be designed as preserving the barycentric coordinates $\boldsymbol{u} \in \mathbb{R}^2$:

$$\begin{aligned} \boldsymbol{p} &= U(\boldsymbol{u}, \theta_0), \\ \boldsymbol{x} &= \mathsf{X}(\boldsymbol{p}, \theta) = U(\boldsymbol{u}, \theta) = U(U^{-1}(\boldsymbol{p}, \theta_0), \theta), \end{aligned} \tag{3}$$

where $U(\theta)$ is a matrix in $\mathbb{R}^{3\times 2}$ defined by the triangle containing $\boldsymbol{x}$. Then we can differentiate Eqn. (2) as:

$$\begin{aligned} &\frac{\partial \hat{\boldsymbol{h}}_i}{\partial \boldsymbol{x}_{i-1}}(U'_{i-1}\boldsymbol{u}_{i-1} + U_{i-1}\boldsymbol{u}'_{i-1}) + \frac{\partial \hat{\boldsymbol{h}}_i}{\partial \boldsymbol{x}_i}(U'_i\boldsymbol{u}_i + U_i\boldsymbol{u}'_i) \\ &\quad + \frac{\partial \hat{\boldsymbol{h}}_i}{\partial \boldsymbol{x}_{i+1}}(U'_{i+1}\boldsymbol{u}_{i+1} + U_{i+1}\boldsymbol{u}'_{i+1}) + \frac{\partial \hat{\boldsymbol{h}}_i}{\partial \theta} = \boldsymbol{0}, \end{aligned} \tag{4}$$

where we use U' and $\boldsymbol{u}'$ to express the derivatives with respect to θ for brevity. The velocity of $\boldsymbol{u}_1$ is induced by the motion of other objects under the specular constraints, and must be computed by Eqn. 4, which requires solving a linear system:

$$\begin{pmatrix} \frac{\partial \hat{\boldsymbol{h}}_1}{\partial \boldsymbol{x}_1}U_1 & \frac{\partial \hat{\boldsymbol{h}}_1}{\partial \boldsymbol{x}_2}U_2 & & & \\ \frac{\partial \hat{\boldsymbol{h}}_2}{\partial \boldsymbol{x}_1}U_1 & \frac{\partial \hat{\boldsymbol{h}}_2}{\partial \boldsymbol{x}_2}U_2 & \frac{\partial \hat{\boldsymbol{h}}_2}{\partial \boldsymbol{x}_3}U_3 & & \\ & \frac{\partial \hat{\boldsymbol{h}}_3}{\partial \boldsymbol{x}_2}U_2 & \frac{\partial \hat{\boldsymbol{h}}_3}{\partial \boldsymbol{x}_3}U_3 & \frac{\partial \hat{\boldsymbol{h}}_3}{\partial \boldsymbol{x}_4}U_4 & \\ & & & \cdots & \\ & & & \frac{\partial \hat{\boldsymbol{h}}_n}{\partial \boldsymbol{x}_{n-1}}U_{n-1} & \frac{\partial \hat{\boldsymbol{h}}_n}{\partial \boldsymbol{x}_n}U_n \end{pmatrix} \begin{pmatrix} \boldsymbol{u}'_1 \\ \boldsymbol{u}'_2 \\ \boldsymbol{u}'_3 \\ \dots \\ \boldsymbol{u}'_n \end{pmatrix} = R, \tag{5}$$

where $\boldsymbol{u}'_{i+1} = 0$. R is computed by automatic differentiation. We rely on Eigen [Guennebaud et al. 2010] to solve this system. Then we can get the velocity $\boldsymbol{v}_1 = U_1\boldsymbol{u}'_1$.

We also need to compute the divergence $\nabla \cdot \boldsymbol{v}_1$ with respect to $\boldsymbol{x}_1$. We only need the derivatives with respect to the two variables $\boldsymbol{u}_1$, since the divergence of $\boldsymbol{v}_1(\boldsymbol{x}_1) = \boldsymbol{v}_1(U_1\boldsymbol{u}_1)$ can be computed by the chain rule. For $\boldsymbol{u}_1 = (u_1, u_2)^T$, we have

$$\frac{\partial}{\partial u_k}\left(\sum_{j=i-1}^{i+1}\frac{\partial \hat{\boldsymbol{h}}_i}{\partial \boldsymbol{x}_j}(U'_j\boldsymbol{u}_j + U_j\boldsymbol{u}'_j)\right) = \boldsymbol{0}, \ k = 1, 2. \tag{6}$$

