

Differentiable Stroke Planning with Dual Parameterization for Efficient and High-Fidelity Painting Creation

Supplementary Material

6. Physically-Grounded Height Field Construction per Stroke

To achieve physically plausible rendering, our method incorporates a height field reconstruction loss (Sec. 3.4). We precompute a reference height field H_{gt} that encodes both global depth relations and local surface undulations. During rendering, we splat paint strokes to generate the color field I_{render} and, concurrently, form a height field H_{render} from the per-stroke height parameters h_i at negligible computational overhead. We then minimize the MSE loss between H_{render} and H_{gt} , formulated as:

$$\mathcal{L}_{height} = \|H_{render} - H_{gt}\|_2, \quad (10)$$

which guides each h_i toward a physically interpretable surface elevation.

The target height field H_{gt} is constructed following a structure–texture decomposition principle, where depth supplies coarse ordering of objects in the scene and the image texture contributes high-frequency surface relief (see Fig. 6). We first obtain a **depth-based height field** H_{depth} from a monocular depth estimation model (e.g., Depth Anything V2). In parallel, we derive a **texture-based height field** $H_{texture}$ by converting the input image I_{gt} to the CIELAB color space and extracting the high-frequency component from its luminance channel. We fuse these two complementary components into the final H_{gt} :

$$H_{gt} = \lambda_h H_{depth} + (1 - \lambda_h) H_{texture}, \quad (11)$$

where λ_h is a blending weight, which is typically set to 0.6.

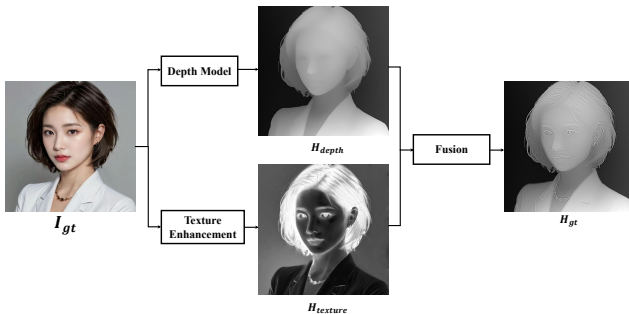


Figure 6. Overview of our height field construction pipeline. We fuse a depth-based height field, representing global scene structure, with a texture-based height field, capturing local surface details, to generate the final physically-plausible height field H_{gt} .

7. Lighting for Painterly Rendering

To reproduce the tactile character of real oil paintings, we construct a height-field representation that models both the canvas substrate and the impasto accumulation of paint (see Fig. 7). For each stroke, we define a per-pixel height contribution $h_i(\mathbf{x})$, which is procedurally perturbed to emulate bristle-induced ridges and grooves. These stroke-level height fields are composited with the canvas micro-geometry using the same front-to-back alpha blending mechanism described in Sec. 3.4, ensuring that height accumulation follows the identical ordering and transparency behavior as color splatting.

The canvas micro-geometry is synthesized procedurally using composite noise. We construct a base albedo field

$$C(\mathbf{x}) = C_0 + \Delta C \left[\text{fbm}(\mathbf{x}) + w_{\text{weave}} \sin(20 \mathbf{x} \cdot d_x) \sin(20 \mathbf{x} \cdot d_y) \right], \quad (12)$$

where $C_0 = (0.8, 0.75, 0.7)$, $\Delta C = 0.3$, and $d_x \perp d_y$ specify the weave directions. Here $\text{fbm}(\cdot)$ denotes fractional Brownian motion noise, which introduces multi-scale stochastic variation typical in procedural texture synthesis. We find that $w_{\text{weave}} = 0.45$ produces a visually plausible balance between FBM noise and weave detail across all canvas resolutions. The canvas height $h_c(\mathbf{x})$ is derived from the luminance of $C(\mathbf{x})$, producing fabric-like micro-relief.

Each stroke’s height is enhanced through a procedural impasto model. The perturbed height map is defined as

$$\tilde{h}_i(\mathbf{x}) = h_i(\mathbf{x}) + A(r_i) \sum_{k=1}^2 w_k \sin(\kappa_k f_0 r_i^{-1} l_i(\mathbf{x}) + \phi_k), \quad (13)$$

where r_i is the stroke radius, $l_i(\mathbf{x})$ the coordinate along the stroke direction, $A(r_i) = 0.1r_i$ controls amplitude, $f_0 = 0.5$ is a base frequency, $(w_1, w_2) = (0.65, 0.35)$ are harmonic weights, $(\kappa_1, \kappa_2) = (1.0, 1.8)$ specify frequency ratios, and ϕ_k are random per-stroke phases to avoid repetition. This formulation generates layered ridge patterns characteristic of real brushwork.

