Supplemental Document: Multirate Shading with Piecewise Interpolatory Approximation

In the main paper, we have analyzed piecewise interpolation errors on general convex polygons and introduced methods to compute subdivision parameters for several shading functions including Lambertian, Blinn-Phong and Microfacet BRDF. The full derivation is lengthy. In this supplemental document, we present a complete derivation with more details and proofs to help implement our proposed approach.

1 ERROR ESTIMATION FOR SHADING FUNCTIONS

1.1 Error of piecewise linear interpolation on a convex polygon

Let $P \in \mathbb{R}^2$ be a convex polygon with vertices $v_1, v_2, ..., v_m$. Suppose we have a subdivision operator T that uniformly subdivides P into a set of N convex sub-polygons $\mathcal{P} = \{P_i\}_{i=1}^{N}$, each of convex sub-polygons in $\mathcal{P}$ are composed by m vertices noted as $\Theta_i = \{v_{i_k}\}_{k=1}^{m}$, where v_{i_k} is one vertex in the set of all M vertices $\mathcal{V} = \{v_k\}_{k=1}^{M}$ of subdivision surface.

Given these subdivided convex sub-polygons, a continuous function $f \in C(P) : \mathbb{R}^2 \to \mathbb{R}$ defined on convex polygon P can be piecewise linearly approximated by a set of values computed at the vertices of subdivided polygons, $\mathcal{F} = \{f(v_k)\}_{k=1}^{M}$. Specifically, for one point $v(x, y) \in P$, the function $f(x, y)$ can be interpolated as:

$$f(x,y) \approx \sum_{i}^{N} \mu_i(x,y) L_{\Theta_i} f(x,y) = \sum_{i}^{N} \mu_i(x,y) \sum_{k=1}^{m} f(v_{i_k}) \lambda_{i_k}(x,y) \tag{1}$$

where $\mu_i(x, y)$ is a discriminant function for P_i where $(x, y) \in P_i$, $\mu_i(x, y) = 1$, otherwise $\mu_i(x, y) = 0$. Meanwhile L_{Θ_i} is a linear interpolation operator that interpolates the values sampled from f at the vertex set $\Theta_i = \{v_{i_k}\}_{k=1}^{m}$ of the convex sub-polygon P_i, and $\lambda_{i_k}(x, y)$ is the linear interpolation coefficient on the convex sub-polygon P_i (e.g., barycentric coordinate) which satisfies Lagrange condition $\sum_{k=1}^{m} \lambda_{i_k}(x, y) = 1$ and linear precision $v(x, y) = \sum_{k=1}^{m} \lambda_{i_k}(x, y) v_{i_k}$.

Approximating a non-linear function $f(x, y)$ by piecewise linear interpolation will introduce error. The error can be reduced by a finer subdivision with denser sampling points generated. To precisely measure the difference, we define L_∞ norm of interpolation error $e(f)$ on the convex polygon P as follows:

$$\|e(f)\|_{\infty,P} = \sup_{P_i \in \mathcal{P}} \|e(f)\|_{\infty,P_i} = \sup_{P_i \in \mathcal{P}} \|f - L_\Theta f\|_{\infty,P_i} \tag{2}$$

Given that the L_∞ error on the convex polygon P is the maximum L_∞ error among all convex sub-polygons P_i, this error can be regarded as an error function depending on the subdivision operator $T(n)$ where n is a parameter controlling the granularity of the subdivision. To control the error within in a threshold ϵ, we find an optimal granularity of subdivision $T(n)$:

$$\begin{aligned} &\arg\min_{n} \; e(f(x,y), T(n)) \quad \forall (x,y) \in P \\ &\text{s.t. } \|e(f)\|_{\infty,P} \le \epsilon \end{aligned} \tag{3}$$

However, analytical solution for Eq. (3) is intractable, therefore we compute an appropriate parameter n based on the interpolation error bound.

1.1.1 A General Estimation on $T(n)$ and Error Bound. For an arbitrary convex polygon P with m vertices, the L_∞ error bound of linear interpolation can be proved as [Guessab and Schmeisser 2005]

$$\|e(f)\|_{\infty,P} = \|f - L_{\Theta_i} f\|_{\infty,P} \le \frac{(r^{sc})^2}{2} |f|_{2,\infty,P}, \forall f \in C^2(P) \tag{4}$$

where r^{sc} and v^{sc} specify the smallest circle P^{sc} which contains P:

$$P^{sc} =: \{v \in \mathbb{R}^2 : \|v - v^{sc}\| \le r^{sc}\} \quad \forall v \in P \tag{5}$$

and $|f|_{2,\infty,P}$ is the second order L_∞ semi-norm that is defined as follows:

$$|f|_{2,\infty,P} = \|\ |D^2 f|\ \|_{\infty,P} \tag{6}$$

and

$$|D^2 f|(x,y) = \sup_{\xi \in \mathbb{R}^2, \|\xi\|_2 = 1} |D^2_\xi f(x,y)| \tag{7}$$

by which $|D^2 f|(x,y)$ is defined as the supremum of the second derivative of f in the arbitrary direction $\xi = [\xi_x, \xi_y]^T$ for all $(x,y) \in P$.

We now define $t = T(n)$ as a uniform subdivision process that let r_i^{sb}, the radius of circumcircle of subdivided convex polygon P_i (defined as Eq. (5) likewise) be $r_i^{sb} \le \frac{r^{sb}}{n}$ for all $i = 1...N$. The L_∞ piecewise interpolation error bound on the subdivided domain will be declined to:

$$\begin{aligned} \|e(f,t)\|_{\infty,P} &= \sup_{P_i \in \mathcal{P}} \|e(f)\|_{\infty,P_i} \\ &\le \sup_{P_i \in \mathcal{P}} \Big(\frac{(r_i^{ab})^2}{2} |f|_{2,\infty,P_i}\Big) \le \frac{(r^{ab})^2}{2n^2} |f|_{2,\infty,P} \end{aligned} \tag{8}$$

Such inequality provides a conservative solution of Eq. (3), that is

$$n \ge r^{ab} \sqrt{\frac{1}{2\epsilon} |f|_{2,\infty,P}}. \tag{9}$$

Specifically, in the context of computer graphics, we have geometries represented by triangle meshes. Linear interpolation error bound on triangular domain T is studied for a sharper bound [Subbotin 1989; Waldron 1998]. Similarly, we define a subdivision process $t = T(n)$ that evenly reduces the diameter h (the length of the longest edge) of the triangle. We can derive

$$\|e(f,t)\|_{\infty,T} \le \frac{1}{6} \frac{h^2}{n^2} |f|_{2,\infty,T} \quad \forall f \in C^2(T) \tag{10}$$

when the diameters of sub-triangles are all less than $\frac{h}{n}$. Likewise, we conservatively estimate parameter n under an error threshold ϵ as

$$n \ge h \sqrt{\frac{1}{6\epsilon} |f|_{2,\infty,T}}. \tag{11}$$

We now provide the details about how to compute such an error bound, for instance, of a shading function on a triangle surface. Without loss of generality, let us consider the shading process f of points on a 2D triangle T. Fig. (1a) shows such a triangle. For the simplicity of derivation, we set one vertex of the triangle as the original point, and one edge is along the x-axis. In this way, three variables, (a,b,c), are enough to represent three vertices of a triangle as (0,0), $(a,0)$ and (b,c).