We need to consider the mapping $\boldsymbol{u}_i = \mathsf{rayIntersect}(\boldsymbol{u}_1)$ representing that $\boldsymbol{u}_i$ is given by the intersection of the object and the specular light path beginning from $\boldsymbol{u}_1$. $\partial \boldsymbol{u}_i/\partial \boldsymbol{u}_1$ can be computed recursively:

$$\frac{\partial \boldsymbol{u}_i}{\partial \boldsymbol{u}_1} = \prod_{j=i-1}^{1}\frac{\partial \boldsymbol{u}_{j+1}}{\partial \boldsymbol{u}_j}. \tag{7}$$

Then we get

$$\sum_{j=i-1}^{i+1}\left(\frac{\partial^2 \hat{\boldsymbol{h}}_i}{\partial \boldsymbol{x}_j \partial u_k}(U'_j\boldsymbol{u}_j + U_j\boldsymbol{u}'_j) + \frac{\partial \hat{\boldsymbol{h}}_i}{\partial \boldsymbol{x}_j}(U'_j\frac{\partial \boldsymbol{u}_j}{\partial u_k} + U_j\frac{\partial \boldsymbol{u}'_j}{\partial u_k})\right) = \boldsymbol{0}. \tag{8}$$

Note that $\partial^2 \hat{\boldsymbol{h}}_i/(\partial \boldsymbol{x}_j \partial u_k)$ is still a 2×3 matrix. We write $\boldsymbol{x}_j = (x_{j,1}, x_{j,2}, x_{j,3})^T$, then

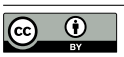
$$\frac{\partial^2 \hat{\boldsymbol{h}}_i}{\partial \boldsymbol{x}_j \partial u_k} = \sum_{m=i-1}^{i+1}\sum_{n=1}^{3}\frac{\partial^2 \hat{\boldsymbol{h}}_i}{\partial \boldsymbol{x}_j \partial x_{m,n}}\frac{\partial x_{m,n}}{\partial u_k}. \tag{9}$$

$\partial^2 \hat{\boldsymbol{h}}_i/(\partial \boldsymbol{x}_j \partial x_{m,n})$ can be computed explicitly, and $\partial x_{m,n}/\partial u_k$ can be computed by $\partial \boldsymbol{u}_i/\partial \boldsymbol{u}_1$. Eqn. (8) also has a recursive form like Eqn. (4), and we can construct a linear system similar to Eqn. (5). Then for a specular light path, after each intersection at $\boldsymbol{x}_j$ we store the matrix $\partial \boldsymbol{u}_j/\partial \boldsymbol{u}_{j-1}$ along with the matrices of $\hat{\boldsymbol{h}}_i$ for the computation of the divergence.

In practice, we find that relying on automatic differentiation to compute the divergence is also feasible, which reduces the complex-

*Corresponding authors.

Authors' Contact Information: Ruizeng Li, University of Science and Technology of China, China and Laoshan Laboratory, China, pb17061297@mail.ustc.edu.cn; Peiqi Wang, University of Science and Technology of China, China, pecky@mail.ustc.edu.cn; Beibei Wang, School of Intelligence Science and Technology, Nanjing University, China, beibei.wang@nju.edu.cn; Ligang Liu, University of Science and Technology of China, China and Laoshan Laboratory, China, lgliu@ustc.edu.cn.



ACM 1557-7368/2026/12-ART198
https://doi.org/10.1145/3842514

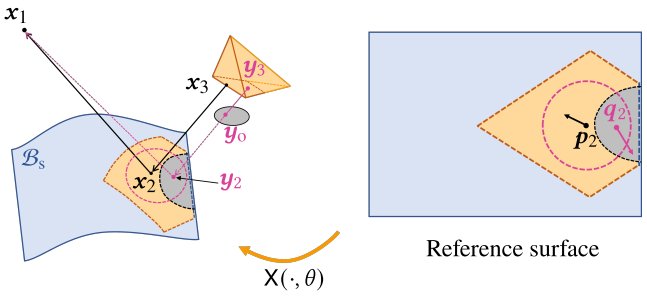


Fig. 1. Velocity computation for visibility boundaries on the PSM.