To couple paint and canvas geometry, we apply thickness-aware modulation:

$$\hat{h}_i(\mathbf{x}) = \tilde{h}_i(\mathbf{x}) \left[1 + (1 - \gamma(\tilde{h}_i(\mathbf{x}))) h_c(\mathbf{x}) \right], \quad (14)$$

$$\gamma(t) = \alpha \min(t/h_t, 1), \quad (15)$$

using $\alpha = 0.8$ and $h_t = 40$. Thin strokes inherit more canvas-induced roughness through $h_c(\mathbf{x})$, while thick impasto strokes ($\gamma \rightarrow 1$) gradually override the canvas texture.

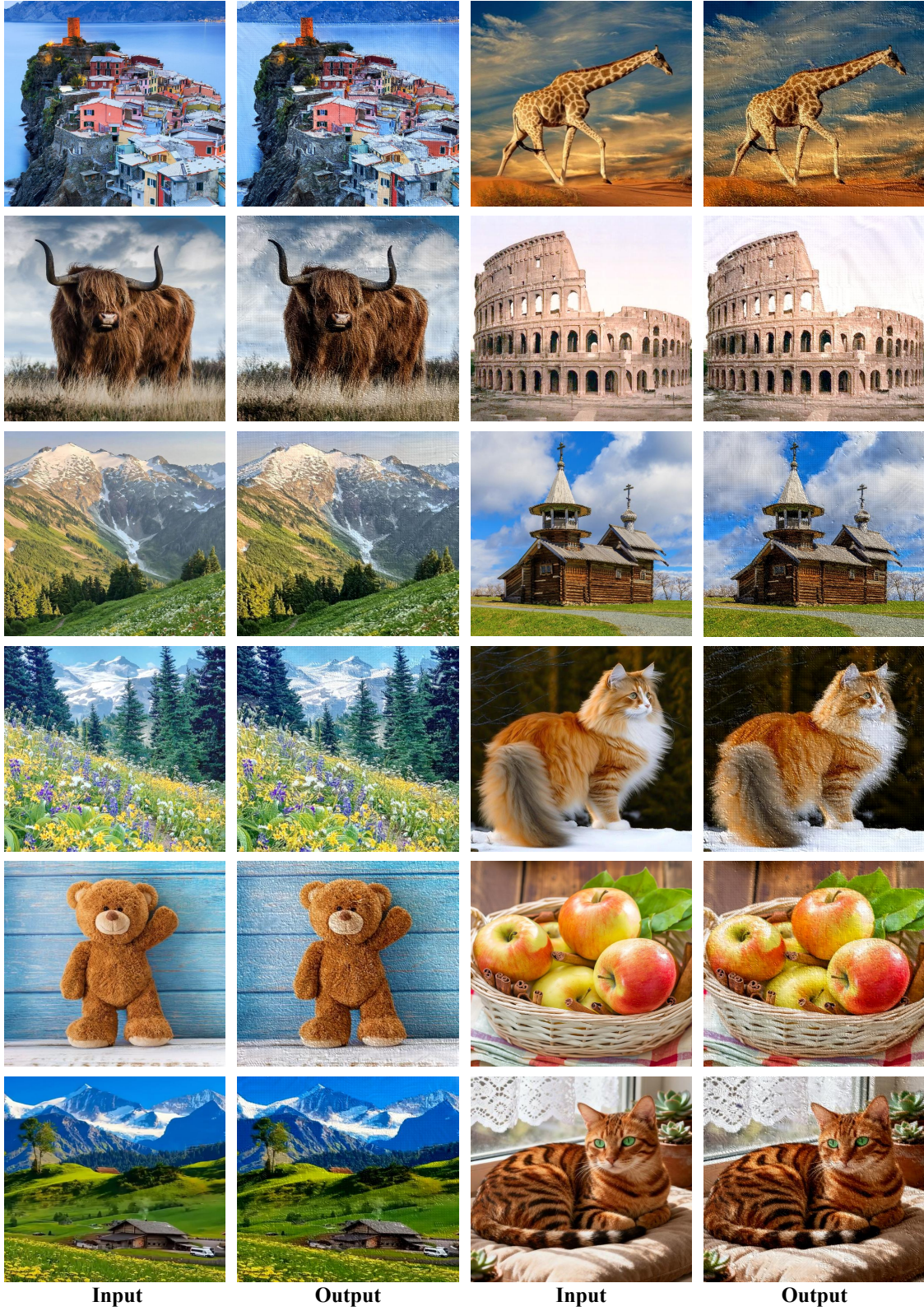


Figure 7. Relighting results using our height field representation. Our rendered result with canvas texture and impasto effects under directional lighting. The height field captures both macroscopic paint accumulation and microscopic canvas weave.