A shading process can be regarded as a combination of two sub-functions. The first one is a mapping function g that interpolates attributes from vertices, such as positions, normals, texture coordinates, etc., to the shading point (x,y) on the triangle. To be specific, attributes $\{A_0, A_1, A_2\}$ at vertices of a triangle shown in Fig. (1a) is interpolated using a barycentric mapping as

$$\begin{aligned} g(x,y) =& (-\frac{A_0}{a} + \frac{A_1}{a})x + (\frac{b-a}{ac}A_0 - \frac{b}{ac}A_1 + \frac{A_2}{c})y + A_0 \\ =& Cx + Dy + E \\ =& A \end{aligned} \tag{12}$$

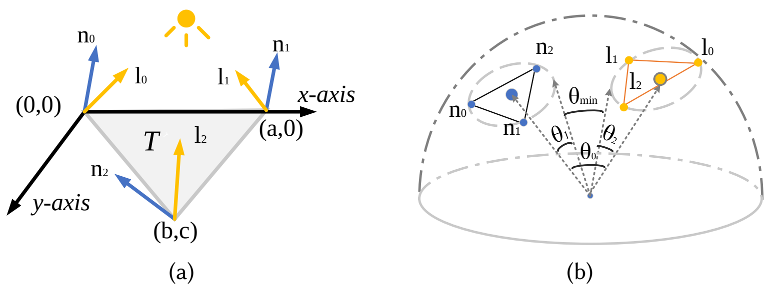

Fig. 1. (a) A vertex of the triangle is fixed on (0,0) in its local 2D coordinate system and one side is fixed along X axis. Factors *a*,*b*,*c* depends on particular shape of a specific triangle. Shading attributes such as **n** and **l** are defined on each of its vertex. (b) Normalized **n** (blue) and normalized **l** (yellow) are distributed on a sphere, and construct two cones. The cosine of $\theta_{\min}$ is a conservative estimation of possible $\max\{\cos\langle \mathbf{n}, \mathbf{l}\rangle\}$.

where $(0, 0)$, $(a, 0)$, (b, c) are coordinates of vertices, and A is the set of interpolated attributes at (x, y), such as normals and light directions.

The second function is the shading function using the attribute A to compute shading values. We denote it as $\hat{f}$, therefore the entire shading process f can be represented as:

$$f(x, y) = \hat{f}(g(x, y)), \ (x, y) \in \ T \tag{13}$$

For the entire shading process, we can further compute the second derivative of it, i.e., $|D^2 f|$, as:

$$|D^2 f| = \rho(H_f) = \rho(H_{\hat{f}(g(x,y))}) = \rho(\begin{bmatrix} C^T \frac{\partial^2 \hat{f}}{\partial A^2} C & D^T \frac{\partial^2 \hat{f}}{\partial A^2} C \\ C^T \frac{\partial^2 \hat{f}}{\partial A^2} D & D^T \frac{\partial^2 \hat{f}}{\partial A^2} D \end{bmatrix}) \tag{14}$$

where H_f is the Hessen matrix of shading function f, and $\rho(H)$ is spectral radius of matrix H.

By plugging Eq. (14) into Eq. (6), we can get the second order L_∞ semi-norm. Using Eq. (10) and Eq. (11), we are able to estimate error bound of shading process f, or vice verse, to compute the subdivision parameter n under a given error threshold.

1.1.2 A Specific Estimation on $T(n)$ and Error Bound. Vector normalization is a fundamental, widely-used operator in shading computations, simple but involves high non-linearity. Performing interpolation to approximate vector normalization may encounter considerable error. On the other hand, due to the complexity of evaluating the second order semi-norms in Eq. (4), direct error analysis using Eq. (10) on vector normalization is impractical at runtime.

To simplify computation, we derive a specific error estimation in vector space. First, w.l.o.g, we consider vector normalization on a triangle T. A vector $\mathbf{w}$ is interpolated by three normalized vectors $\mathbf{w}_0$,$\mathbf{w}_1$ and $\mathbf{w}_2$ at three vertices of T as $\mathbf{w} = \sum_{k=1}^{3} \mathbf{w}_k \lambda_k(x, y)$. Its normalized vector is computed as $\frac{\mathbf{w}}{\|\mathbf{w}\|_2}$. Hence, the L_∞ error of the length between the linear interpolated vector $\mathbf{w}$ and its normalized vector can be computed as follows:

$$\Big\| \, \| \frac{\mathbf{w}}{\|\mathbf{w}\|_2} - \mathbf{w} \|_2 \, \Big\|_\infty \leq \max_T \{1 - \|\mathbf{w}\|_2\}. \tag{15}$$

where $\max\{1-\|\mathbf{w}\|_2\}$ represents the maximum difference between normalized $\mathbf{w}$ and unnormalized $\mathbf{w}$ in their length. Seen as Fig. (2a), three normals construct triangle ΔABC on a unit sphere whose center is O, where AO, BO, CO are three normalized normals. We define h^* is the diameter of ΔABC (we assume h^* is AB w.l.o.g.). P is the center of circumcircle of triangle ΔABC where

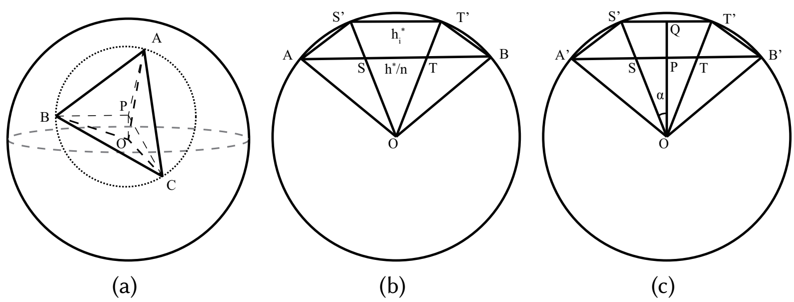

Fig. 2

$AP = BP = CP = R$ is the radius of the circumcircle. Noticing that $OP \perp \Delta ABC$, therefore we can derive that

$$\max_T\{1 - \|L_\Theta \mathbf{w}\|_2\} = 1 - OP = 1 - \sqrt{1 - R^2} = 1 - \sqrt{1 - (\frac{h^*}{2sinC})^2} \tag{16}$$