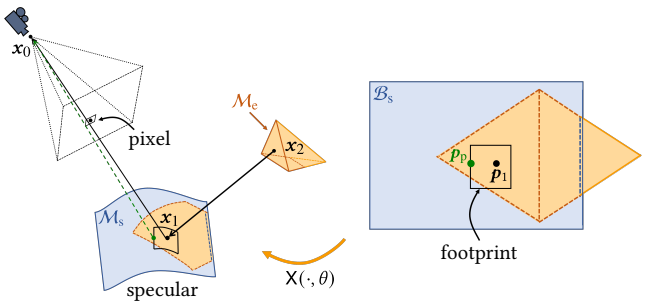


Fig. 2. Velocity computation for pixel boundaries on the PSM.

ity of recursive computation. $\boldsymbol{v}_1$ can be expressed as:

$$\boldsymbol{v}_1 = \sum_{i=1}^{n+1} K_i U_i' \boldsymbol{u}_i, \tag{10}$$

where K_i is the corresponding matrix calculated by Eqn. (5). For the matrix $K_i = K_i(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_i)$, we have

$$\frac{\partial K_i}{\partial \boldsymbol{u}_1} = \sum_{j=1}^{i} \frac{\partial K_i}{\partial \boldsymbol{u}_j} \frac{\partial \boldsymbol{u}_j}{\partial \boldsymbol{u}_1}. \tag{11}$$

$\partial K_i / \partial \boldsymbol{u}_j$ can be computed by automatic differentiation.

For short specular paths, we find that the computation above does not introduce significant computational overhead. However, the cost increases with path length. Designing more efficient velocity computation methods is left as future work.

Visibility boundary. To compute the velocity from visibility boundaries, we sample around $\boldsymbol{x}_2$ on the PSM to find the occluder. For a sampled point $\boldsymbol{q}_2$ corresponding to a occluded light path, we need to compute both $\boldsymbol{v}_\mathrm{m}$ and $\boldsymbol{v}_\mathrm{vis}$ on the PSM. In Fig. 1, we take the occluder between $\boldsymbol{x}_2$ and $\boldsymbol{x}_3$ as an example. The specular path intersects the occluder at $\boldsymbol{y}_\mathrm{o}$, with the sampled material point $\boldsymbol{q}_2$ and $\boldsymbol{y}_2 = X(\boldsymbol{q}_2, \theta)$. Then the velocity from the occluder can be computed under the constraint:

$$\boldsymbol{h}(\boldsymbol{x}_1, \boldsymbol{y}_2, \boldsymbol{y}_\mathrm{o}, \theta) = \boldsymbol{h}. \tag{12}$$

This is the velocity of the occluder under the constraints imposed by the PSM. Meanwhile, the velocity from manifold boundaries is computed by

$$\boldsymbol{h}(\boldsymbol{x}_0, \boldsymbol{y}_2, \boldsymbol{y}_3, \theta) = \boldsymbol{h}. \tag{13}$$

We can then compute the velocities using matrix operations as described above.

Pixel boundary. For the velocity of the pixel boundary, we find the point closest to $\boldsymbol{x}_0\boldsymbol{x}_1$ on the filter support and project it onto $\mathcal{M}_\mathrm{s}$ to obtain $\boldsymbol{x}_\mathrm{p}$ and the corresponding material point $\boldsymbol{p}_\mathrm{p}$ (Fig. 2). If such intersection exists, B_p is computed as the distance between $\boldsymbol{p}_\mathrm{p}$ and $\boldsymbol{p}_1$. Otherwise B_p equals to $+\infty$. The velocity field caused by footprint is calculated under the fixed direction constraint:

$$\omega(\boldsymbol{x}_0, \boldsymbol{x}_1) = \mathbf{const}. \tag{14}$$

ω is the normalized direction. Then we can compute the velocity by performing the matrix operations as above. Note that in this way, we still preserve the material-form reparameterization on $\mathcal{B}_\mathrm{s}$, with the need of handling the discontinuity induced by the pixel boundary.

