Lifting Lines and Tone: Image-space Stylization in Path-space (Supplemental)

REX WEST, Aoyama Gakuin University, Japan
SAYAN MUKHERJEE, The University of Tokyo, Japan
YONGHAO YUE, Aoyama Gakuin University, Japan

ACM Reference Format:
Rex West, Sayan Mukherjee, and Yonghao Yue. 2026. Lifting Lines and Tone: Image-space Stylization in Path-space (Supplemental). *ACM Trans. Graph.* 45, 4, Article 138 (July 2026), 11 pages. https://doi.org/10.1145/3811359

This supplemental document accompanies Lifting Lines and Tone: Image-space Stylization in Path-space. It collects proofs, implementation details, and extended comparisons. Section S1 contains discussion about our feature line method, and Section S2 describes details about the tone rendering method developed in the main paper.

S1 FEATURE-LINE SUPPLEMENTAL DETAILS

This section provides further details on the feature-line construction in Section 4 of the main paper. We begin with the continuous derivation behind the parallel-transport operator used for conditionally specular paths, then collect the lower-bound and convergence arguments behind the finite-difference search, then provide implementation details and finally close with a few practical discussion points for implementation.

S1.1 Parallel transport of half-vectors along conditionally specular paths

Given a conditionally specular light path defined by a sequence of microfacet half-vectors, our goal is to show that neighboring image-space samples induce a unique *parallel* path obtained by transporting each half-vector along the (possibly curved) surface and reapplying specular reflection. This construction provides a well-defined correspondence between nearby image-space coordinates even under non-specular (microfacet) scattering.

Setup. Let $S \subset \mathbb{R}^3$ be a smooth surface with unit normal field $\mathbf{n}(x)$ and induced Levi–Civita connection. Let $\gamma : [0,1] \to S$ be a C^1 curve with $\gamma(0) = x$ and $\gamma(1) = x'$, and denote $\dot\gamma = \frac{\mathrm{d}\gamma}{\mathrm{d}s}$.

Fix a viewpoint $\mathbf{E} \in \mathbb{R}^3$ and define the (unit) view direction

$$\mathbf{v}(s) := \frac{\mathbf{E} - \gamma(s)}{\|\mathbf{E} - \gamma(s)\|}. \tag{S1}$$

At each s, a microfacet interaction is represented by a unit half-vector $\mathbf{h}(s) \in \mathbb{S}^2$, which induces a reflected direction

$$\mathbf{d}(s) := 2\big(\mathbf{h}(s)\cdot\mathbf{v}(s)\big)\,\mathbf{h}(s) - \mathbf{v}(s). \tag{S2}$$

Authors' addresses: Rex West, Aoyama Gakuin University, Japan; Sayan Mukherjee, The University of Tokyo, Japan; Yonghao Yue, Aoyama Gakuin University, Japan.

Fixed-tilt constraint. Under a microfacet model, conditioning on a facet selection fixes the cosine tilt

$$c := \mathbf{h}\cdot\mathbf{n} \in [-1, 1]. \tag{S3}$$

Accordingly, admissible half-vectors at a surface point with normal $\mathbf{n}$ lie on the circle

$$\mathcal{F}(\mathbf{n}, c) := \{\mathbf{h} \in \mathbb{S}^2 \mid \mathbf{h}\cdot\mathbf{n} = c\}. \tag{S4}$$

Along the curve γ, we require

$$\mathbf{h}(s) \in \mathcal{F}(\mathbf{n}(s), c) \quad \text{for all } s, \tag{S5}$$

i.e. the selected microfacet tilt is preserved.

Parallel transport of half-vectors. Write the half-vector as

$$\mathbf{h} = c\,\mathbf{n} + s\,\mathbf{u}, \qquad s := \sqrt{1 - c^2}, \qquad \mathbf{u} \in T_{\gamma(s)}S,\ \|\mathbf{u}\| = 1. \tag{S6}$$

Parallel transport of $\mathbf{h}$ along γ is defined by requiring that its tangential component $\mathbf{u}$ is parallel under the Levi–Civita connection:

$$\nabla_{\dot\gamma}\,\mathbf{u} = 0. \tag{S7}$$

This condition preserves both $\|\mathbf{h}\| = 1$ and the tilt constraint (S5).

Using the Gauss–Weingarten relation $\frac{\mathrm{d}\mathbf{n}}{\mathrm{d}s} = -\mathbf{S}(\dot\gamma)$, where $\mathbf{S}$ is the shape operator, (S7) is equivalent to the ambient-space ODE

$$\frac{\mathrm{d}\mathbf{h}}{\mathrm{d}s} = c\,\frac{\mathrm{d}\mathbf{n}}{\mathrm{d}s} - \Big(\frac{\mathrm{d}\mathbf{n}}{\mathrm{d}s}\cdot\mathbf{h}\Big)\,\mathbf{n}. \tag{S8}$$

This equation uniquely defines the transported half-vector along γ.

Induced transport of reflected directions. Given the transported half-vector $\mathbf{h}(s)$, we define the reflected direction by

$$\mathbf{d}(s) := 2\big(\mathbf{h}(s)\cdot\mathbf{v}(s)\big)\,\mathbf{h}(s) - \mathbf{v}(s). \tag{S9}$$

Differentiating yields

$$\frac{\mathrm{d}\mathbf{d}}{\mathrm{d}s} = 2\Big(\frac{\mathrm{d}\mathbf{h}}{\mathrm{d}s}\cdot\mathbf{v} + \mathbf{h}\cdot\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}s}\Big)\,\mathbf{h} + 2\big(\mathbf{h}\cdot\mathbf{v}\big)\,\frac{\mathrm{d}\mathbf{h}}{\mathrm{d}s} - \frac{\mathrm{d}\mathbf{v}}{\mathrm{d}s}, \tag{S10}$$

where

$$\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}s} = -\frac{\Pi_{\mathbf{v}^\perp}\dot\gamma}{\|\mathbf{E} - \gamma\|}, \qquad \Pi_{\mathbf{v}^\perp} = \mathbf{I} - \mathbf{v}\mathbf{v}^\top. \tag{S11}$$

Substituting (S8) gives a closed evolution for $\mathbf{d}$ driven by surface geometry and camera motion.

Consequence for path correspondence. Conditioned on a sequence of microfacet half-vectors $\{\mathbf{h}_i\}$, a light path is purely specular. For a neighboring image-space coordinate, parallel transport of each $\mathbf{h}_i$ according to (S8), followed by reflection via (S9), produces a unique auxiliary path. This construction preserves image-space parallelism at each bounce and defines a consistent correspondence between nearby samples under glossy reflection and refraction.

Importantly, the transport rule depends only on local surface geometry and does not require global smoothness. In practice, we

apply the same construction across chart boundaries and discrete surface representations, treating (S8) as a local update rule for half-vectors along the path.

S1.2 Finite differences and Lipschitz constants

This subsection supports the lower-bound and convergence claims made in Section 4.5 of the main paper. Fix a conditioning realization ξ and write $f_\xi(s) := f(s, \xi)$. We noted in the main text that the finite differences used by the line detector are not just heuristic approximations to a gradient. Each sampled two-point slope gives us a lower bound on the maximal gradient somewhere in the sampled region, and exhaustive search over such pairs is *consistent* under mild regularity assumptions.

Finite differences yield valid lower bounds. Let $S \subset \mathbb{R}^2$ be a bounded region of image-space and let $f_\xi : S \to \mathbb{R}$ be locally Lipschitz, hence differentiable almost everywhere. For distinct points $u, v \in S$ with the segment $[u, v] \subset S$, define the two-point finite difference

$$D(u, v) := \frac{f_\xi(v) - f_\xi(u)}{\|v - u\|}. \tag{S12}$$

Write $\gamma(t) = u + t(v - u)$ for $t \in [0, 1]$ and let $g(t) = f_\xi(\gamma(t))$. Since f_ξ is locally Lipschitz, so is g. By the classical mean value theorem in the smooth case, and by the generalized Clarke mean value theorem in the locally Lipschitz case, there exists $t^* \in (0, 1)$ such that

$$g'(t^*) = f_\xi(v) - f_\xi(u). \tag{S13}$$

Setting $s^* = \gamma(t^*)$ and dividing by $\|v - u\|$ gives

$$\left| D^{(s)}_{\mathbf{u}\to\mathbf{v}} f_\xi(s^*) \right| = \frac{|f_\xi(v) - f_\xi(u)|}{\|v - u\|}. \tag{S14}$$

At points of differentiability,

$$D^{(s)}_{\mathbf{u}\to\mathbf{v}} f_\xi(s^*) = \nabla_s f_\xi(s^*) \cdot \frac{v - u}{\|v - u\|}, \tag{S15}$$

and Cauchy–Schwarz therefore yields

$$\frac{|f_\xi(v) - f_\xi(u)|}{\|v - u\|} \le \|\nabla_s f_\xi(s^*)\| \le \sup_{y \in S} \|\nabla_s f_\xi(y)\|. \tag{S16}$$

Corner cases. If f_ξ is continuous but not differentiable everywhere, the same conclusion still holds in the generalized-gradient sense described above. If f_ξ has a jump discontinuity, then the local Lipschitz constant is infinite in every neighborhood that contains the jump, and finite differences across the discontinuity grow without bound as the sample spacing shrinks. In our application, this is the desired behavior: silhouettes and other transport discontinuities should act like gradients of arbitrarily large magnitude rather than exceptions to the detector.