The final height field $H(\mathbf{x})$ is obtained by splatting the modulated stroke heights $\hat{h}_i(\mathbf{x})$ using the same front-to-back compositing as color (Eq. 9), with height values replacing the per-segment color. This shared compositing mechanism ensures consistent ordering and enables height accumulation without additional computational cost.

Surface normals are computed from the resulting height field via spatial gradients:

$$\mathbf{n}(\mathbf{x}) = \text{normalize}([-s \nabla H(\mathbf{x}), 1]^\top), \quad (16)$$

where s is a height-to-slope scaling factor converting pixel-level height variation into shading-consistent surface slopes.

Shading is performed using a GGX microfacet reflectance model:

$$L = \frac{\rho}{\pi} L_d + \frac{D F G}{4(\mathbf{n} \cdot \mathbf{v})(\mathbf{n} \cdot \mathbf{l})} L_s, \quad (17)$$

with roughness $\alpha_r = 0.3$, Fresnel base reflectance $F_0 = 0.08$ (typical for non-metallic pigments), and lighting intensities $(L_d, L_s) = (1.0, 0.8)$. This reflectance model accentuates both canvas microstructure and impasto depth under directional illumination.

8. Theoretical Analysis of the Bézier Intermediate Representation

In our stroke-based rendering (SBR) framework, we parameterize strokes using a set of vertices θ . Optimizing θ directly in the pixel domain is ill-posed and prone to high-frequency noise. To address this, we employ a Bézier curve as a differentiable geometric proxy. This process can be formally understood as a variant of *Projected Gradient Descent*, where the optimization trajectory is constrained to the manifold of smooth cubic polynomials.

Without loss of generality, we present the analysis for a single cubic Bézier segment. In practice, complex strokes are modeled as piecewise chains of such segments (splines), where this analysis applies locally to each segment.

8.1. The Bézier Proxy as a Projection Operator

Let a stroke segment be represented by N vertices $P \in \mathbb{R}^{N \times d}$, flattened into a parameter vector $\theta = \text{vec}(P) \in \mathbb{R}^{Nd}$. We define a differentiable rendering loss $L(\theta)$.

The projection begins by fitting a cubic Bézier curve to the vertices. We assign scalar parameters $\{t_i\}_{i=0}^{N-1}$ based on cumulative chord length and construct the Bernstein basis matrix $B \in \mathbb{R}^{N \times 4}$. The control points $C \in \mathbb{R}^{4 \times d}$ are obtained via least-squares fitting:

$$C = (B^\top B)^{-1} B^\top P. \quad (18)$$

To express this rigorously in vector form, we utilize the Kronecker product \otimes . Let $c = \text{vec}(C) \in \mathbb{R}^{4d}$. The fitting operation is a linear projection:

$$c = F\theta, \quad \text{where} \quad F = ((B^\top B)^{-1} B^\top) \otimes I_d. \quad (19)$$

Here, I_d is the $d \times d$ identity matrix, ensuring the projection applies independently to each spatial dimension.

Following projection, we define a reconstruction operator S that samples the curve at fixed intervals to produce a refined polyline $\hat{\theta}$. The composite operator $M = SF$ projects the arbitrary vertex vector θ onto the Bézier subspace. The optimization objective becomes $\tilde{L}(\theta) = L(M\theta)$.

8.2. Decoupling Normal and Tangential Flow

A critical aspect of our formulation is the handling of the parameterization $\{t_i\}$. Since t_i depends on the vertex positions P , the matrix M is technically a function of θ . By the chain rule, the exact gradient is:

$$\nabla_\theta \tilde{L} = M^\top \nabla_{\hat{\theta}} L + (\nabla_\theta M)^\top \nabla_{\hat{\theta}} L. \quad (20)$$

However, we explicitly discard the second term $(\nabla_\theta M)^\top$. This is not merely a computational simplification, but a deliberate design choice to **decouple geometric deformation from parameterization drift**. In geometric optimization, the gradient of the loss typically contains two components: a *normal* component that changes the shape, and a *tangential* component that merely slides vertices along the curve without altering geometry. The term $(\nabla_\theta M)^\top$ couples these effects. By treating the parameterization $\{t_i\}$ (and thus M) as locally frozen, we enforce a robust inductive bias: the optimizer focuses solely on minimizing the rendering loss via shape deformation, preventing pathological vertex clustering or tangential drift.