2 Sampling-based Solution

We design a sampling-based method for manifold boundaries, where the situation is different from visibility boundaries. For the visibility boundary, the effective integration domain is the complement of $\mathcal{B}_\mathrm{e}$ and the occluded region. Therefore, previous methods [Bangaru et al. 2020; Xu et al. 2023] define two velocity fields on $\mathcal{B}_\mathrm{e}^\mathrm{wa}$ and $\mathcal{B}_\mathrm{e} \backslash \mathcal{B}_\mathrm{e}^\mathrm{wa}$ and smoothly interpolate them near the visibility boundary. However, in our situation the PSM can be regarded as the intersection of two objects (see Fig. 5 in the main paper). Therefore, on the integrand region, i.e. the PSM, we can obtain two velocity fields $\boldsymbol{v}_\mathrm{s} = \boldsymbol{0}$ and $\boldsymbol{v}_\mathrm{e}$, where $\boldsymbol{v}_\mathrm{e}$ results from mapping the velocity field on $\mathcal{M}_\mathrm{e}$ to $\mathcal{B}_\mathrm{s}$ under geometric constraints, and $\boldsymbol{v}_\mathrm{e} = \boldsymbol{v}_\mathrm{e}^\partial$ on $\Delta\mathcal{S}_\mathrm{p}^\mathrm{e}$. We need to interpolate the two velocity fields to satisfy Eqn. (26) in the main paper. By extending the design of the warped-area sampling (WAS) method, we can construct the $\boldsymbol{v}_\mathrm{m}$ as:

$$\begin{aligned}
\boldsymbol{v}_\mathrm{m}(\boldsymbol{p}) &= \frac{\int_{\mathcal{S}_\mathrm{p}} [\mathrm{w}_\mathrm{s}(\boldsymbol{q};\boldsymbol{p})\boldsymbol{v}_\mathrm{s}(\boldsymbol{q}) + \mathrm{w}_\mathrm{e}(\boldsymbol{q};\boldsymbol{p})\boldsymbol{v}_\mathrm{e}(\boldsymbol{q})] dA(\boldsymbol{q})}{\int_{\mathcal{S}_\mathrm{p}} [\mathrm{w}_\mathrm{s}(\boldsymbol{q};\boldsymbol{p}) + \mathrm{w}_\mathrm{e}(\boldsymbol{q};\boldsymbol{p})] dA(\boldsymbol{q})}, \\
\mathrm{w}_\mathrm{s}(\boldsymbol{q};\boldsymbol{p}) &= \frac{1}{d(\boldsymbol{q};\boldsymbol{p}) + B(\boldsymbol{q}, \Delta\mathcal{S}_\mathrm{p}^\mathrm{s})}, \\
\mathrm{w}_\mathrm{e}(\boldsymbol{q};\boldsymbol{p}) &= \frac{1}{d(\boldsymbol{q};\boldsymbol{p}) + B(\boldsymbol{q}, \Delta\mathcal{S}_\mathrm{p}^\mathrm{e})},
\end{aligned} \tag{15}$$

where $B(\boldsymbol{q}, \Delta\mathcal{S}_\mathrm{p}^\mathrm{s})$ and $B(\boldsymbol{q}, \Delta\mathcal{S}_\mathrm{p}^\mathrm{e})$ are boundary-test functions for $\mathcal{B}_\mathrm{s}$ and $\mathcal{B}_\mathrm{e}$, respectively.

The sampling-based solution can be viewed as a direct generalization of WAS. w_s and w_e are spatially varying kernels that behave like delta functions near corresponding boundaries, making velocity conditions satisfied in theory. In practice, however, since w_s and w_e can only approximate delta functions, the conditions are satisfied only approximately.

Sampling on PSM. For a material point $\boldsymbol{p} \in \mathcal{B}_\mathrm{s}$, we sample points $\boldsymbol{q}$ around $\boldsymbol{p}$ and trace reflected or refracted rays according to the geometric constraints. If a ray hits $\mathcal{M}_\mathrm{e}$, we calculate corresponding velocities and spatially varying kernels. $\boldsymbol{v}_\mathrm{m}$ is computed by the Monte Carlo estimate:

$$\langle \boldsymbol{v}_\mathrm{m} \rangle = \frac{\langle \mathrm{w}_\mathrm{s}\boldsymbol{v}_\mathrm{s} + \mathrm{w}_\mathrm{e}\boldsymbol{v}_\mathrm{e} \rangle}{\langle \mathrm{w}_\mathrm{s} + \mathrm{w}_\mathrm{e} \rangle}. \tag{16}$$

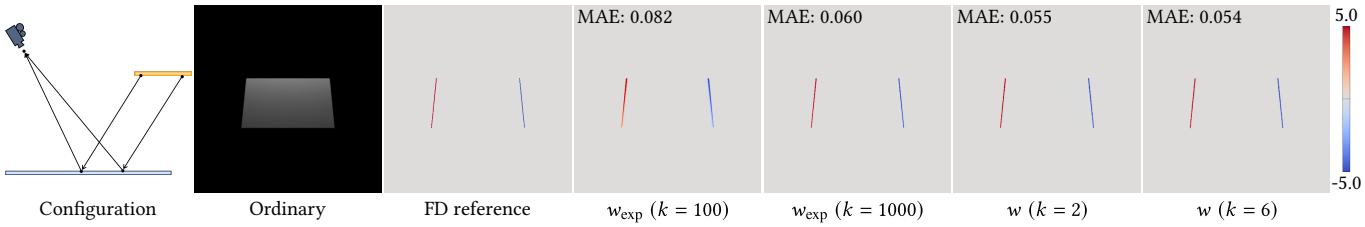


Fig. 3. Comparisons between different weight functions.