Consequence for stochastic search. Each sampled finite difference is therefore a valid lower bound on the gradient magnitude at some point along the sampled segment. For any collection of sample pairs (u_i, v_i),

$$\max_i \frac{|f_\xi(v_i) - f_\xi(u_i)|}{\|v_i - u_i\|} \le \sup_{y \in S} \|\nabla_s f_\xi(y)\|. \tag{S17}$$

The line search used in the main paper can fail to find the maximal gradient, but it cannot overestimate it.

Setup for exhaustive search. To make the smooth-case consistency statement mathematically precise, let us now assume that $S \subset \mathbb{R}^2$ is convex with nonempty interior and that f_ξ is continuously differentiable on an open set U containing the closure $\overline{S}$. Define the global Lipschitz constant of f_ξ over S by

$$\mathrm{L}_S(f_\xi) := \sup_{\substack{u, v \in S \\ u \neq v}} \frac{|f_\xi(v) - f_\xi(u)|}{\|v - u\|}.$$

By definition, $\mathrm{L}_S(f_\xi)$ is the smallest $L \in [0, \infty]$ such that $|f_\xi(v) - f_\xi(u)| \le L\|v - u\|$ for all $u, v \in S$.

Theorem S1.1. *If S is convex and $f_\xi \in C^1(U)$ for an open $U \supset \overline{S}$, then*

$$\mathrm{L}_S(f_\xi) = \sup_{y \in S} \|\nabla_s f(y, \xi)\|. \tag{S18}$$

Proof. Let $M := \sup_{y \in S} \|\nabla_s f(y, \xi)\|$.

For the upper bound, pick any distinct $u, v \in S$ and consider the line segment $\gamma : [0, 1] \to S$ given by $\gamma(t) = u + t(v - u)$. Convexity of S ensures $\gamma(t) \in S$ for all $t \in [0, 1]$. By the fundamental theorem of calculus,

$$f_\xi(v) - f_\xi(u) = \int_0^1 \frac{d}{dt} f_\xi(\gamma(t))\, dt = \int_0^1 \nabla_s f_\xi(\gamma(t)) \cdot (v - u)\, dt.$$

Taking absolute values and applying Cauchy–Schwarz yields

$$|f_\xi(v) - f_\xi(u)| \le \int_0^1 \|\nabla_s f_\xi(\gamma(t))\|\, \|v - u\|\, dt \le M\, \|v - u\|.$$

Dividing by $\|v - u\|$ and taking the supremum over $u \neq v$ gives $\mathrm{L}_S(f_\xi) \le M$.

For the lower bound, fix any point y in the interior S° of S and any unit direction $w \in \mathbb{S}^1$. Define $v_\varepsilon := y + \varepsilon w$. For sufficiently small $\varepsilon > 0$, we have $v_\varepsilon \in S$ since y is interior. Then

$$\frac{|f_\xi(v_\varepsilon) - f_\xi(y)|}{\|v_\varepsilon - y\|} = \left| \frac{f_\xi(y + \varepsilon w) - f_\xi(y)}{\varepsilon} \right|.$$

Taking $\varepsilon \to 0^+$ and using differentiability gives

$$\lim_{\varepsilon \to 0^+} \frac{|f_\xi(y + \varepsilon w) - f_\xi(y)|}{\varepsilon} = |\nabla_s f_\xi(y) \cdot w|.$$

Therefore, for each such y,

$$\mathrm{L}_S(f_\xi) \ge \sup_{\|w\| = 1} |\nabla_s f_\xi(y) \cdot w| = \|\nabla_s f_\xi(y)\|.$$

Taking the supremum over $y \in S^\circ$ yields $\mathrm{L}_S(f_\xi) \ge \sup_{y \in S^\circ} \|\nabla_s f_\xi(y)\|$. Since $f_\xi \in C^1(U)$, $\|\nabla_s f_\xi\|$ is continuous, and therefore

$$\sup_{y \in S^\circ} \|\nabla_s f_\xi(y)\| = \sup_{y \in S} \|\nabla_s f_\xi(y)\| = M.$$

Hence $\mathrm{L}_S(f_\xi) \ge M$, completing the proof. □

Consequence for an ideal line detector. In the C^1 setting above, the supremum of finite differences over S coincides with the supremum of gradient magnitudes:

$$\sup_{\substack{u,v\in S\\ u\neq v}} \frac{|f_\xi(v)-f_\xi(u)|}{\|v-u\|} = \sup_{y\in S} \|\nabla_s f_\xi(y)\| = \mathrm{L}_S(f_\xi). \tag{S19}$$

Therefore, if we define an *ideal* line detector over a neighborhood S as the threshold test $\sup_{y\in S}\|\nabla_s f_\xi(y)\| \geq t$, then under the C^1 assumptions this is equivalent to testing whether there exists a pair $(u,v) \in S\times S$ such that $\frac{|f_\xi(v)-f_\xi(u)|}{\|v-u\|} \geq t$. Consequently, as we sample more and more pairs (u,v) from a distribution with full support, our stochastic detector converges to this ideal detector (up to the measure-zero boundary case $t = \mathrm{L}_S(f_\xi)$).

Beyond C^1: what "exactness" means for C^0 or discontinuous signals. When f_ξ is not C^1, the right-hand side $\sup_{y\in S}\|\nabla_s f_\xi(y)\|$ may be undefined, so the equality in Eq. (S19) cannot be interpreted literally. However, the left-hand side remains well defined and characterizes the maximum slope in a generalized sense: it defines the (possibly infinite) Lipschitz constant $\mathrm{L}_S(f_\xi)$.

If f_ξ is *Lipschitz* but not everywhere differentiable (e.g. piecewise smooth with kinks), then f_ξ is differentiable almost everywhere by Rademacher's theorem [Rademacher 1919; Evans and Gariepy 2025], and $\mathrm{L}_S(f_\xi)$ coincides with the *essential supremum* of $\|\nabla_s f_\xi\|$. In this regime, the finite-difference search remains consistent with the intuitive "largest local slope" criterion even though classical gradients fail on a measure-zero set.

If f_ξ has jump discontinuities on S (as can occur at visibility boundaries, silhouettes, or other topological changes in path geometry), then $\mathrm{L}_S(f_\xi) = \infty$, and finite differences can become arbitrarily large for point pairs straddling the jump. Accordingly, for any finite threshold t the detector becomes exact in the limit as well: it will almost surely, under dense sampling, succeed in finding pairs whose finite difference exceeds t. This matches our practical interpretation that discontinuities correspond to "arbitrarily large" gradients and should robustly trigger line responses, even though classical derivatives do not exist there.

S1.3 Finite difference search convergence

The problem in Section 4.5 of the main paper is an extremal decision problem rather than an integration problem: we want to determine whether there exists a point in S at which the gradient magnitude exceeds a threshold t. Let

$$G(y) := \|\nabla_s f_\xi(y)\|, \qquad A_t := \{y \in S : G(y) > t\}. \tag{S20}$$

In pointwise gradient estimation [West 2021], i.i.d. points $y_i \sim \mathrm{Uniform}(S)$ are drawn and $G(y_i)$ is estimated by some local procedure. The probability of detection in a single trial is therefore governed by the ratio of the measures of A_t and S:

$$q := \mathbb{P}(y \in A_t) = \frac{\mu(A_t)}{\mu(S)}. \tag{S21}$$

After N samples, the detection probability is $1-(1-q)^N$. Thus, convergence is controlled directly by the area of the high-gradient set.

In finite-difference lower-bound search, we instead draw pairs (u,v) and evaluate the slope $\frac{|f_\xi(v)-f_\xi(u)|}{\|v-u\|}$. The exact success probability now depends on the sampling distribution of segments and on how they align with the high-gradient region, but crucially it is controlled by the chance that a sampled segment *intersects* a high-gradient neighborhood, rather than by the area of A_t alone. When A_t is thin, sparse, or concentrated near lower-dimensional structures, this event has a much higher chance of occuring.

In the worst case, A_t may have arbitrarily small or even zero area, so q can vanish, while the probability that a random segment probes a region around A_t remains substantially larger. Consequently, the expected number of samples required for detection behaves like

$$\mathbb{E}[N_{\text{search}}] \sim \frac{1}{p}, \qquad \mathbb{E}[N_{\text{estimate}}] \sim \frac{1}{q}, \tag{S22}$$

with $p \gg q$ in precisely the rare-event regimes that matter for line detection. This is why the search formulation in the main paper converges more reliably than directly estimating gradient magnitudes at random points. See the exposition in Section S1.4 for further details about performance.

S1.4 Implementation details

Sampling auxiliary paths. For each base path we want to detect lines for, our feature-line estimator relies on finite differences evaluated between a base path and a set of image-space-offset auxiliary paths. A direct implementation of the mathematical construction would generate a full set of auxiliary paths *from scratch* at every stylization evaluation, which quickly becomes prohibitive as style functions are evaluated at many vertices along a path. Instead, we construct and reuse auxiliary *edges* incrementally inside the same depth-first recursion used to sample the SRE path tree (see Fig. 1). This technically can violate the independence requirement of samples used to estimate the weight terms in Eq. (34), however we observed the effect is minor, and further masked by the early-exit algorithm discussed below, where it is very rare for samples to be shared across more than a few weight term estimates.