On the other hand, if P is not in triangle ΔABC (when ΔABC is an obtuse triangle). OP is a conservative but not an accurate solution of $\max\{1 - \|L_\Theta \mathbf{w}\|_2\}$. It can be proved that when ΔABC is an obtuse triangle, we have

$$\max_T\{1 - \|\mathbf{w}\|_2\} = 1 - \sqrt{1 - (\frac{h^*}{2})^2} \tag{17}$$

Hence, for all triangles, we have

$$\max_T\{1 - \|\mathbf{w}\|_2\} \leq 1 - \sqrt{1 - \frac{{h^*}^2}{3}} \tag{18}$$

Now we further consider the error between normalized $\mathbf{w}$ and unormalized $\mathbf{w}$ on sub-triangle i. With a subvision parameter n, h^* will decrease to $\frac{h^*}{n}$ on original triangle domain. However, since new interpolated normals on each sub-triangle should be re-normalized in piecewise interpolation, therefore $\frac{h^*}{n}$ has to be scaled to h_i^* on i-th sub-triangle shown as Fig. (2b), where SO and TO is the new interpolated normals and ST decrease to $\frac{h^*}{n}$. SO and TO ought to be re-normalized to $S'O$ and $T'O$ where $S'T'$ is the true h_i^*. To calculate the maximum possible h_i^*, we can construct a special circumstance shown as Fig. (2c), where P is the center of circumcircle of triangle ΔABC and P evenly divide $ST(SP = PT)$. Under such circumstance, $h_i^* = S'T'$ can get the maximum value at arbitrary subdivision pattern. When P is not in the triangle ΔABC, it is still a conservative estimation. By letting $OP = \sqrt{1 - R^2}$, $SP = \frac{h^*}{2n}$, $\alpha = \arctan(\frac{SP}{OP})$ and $S'Q = \sin(\alpha)$, we can derive

$$h_i^* = S'T' = 2S'Q = 2\sin\left(\arctan(\frac{h^*}{2n\sqrt{1 - R^2}})\right) \tag{19}$$

Given h_i^*, a subdivision parameter n can be computed as

$$n \geq \frac{h^*\sqrt{1 - \frac{{h_i^*}^2}{4}}}{h_i^*\sqrt{1 - R^2}} \tag{20}$$

Hence, providing an error threshold ϵ, we first compute a proper h_i^* on sub-triangles by taking Eq. (18) and apply Eq. (19) to calculate the subdivision parameter n:

$$n \geq \frac{\sqrt{1+3(1-\epsilon)^2}h^*}{2\sqrt{3-3(1-\epsilon)^2}\sqrt{1-R^2}}. \tag{21}$$

1.2 Example: Lambertian model

The Lambertian model is one of the simplest shading functions that requires normals and light directions as attributes to be interpolated from m vertices of a polygon P to other coordinates. We consider triangle primitives whose $m = 3$. We use $\mathbf{n}$ and $\mathbf{l}$ to denote the linear interpolated normals and light directions, which are computed as $\mathbf{n} = \sum_{k=1}^{m} \mathbf{n}_k \lambda_k(x, y)$ and $\mathbf{l} = \sum_{k=1}^{m} \mathbf{l}_k \lambda_k(x, y)$, where $\mathbf{n}_k$ and $\mathbf{l}_k$ denote the normals and light directions at each vertex. The interpolated $\mathbf{n}$ and $\mathbf{l}$ are unnormalized. The entire shading function of the Lambertian model is computed as

$$f(x, y) = \hat{f}(\mathbf{n}, \mathbf{l}) = K_d \cdot \frac{\mathbf{n}}{\|\mathbf{n}\|_2} \cdot \frac{\mathbf{l}}{\|\mathbf{l}\|_2}, \tag{22}$$

where K_d is the diffuse coefficient, while $\frac{\mathbf{n}}{\|\mathbf{n}\|_2}$ and $\frac{\mathbf{l}}{\|\mathbf{l}\|_2}$ are the normalized normal and lighting direction at the shading point respectively.

1.2.1 An Attempt to Apply General Error Estimation Directly. We show a straightforward attempt to apply the general formula using Eq. (10) for clarifying the reason of requiring our special error estimation method for vector normalization.

Suppose existing a barycentric mapping from (x, y) to normal $\mathbf{n}$ and light direction $\mathbf{l}$, we denote the barycentric mapping using Eq. (12) as:

$$\begin{aligned} \mathbf{n} &= C_1 x + D_1 y + \mathbf{n}_0 \quad \text{where} \quad C_1 = -\frac{\mathbf{n}_0}{a} + \frac{\mathbf{n}_1}{a}, D_1 = -\frac{b-a}{ac}\mathbf{n}_0 - \frac{b}{ac}\mathbf{n}_1 + \frac{\mathbf{n}_2}{c} \\ \mathbf{l} &= C_2 x + D_2 y + \mathbf{l}_0 \quad \text{where} \quad C_2 = -\frac{\mathbf{l}_0}{a} + \frac{\mathbf{l}_1}{a}, D_2 = -\frac{b-a}{ac}\mathbf{l}_0 - \frac{b}{ac}\mathbf{l}_1 + \frac{\mathbf{l}_2}{c} \end{aligned} \tag{23}$$

and $f(x, y) = \frac{\mathbf{n}}{\|\mathbf{n}\|_2} \cdot \frac{\mathbf{l}}{\|\mathbf{l}\|_2}$.

According to Eq. (13) and Eq.(14), the semi-norm of f is computed as

$$|f|_{2,\infty,T_i} = \frac{1}{2} \sup_{x,y \in T_i} \{ |\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}| + \sqrt{(\frac{\partial^2 f}{\partial x^2} - \frac{\partial^2 f}{\partial y^2})^2 + 4(\frac{\partial^2 f}{\partial xy})^2} \} \tag{24}$$

where we have

$$\begin{aligned} \frac{\partial^2 f}{\partial x^2} &= -\frac{2C_1 \cdot \mathbf{n}(C_1 \cdot \mathbf{l} + \mathbf{n} \cdot C_2)}{\sqrt{\mathbf{l} \cdot \mathbf{l}}(\mathbf{n} \cdot \mathbf{n})^{3/2}} + \frac{2C_1 \cdot \mathbf{n} C_2 \cdot \mathbf{l} \mathbf{n} \cdot \mathbf{l}}{(\mathbf{l} \cdot \mathbf{l})^{3/2}(\mathbf{n} \cdot \mathbf{n})^{3/2}} - \frac{2C_2 \cdot \mathbf{l}(C_1 \cdot \mathbf{l} + \mathbf{n} \cdot C_2)}{(\mathbf{l} \cdot \mathbf{l})^{3/2}\sqrt{\mathbf{n} \cdot \mathbf{n}}} \\ &+ \frac{2C_1 \cdot C_2}{\sqrt{\mathbf{l} \cdot \mathbf{l}}\sqrt{\mathbf{n} \cdot \mathbf{n}}} + \frac{3(C_1 \cdot \mathbf{n})^2 \mathbf{n} \cdot \mathbf{l}}{\sqrt{\mathbf{l} \cdot \mathbf{l}}(\mathbf{n} \cdot \mathbf{n})^{5/2}} - \frac{C_1 \cdot C_1 \mathbf{n} \cdot \mathbf{l}}{\sqrt{\mathbf{l} \cdot \mathbf{l}}(\mathbf{n} \cdot \mathbf{n})^{3/2}} - \frac{C_2 \cdot C_2 \mathbf{n} \cdot \mathbf{l}}{(\mathbf{l} \cdot \mathbf{l})^{3/2}\sqrt{\mathbf{n} \cdot \mathbf{n}}} + \frac{3(C_2 \cdot \mathbf{l})^2 \mathbf{n} \cdot \mathbf{l}}{(\mathbf{l} \cdot \mathbf{l})^{5/2}\sqrt{\mathbf{n} \cdot \mathbf{n}}} \end{aligned} \tag{25}$$