For more details, we refer to Xu et al. [2023]'s work.

Slow Convergence. As stated in previous works, the estimator in Eqn. (16) is biased but consistent because the computation of the expectation does not satisfy $\mathbb{E}[1/f] = 1/\mathbb{E}[f]$. In our case, we find that interpolation leads to slower convergence and introduces larger errors. As shown in Fig. 15 in the main paper, the sampling-based method requires a large number of auxiliary samples to converge to a relatively correct result, which is time-consuming and not useful. Consequently, this motivates us to seek the direct-interpolation-based approach to compute the velocity field more efficiently.

3 Weight Functions and Error Analysis

Other weight functions. Our weight function for velocity interpolation is designed as

$$w(\boldsymbol{p}) := \frac{B^k(\boldsymbol{p}, \Delta\mathcal{S}^{\text{s}}_{\text{p}})}{B^k(\boldsymbol{p}, \Delta\mathcal{S}^{\text{s}}_{\text{p}}) + B^k(\boldsymbol{p}, \Delta\mathcal{S}^{\text{e}}_{\text{p}})}, \tag{17}$$

for the two-bounce case. We test another type of weight function:

$$w_{\text{exp}}(\boldsymbol{p}) := \frac{e^{-kB(\boldsymbol{p}, \Delta\mathcal{S}^{\text{e}}_{\text{p}})}}{e^{-kB(\boldsymbol{p}, \Delta\mathcal{S}^{\text{e}}_{\text{p}})} + e^{-kB(\boldsymbol{p}, \Delta\mathcal{S}^{\text{s}}_{\text{p}})}}, \tag{18}$$

which approximates 1 as $B(\boldsymbol{p}, \Delta\mathcal{S}^{\text{e}}_{\text{p}})) = 0$ with a large k. In Fig. 3, we make comparisons about different weight functions with different k. Our choice w, which satisfies $w = 0$ as $B(\boldsymbol{p}, \Delta\mathcal{S}^{\text{e}}_{\text{p}})) = 0$ in theory, introduces less error. Since there are infinitely many weight functions satisfying the boundary conditions, we choose our w that performs well in practice, and leave the selection of the optimal weight function for future work.

Error Analysis. The error induced by our direct-interpolation-based method can be computed as (see Eqn. (24) in the main paper):

$$\int_{\Delta\mathcal{S}_{\text{p}}} |\widehat{F}(\boldsymbol{v}_{\text{m}} - \boldsymbol{v}^{\partial}) \cdot \boldsymbol{n}^{\partial}| dl(\boldsymbol{p}_2), \tag{19}$$

which describes the error between the interpolated velocity and $\boldsymbol{v}^{\partial}$ on $\Delta\mathcal{S}_{\text{p}}$. Due to the discontinuity of boundary conditions, direct interpolation introduces errors near the intersections of different boundaries. Designing an unbiased method is left for future work.

4 More Explanations

4.1 Boundary-test Function

For the two-bounce case, we state that $\boldsymbol{p}$ belongs to manifold boundaries is equivalent to $B(\boldsymbol{p}, \Delta\mathcal{S}^{\text{e}}_{\text{p}})B(\boldsymbol{p}, \Delta\mathcal{S}^{\text{s}}_{\text{p}}) = 0$. The proof is as follows:

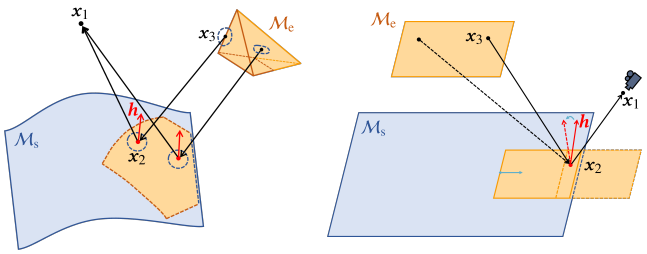


Fig. 4. Left: proof about the properties of boundary-test functions. Right: continuity of the integral on PSM.