At the first camera edge, we draw a fixed number N_{aux} of auxiliary image-space coordinates s' in a line stencil (e.g. disk or other radially symmetric shape) around the base pixel s (we use stratified sampling for variance reduction). Each auxiliary pixel s' spawns an auxiliary camera ray that shares the *same camera-side conditional event* as the base ray (e.g., the same thin-lens aperture sample when depth of field is enabled), differing only in its image coordinate. We intersect these auxiliary rays once to obtain a set of auxiliary first-hit vertices. The resulting set of pairs $(\mathbf{x}_0, \mathbf{x}'_1)$ is stored as an `AuxiliaryPathFrame`.

Amortized sampling cost via auxiliary edge reuse. As the base path is extended by BSDF sampling or next-event estimation, we extend the auxiliary set *one edge at a time*: for each new base edge $(\mathbf{x}_i, \mathbf{x}_{i+1})$, we generate N_{aux} corresponding auxiliary continuation edges $(\mathbf{x}'_i, \mathbf{x}'_{i+1})$ and store them in a new `AuxiliaryPathFrame` pushed on a stack. Since we sample paths depth-first, pushing/popping these frames naturally matches recursion: auxiliary edges for a prefix are reused by all subpaths of that prefix, and memory remains $O(N_{\text{aux}} \cdot \text{depth})$.

Preserving conditional events via random-number replay. A key requirement of our formulation is that finite differences correspond to a *partial* variation in image-space at fixed conditional realization. In the implementation, this is enforced by reusing the same random variables ξ that produced each scattering event on the base path when generating the auxiliary continuation edges. In addition to lens sample and half-vector copy, we record the sampler position immediately before sampling the base event, and for each auxiliary edge we *rewind* the sampler to that position, sample the auxiliary event, and then restore the sampler. This "replay" strategy keeps camera and BSDF-side conditional variables fixed while allowing image-space coordinates to change.

Constructing auxiliary continuation edges. Given a base vertex $\mathbf{x}_i$ with incoming direction ω_i and sampled outgoing direction ω_o, we generate the auxiliary outgoing direction at the corresponding auxiliary vertex $\mathbf{x}'_i$ by preserving the realized scattering *event* as follows:

- **Microfacet / conditionally specular events.** For glossy events treated as conditionally specular, the base scattering is represented by a microfacet half-vector $\mathbf{h}$ derived from (ω_i, ω_o). We express $\mathbf{h}$ in an incoming-direction–oriented local basis at $\mathbf{x}_i$ and reconstruct it in the corresponding basis at $\mathbf{x}'_i$ (with the auxiliary incoming direction and auxiliary shading normal). This preserves the tilt $c = \mathbf{h}\cdot\mathbf{n}$ and canonically identifies the azimuthal degree by transporting the tangential component through the change of local frame. The auxiliary outgoing direction then follows from specular reflection about the transported half-vector.
- **Delta-specular events.** For perfectly specular materials, we sample the auxiliary event using the same random variables as the base event (via replay), so the reflection/refraction branch and any additional discrete choices match by construction.
- **Non-microfacet / diffuse-like events.** For diffuse-like cases (where a conditional-specular interpretation is not available), we use a geometric fallback that transports the outgoing direction by the minimal rotation aligning base and auxiliary shading normals. This does not impose a physical microfacet event, but yields a stable correspondence for finite differences in practice.

For robustness, we terminate auxiliary continuations when the correspondence becomes unreliable (e.g., if an auxiliary ray hits a different object than the base ray at that depth, or if shading normals differ beyond a threshold), and simply exclude those auxiliary edges from subsequent finite-difference tests.

Computing line styles. In our renderer, line rendering is implemented as an ordinary SRE style function that can be applied *in addition to* any per-surface, per-material, or global stylization. We support multiple line types through a dictionary-style composition: each line type specifies its own metric function, stencil/region in image-space, and line appearance parameters. During evaluation, we apply (i) the material's local style function (including any material-bound line styles), followed by (ii) an optional global "feature line dictionary" that decides whether any line type triggers for the current edge. Priority between line types is artist-controlled; a useful default is front-to-back ordering so nearer occluding lines override farther ones.

Different line types may require different stencil sizes. In the current implementation we draw a single auxiliary set from a disk whose radius is the maximum stencil radius across the active line types (and any additional line-width constraints). This guarantees that every line type can reuse the same auxiliary samples, and each line type simply ignores auxiliary pixels that fall outside its own stencil. While this "bounding disk" strategy can waste samples when stencils differ greatly, it keeps the estimator simple and amortizes auxiliary tracing across all line types. More specialized stencil-aware densities (e.g., mixtures over stencils) are compatible with the same framework and can be added as an importance-sampling refinement.

Each line type uses the same core estimator: draw n many finite-difference comparisons over all the samples available in the current `AuxiliaryPathFrame` (plus the base edge), and early-exit as soon as any comparison exceeds threshold (since a single "hit" makes the product estimator zero). Dictionary evaluation therefore short-circuits quickly in the common case where strong line-producing gradients exist, and the dominant overhead becomes the cost of tracing N_{aux} auxiliary edges per base edge—comparable in spirit to cone/beam-based auxiliary sampling, but here generated by an event-preserving, curvature-aware transport rule. Our overall performance overhead is similar to West [2021], with one exception: [West 2021] compare the base path with each auxiliary path, while we compare *every pair* of paths within the stencil. This allows our line detection to maximize the benefit of treating line rendering as a lower bound-search, but comes at increased cost for line detection.

Cone sampling of West [2021]. Since our method generalizes prior ray-based methods, we can easily implement West [2021] under our framework. The only missing part is the cone sampling of West [2021], which we can do using the same `AuxiliaryPathFrame` / tree-sampling algorithm above. For a given base edge, we can compute the summed edge length back to the camera, and place the apex of a virtual cone at that distance behind the base edge. To sample a next set of auxiliary edges, we can copy the end vertex of each previous auxiliary edge as the start vertex of the new auxiliary edges, and ray cast from these new start vertices outward in the direction of the cone apex towards the new edge start vertices to find the new auxiliary edge end vertices.

S1.5 Discussion

Anisotropic microfacet models. For simplicity, the parallel transport of half-vectors discussed in the main paper assumes an isotropic microfacet, for which the normal distribution function is invariant under rotations in the tangent plane. For anisotropic microfacet models, the distribution is instead defined relative to a local tangent frame $(\mathbf{t}_1^*, \mathbf{t}_2^*) \in T_{\mathbf{x}}S$, and conditioning on a microfacet realization implicitly fixes both the tilt of the half-vector and its azimuth relative to this frame.

In this case, parallel transport must be applied to the full tangent frame rather than a single tangential direction. Concretely, at a base vertex $\mathbf{x}_i$ we condition on a half-vector expressed as,

$$\mathbf{h}_i = c_i\,\mathbf{n}_i + s_i(\cos\alpha_i\,\mathbf{t}^*_{1,i} + \sin\alpha_i\,\mathbf{t}^*_{2,i}), \tag{S23}$$

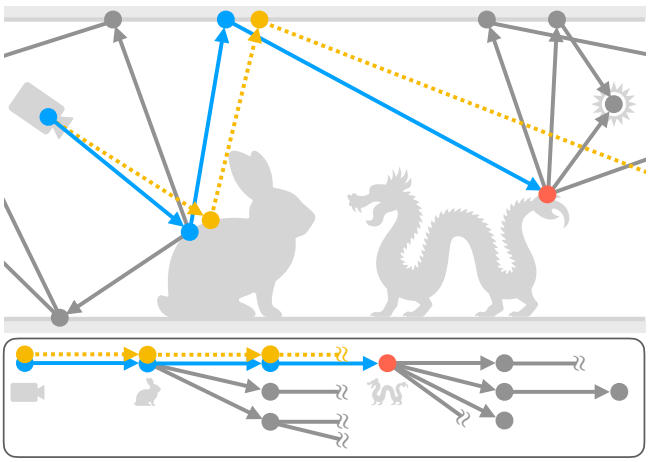

Fig. 1. As we traverse an SRE path-tree sample (gray), we follow one *geometric path* (blue) and construct its image-space–offset *auxiliary* counterparts (orange, dotted) *geometrically*, edge-by-edge, in lockstep with the base path. A fixed set of auxiliary image-space offsets is spawned at the camera and then extended whenever the base path advances, so auxiliary prefixes are shared by all descendant subpaths in the depth-first recursion—amortizing auxiliary tracing and keeping memory proportional to the number of auxiliary samples and the path depth.

and transport both basis vectors $\mathbf{t}^*_{1,i}$ and $\mathbf{t}^*_{2,i}$ via Levi–Civita parallel transport to the auxiliary vertex $\mathbf{x}'_i$. The transported half-vector $\mathbf{h}'_i$ is then reconstructed using the same coefficients (c_i, s_i, α_i) in the transported frame.

The specular case. Purely specular paths require no special handling: specular reflection has no azimuthal degree of freedom, and the operator $\mathcal{T}_{\delta s}$ reduces to classical image-space parallelism in the specular limit. Our formulation therefore recovers prior ray-based line rendering behavior as a special case.

