

Supplementary: A Theory of Topological Derivatives for Inverse Rendering of Geometry

Ishit Mehta Manmohan Chandraker Ravi Ramamoorthi

University of California San Diego

1. Outline

We organize the supplementary according to the section numbers used in the main manuscript. In Section 4.2, the derivation for topological derivative (TD) with respect to phase nucleation is outlined. We show a comparison between area-sampling and edge-sampling in Figure 1. In Figure 3 we show additional results for 2D vector graphics recovery using score-distillation-sampling [7]. We test our proposed theory on real-world data for 3D reconstruction in Figure 2, with discussion in Section 5.2. We discuss the relevance of curvature in the same section. A more detailed derivation for secondary visibility is in Section 5.2.1 with additional illustrations for the optimization results shown in the main paper. We provide implementation details for all the experiments in Section 7.

4.2. Topological Derivatives in 2D

We first outline the derivation for TD with respect to nucleating phases in the exterior of closed 2D curves.

Result 4 *Let Γ be a closed curve with Ω_Γ as its interior. For a point $\mathbf{x} \in \mathbb{R}^2 \setminus \Omega_\Gamma$ the topological derivative with respect to phase nucleation is,*

$$D_\tau(\mathbf{x}) = g_F(\mathbf{x}) - g_B(\mathbf{x}).$$

We follow the same derivation structure as in the case of hole nucleation. The main difference comes from how we define the notion of a perturbed shape. Consider a point $\mathbf{u} \in \mathcal{I}$ in the image plane, such that the corresponding point on the 2D plane $\mathbf{x} \in \mathbb{R}^2$ lies in the exterior of the given curve. We introduce a circular disk B_ϵ of radius ϵ centered at \mathbf{x} . The topological derivative with respect to this perturbation is then,

$$D_\tau(\mathbf{x}, \Omega_\Gamma) := \lim_{\epsilon \rightarrow 0} \frac{I(\Omega_\Gamma \cup \overline{B_\epsilon(\mathbf{x})}) - I(\Omega_\Gamma)}{V(B_\epsilon)}. \quad (1)$$

Note that the perturbed image functional is defined for the union of the original shape and the circular disk. The asymptotic expansion of (1) is,

otic expansion of (1) is,

$$I(\Omega_\Gamma \cup \overline{B_\epsilon}) = I(\Omega_\Gamma) + V(B_\epsilon)D_\tau(\mathbf{x}, \Omega_\Gamma) + o(V(B_\epsilon)). \quad (2)$$

We now construct a similar normal velocity as in the case of TD for hole nucleation. Let v be a continuous normal velocity such that $v = 1$ on the boundary ∂B_ϵ of the added phase and $v = 0$ on the original curve Γ . Intuitively, this function expands the radius of the circular disk by 1. Our goal is to measure the change in the functional I as we expand the nucleated phase and $\epsilon \rightarrow 0$. We take the Gâteaux derivative of (2) in the direction v . By definition, we know:

$$dI(\Omega_\Gamma)v = 0. \quad (3)$$

From (2) and (3), we can redefine the TD as,

$$\begin{aligned} D_\tau(\mathbf{x}, \Omega_\Gamma) &= \lim_{\epsilon \rightarrow 0} \frac{1}{V'(B_\epsilon)} dI(\Omega_\Gamma \cup \overline{B_\epsilon})v \\ &= \lim_{\epsilon \rightarrow 0} \frac{1}{V'(B_\epsilon)} \int_{\Gamma \cup \partial B_\epsilon} (g_F - g_B)v \, d\sigma \\ &= \lim_{\epsilon \rightarrow 0} \frac{1}{V'(B_\epsilon)} \int_{\partial B_\epsilon} (g_F - g_B) \, d\sigma \\ &= g_F(\mathbf{x}) - g_B(\mathbf{x}). \end{aligned} \quad (4)$$

The last step in the derivation is implied from: as $\epsilon \rightarrow 0$, g_F and g_B correspond to $g_F(\mathbf{x})$ and $g_B(\mathbf{x})$, and $\int_{\partial B_\epsilon} (g_F - g_B) \, d\sigma = 2\pi\epsilon \times (g_F(\mathbf{x}) - g_B(\mathbf{x}))$.