If $\boldsymbol{p}$ belongs to manifold boundaries but $B(\boldsymbol{p}, \Delta\mathcal{S}^{\text{e}}_{\text{p}}) \cdot B(\boldsymbol{p}, \Delta\mathcal{S}^{\text{s}}_{\text{p}}) \neq 0$, then there exists a small neighborhood $\mathcal{N}(\boldsymbol{p})$ around $\boldsymbol{p}$ such that for each $\boldsymbol{q} \in \mathcal{N}(\boldsymbol{p})$, $B(\boldsymbol{q}, \Delta\mathcal{S}^{\text{e}}_{\text{p}})B(\boldsymbol{q}, \Delta\mathcal{S}^{\text{s}}_{\text{p}}) \neq 0$ (Fig. 4). Consider the specular path $\boldsymbol{x}_0\boldsymbol{x}_1\boldsymbol{x}_2\boldsymbol{x}_3$ corresponding to $\boldsymbol{p}$, where $\boldsymbol{x}_2 = \mathsf{X}(\boldsymbol{p}, \theta)$. By continuity of the constraint function and definitions of boundary-test functions, there exist neighborhoods around $\boldsymbol{x}_2 \in \mathcal{M}_{\text{s}}$ and $\boldsymbol{x}_3 \in \mathcal{M}_{\text{e}}$, such that they do not intersect the discontinuity curves on $\mathcal{M}_{\text{s}}$ and $\mathcal{M}_{\text{e}}$, respectively. Then $\boldsymbol{p}$ is contained in the interior of $\mathcal{S}_{\text{p}}$ and not located on manifold boundaries. This leads to a contradiction. Therefore, $B(\boldsymbol{p}, \Delta\mathcal{S}^{\text{e}}_{\text{p}})B(\boldsymbol{p}, \Delta\mathcal{S}^{\text{s}}_{\text{p}}) = 0$.

On the other hand, if $B(\boldsymbol{p}, \Delta\mathcal{S}^{\text{e}}_{\text{p}})B(\boldsymbol{p}, \Delta\mathcal{S}^{\text{s}}_{\text{p}}) = 0$, without loss of generality we assume $B(\boldsymbol{p}, \Delta\mathcal{S}^{\text{e}}_{\text{p}}) = 0$. Then $\boldsymbol{x}_3$ belongs to object or normal boundaries on $\mathcal{M}_{\text{e}}$. By the definition of manifold boundaries, $\boldsymbol{p}$ is located on them.

4.2 Continuity of the Integral on PSM

For the two-bounce case with a glossy BSDF, we have

$$I = \int_{\mathcal{H}} \underbrace{\int_{\mathcal{B}_{\text{s}}} D(\boldsymbol{h}) \widehat{F}_{\text{h}} dA(\boldsymbol{p}_2)}_{=: F_{\text{int}}(\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{h}, \theta)} dA(\boldsymbol{h}). \tag{20}$$

F_{int} is continuous on $\mathcal{H}$. This is because the area of the effective integration region varies smoothly with $\boldsymbol{h}$ (Fig. 4). Therefore, F_{int} is a continuous function of $\boldsymbol{h}$ and it does not jump to zero discontinuously.

5 Relationship with EPSM

Our definition of PSM is different from the EPSM. EPSM is defined as a higher-dimensional manifold that couples the path with scene parameters, where different constraints including material-form reparameterization can be imposed on each vertex. In contrast, our PSM focuses on the specular path segment, specifically applying

material-form reparameterization to the first specular object and projecting the constraint manifold onto this surface. This facilitates the explicit computation of the boundary integral. In Sec. 6.3 in the main paper, we show that our PSM can be extended to support more types of constraints.

References

Sai Praveen Bangaru, Tzu-Mao Li, and Frédo Durand. 2020. Unbiased warped-area sampling for differentiable rendering. *ACM Trans. Graph.* 39, 6, Article 245 (Nov. 2020), 18 pages.

Gaël Guennebaud, Benoît Jacob, et al. 2010. Eigen. https://libeigen.gitlab.io.

Peiyu Xu, Sai Bangaru, Tzu-Mao Li, and Shuang Zhao. 2023. Warped-Area Reparameterization of Differential Path Integrals. *ACM Trans. Graph.* 42, 6, Article 213 (Dec. 2023), 18 pages.