Relation to half-vector copy. Under a fixed conditional-specular parametrization and infinitesimal image-space perturbations, parallel transport of microfacet half-vectors is equivalent to half-vector copy between local charts [Jakob and Marschner 2012; Kettunen et al. 2015]. Outside of this regime (e.g., non-view-oriented charts, or varying parametrizations) the two constructions differ. Framing the construction as a designed operator $\mathcal{T}_{\delta s}$ clarifies its geometric meaning and ensures consistent correspondence beyond a particular chart choice or infinitesimal regime.

Non-microfacet materials. Our formulation is principled for scattering models whose randomness can be parameterized by conditional variables that admit a conditional-specular interpretation, including microfacet BSDFs and thin-lens cameras. For materials such as Lambertian diffuse reflection, there is no such conditional-specular parameterization that can be preserved by $\mathcal{T}_{\delta s}$ in the same sense. In such cases, we adopt a pragmatic approximation: for glossy-like materials we copy the half-vector as if they were based on an NDF, and for diffuse-like materials we transport the outgoing direction between the local charts of the base and auxiliary paths. While this approximation lacks a strict physical interpretation, it maintains stable finite-difference comparisons in practice. Exploration into theoretically-grounded support for materials that do not have a conditionally-specular interpretation is left to future work.

Computing n_θ for target blend weights w_θ. Directly controlling line opaqueness through the sample count n_θ can be unintuitive. A more artist-intuitive approach is to pick a target blend weight w_θ, approximate the failure rate for a single finite difference $M_\theta = \int_{S^2_\theta} m_\theta(u,v)p_\theta(u,v)\,\mathrm{d}u\mathrm{d}v$, and solve for $n_\theta = \log_{M_\theta}(w_\theta)$. Non-natural number sample counts n_θ can then be estimated via a power-series estimator or, as a rough but monotonic approximation, stochastically rounded at each evaluation.

Line stencils. Our gradient search-based line detection leveraged symmetry in the line stencil region S to produce line renderings without actually ever knowing *where* the discontinuities are in the image-plane. This lets us use any radially symmetric region S as a line stencil. (e.g. in the second robot from the left in Fig. 11 we use a square line stencil for sharper corners). An interesting avenue of future work would be how to incorporate non-radially symmetric stencils.

Line metric function. The metric function determines where lines should appear and can be any function that evaluates to 0 at edges and 1 elsewhere. In practice, the essential point is that it can be used to compute where lines should be drawn (and can be used to compute an expectation of "failure"). For example, we can use a metric that also considers a minimum threshold on the primal value to prevent fine details from producing lines.

Line color. The line color $c_\theta(\mathbf{y}, \mathbf{z})$ need not be constant, and is fully parameterized by θ and operates over the edge $(\mathbf{y}, \mathbf{z})$. For example, we can modulate the brightness of line proportional to the exitant radiance, or produce lines that act like emitters [West 2021] (see Fig. 11).

Line detection deeper in a path. It can be desirable to test for lines only at a certain path depth. In this situation we can filter out auxiliary paths that would have produced a line when compared to the base path at earlier edges. The resulting set of auxiliary paths are generally strongly correlated and often traverse exactly the same seqeunce of smooth (or near-smooth) manifolds. We leverage this when drawing lines in reflections to reduce spurious artifacts from de-correlated paths.

Integration with NL-NRC. As our line and tone rendering methods are ordinary style functions they are (mostly) compatible with NL-NRC [Tong and Hachisuka 2025] out-of-the-box.

Required changes would be adding the auxiliary path sampling (line rendering) and the anchor mapping and local inversion (tone rendering), and providing the auxiliary path data and inverted image-space coordinate to the style function's parametrization.

However, training for line rendering may be slow. The parametrization $\mathcal{S}$ (in [Tong and Hachisuka 2025] notation) is unique to the vertex (e.g. $v.\mathcal{S}$ in Listing 3). For our line and tone rendering, the parametrization is unique to the path taken up to v, and the resulting stylization is sensitive to even small changes in that path. As a

result, training can take longer than style functions that are only dependent on e.g. position and path depth.

S2 TONE FOUNDATIONS AND IMPLEMENTATION DETAILS

This section supports the tone construction in Section 5 of the main paper. We begin with the analytic continuation theorem cited from the main text, then turn to the local approximation machinery used in practice, followed by specific remarks on Möbius extrapolation, Linear MLS, and the implementation of the tone pipeline.

S2.1 Analytic continuation for holomorphic mappings from image-space to path-space

This subsection provides the formal theorem for the existence and uniqueness of an extension map $\widetilde{\phi_k}$ referenced in Section 5 of the main paper.

Theorem S2.1. *Let M be a connected oriented Riemann surface, which is a subspace of the surface manifold $\mathcal{M}$, and let $S \subseteq \mathbb{C}$ be a connected region. Let $\phi : S \to M$ be a holomorphic, orientation-preserving conformal map. Write $\phi(S) \subseteq M$ for its image. Then there exists a connected oriented Riemann surface $S_{\max}$ and a holomorphic embedding $\iota : S \hookrightarrow S_{\max}$, together with a holomorphic map $\widetilde{\phi} : S_{\max} \to M$ such that*

- *$\widetilde{\phi} \circ \iota = \phi$ on S;*
- *if $\psi : S' \to M$ is any holomorphic map extending ϕ, then there is a unique holomorphic $h : S' \to S_{\max}$ with $\psi = \widetilde{\phi} \circ h$;*
- *$\widetilde{\phi} : S_{\max} \to M$ is onto;*
- *$\widetilde{\phi}$ is unique up to deck transformations.*

Proof. Choose a holomorphic atlas $\{(U_\alpha, z_\alpha)\}_{\alpha \in I}$ on M, so the transitions $z_\beta \circ z_\alpha^{-1}$ are holomorphic for intersecting U_α and U_β. Since ϕ is conformal and orientation-preserving, for each chart we set

$$f_\alpha := z_\alpha \circ \phi : \ \phi^{-1}(U_\alpha) \cap S \longrightarrow \mathbb{C}.$$

Note that each f_α is holomorphic and has non-vanishing derivative.

Now let us fix a $p_0 \in S$. For any point $x \in M$, take a piecewise analytic curve $\gamma : [0,1] \to M$ with $\gamma(0) = \phi(p_0)$ and $\gamma(1) = x$. Cover γ by chart domains $U_{\alpha_1}, U_{\alpha_2}, \cdots$. By one-variable analytic continuation applied to the f_{α_j} and the holomorphic transition maps $z_{\alpha_{j+1}} \circ z_{\alpha_j}^{-1}$, the germ of f_{α_1} uniquely extends along γ as long as f_{α_j} has nonzero derivative. The identity theorem implies that for any other curve $\bar{\gamma}$ with $\bar{\gamma}(0) = p_0$ and $\bar{\gamma}(1) = x$, the extensions along γ and $\bar{\gamma}$ match up to deck transformations.

Next, we glue all such germs of the analytic continuations of ϕ to construct a connected Riemann surface $S_{\max}$ together with a natural projection

$$\widetilde{\phi} : S_{\max} \longrightarrow M,$$

sending a germ to its endpoint in M. By construction $\widetilde{\phi}$ is holomorphic with non-vanishing derivative, hence a local biholomorphism. The original map ϕ identifies S with an open subset of $S_{\max}$, providing the embedding ι and the extension property $\widetilde{\phi} \circ \iota = \phi$.

Now, if $\psi : S' \to M$ were to be another holomorphic extension of ϕ, then ψ agrees germ-by-germ with the continuations used to build $S_{\max}$; this produces a unique holomorphic map $h : S' \to S_{\max}$ with $\psi = \widetilde{\phi} \circ h$.

By construction we continued along every path in M starting at $\phi(p_0)$. The image $\widetilde{\phi}(S_{\max})$ is therefore the connected component of M containing $\phi(S)$; since M is connected and $\phi(S) \neq \emptyset$, the map $\widetilde{\phi}$ is onto.

Finally, $\widetilde{\phi}$ is a covering map, and uniqueness up to deck transformations follows from the usual uniqueness of coverings with the same germ on S. □

Uniformization and special cases for $S_{\max}$. By the Uniformization theorem, there is a universal covering $\pi : \widetilde{M} \to M$, where the universal cover $\widetilde{M}$ is conformally equivalent to either the Riemann sphere $\mathbb{S}^2$, complex plane $\mathbb{C}$ or the open unit disk $\mathbb{D}$. Since $\widetilde{\phi} : S_{\max} \to M$ is a covering, there exists a covering $\widetilde{\pi} : \widetilde{M} \to S_{\max}$ with $\widetilde{\phi} \circ \widetilde{\pi} = \pi$. Equivalently, $S_{\max} \cong \widetilde{M}/\Gamma$ for some subgroup Γ of the deck group of π.

In the context of geometry processing and mapping, we can view $S_{\max}$ as a maximally "unwrapped" parameterization domain. If the monodromy subgroup Γ is trivial (i.e., there are no periodic boundary conditions or gluing at the borders), then $S_{\max} \cong \widetilde{M}$ itself. Intuitively, this means that the maximal continuation domain flattens out naturally to cover the entire ambient space ($\mathbb{S}^2$, $\mathbb{C}$ or $\mathbb{D}$) without overlapping with itself.