$$\begin{aligned} \frac{\partial^2 f}{\partial y^2} &= -\frac{2D_1 \cdot \mathbf{n}(D_1 \cdot \mathbf{l} + \mathbf{n} \cdot D_2)}{\sqrt{\mathbf{l} \cdot \mathbf{l}}(\mathbf{n} \cdot \mathbf{n})^{3/2}} + \frac{2D_1 \cdot \mathbf{n}(D_2 \cdot \mathbf{l})\mathbf{n} \cdot \mathbf{l}}{(\mathbf{l} \cdot \mathbf{l})^{3/2}(\mathbf{n} \cdot \mathbf{n})^{3/2}} - \frac{2D_2 \cdot \mathbf{l}(D_1 \cdot \mathbf{l} + \mathbf{n} \cdot D_2)}{(\mathbf{l} \cdot \mathbf{l})^{3/2}\sqrt{\mathbf{n} \cdot \mathbf{n}}} \\ &+ \frac{2D_1 \cdot D_2}{\sqrt{\mathbf{l} \cdot \mathbf{l}}\sqrt{\mathbf{n} \cdot \mathbf{n}}} + \frac{3(D_1 \cdot \mathbf{n})^2 \mathbf{n} \cdot \mathbf{l}}{\sqrt{\mathbf{l} \cdot \mathbf{l}}(\mathbf{n} \cdot \mathbf{n})^{5/2}} - \frac{D_1 \cdot D_1 \mathbf{n} \cdot \mathbf{l}}{\sqrt{\mathbf{l} \cdot \mathbf{l}}(\mathbf{n} \cdot \mathbf{n})^{3/2}} - \frac{D_2 \cdot D_2 \mathbf{n} \cdot \mathbf{l}}{(\mathbf{l} \cdot \mathbf{l})^{3/2}\sqrt{\mathbf{n} \cdot \mathbf{n}}} + \frac{3(D_2 \cdot \mathbf{l})^2 \mathbf{n} \cdot \mathbf{l}}{(\mathbf{l} \cdot \mathbf{l})^{5/2}\sqrt{\mathbf{n} \cdot \mathbf{n}}} \end{aligned} \tag{26}$$

$$\frac{\partial^2 f}{\partial x\partial y}=\frac{\partial^2 f}{\partial y\partial x}=\frac{C_1\cdot D_2+D_1\cdot C_2}{\sqrt{\mathbf{l}\cdot\mathbf{l}}\sqrt{\mathbf{n}\cdot\mathbf{n}}}-\frac{D_1\cdot\mathbf{n}(C_1\cdot\mathbf{l}+\mathbf{n}\cdot C_2)}{\sqrt{\mathbf{l}\cdot\mathbf{l}}(\mathbf{n}\cdot\mathbf{n})^{3/2}}-\frac{D_2\cdot\mathbf{l}(C_1\cdot\mathbf{l}+\mathbf{n}\cdot C_2)}{(\mathbf{l}\cdot\mathbf{l})^{3/2}\sqrt{\mathbf{n}\cdot\mathbf{n}}}-\frac{C_1\cdot\mathbf{n}(D_1\cdot\mathbf{l}+\mathbf{n}\cdot D_2)}{\sqrt{\mathbf{l}\cdot\mathbf{l}}(\mathbf{n}\cdot\mathbf{n})^{3/2}}$$
$$-\frac{C_1\cdot D_1\mathbf{n}\cdot\mathbf{l}}{\sqrt{\mathbf{l}\cdot\mathbf{l}}(\mathbf{n}\cdot\mathbf{n})^{3/2}}+\frac{3C_1\cdot\mathbf{n}D_1\cdot\mathbf{nn}\cdot\mathbf{l}}{\sqrt{\mathbf{l}\cdot\mathbf{l}}(\mathbf{n}\cdot\mathbf{n})^{5/2}}+\frac{C_1\cdot\mathbf{n}D_2\cdot\mathbf{ln}\cdot\mathbf{l}}{(\mathbf{l}\cdot\mathbf{l})^{3/2}(\mathbf{n}\cdot\mathbf{n})^{3/2}}-\frac{C_2\cdot\mathbf{l}(D_1\cdot\mathbf{l}+\mathbf{n}\cdot D_2)}{(\mathbf{l}\cdot\mathbf{l})^{3/2}\sqrt{\mathbf{n}\cdot\mathbf{n}}}+\frac{C_2\cdot\mathbf{l}D_1\cdot\mathbf{nn}\cdot\mathbf{l}}{(\mathbf{l}\cdot\mathbf{l})^{3/2}(\mathbf{n}\cdot\mathbf{n})^{3/2}}$$
$$-\frac{C_2\cdot D_2\mathbf{n}\cdot\mathbf{l}}{(\mathbf{l}\cdot\mathbf{l})^{3/2}\sqrt{\mathbf{n}\cdot\mathbf{n}}}+\frac{3C_2\cdot\mathbf{l}D_2\cdot\mathbf{ln}\cdot\mathbf{l}}{(\mathbf{l}\cdot\mathbf{l})^{5/2}\sqrt{\mathbf{n}\cdot\mathbf{n}}} \tag{27}$$

These lengthy derivations show that although we have an analytic form of the general error estimation on the Lambertian model by directly computing (Eq. (10)) , it is impractical to allow a runtime fast evaluation.