8.3. Spectral Analysis: Subspace Constraint

The smoothing effect of our method stems from the rank-deficient nature of the projection F . The matrix B maps the 4-dimensional control point space to the N -dimensional vertex space. Consider the Singular Value Decomposition (SVD) of the basis matrix B . Since B has at most rank 4, the projection operator F has a null space of dimension $N - 4$ (assuming $N > 4$).

This implies that our proxy acts as a strict **low-rank geometric filter**. Any high-frequency variation in the vertex gradient $\nabla_\theta L$ that falls into this null space is mathematically eliminated during the backpropagation step $\nabla_\theta \tilde{L} \approx F^\top(\dots)$. Unlike a standard low-pass filter which attenuates frequencies based on a cutoff, the Bézier projection fundamentally restricts the degrees of freedom of the solution to the cubic polynomial subspace. This improves the condition number of the optimization problem by removing high-frequency noise directions from the loss landscape.

8.4. Implementation Details and Robustness

Our fitting procedure adapts to the stroke complexity to ensure numerical stability. For strokes with sparse vertices ($N \leq \text{degree} + 1$), global least-squares fitting is ill-conditioned. In these cases, we construct a Catmull–Rom

spline through the vertices and convert each segment into an equivalent cubic Bézier curve to ensure C^1 continuity. For strokes with sufficient vertices ($N > \text{degree} + 1$), we perform a global least-squares fit of an n -th degree Bézier curve. We solve the linear system $Bc \approx P$ using a normalized chord-length parameterization. Throughout optimization, we monitor the symmetric Chamfer distance between the polyline and the Bézier proxy to ensure the representation remains faithful.

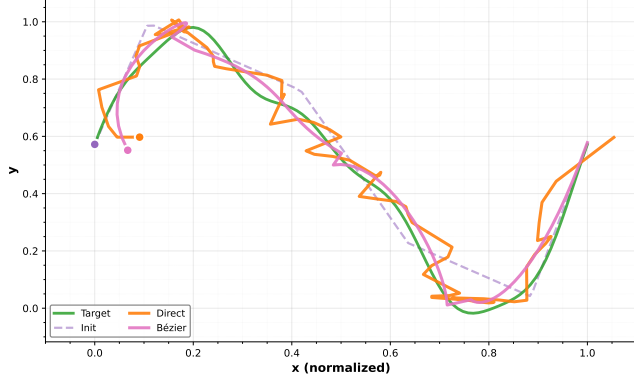


Figure 8. **Trajectory comparison.** Direct optimization (orange) overfits to noise. Bézier proxy (magenta) projects the solution onto the cubic polynomial subspace.

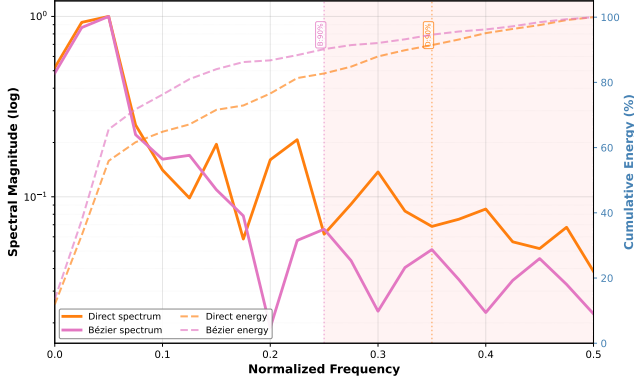


Figure 9. **Gradient Spectrum.** Spectral analysis confirms that the proxy operator M^\top nullifies gradient components associated with high-frequency spatial variation.

To empirically demonstrate the smoothing and regularizing effect of the Bézier proxy, we conduct a toy experiment. We define a smooth ground-truth stroke and add i.i.d. Gaussian noise ($\mathcal{N}(0, 0.05^2)$) to its vertices, creating a noisy target. We then initialize a stroke and optimize its control points for 200 iterations (learning rate 0.3) to fit this noisy target, comparing two methods: (i) direct optimization of the vertices, and (ii) optimization through our Bézier proxy.

As shown in Fig. 8, direct optimization quickly overfits

to the noise, producing a visually jagged stroke. The Bézier proxy, however, ignores the high-frequency jitter and recovers a smooth curve that faithfully captures the underlying structure of the clean target. The gradient spectrum analysis (Fig. 9) confirms our theoretical intuition: the Bézier proxy significantly dampens high-frequency components of the gradient update, thus stabilizing the optimization and promoting structurally sound results.