Real analytic metric instead of holomorphic. While Theorem S2.1 relies on the complex analytic formulation of holomorphic functions, in practice, tone mapping operates on real-valued domains. Fortunately, our framework can be directly exended to real spaces: if our initial ϕ is a local isometry for a real-analytic metric, the argument goes through word-by-word with "holomorphic" replaced by "local isometry." By using unique continuation for real-analytic elliptic systems, one again obtains a maximal extension that is a covering map unique up to deck transformations. Intuitively, this guarantees that our tone mapping extensions remain consistent even when operating entirely within real-valued spaces.

S2.2 Local approximations of analytic continuity

We discussed in Section 5.2 of our main paper that an analytic continuation of the mirror reflection $\widetilde{\phi_k}$ theoretically exists. Directly computing it, however, is impractical. Extending ϕ_k explicitly following the steps of Theorem S2.1 would require repeatedly solving the Cauchy–Riemann equations over small steps, carefully managing step sizes, boundary conditions, and numerical stability. Errors accumulate rapidly with distance from the known domain, making the resulting mapping unreliable and computationally expensive. For these reasons, we do not pursue exact analytic continuation, but instead develop a local, sampling-based approximation that captures its essential behavior while remaining practical for Monte Carlo rendering. Fortunately, this problem is simply an interpolation/extrapolation problem which we can write precisely as follows: Given a path vertex $\mathbf{x}$ with no known $\phi_k^{-1}(\mathbf{x})$, and "close-by" vertices $\mathbf{x}_1, \ldots, \mathbf{x}_\ell \in M$ with known pre-images $\phi_k^{-1}(\mathbf{x}_1), \ldots, \phi_k^{-1}(\mathbf{x}_\ell) \in \mathbb{R}^2$, find a consistent extension $\tilde{\phi}_k$ and an image-space coordinate $\tilde{\phi}_k^{-1}(\mathbf{x}) \in \mathbb{R}^2$.

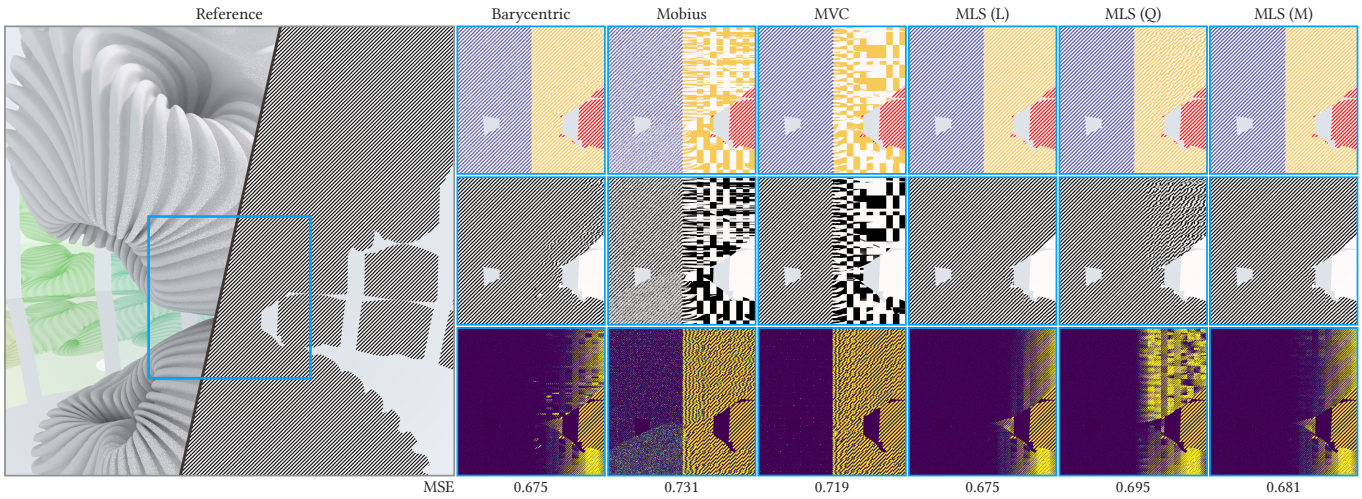

Fig. 2. Comparison of different interpolation/extrapolation methods introduced in Section S2.2.1. Samples used in tone interpolation/extrapolation are only generated for the left side of the image-space, such that the further right on the image-space, the farther the extrapolation distance. The reference scene shows a geometrically complex object and a hatching pattern on the image-space that lives on the object. This object is placed on a spherical mirror, with more curved mirrors in the background so that reflected hatching patterns should be successively curved. First row of the right half shows three regions in blue, yellow and red, corresponding to regions where the method interpolates from nearby samples, extrapolates from nearby samples, or fails to do either due to local degeneracy. Second row shows the interpolated/extrapolated tones, and third row the error heatmaps. All methods except Möbius work comparably well on the interpolation region, whereas Möbius is unable to get a consistent extension mapping due to the complex geometry of the manifold. Barycentric, while simple, shows warping in the extrapolation region when the choice of samples are ambiguous. Möbius and MVC both fail to provide good extrapolations due to degeneracy to sample singularities at farther distances. The MLS methods visually give the most appealing results, although incur more error at farther distances due to non-conformality. While all the MLS methods and Barycentric have similar MSE, we note that MLS (L) gives the most visually consistent and appealing tone extension.

Many works in computer graphics have adapted different techniques to attack this problem. Here we describe these techniques, and also suggest a conformal extension technique which we call the Möbius extrapolation. In the rest of this exposition, we assume ℓ is large enough for executing each of the methods (for example, $\ell \geq 3$ for Barycentric). When fewer than the required number of samples are available, our implementation gracefully falls back to an alternative with smaller sample requirement. When interpolation or extrapolation is impossible due to local degeneracy (i.e. no suitable nearby samples or methods requiring inversion of degenerate matrices), our implementation falls back and returns the path's originating image-space coordinate.

S2.2.1 Practical Interpolation/Extrapolation Methods. We now go over the different extrapolation techniques used in this work.

Barycentric extrapolation. Perhaps the simplest way to find a image-space coordinate corresponding to a path vertex $\mathbf{x}$ is to take the three "nearest" points $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$, project them onto the plane through $\mathbf{x}$ normal to the ray entering $\mathbf{x}$, and if π is this projection satisfying

$$\mathbf{x} = c_1\pi(\mathbf{x}_1) + c_2\pi(\mathbf{x}_2) + c_3\pi(\mathbf{x}_3), \tag{S24}$$

and then define $\tilde{\phi}_k^{-1}(\mathbf{x}) := c_1\phi_k^{-1}(\mathbf{x}_1) + c_2\phi_k^{-1}(\mathbf{x}_2) + c_3\phi_k^{-1}(\mathbf{x}_3)$.

Möbius extrapolation. Since the Barycentric extrapolation is not conformal, one can instead try to take the three closest points $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ and construct the unique Möbius map (complex mapping of the form $f(z) = \frac{az+b}{cz+d}$) that takes points $\phi_k^{-1}(\mathbf{x}_1), \phi_k^{-1}(\mathbf{x}_2), \phi_k^{-1}(x_3)$ to $\pi(\mathbf{x}_1), \pi(\mathbf{x}_2), \pi(\mathbf{x}_3)$ respectively. This map is invertible and angle-preserving by definition. More details about first order and second order Möbius extrapolation is provided in Section S2.4.

Mean Value Coordinates. A similar technique to barycentric coordinates, known as Mean Value Coordinates was introduced in computer graphics by Floater [2003], where Eq. (S24) is simply generalized to ℓ vertices:

$$\mathbf{x} = c_1\pi(\mathbf{x}_1) + \cdots + c_\ell\pi(\mathbf{x}_\ell), \tag{S25}$$

and the coefficients c_i are now derived from the mean value theorem for harmonic functions.

Linear Moving Least Squares (MLS (L)). Linear MLS constructs $\tilde{\phi}_k^{-1}(\mathbf{x})$ by fitting, in a local neighborhood of $\mathbf{x}$, a low-degree polynomial that approximates the known image-space coordinates in a weighted least-squares sense [Lancaster and Salkauskas 1981; Levin 1998]. Let $\mathbf{p}_i = \pi(\mathbf{x}_i) \in \mathbb{R}^2$ denote the projected positions, $s_i = \phi_k^{-1}(\mathbf{x}_i) \in \mathbb{R}^2$ the corresponding image-space samples, and $\mathbf{p} = \pi(\mathbf{x})$ the projection of the query point. In *linear MLS*, we fit an affine map $\mathbf{u}(\mathbf{p}) \approx \mathbf{A}\mathbf{p} + \mathbf{b}$ by minimizing

$$\min_{\mathbf{A},\mathbf{b}} \sum_{i=1}^{k} w_i(\mathbf{x}) \left\| \mathbf{A}\,\mathbf{p}_i + \mathbf{b} - \mathbf{u}_i \right\|^2, \tag{S26}$$

where $w_i(\mathbf{x})$ are spatial weights that decay with $\|\mathbf{p}-\mathbf{p}_i\|$. We provide the details of our exact Linear MLS implementation in Section S2.5.