Eliminating Edge Sampling We show that the SD and TD for a closed curve is the same in their functional form. A level-set PDE which uses only shape derivatives has the evolution dynamics defined only on the curve. For numerical optimization, this requires explicit sampling of the boundaries as in [4]. Using the TD, the level-set PDE is defined everywhere on the image plane and we can evolve curves using area sampling. In practice, for complex shapes, this will lead to faster optimization. In Figure 1, we compare

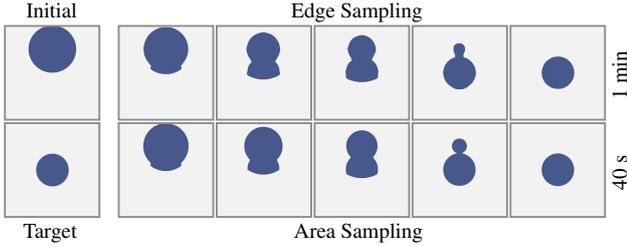


Figure 1. **Area sampling v/s edge sampling.** We can evolve curves using only area sampling (*bottom*) and get similar curve evolution as from a differentiable rasterizer [4] (*top*) which uses edge sampling. In practice, this leads to reduced optimization runtimes.

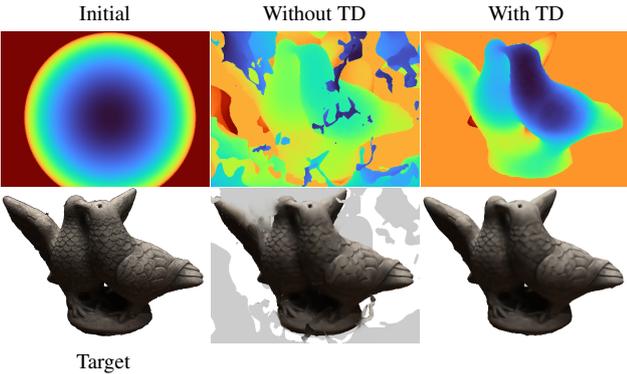


Figure 2. **Preliminary results on 3D shape recovery with real data.** We test our proposed theory on topological derivatives for multi-view 3D reconstruction using real data. Given 64 images for `scan_106` from DTU dataset [2], we reconstruct shape and reflectance starting from spherical initialization. On the top we show the depth map for the recovery and the textured mesh is shown on the bottom. We note that this result is still a proof-of-concept that demonstrates the potential of using topological derivatives in the presence of irregularities in real-world data.

shape optimization using edge and area sampling. The area sampling example uses a single sample per pixel and only compares colors with the target at the centre of the pixel.

Generative Vector Graphics We provide additional results for generating vector graphics using SDS [7] in Figure 3.

5.2. Initial Demonstration on Real Data

We test the feasibility of using topological derivatives with real data. Given a set of images, with a known background for each image, we reconstruct the object of interest and its reflectance. We use a neural reflectance function $f_\theta : \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^3 \rightarrow \mathbb{R}^3$, which takes the position, normal and viewing direction as input and estimates the outgoing radiance. θ is the set of parameters for the neural network. Figure 2 shows the recovery using our method with an ablation on TD. Our initial results indicate that TD is useful

for taking large steps to recover a coarse shape and resolve some of the shape/radiance ambiguities. Using only shape derivatives, we find the optimization to be biased towards optimizing reflectance parameters, which results in floating artifacts. We note that these results are preliminary and act as a proof of concept for using TD in the presence of non-idealities such as sensor noise, error in camera positions and distortion.

Curvature κ The TD for 3D depends on the curvature of the perturbed surface, which is not the same as the curvature at a point \mathbf{x} on the unperturbed surface. It depends on how we define the cone of perturbation. In practice, we find that we are able to achieve good recovery by assuming the curvature of the unperturbed shape to be constant across the *entire* surface. To understand this relevance of κ during optimization, we revisit the level-set PDE used for surface evolution:

$$\frac{\partial \phi}{\partial t} = -dI(\Gamma)|\nabla \phi| - \lambda D_\tau(\mathbf{x}, \Gamma)|\nabla \phi|. \quad (5)$$

Note that there is an additional factor λ in this PDE in comparison to the equation (24) in the main paper. We use λ as a weighting term to balance shape and topological derivatives during optimization. By assuming constant curvature, we can absorb κ into the weighting term λ .