1.2.2 Our Simplified Derivation. As shown before, directly computing the semi-norm $|f|_{2,\infty,P}$ of the Lambertian model is complicated because of vector normalization terms. However, we can split the interpolation error of f into two simpler terms, and compute the separated error bounds. Theoretically, after the subdivision, the error bound on a convex sub-polygon P_i can be computed as follows:

$$\begin{aligned}
&\|f-L_{\Theta_i}f\|_{\infty,P_i}=\|\hat{f}(n,l)-L_{\Theta_i}\hat{f}(n,l)\|_{\infty,P_i}\\
=&\|\frac{\mathbf{n}}{\|\mathbf{n}\|_2}\cdot\frac{\mathbf{l}}{\|\mathbf{l}\|_2}-L_{\Theta_i}(\frac{\mathbf{n}}{\|\mathbf{n}\|_2}\cdot\frac{\mathbf{l}}{\|\mathbf{l}\|_2})\|_{\infty,P_i}\\
=&\|\frac{\mathbf{n}}{\|\mathbf{n}\|_2}\cdot\frac{\mathbf{l}}{\|\mathbf{l}\|_2}-L_{\Theta_i}(\mathbf{n}\cdot\mathbf{l})\|_{\infty,P_i}\\
\leq&\|\frac{\mathbf{n}}{\|\mathbf{n}\|_2}\cdot\frac{\mathbf{l}}{\|\mathbf{l}\|_2}-\mathbf{n}\cdot\mathbf{l}\|_{\infty,P_i}+\|\mathbf{n}\cdot\mathbf{l}-L_{\Theta_i}(\mathbf{n}\cdot\mathbf{l})\|_{\infty,P_i}
\end{aligned} \tag{28}$$

For the first term of the error, by applying the specific error estimation on the length of interpolating normalized vectors (Eq. (15)), it can be computed as follows:

$$\begin{aligned}
&\|\frac{\mathbf{n}}{\|\mathbf{n}\|_2}\cdot\frac{\mathbf{l}}{\|\mathbf{l}\|_2}-\mathbf{n}\cdot\mathbf{l}\|_{\infty,P_i}\\
=&\|(1-\|\mathbf{n}\|_2\cdot\|\mathbf{l}\|_2)\cos\langle\mathbf{n},\mathbf{l}\rangle\|_{\infty,P_i}\\
\leq&\Big\|\Big(\Big\|\|\frac{\mathbf{n}}{\|\mathbf{n}\|_2}-\mathbf{n}\|_2\Big\|_\infty+\Big\|\|\frac{\mathbf{l}}{\|\mathbf{l}\|_2}-\mathbf{l}\|_2\Big\|_\infty\Big)\cdot\cos\langle\mathbf{n},\mathbf{l}\rangle\Big\|_{\infty,P_i}\\
\leq&\big(\max\{1-\|\mathbf{n}\|_2\}+\max\{1-\|\mathbf{l}\|_2\}\big)\cdot\max\{\cos\langle\mathbf{n},\mathbf{l}\rangle\}
\end{aligned} \tag{29}$$

where $\Big\|\|\frac{\mathbf{n}}{\|\mathbf{n}\|_2}-\mathbf{n}\|_2\Big\|_{\infty,P_i}$ and $\Big\|\|\frac{\mathbf{l}}{\|\mathbf{l}\|_2}-\mathbf{l}\|_2\Big\|_{\infty,P_i}$ are the errors from the normalization function, and $\max\{\cos\langle\mathbf{n},\mathbf{l}\rangle\}$ is the potential maximum shading value on the sub-triangle. The inequality in Eq. (29) can be directly derived from Eq. (15).

The potential maximum shading value on the convex polygon P, $\max\{\cos\langle\mathbf{n},\mathbf{l}\rangle\}$, can be calculated by finding the minimum possible angle between $\mathbf{n}$ and $\mathbf{l}$. For instance, when the convex polygon is a triangle, shown as Fig. (1b), all $\mathbf{n}$ and $\mathbf{l}$ form two spherical triangles in the hemisphere vector space. For simplicity, we construct two circumcircles to include $\mathbf{n}$ and $\mathbf{l}$ respectively, and denote the angle between these circumcircles as θ_0, and the interior angles of each circumcircle as θ_1 and θ_2. In this way, $\max\{\cos\langle n,l\rangle\}$ can be estimated as

$$\max\{\cos\langle\mathbf{n},\mathbf{l}\rangle\}=\cos(\max\{0,\theta_0-\theta_1-\theta_2\}). \tag{30}$$

Providing an error threshold ϵ, by letting $\epsilon'=\frac{\epsilon}{\max\{\cos\langle\mathbf{n},\mathbf{l}\rangle\}}$, we can use Eq. (21) to calculate appropriate subdivision parameters $n_\mathbf{n}$ and $n_\mathbf{l}$ for $\mathbf{n}$ and $\mathbf{l}$ respectively. For example, $n_\mathbf{n}$ can be computed as

$$n_{\mathbf{n}} \geq \frac{\sqrt{1+3(1-\epsilon')^2}h^*_{\mathbf{n}}}{2\sqrt{3-3(1-\epsilon')^2}\sqrt{1-R^2_{\mathbf{n}}}}, \tag{31}$$

where $h^*_{\mathbf{n}} = \max\{\|\mathbf{n}_0 - \mathbf{n}_1\|_2, \|\mathbf{n}_0 - \mathbf{n}_2\|_2, \|\mathbf{n}_1 - \mathbf{n}_2\|_2\}$ and $R_{\mathbf{n}}$ is the radius of the circumcircle of $\mathbf{n}_0$, $\mathbf{n}_1$ and $\mathbf{n}_2$.

The second term of the error in Eq. (28) is estimated by the general error estimation formula Eq. (11) where the interpolated function is an inner product because $f(x,y) = \hat{f}(\mathbf{n},\mathbf{l}) = \mathbf{n}\cdot\mathbf{l}$. The attributes in this function are normals and light directions, i.e., $A = [\mathbf{n}^T, \mathbf{l}^T]^T$. By using Eq. (14) to compute the second order derivative $|D^2 f|$, we find $\frac{\partial^2 \hat{f}}{\partial A^2}$ is a constant matrix which significantly simplifies the computation:

$$\begin{aligned}
&\|f(x,y)) - L_{\Theta_i} f(x,y)\|_{\infty,T_i}\\
=&\|\hat{f}(\mathbf{n},\mathbf{l}) - L_{\Theta_i}\hat{f}(\mathbf{n},\mathbf{l})\|_{\infty,T_i}\\
\leq&\frac{1}{6}\frac{h^2}{n^2}\|D^2 f\|_{\infty,T_i}\\
\leq&\frac{1}{6}\frac{h^2}{n^2}\left(|C_1^T C_2 + D_1^T D_2| + \sqrt{(C_1^T C_2 - D_1^T D_2)^2 + (C_1^T D_2 + C_2^T D_1)^2}\right)\\
&\text{where} \quad C_1 = -\frac{\mathbf{n}_0}{a} + \frac{\mathbf{n}_1}{a}, D_1 = \frac{b-a}{ac}\mathbf{n}_0 - \frac{b}{ac}\mathbf{n}_1 + \frac{\mathbf{n}_2}{c}\\
&\text{and} \quad C_2 = -\frac{\mathbf{l}_0}{a} + \frac{\mathbf{l}_1}{a}, D_2 = \frac{b-a}{ac}\mathbf{l}_0 - \frac{b}{ac}\mathbf{l}_1 + \frac{\mathbf{l}_2}{c}
\end{aligned} \tag{32}$$

Given a L_∞ error threshold ϵ', the subdivision parameter $n_{\mathbf{n}\cdot\mathbf{l}}$ can be calculated as:

$$n_{\mathbf{n}\cdot\mathbf{l}} \geq h\sqrt{\frac{1}{6\epsilon'}\left(|C_1^T C_2 + D_1^T D_2| + \sqrt{(C_1^T C_2 - D_1^T D_2)^2 + (C_1^T D_2 + C_2^T D_1)^2}\right)} \tag{33}$$