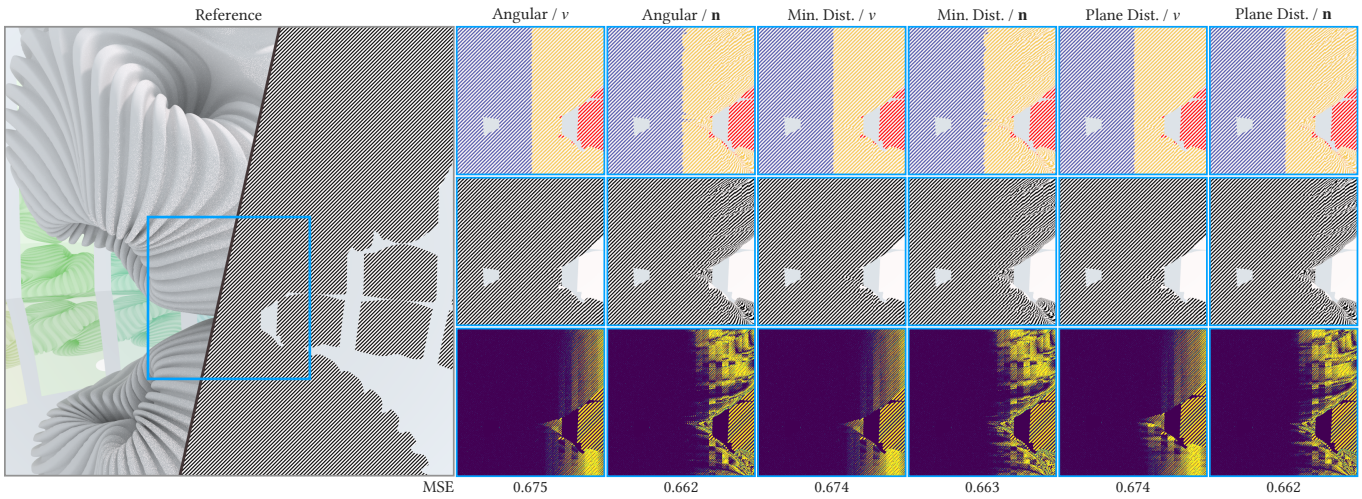

Fig. 3. Comparison of the three sorting methods and two plane orientation choices described in Section S2.2.1. All combinations are robust in interpolation tasks, and have similar errors in this inset, but the most visually accurate render is obtained by using the combination of angular distance $d_\angle$ and local chart oriented along the incoming view direction v.

Quadratic Moving Least Squares (MLS (Q)). Quadratic MLS extends this construction by computing a fit over a quadratic (instead of linear) polynomial basis in $\mathbf{p}$ (including terms such as p_x^2, $p_x p_y$, p_y^2), yielding higher-order accuracy at the cost of a slightly larger linear system [Joldes et al. 2015].

Mixed Moving Least Squares (MLS (M)). Since quadratic MLS extrapolation leads to geodesic artifacts and is generally unstable in extrapolation at large distances, one can also weight linear MLS and quadratic MLS according to the distance from the query point $\mathbf{x}$ to devise a *weighted linear-quadratic MLS* extrapolation.

Fig. 2 compares the performances of each of these local intrapolation/extrapolation techniques on a scene with a geometrically complex object placed on a specular curved surface, with several curved mirrors in the background so that reflected hatching patterns should be curved conformally in the reflections. While its form is simple, Barycentric does not preserve angles, and moreover, warp artifacts appear near boundaries where the choice of the "nearest three" out of all nearby samples becomes ambiguous. Möbius extrapolation and MVC, while theoretically appealing, bring far away points to one of the three vertices of the triangle (Möbius), or ℓ vertices of the polygon (MVC). This is why these methods effectively put all their weight on a single vertex, and therefore are unsuitable for extrapolation. The family of MLS extrapolators seem to perform the best in practice. Linear MLS is highly performant, and while it is not conformal, leading to an increase in rotational error for more distant extrapolation, the visuals produced are stable and visually indistinguishable. Quadratic MLS introduces geodesic warping artifacts, which are mitigated by MLS (M). This is why in practice, we use Linear MLS in all our renders.

S2.2.2 Choosing adequate nearby samples. Now that we have seen different ways one can locally approximate the analytic continuation map, we turn our attention to the problem of selecting the nearest ℓ samples to provide as inputs to these extrapolation algorithms. Given a path vertex $\mathbf{x}$ with no known $\phi_k^{-1}(\mathbf{x})$, we can have several other specular paths with the same prefix in the scene, say $\{\bar{\mathbf{y}}^{(i)}\}_{i=1}^N$ which carry image-space coordinates down to depth k, and lie within a bounded search radius. Moreover, since Linear MLS fits a local geometric chart around the query vertex $\mathbf{x}$ and builds an approximate ϕ_k^{-1} to the image-space, in practice one needs to fix not only a local chart around $\mathbf{x}$, but also a metric to compare $\mathbf{x}$ and the other N candidate vertices. In this subsection, we evaluate the performance of combinations of choices of local charts and metrics to select the nearest ℓ samples, which are then fed into our Linear MLS pipeline to produce visuals in Fig. 3.

There are two ways we can select a local chart around $\mathbf{x}$: one is the incoming direction v at $\mathbf{x}$ given by the k'th edge of the path (normal direction v), and another by simply considering the tangent plane to the manifold at $\mathbf{x}$ (normal direction $\mathbf{n}$). This choice alone is not enough for Linear MLS since N can technically be a large number. Therefore, we compare three simple ways to compute distances between the query point $\mathbf{x}$ and any other candidate vertex $\mathbf{y}_k$ (see Fig. 4).

(1) **Angular distance:** Let $\mathbf{x}'$ denote the $(k-1)$'th vertex on the path leading to $\mathbf{x}$. Then, the angular distance is defined as $d_\angle(\mathbf{x}, \mathbf{y}_k) = \angle(\mathbf{x}, \mathbf{x}', \mathbf{y}_k)$.
(2) **Minimum distance:** The minimum distance $d_m(\mathbf{x}, \mathbf{y}_k)$ is given by the orthogonal distance between the point $\mathbf{x}$ and ray $\mathbf{y}_{k-1}\mathbf{y}_k$.
(3) **Plane distance:** For a fixed plane P through $\mathbf{x}$ (in our case, P either has v or $\mathbf{n}$ as its normal), we compute the plane distance $d_P(\mathbf{x}, \mathbf{y}_k)$ by first intersecting the ray $\mathbf{y}_{k-1}\mathbf{y}_k$ with P, and then computing the Euclidean distance between $\mathbf{x}$ and the intersection point.

Fig. 3 compares the performance of the 6 different combinations of metrics and local charts. Although the numerical errors incurred by

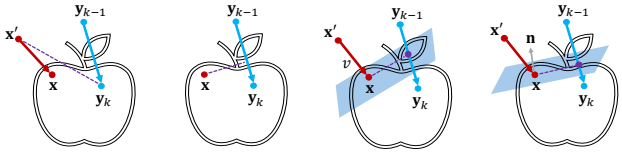

Fig. 4. Different ways of measuring the distance between a query point $\mathbf{x}$ and a candidate vertex $\mathbf{y}_k$. Here $\mathbf{x}'$ and $\mathbf{y}_{k-1}$ denote the $(k-1)$'th vertex on the path leading to $\mathbf{x}$ and $\mathbf{y}_k$, respectively.

each of these methods are similar, the angular distance $d_\angle$ and plane orientation v gives the most visually accurate results. Intuitively, since the mapping is effectively evolving through the scene in a view-dependent manner, the highly view-correlated angular distance measure and view-oriented plane v is stable over this evolution. Thus, we use this particular combination in our renders.

S2.3 Implementation details for Tone Rendering

Our tone pipeline requires evaluating an approximation of ϕ^{-1} at arbitrary path vertices. Because ϕ is defined implicitly by path tracing from the camera outwards and is generally partial, we implement ϕ^{-1} through a cache of mirror-anchor samples and a local, numerically stable continuation scheme.

Prepass: caching mirror-anchor samples. We begin with a prepass that traces one (or a small number of) mirror-reflection projection path(s) per pixel. Each projection path is generated by treating all interactions as perfectly specular yielding samples of the anchor map ϕ. If the original material supports transmission in addition to reflection we generate samples for both branches. For each visited vertex $\mathbf{x}$ along a projection path, we store a correspondence pair $(\mathbf{x}, s)$ where s is the originating image coordinate.

In practice we organize these samples by the *path prefix* (the sequence of intersected surfaces) and build a hybrid auxiliary spatial acceleration structure of a tree and spatial hashes at each node for fast candidate lookup. Specular path samples are stored vertex by vertex into a tree structure that separates each next level by approximately connected manifolds (we found separating by surface ID or submesh ID tends to work well without the need to compute connectivity). Tree nodes that exceed a certain number of stored vertices populate a spatial hash for the edges to reduce lookup overhead. Slightly counter-intuitively, we found using a 2D spatial hash based on image-space coordinate tended to work better than a 3D spatial hash based on vertex positions.

Query-time inversion via local continuation. When rendering a general path and encountering a vertex $\mathbf{x}$, we query the cache for candidate samples that (i) share a compatible prefix (i.e. follows the same traversal though the tree cache) and (ii) lie within a bounded search radius of $\mathbf{x}$. In practice we found a maximum search radius of 512 pixels for 1920x1080 renders tends to work well for all but extremely rough reflections on manifold boundaries. Each retrieved candidate then provides a nearby pair $(\mathbf{x}_i, s_i)$. We then approximate ϕ^{-1} locally by fitting an affine map from a local geometric chart around $\mathbf{x}$ to image-space via moving least squares (MLS).