5.2.1 Secondary Visibility

Although our notation for topological derivatives follows the standard definition [10], for brevity, we assume that the TD is conditional to a given camera position and its viewing direction. For primary visibility this is implicitly defined using the cone of perturbation where the apex is the camera’s origin. This relationship between the camera position and the topological derivative is different in the case of secondary visibility. Here, the apex is at the shading point \mathbf{x} which is the first intersection point for a ray originating from the camera. For this case, the definition of the scene functions g and g_B also differ. For primary visibility, the function g represents the radiance error along a ray originating at the camera center, whereas for secondary visibility, g corresponds to the radiance error for a light path that begins with $\mathbf{u} \rightarrow \mathbf{x} \rightarrow \mathbf{x}'$. Similarly the function g_B defines the error when the first occluder does not exist in case of primary visibility — and for secondary visibility, the error for path starting with $\mathbf{u} \rightarrow \mathbf{x} \rightarrow \mathbf{x}'$ but without the occluder corresponding to the secondary intersection. Apart from the differences in these definitions, the derivation for the TD for secondary visibility follows similarly from primary visibility. We assume a virtual camera centered at the shading point x and from the TD definition in Result 3 from

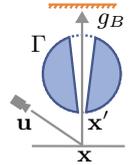




Figure 3. **Generating complex vector graphics using score-distillation** [7] We show additional results for the task of reconstructing vector images using a non-standard reconstruction loss. We use score-distillation-sampling loss from [7] with a stable diffusion model [9]. The shape initialization is the same as shown in Figure 4 in the main paper. In the order of top to bottom and left to right, the following text-prompts are used for generation: mouse, horse, penguin, eagle, mantis, pufferfish, cow, meerkat, rabbit, fennic fox, owl, shoebill, frog, koala, tiger. We augment the prompts with “Frontal face of <tag>. Minimal line drawing. Trending on artstation. Plain white background. Black and white.”

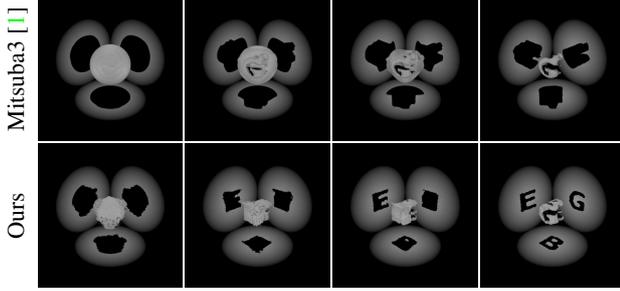


Figure 4. **Reconstructing shape ambigrams** We aid the GEB illustration (Figure 9) shown in the manuscript with mid-optimization renderings for NIE [6] + Mitsuba 3 [1] and our method. The recovery starts from a spherical initialization and progresses as shown here (from left to right).

the main manuscript, we arrive at this result,

$$D_{\tau}(\mathbf{x}', \Gamma) = (g(\mathbf{x}') - g_B(\mathbf{x}')) \frac{\kappa(\mathbf{x}' - \mathbf{x})^t (\mathbf{x}' - \mathbf{x})}{(\mathbf{x}' - \mathbf{x})_z^3}. \quad (6)$$

The change of terms from \mathbf{x} to $\mathbf{x}' - \mathbf{x}$ is due to the repositioning of the camera from the origin to \mathbf{x} . To be more specific, $g(\mathbf{x}')$ can be expanded to $g(\mathbf{u} \rightarrow \mathbf{x} \rightarrow \mathbf{x}')$. In Figure 4 we show the optimization progression for the results shown in Figure 9 in the main manuscript.

7. Implementation Details

Image Vectorization The image vectorization experiment uses a 128×128 2D grid to discretely represent the level-set function ϕ . At initialization, ϕ is a signed distance function (SDF). During optimization, we do not enforce ϕ to be an SDF. The raster input is a 128×128 image upsampled to 512×512 . At each iteration, we extract the 0-isocontours from the level-set function. The point-set for these contours are then used as closed bezier paths that we rasterize at 512×512 resolution. The scene functions $g_F(\mathbf{x})$ and $g_B(\mathbf{x})$ are defined as $|F(\mathbf{x}) - c(\mathbf{x})|$ and $|B(\mathbf{x}) - c(\mathbf{x})|$ respectively. $F(\mathbf{x})$ and $B(\mathbf{x})$ correspond to foreground and background colors at \mathbf{x} , and $c(\mathbf{x})$ is the color in the target raster image. We use a single sample per pixel and evaluate these functions at the center of the pixel. We use Adam as the optimizer and the learning rate for both shape derivative and topological derivative methods is 0.01.