In summary, the initial interpolation error of f (Eq. (22)) is expanded as

$$\begin{aligned}
&\|f - L_{\Theta_i} f\|_{\infty,P_i}\\
=&\max\{1 - \|\mathbf{n}\|_2\}\cdot\max\{\cos\langle\mathbf{n},\mathbf{l}\rangle\} + \max\{1 - \|\mathbf{l}\|_2\}\cdot\max\{\cos\langle\mathbf{n},\mathbf{l}\rangle\} + \|\mathbf{n}\cdot\mathbf{l} - L_{\Theta_i}(\mathbf{n}\cdot\mathbf{l})\|_{\infty,P_i}
\end{aligned} \tag{34}$$

from which, we obtain three subdivision parameters, namely, $n_{\mathbf{n}}$, $n_{\mathbf{l}}$ and $n_{\mathbf{n}\cdot\mathbf{l}}$. Once given an error threshold ϵ^*, we evenly divide it into three bounds, $\epsilon = \frac{\epsilon^*}{3}$, and individually estimate the corresponding subdivision parameters. We select the maximum value as our final subdivision parameter:

$$n = \max\{n_{\mathbf{n}}, n_{\mathbf{l}}, n_{\mathbf{n}\cdot\mathbf{l}}\}. \tag{35}$$

1.3 Example: Blinn-Phong model

Similar to Lambertian model, Blinn-Phong model requires normals and half-vectors (instead of light direction) as attributes and has an additional power operation. We denote normal and half-vector as $\mathbf{n}$ and $\mathbf{h}$. The entire shading function using Blinn-Phong model is computed as

$$f(x,y) = \hat{f}(\mathbf{n},\mathbf{h}) = K_s\cdot\left(\frac{\mathbf{n}}{\|\mathbf{n}\|_2}\cdot\frac{\mathbf{h}}{\|\mathbf{h}\|_2}\right)^\alpha, \tag{36}$$

where K_s is the specular coefficient and α is the shininess coefficient. To simplify the derivation, we introduce a new variable t as

$$t = \frac{\mathbf{n}}{\|\mathbf{n}\|_2}\cdot\frac{\mathbf{h}}{\|\mathbf{h}\|_2} \tag{37}$$

By plugging Eq. (37) into Eq. (36), f is simplified as

$$f(x,y) = \hat{f}(t) = K_s \cdot t^{\alpha} \tag{38}$$

Note that t is not linearly distributed on the surface. However, we can assume that there exists a linear interpolation of t, $L_{\Theta_i} t = \sum_{k=1}^{m} t_{i_k} \lambda_{i_k}(x)$, where t_{i_k} denotes the values computed at vertices of the convex sub-polygon P_i. We leverage the linear interpolation of t to estimate the error of Blinn-Phong model as

$$\begin{aligned} &\|f(x,y) - L_{\Theta_i} f(x,y)\|_{\infty,P_i} = \|\hat{f}(t) - L_{\Theta_i}\hat{f}(t)\|_{\infty,P_i} \\ \leq &\|\hat{f}(t) - \hat{f}(L_{\Theta_i} t)\|_{\infty,P_i} + \|\hat{f}(L_{\Theta_i} t) - L_{\Theta_i}\hat{f}(t)\|_{\infty,P_i} \end{aligned} \tag{39}$$

The first term of the inequality is the error introduced by the assumed linear interpolation of t, while the second term is the error caused by the interpolation of the new function $\hat{f}(t)$ on the convex sub-polygon P_i.

1.3.1 Estimation on $\|\hat{f}(L_{\Theta_i} t) - L_{\Theta_i}\hat{f}(t)\|_{\infty,P_i}$. The second term in Eq. (39) is easy to compute. Note that $L_{\Theta_i}\hat{f}(t) = L_{\Theta_i}\hat{f}(L_{\Theta_i} t)$, which suggests that this error is caused by the interpolation of the power function in Blinn-Phong model. Using the new variable t, we can directly apply the error formula Eq. (10) to derive a close-form solution:

$$\begin{aligned} &\|\hat{f}(L_{\Theta_i} t) - L_{\Theta_i}\hat{f}(t)\|_{\infty,P_i} \\ =&\|\hat{f}(L_{\Theta_i} t) - L_{\Theta_i}\hat{f}(L_{\Theta_i} t)\|_{\infty,P_i} \\ \leq&\frac{1}{6}\frac{h^2}{n^2}\|(C_t^2 + D_t^2)\frac{\partial^2 \hat{f}}{\partial t^2}\|_{\infty,P_i} \\ \leq&\frac{1}{6}\frac{h^2}{n^2}\|(C_t^2 + D_t^2)K_s\alpha(\alpha-1)t^{\alpha-2}\|_{\infty,P_i} \\ \leq&\frac{1}{6}\frac{K_s\alpha(\alpha-1)(C_t^2 + D_t^2)h^2}{n^2}(L_{\Theta} t)_{\max}^{\alpha-2} \end{aligned} \tag{40}$$

where $C_t = -\frac{t_0}{a} + \frac{t_1}{a}$, $D_t = \frac{b-a}{ac}t_0 - \frac{b}{ac}t_1 + \frac{t_2}{c}$ and $(L_{\Theta} t)_{\max} = \max\{t_0, t_1, t_2\}$. t_0, t_1, t_2 are the value of t calculated on each vertex of P_i. Hence, given an error threshold ϵ', we compute a subidivion parameter n by:

$$n \geq \frac{K_s\alpha(\alpha-1)(C_t^2 + D_t^2)h^2}{6\epsilon'}(L_{\Theta} t)_{\max}^{\alpha-2} \tag{41}$$

1.3.2 Estimation on $\|\hat{f}(t) - \hat{f}(L_{\Theta_i} t)\|_{\infty,P_i}$. We can further simplify and expand the first term in Eq. (39) as follows by using the Mean Value Theorem:

$$\begin{aligned} &\|\hat{f}(t) - \hat{f}(L_{\Theta_i} t)\|_{\infty,P_i} \\ =&\|\hat{f}_t'(\nu)(t - L_{\Theta_i} t)\|_{\infty,P_i} \qquad \text{where } L_{\Theta_i} t \leq \nu \leq t \\ \leq&|\hat{f}_t'(t_{max})| \cdot \|t - L_{\Theta_i} t\|_{\infty,P_i} \end{aligned} \tag{42}$$

The above inequality is always satisfied because the first derivative of Eq. (38) is a monotonically increasing function, and t_{max}, which denotes the maximum value of t on the convex sub-polygon P_i, can be calculated by finding the minimum possible angle between $\mathbf{n}$ and $\mathbf{h}$ similar to Eq. (30).