Tangent-plane framing and weights. We found that the choice of local chart strongly affects stability under reflection/refraction and near grazing configurations. Rather than using the surface normal to define the tangent plane, we frame the chart using the *incoming direction* of the geometric path at $\mathbf{x}$. This produces a smoothly varying chart along transport and avoids chart discontinuities caused by normal variation (including shading normals) that are unrelated to the anchor mapping. For MLS weights in Eq. (S26) we use an isotropic Gaussian $w_i(\mathbf{x}) = \exp\big(-\|\pi(\mathbf{x}) - \pi(\mathbf{x}_i)\|^2/\sigma^2\big)$ in the local chart, with σ set adaptively to the mean distance of the selected candidates. For $\pi(\mathbf{x}_i)$ we use the position of the candidates on the local tangent plane. This provides continuous scale-adaptive smoothing without requiring hand-tuning per scene.

Fallback hierarchy for sparse or degenerate neighborhoods. The number and configuration of available candidates is stochastic and can become sparse near occlusion boundaries, silhouette regions, or under limited projection coverage. However, in order to compute the tone at a query point, it is necessary to specify an image-space coordinate via an approximate local inverse ϕ^{-1}. We therefore employ a hierarchy of fallbacks: directly viewed objects and specular paths from the camera are inverted in closed form; when enough well-conditioned samples are available, we use linear MLS; when samples are insufficient for a stable fit or there critical mathematical failures (e.g. the matrix in MLS is singular), we fall back to simpler local rules (affine/barycentric extrapolation for minimal sample sets, and nearest-neighbor assignment in degenerate cases). If no compatible candidates are found, we fall back to a conservative image-coordinate estimate (e.g., the first-edge projection), ensuring that tone evaluation remains continuous and well-defined even when ϕ^{-1} cannot be reliably inferred. However, the first-edge fallback and produce undesirable clear image-space structure in very rough reflections (see Fig. 5 for the performance and failure modes of our fallback hierarchy). Finding a more robust fallback regime would be a compelling avenue of future work.

The computed image-space coordinate. The computed image-space coordinate is then made available to style functions as a part of the parameters θ. This means that not just tone, but any style function has access to a anchor mapping image-space coordinate (e.g. effects like adjusting cel shading cutoffs based on image-space coordinates is now possible).

S2.4 Möbius extrapolation

A Möbius transform is the simplest class of conformal map from $\mathbb{C}$ to $\mathbb{C}$, which is a rational map of the form $z \mapsto \frac{az+b}{cz+d}$. We propose a way to approximate the analytic continuation of ϕ_k with a Möbius extrapolation as follows.

First, given the query vertex $\mathbf{x}$, we select three other path vertices $\mathbf{y}_k^{(1)}$, $\mathbf{y}_k^{(2)}$, $\mathbf{y}_k^{(3)}$ on the scene manifold. Let P be the plane perpendicular to the view direction at $\mathbf{x}$, and compute the ray-plane intersections $\mathbf{z}_i = \mathbf{y}_{k-1}^{(i)}\mathbf{y}_k^{(i)} \cap P$. Then, we think of $\{z_i\}$ lie on P as three points on the complex plane, and their preimages $\{\phi_k^{-1}(\mathbf{z}_i)\}$ as another three points on another copy of the complex plane. By a shift of origin, we can assume that they both have centroids at the origin (i.e. $\mathbf{z}_1 + \mathbf{z}_2 + \mathbf{z}_3 = 0$ and $\phi_k^{-1}(\mathbf{z}_1) + \phi_k^{-1}(z_2) + \phi_k^{-1}(z_3) = 0$.

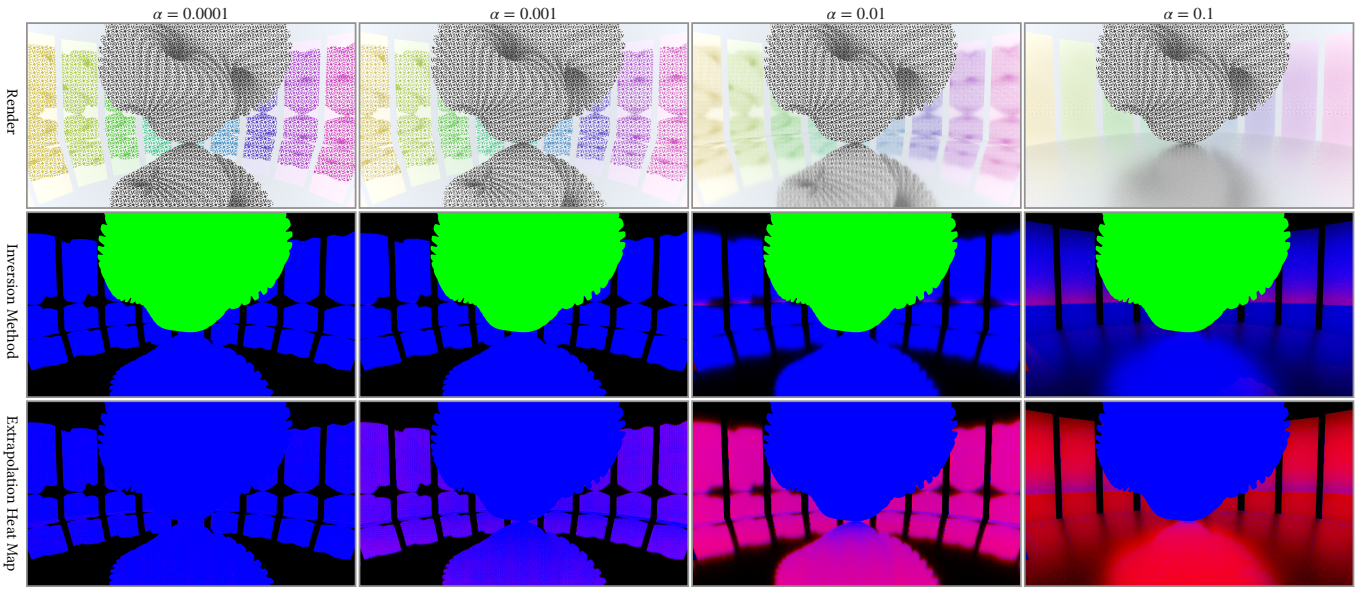

Fig. 5. As we increase roughness α in a Trowbridge-Reitz microfacet distribution the likelihood of selecting a fallback inversion strategy, as well as sample extrapolation, increases. In the top row we show the rendered result using our tone rendering method. In the far right column we see the tone pattern is (incorrectly) clearly visible in the rough reflection due to inversion failure. This failure is visualized as a color map in the second row, where among the analytic inversion (green) and approximate MLS inversion (blue), we see a clear band of red corresponding to inversion failure. In the bottom row, we see that the likelihood of a path sample inverted via MLS using interpolation (blue) goes down and extrapolation (red) goes up. Note that inversion failure is counted as "interpolation" so we see a blue band at the region of inversion failure.

Algorithm 1 then computes the unique Möbius transformation that maps $\mathbf{z}_i$ to $\phi_k^{-1}(\mathbf{z}_i)$.

Algorithm 1: Compute Möbius maps f and f^{-1}

Input : $x_1, x_2, x_3 \in \mathbb{C}$ with $x_1 + x_2 + x_3 = 0$ and $z_1, z_2, z_3 \in \mathbb{C}$ with $z_1 + z_2 + z_3 = 0$

Output: Möbius map f with $f(x_i) = z_i$, and its inverse f^{-1}

Set $a_1 \leftarrow (x_2 - x_3),\ b_1 \leftarrow -(x_2 - x_3)x_1$;

Set $c_1 \leftarrow (x_2 - x_1),\ d_1 \leftarrow -(x_2 - x_1)x_3$;

Set $a_2 \leftarrow z_1(z_2 - z_3),\ b_2 \leftarrow -z_2(z_1 - z_3)$;

Set $c_2 \leftarrow (z_2 - z_3),\ d_2 \leftarrow -(z_1 - z_3)$;

Set $\begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} a_2 & b_2 \\ c_2 & d_2 \end{pmatrix} \begin{pmatrix} a_1 & b_1 \\ c_1 & d_1 \end{pmatrix}$;

Set $f(x) \leftarrow \dfrac{ax + b}{cx + d}$ and $f^{-1}(w) \leftarrow \dfrac{dw - b}{-cw + a}$;

return *Maps* f, f^{-1}

Quadratic approximation. Since the Möbius extension technically fits a *local plane* around the point $\mathbf{x}$, it does not account for the curvature of the scene manifold M at $\mathbf{x}$. In general, the closest cached manifold vertices $\mathbf{y}_k^{(1)}, \mathbf{y}_k^{(2)}, \mathbf{y}_k^{(3)}$ are not coplanar in general due to curvature of the manifold, which is why we had to perform a ray-plane intersection to compute the coordinates $\mathbf{z}_i$ in the usual Möbius approximation.

Lemma S2.1 describes how using local curvature information, we can theoretically approximate the scene manifold M in a quadratic fashion similar in flavor to quadratic MLS. Implementation and analysis of quadratic and higher order Möbius approximations, and how they affect the scene is left as future work.