Generating Vector Graphics We use a 128×128 level-set grid. The score-distillation-sampling loss is from [7]:

$$\nabla \mathcal{L} = \mathbb{E}_{t, \epsilon} \left[w(t) (\hat{\epsilon}(z_t : y, t) - \epsilon) \frac{\partial z}{\partial x} \right], \quad (7)$$

where t is the diffusion time step, w is a weighting term, z_t is the noisy latent variable, y is text-embeddings, $\hat{\epsilon}$ is



Figure 5. Samples from the set of images used for multi-view reconstruction. We use the kloppenheim_06 known environment map by Greg Zaal [13].

the estimated noise and ϵ is the ground-truth noise added to the latent variable. We use this gradient with our level-set PDE to optimize closed bezier curves given a text-prompt. The method using Li *et al.*'s [4] differentiable rasterizer is initialized with 128 randomly initialized disks with 12 control points. Our method is initialized with an SDF of a disk. The images are rasterized at 512×512 resolution. We use the learning rate scheduler from [12]. Our method is run for 2000 iterations and Li *et al.*'s method is run for 400 iterations. We find that the quality of output degrades significantly after 400 iterations with Li *et al.*'s method.

Multi-View 3D Reconstruction Each example is recovered from a set of 100 images of 256×256 resolution, except for helix, which is rendered at 512×512 . We use a level-set function ϕ defined on a 3D grid of $128 \times 128 \times 128$ resolution. Each shape is assumed to have a constant diffused BRDF. We use a known environment map [13] and use low-frequency illumination using spherical harmonics [8]. We finetune recovered meshes for each example after 1200 iterations and the total number of iterations is 2000. We employ a Laplacian regularizer for smoother recovery. At each step, we extract a mesh using Marching Cubes [5] at 128^3 resolution. We use nvdiffrast to estimate the shading and edge visibility gradients [3]. The scene functions g and g_B correspond to L2 error between target - foreground and target - background. We use a single sample at the centre of each pixel to estimate the topological derivative. The weighting term as discussed in (5) is 0.002, which we reduce to 0 after 1200 iterations.

Secondary Visibility We use a single image of 512×512 resolution to recover the shape. We use Mitsuba 3 [1] to estimate the shading and edge visibility gradients. To estimate the topological derivatives, we use a custom integrator for a two-bounce rendering model. Three spotlights are placed in the scene, one opposite to each of three orthogonal planes shown in Figure 4. We use emitter importance sampling for the integrator. The shape is extracted using marching cubes [5] at each iteration at 128^3 resolution. We use $\lambda = 0.001$.

Initial Experiments with Real Data We use 64 images for scan_106 from the DTU dataset [2]. Each image is

downsampled to $300 \times 400 \times 3$. For a more controlled setup, we segment out the background and replace it with a constant color. We use a 3 layered, 256 neurons-wide MLP with ReLU activations as the neural BRDF. The input position to the network is mapped to fourier features [11]. We fine-tune the extracted mesh after 2000 iterations and the total number of iterations we use is 4000. The total optimization time is ~ 1 hr.

References

- [1] Wenzel Jakob, Sébastien Speierer, Nicolas Roussel, Merlin Nimier-David, Delio Vicini, Tizian Zeltner, Baptiste Nicolet, Miguel Crespo, Vincent Leroy, and Ziyi Zhang. Mitsuba 3 renderer, 2022. <https://mitsuba-renderer.org>.
- [2] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engil Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 406–413. IEEE, 2014.
- [3] Samuli Laine, Janne Hellsten, Tero Karras, Yeongho Seol, Jaakko Lehtinen, and Timo Aila. Modular primitives for high-performance differentiable rendering. *ACM Transactions on Graphics*, 39(6), 2020.
- [4] Tzu-Mao Li, Michal Lukáč, Michaël Gharbi, and Jonathan Ragan-Kelley. Differentiable vector graphics rasterization for editing and learning. 39(6):1–15.
- [5] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM siggraph computer graphics*, 21(4):163–169, 1987.
- [6] Ishit Mehta, Manmohan Chandraker, and Ravi Ramamoorthi. A level set theory for neural implicit evolution under explicit flows. In *European Conference on Computer Vision (ECCV)*, pages 711–729. Springer, 2022.
- [7] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.
- [8] Ravi Ramamoorthi and Pat Hanrahan. An efficient representation for irradiance environment maps. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 497–500, 2001.
- [9] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021.
- [10] Jan Sokolowski and Antoni Zochowski. On the topological derivative in shape optimization. *SIAM journal on control and optimization*, 37(4):1251–1272, 1999.
- [11] Matthew Tancik, Pratul P Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *arXiv preprint arXiv:2006.10739*, 2020.
- [12] Jiayang Tang. Stable-dreamfusion: Text-to-3d with stable-diffusion, 2022. <https://github.com/ashawkey/stable-dreamfusion>.
- [13] Greg Zaal.