By replacing $\mathbf{l}$ with $\mathbf{h}$, $\|t - L_{\Theta_i} t\|_{\infty,P_i}$ takes a form identical to Lambertian model in Eq. (28). We can apply the same derivation to evaluate this error bound.

1.3.3 Final Subdivision. Given an error threshold ϵ, we apply the same strategy in the Lambertian model to combine different terms. We evenly divide ϵ into smaller thresholds. Let each term in Eq. (39) satisfy the split error bound and take the maximum parameters as the conservative final estimation.

1.4 Example: Microfacet model

Now we deal with a more sophisticated BRDF model, Microfacet model, which is widely-used in current graphics applications. We denote normal, light direction, half-vector and view direction as $\mathbf{n}, \mathbf{l}, \mathbf{h}$, and $\mathbf{v}$ respectively. The Microfacet model is computed as

$$f(x,y) = \hat{f}(\mathbf{n}, \mathbf{l}, \mathbf{h}, \mathbf{v}) = \frac{D(\mathbf{n}, \mathbf{h})F(\mathbf{v}, \mathbf{h})V(\mathbf{l}, \mathbf{v})}{4}, \tag{43}$$

where $D(\mathbf{n}, \mathbf{h})$ is a GGX normal distribution function [Walter et al. 2007], $F(\mathbf{v}, \mathbf{h})$ is the Fresnel term using the Schlick approximation [Smith 1967], and $V(\mathbf{l}, \mathbf{v})$ is the Schlick geometry term [Smith 1967].

For such a complex shading function with high-dimension and non-linear properties, we introduce the error propagation formula:

$$\Delta f = |\frac{\partial f}{\partial x_0}|\Delta x_0 + |\frac{\partial f}{\partial x_1}|\Delta x_1 + \cdots + |\frac{\partial f}{\partial x_n}|\Delta x_n, f = f(x_0, \cdots, x_n) \tag{44}$$

On the convex sub-polygon P_i, by letting $\|f - L_{\Theta_i} f\|_{\infty,P_i} = \|\hat{f} - L_{\Theta_i}\hat{f}\|_{\infty,P_i} = \|\Delta\hat{f}\|_{\infty,P_i}$, and applying Eq. (44) to the interpolation error of $\hat{f}$, we obtain:

$$\begin{aligned}
\|\Delta\hat{f}\|_{\infty,P_i} &= \| \, |\frac{\partial\hat{f}}{\partial D}|\Delta D + |\frac{\partial\hat{f}}{\partial F}|\Delta F + |\frac{\partial\hat{f}}{\partial V}|\Delta V\|_{\infty,P_i} \\
&= \|\frac{\hat{f}}{D}(D - L_{\Theta_i}D) + \frac{\hat{f}}{F}(F - L_{\Theta_i}F) + \frac{\hat{f}}{V}(V - L_{\Theta_i}V)\|_{\infty,P_i} \\
&\leq \sum_{I\in D,L,V} \|\frac{\hat{f}}{I}\|_{\infty,P_i} \cdot \|I - L_{\Theta_i}I\|_{\infty,P_i}.
\end{aligned} \tag{45}$$

which shows the total interpolation error of $\hat{f}$ propagates from the interpolation error from three components $D(\mathbf{n}, \mathbf{h})$, $F(\mathbf{v}, \mathbf{h})$ and $V(\mathbf{l}, \mathbf{v})$. Each term is a function of the dot product of normalized vectors, therefore we can extend Eq. (39) to compute interpolation errors for $\|I - L_{\Theta_i}I\|_{\infty,P_i}, I \in D, L, V$. We show some examples below.

1.4.1 Error estimation on $\|D - L_{\Theta_i}D\|_{\infty,P_i}$. We model the normalized distribution function $D(\mathbf{n}, \mathbf{h})$ using GGX distribution

$$D(\mathbf{n}, \mathbf{h}) = \frac{\alpha^2}{\pi((\mathbf{n}\cdot\mathbf{h})^2(\alpha^2 - 1) + 1)^2} \tag{46}$$

Similar to how we handle the Blinn-Phong model, we introduce a new variable t

$$t = \frac{\mathbf{n}}{\|\mathbf{n}\|_2} \cdot \frac{\mathbf{h}}{\|\mathbf{h}\|_2} \tag{47}$$

by which $D(\mathbf{n}, \mathbf{h})$ becomes

$$D(t) = \frac{\alpha^2}{\pi(t^2(\alpha^2 - 1) + 1)^2} \tag{48}$$

We expand the interpolation error of D as follows:

$$\begin{aligned}
&\|D(\mathbf{n},\mathbf{h}) - L_{\Theta_i} D(\mathbf{n},\mathbf{h})\|_{\infty,P_i} \\
=&\|D(t) - L_{\Theta_i} D(t)\|_{\infty,P_i} \\
=&\|D(t) - D(L_{\Theta_i} t) + D(L_{\Theta_i} t) - L_{\Theta_i} D(t)\|_{\infty,P_i} \\
\leq&\|D(t) - D(L_{\Theta_i} t)\|_{\infty,P_i} + \|D(L_{\Theta_i} t) - L_{\Theta_i} D(t)\|_{\infty,P_i}
\end{aligned} \tag{49}$$

This equation has an identical form as Eq. (39), we thus leverage a similar formula to estimate this error as well as the subdivision parameter.

1.4.2 Estimation on $\|F - L_{\Theta_i} F\|_{\infty,P_i}$ and $\|V - L_{\Theta_i} V\|_{\infty,P_i}$. All of these three sub-terms $D(\mathbf{n},\mathbf{h})$, $F(\mathbf{v},\mathbf{h})$, $V(\mathbf{l},\mathbf{v})$ follow similar forms as Blinn-Phong model. We can introduce a new variable t to replace inner product of two vectors as a single scalar and further expand the interpolation error to simplify computation. Besides, either F or V is monotonic function which can apply the Mean Value Theorem as in Eq. (42).

1.4.3 Estimation on $\|\frac{\hat{f}}{I}\|_{\infty,P_i}, I \in D, L, V$. We can compute an upper bound for each term. For example, we have

$$\begin{aligned}
\|\frac{\hat{f}}{D}\|_{\infty,P_i} =&\|\frac{F(\mathbf{v},\mathbf{h})V(\mathbf{l},\mathbf{v})}{4}\|_{\infty,P_i} \\
=&\|\frac{F_0 + (1-F_0)(1-\mathbf{v}\cdot\mathbf{h})^5}{4((\mathbf{n}\cdot\mathbf{l})(1-k)+k)((\mathbf{n}\cdot\mathbf{v})(1-k)+k)}\|_{\infty,P_i} \\
\leq&\frac{F_0 + (1-F_0)(1-\min\{\mathbf{v}\cdot\mathbf{h}\})^5}{4(\min\{\mathbf{n}\cdot\mathbf{l}\}(1-k)+k)(\min\{\mathbf{n}\cdot\mathbf{v}\}(1-k)+k)}
\end{aligned} \tag{50}$$

Given the monotonicity of the Fresnel and Geometry terms, the above inequality is always satisfied. The minimum values of $\mathbf{v}\cdot\mathbf{h}$, $\mathbf{n}\cdot\mathbf{l}$ and $\mathbf{n}\cdot\mathbf{v}$ can be evaluated efficiently. For example, to compute $\min\{\mathbf{n}\cdot\mathbf{l}\}$, we construct two cones to include $\mathbf{n}$ and $\mathbf{l}$ in the vector space respectively, similar to Fig. 1b. The minimum dot product is calculated as

$$\min\{\mathbf{n}\cdot\mathbf{l}\} = \cos(\min\{\pi, \theta_0 + \theta_1 + \theta_2\}) \tag{51}$$

1.4.4 Final Subdivision. The subdivision parameters are determined in the same way that we evenly divide the error threshold and assign to different terms in Eq. (45) to compute n separately. The final subdivision is the maximum value among them.