Lemma S2.1 (Local quadratic approximation). *Fix a $\mathbf{x} \in M$ for a surface M. Let $W = \{w_1, w_2\}$ be a basis of the tangent space $T_{\mathbf{x}}M$, and $G = \begin{bmatrix} \langle w_1, w_1 \rangle & \langle w_1, w_2 \rangle \\ \langle w_1, w_2 \rangle & \langle w_2, w_2 \rangle \end{bmatrix}$ be the first fundamental form in W-basis. Let $G = U^\top \Lambda U$ be the spectral decomposition of G with $\Lambda = \operatorname{diag}(\lambda_1, \lambda_2)$. Let $G^{-1/2} := U^\top \operatorname{diag}(\lambda_1^{-1/2}, \lambda_2^{-1/2})U$. Then the vectors $\hat{e}_i := \sum_{j \in \{1,2\}} (G^{-1/2})_{ij} w_j$ form an orthonormal basis of $T_{\mathbf{x}}M$.*

Suppose the second fundamental form is given by the matrix $H = \begin{bmatrix} L & M \\ M & N \end{bmatrix}$ in the $\{\hat{e}_1, \hat{e}_2\}$-basis. Suppose $\hat{n}$ is the unit normal chosen so that $\{\hat{e}_1, \hat{e}_2, \hat{n}\}$ is positively oriented. Then, locally,

$$\psi(u, v) \approx y + u\hat{e}_1 + v\hat{e}_2 + \frac{1}{2}(Lu^2 + 2Muv + Nv^2)\hat{n} \tag{S27}$$

gives a map $\psi : \mathbb{R}^2 \to \mathbb{R}^3$ which is locally invertible. The inverse of ψ is given around $\mathbf{x}$ by the map $\psi^{-1}(z) \approx (\langle z - \mathbf{x}, \hat{e_1} \rangle, \langle z - \mathbf{x}, \hat{e_2} \rangle)$.

S2.5 Linear Moving Least Squares for $\tilde{\phi}_k^{-1}$

Given a set of sample correspondences

$$\{(\mathbf{x}_i, \mathbf{s}_i)\}_{i=1}^{\ell}, \quad \mathbf{x}_i \in \mathbb{R}^3, \quad \mathbf{s}_i = (u_i, v_i) \in \mathbb{R}^2,$$

where each $\mathbf{x}_i$ is a 3D point and $\mathbf{s}_i = \tilde{\phi}_k^{-1}(\mathbf{x}_i)$ is its associated image-space coordinate, we seek to estimate the image-space coordinate corresponding to a query point $\mathbf{y} \in \mathbb{R}^3$.

Local reference plane. A local plane is defined through $\mathbf{y}$ with normal

$$\mathbf{n} = -\frac{\mathbf{d}}{\|\mathbf{d}\|},$$

where $\mathbf{d}$ is the incoming direction at the query point.

Let the centroid of the sample points be

$$\mathbf{z}_c = \frac{1}{\ell}\sum_{i=1}^{\ell}\mathbf{x}_i.$$

PCA-based local parameterization. Define centered sample offsets

$$\mathbf{d}_i = \mathbf{x}_i - \mathbf{z}_c.$$

The covariance matrix is

$$\mathbf{C} = \frac{1}{\ell}\sum_{i=1}^{\ell}\mathbf{d}_i\mathbf{d}_i^{\top}.$$

Let $\mathbf{t}_1$ be the eigenvector of $\mathbf{C}$ corresponding to the largest eigenvalue. A tangent basis for the plane is constructed as

$$\mathbf{e}_1 = \frac{\mathbf{t}_1 - (\mathbf{t}_1\cdot\mathbf{n})\mathbf{n}}{\|\mathbf{t}_1 - (\mathbf{t}_1\cdot\mathbf{n})\mathbf{n}\|}, \qquad \mathbf{e}_2 = \mathbf{n}\times\mathbf{e}_1.$$

Projection to plane coordinates. Each sample point is projected into local 2D plane coordinates:

$$\mathbf{p}_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} (\mathbf{x}_i - \mathbf{z}_c)\cdot\mathbf{e}_1 \\ (\mathbf{x}_i - \mathbf{z}_c)\cdot\mathbf{e}_2 \end{bmatrix}.$$

The query point is similarly projected:

$$\mathbf{p}_q = \begin{bmatrix} x_q \\ y_q \end{bmatrix} = \begin{bmatrix} (\mathbf{y} - \mathbf{z}_c)\cdot\mathbf{e}_1 \\ (\mathbf{y} - \mathbf{z}_c)\cdot\mathbf{e}_2 \end{bmatrix}.$$

Linear MLS approximation. We approximate the image-space mapping locally using an affine function

$$f(\mathbf{p}) = \begin{bmatrix} u(\mathbf{p}) \\ v(\mathbf{p}) \end{bmatrix} = \begin{bmatrix} a_0 + a_1 x + a_2 y \\ b_0 + b_1 x + b_2 y \end{bmatrix}.$$

Define the polynomial basis

$$\boldsymbol{\phi}(\mathbf{p}) = \begin{bmatrix} 1 \\ x \\ y \end{bmatrix}.$$

Weighting kernel. Each sample is weighted according to its distance from the query point:

$$w_i = \exp\left(-\frac{\|\mathbf{p}_i - \mathbf{p}_q\|^2}{\sigma^2}\right),$$

where σ is chosen proportional to the average distance of the samples from $\mathbf{p}_q$.

Weighted normal equations. The MLS coefficients are obtained by minimizing the weighted least-squares energies

$$\sum_{i=1}^{\ell} w_i \left(\boldsymbol{\phi}(\mathbf{p}_i)^{\top}\mathbf{a} - u_i\right)^2, \qquad \sum_{i=1}^{\ell} w_i \left(\boldsymbol{\phi}(\mathbf{p}_i)^{\top}\mathbf{b} - v_i\right)^2.$$

This yields the normal equations

$$\mathbf{M}\mathbf{a} = \mathbf{b}_x, \qquad \mathbf{M}\mathbf{b} = \mathbf{b}_y,$$

with

$$\mathbf{M} = \sum_{i=1}^{\ell} w_i\, \boldsymbol{\phi}(\mathbf{p}_i)\boldsymbol{\phi}(\mathbf{p}_i)^{\top},$$

$$\mathbf{b}_x = \sum_{i=1}^{\ell} w_i\, u_i\, \boldsymbol{\phi}(\mathbf{p}_i), \qquad \mathbf{b}_y = \sum_{i=1}^{\ell} w_i\, v_i\, \boldsymbol{\phi}(\mathbf{p}_i).$$

Evaluation at the query point. Finally, the estimated image-space coordinate of the query point is

$$u_q = \boldsymbol{\phi}(\mathbf{p}_q)^{\top}\mathbf{a}, \qquad v_q = \boldsymbol{\phi}(\mathbf{p}_q)^{\top}\mathbf{b}.$$

REFERENCES

Lawrence C. Evans and Ronald F. Gariepy. 2025. *Measure Theory and Fine Properties of Functions* (2nd ed.). CRC Press. https://doi.org/10.1201/9781003583004

Michael S. Floater. 2003. Mean Value Coordinates. *Computer Aided Geometric Design* 20, 1 (2003), 19–27. https://doi.org/10.1016/S0167-8396(03)00002-5

Wenzel Jakob and Steve Marschner. 2012. Manifold Exploration: A Markov Chain Monte Carlo Technique for Rendering Scenes with Difficult Specular Transport. *ACM Trans. Graph.* 31, 4 (Proc. SIGGRAPH 2012), Article 58 (July 2012), 13 pages. https://doi.org/10.1145/2185520.2185554

Grand Roman Joldes, Habibullah Amin Chowdhury, Adam Wittek, Barry Doyle, and Karol Miller. 2015. Modified Moving Least Squares with Polynomial Bases for Scattered Data Approximation. *Appl. Math. Comput.* 266 (2015), 893–902. https://doi.org/10.1016/j.amc.2015.05.150

Markus Kettunen, Marco Manzi, Miika Aittala, Jaakko Lehtinen, Frédo Durand, and Matthias Zwicker. 2015. Gradient-Domain Path Tracing. *ACM Trans. Graph.* 34, 4 (Proc. SIGGRAPH 2015), Article 123 (July 2015), 13 pages. https://doi.org/10.1145/2766997

Peter Lancaster and Kes Salkauskas. 1981. Surfaces Generated by Moving Least Squares Methods. *Mathematics of computation* 37, 155 (1981), 141–158. https://doi.org/10.1090/S0025-5718-1981-0616367-1

David Levin. 1998. The Approximation Power of Moving Least-Squares. *Math. Comput.* 67, 224 (Oct. 1998), 1517–1531. https://doi.org/10.1090/S0025-5718-98-00974-0

Hans Rademacher. 1919. Über partielle und totale Differenzierbarkeit von Funktionen mehrerer Variabeln und über die Transformation der Doppelintegrale. *Math. Ann.* 79, 4 (1919), 340–359. https://doi.org/10.1007/BF01498415

Xiaochun Tong and Toshiya Hachisuka. 2025. Practical Stylized Nonlinear Monte Carlo Rendering. In *SIGGRAPH 2025 Conference Papers*. Article 84, 11 pages. https://doi.org/10.1145/3721238.3730686

Rex West. 2021. Physically-based Feature Line Rendering. *ACM Trans. Graph.* 40, 6 (Proc. SIGGRAPH Asia 2021), Article 246 (Dec. 2021), 11 pages. https://doi.org/10.1145/3478513.3480550