1.5 Discrete Variables: Textures

We extend our derivation to discrete variables which are often encoded as texture in computer graphics. Textures represent spatially-varying coefficients in shading functions e.g., diffuse or specular albedo, shininess or roughness values. Formally, the linear interpolation error of shading function f on a convex sub-polygon P_i is:

$$\begin{aligned}
&\|f(x,y) - L_{\Theta} f(x,y)\|_{\infty,P_i} \\
=&\|\hat{f}(\mathbf{A}(x,y),\alpha(x,y)) - L_{\Theta} f(\mathbf{A}(x,y),\alpha(x,y))\|_{\infty,P_i}
\end{aligned} \tag{52}$$

where $\mathbf{A}$ is a set of attributes defined on the shading function, and α is a sampled value from a texture. We consider one texture for simplicity but the following derivation can be easily applied to multiple textures.

Since texture stores discrete values, we cannot directly obtain an analytical form for $\alpha(x,y)$. However, we observe that most of the shading functions and their second derivatives are usually monotonic functions w.r.t. their coefficients. When evaluating shading functions by sampled α, the maximum interpolation error of $\hat{f}(\mathbf{A}, \alpha)$ for all $\alpha^* \in (\alpha_{\min}, \alpha_{\max})$ is at $\alpha^* = \alpha_{\min}$ or $\alpha^* = \alpha_{\max}$. We split Eq. (52) into two terms:

$$\begin{aligned} &\|\hat{f}(\mathbf{A}, \alpha) - L_{\Theta_i}\hat{f}(\mathbf{A}, \alpha)\|_{\infty, P_i} \\ \leq &\sup_{\alpha^* \in (\alpha_{\min}, \alpha_{\max})} \|\hat{f}(\mathbf{A}, \alpha^*) - L_{\Theta_i}\hat{f}(\mathbf{A}, \alpha^*)\|_{\infty, P_i} \\ &+ \sup_{\alpha^* \in (\alpha_{\min}, \alpha_{\max})} \|L_{\Theta_i}\hat{f}(\mathbf{A}, \alpha^*) - L_{\Theta_i}\hat{f}(\mathbf{A}, \alpha)\|_{\infty, P_i} \end{aligned} \tag{53}$$

where the first term is the linear interpolation error of $\hat{f}(\mathbf{A}, \alpha^*)$ while the second term is the error introduced by replacing α with the fixed value α^*.

Note that interpolation error of $\hat{f}(\mathbf{A}, \alpha^*)$ reach its maximum at $\alpha^* = \alpha_{\min}$ or $\alpha^* = \alpha_{\max}$. The first term can be computed by:

$$\begin{aligned} &\sup_{\alpha^* \in (\alpha_{\min}, \alpha_{\max})} \|\hat{f}(\mathbf{A}, \alpha^*) - L_{\Theta}\hat{f}(\mathbf{A}, \alpha^*)\|_{\infty, P_i} \\ = &\max_{\alpha^* = \alpha_{\min},\ \alpha^* = \alpha_{\max}} \{\|\hat{f}(\mathbf{A}, \alpha^*) - L_{\Theta}\hat{f}(\mathbf{A}, \alpha^*)\|_{\infty, P_i}\} \end{aligned} \tag{54}$$

For the second term, the supremum of $\|L_{\Theta}\hat{f}(\mathbf{A}, \alpha^*) - L_{\Theta}\hat{f}(\mathbf{A}, \alpha)\|_{\infty, P_i}$ can be constrained as

$$\begin{aligned} &\sup_{\alpha^* \in (\alpha_{\min}, \alpha_{\max})} \|L_{\Theta_i}\hat{f}(\mathbf{A}, \alpha^*) - L_{\Theta_i}\hat{f}(\mathbf{A}, \alpha)\|_{\infty, P_i} \\ &\leq |\hat{f}(\mathbf{A}, \alpha_{\max}) - \hat{f}(\mathbf{A}, \alpha_{\min})| \end{aligned} \tag{55}$$

In our error estimation, $\alpha_{\min}$ and $\alpha_{\max}$ are the values on the convex sub-polygon P_i. However, we cannot obtain the precise range of $\alpha_{\min}$ and $\alpha_{\max}$ on individual P_i before subdivision. For conservative estimation, we take $\alpha_{\min}$ and $\alpha_{\max}$ on the original polygonal domain P to obtain $\epsilon'' = |\hat{f}(\mathbf{A}, \alpha_{\max}) - \hat{f}(\mathbf{A}, \alpha_{\min})|$. Given an error threshold ϵ, we compute $\epsilon' = \epsilon - \epsilon''$ and limit the first error term within ϵ':

$$\max_{\alpha^* = \alpha_{\min},\ \alpha^* = \alpha_{\max}} \{\|\hat{f}(\mathbf{A}, \alpha^*) - L_{\Theta}\hat{f}(\mathbf{A}, \alpha^*)\|_{\infty, P_i}\} \leq \epsilon' \tag{56}$$

With a constant α^*, we estimate its subdivision parameter n using previous derivations.

REFERENCES

Allal Guessab and Gerhard Schmeisser. 2005. Sharp error estimates for interpolatory approximation on convex polytopes. *SIAM journal on numerical analysis* 43, 3 (2005), 909–923.

B Smith. 1967. Geometrical shadowing of a random rough surface. *IEEE transactions on antennas and propagation* 15, 5 (1967), 668–671.

Yurii Nikolaevich Subbotin. 1989. The dependence of estimates of a multidimensional piecewise-polynomial approximation on the geometric characteristics of a triangulation. *Trudy Matematicheskogo Instituta imeni VA Steklova* 189 (1989), 117–137.

Shayne Waldron. 1998. The Error in Linear Interpolation at the Vertices of a Simplex. *SIAM J. Numer. Anal.* 35, 3 (1998), 1191–1200.

Bruce Walter, Stephen R. Marschner, Hongsong Li, and Kenneth E. Torrance. 2007. Microfacet Models for Refraction Through Rough Surfaces. In *Proceedings of the 18th Eurographics Conference on Rendering Techniques* (Grenoble, France) *(EGSR'07)*. 195–206.