A Theory of Topological Derivatives for Inverse Rendering of Geometry

Ishit Mehta   Manmohan Chandraker   Ravi Ramamoorthi

University of California San Diego

Abstract

We introduce a theoretical framework for differentiable surface evolution that allows discrete topology changes through the use of topological derivatives for variational optimization of image functionals. While prior methods for inverse rendering of geometry rely on silhouette gradients for topology changes, such signals are sparse. In contrast, our theory derives topological derivatives that relate the introduction of vanishing holes and phases to changes in image intensity. As a result, we enable differentiable shape perturbations in the form of hole or phase nucleation. We validate the proposed theory with optimization of closed curves in 2D and surfaces in 3D to lend insights into limitations of current methods and enable improved applications such as image vectorization, vector-graphics generation from text prompts, single-image reconstruction of shape ambigrams and multiview 3D reconstruction.

1. Introduction

Recovering geometry from images is a central theme for several problems in vision and graphics, where a common approach is to derive the differential of the rendering functional. Depending on the type of surface representation used, such as Bézier paths [25], triangle meshes [10, 24] or level-sets [12], corresponding gradient flow equations are derived. These works formulate image differentials as shape derivatives, which we posit to be restrictive for inverse problems, since deformations induced by shape derivatives (SD) are limited to surface boundaries. This may lead to local minima when recovering geometry with high-genus topology. In Figure 1 we illustrate two inverse problems where shape derivatives do not suffice to recover the optimal shape. In such cases the optimization is required to make updates far away from the boundary by (a) nucleating additional volume in the exterior, and (b) perforating the interior of the shape. Our work theoretically characterizes such shape perturbations in the form of topological derivatives (TD) for inverse rendering.

We derive TDs following the definition proposed by
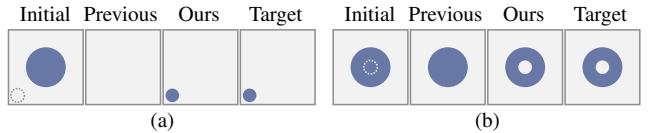
Figure 1. **Pathologies of shape derivatives.** Previous work on inverse surface reconstruction relies on shape derivatives which are inadequate for instances when (a) the target geometry is far from the initialization, and when (b) target has a hole in the interior. We resolve these failure cases with topological derivatives.

Sokolowski and Zochowski [44]. In the case of planar curves, vanishing balls are introduced in their interior or exterior. In the limit, the TD is estimated as the difference in the image functionals for the perturbed and unperturbed shapes. The resulting gradient updates can prompt hole and phase nucleation in regions of high-error (see Figure 1). For closed surfaces in 3D, we introduce TDs with respect to conic perturbations through the interior of the visible shape. We observe that visibility terms in SDs are evaluated only on the apparent contours of the visible shape. In contrast, gradient flows using TDs can encourage visibility changes in the interior, resulting in a more robust and accurate recovery.

Informed by recent successes in inverse rendering [2, 30, 46, 47, 49], we use level-sets [38] for surface representation. We build on the extensive literature on *variational level-sets* [11, 19, 45, 52] for surface reconstruction. Our approach differs from the more recent methods on differentiable rasterization [25] and rendering [2, 17, 24, 46] that use differential calculus to derive geometry gradients. The variational framework lends us a common structure to analyze the motion of continuous surfaces in both 2D and 3D. It also provides a natural way to extend shape derivatives to the formulation for topological derivatives. Additionally, using this approach, we are able to draw theoretical insights across literature spanning 3D reconstruction [6, 12], topology optimization [44], differentiable vector graphics [25] and differentiable rendering [17, 22, 24, 41].

We structure this work as three theoretical results interspersed with empirical observations for closed curves in 2D (§ 4) and surfaces in 3D (§ 5). Previous work on differentiable rasterization [25] proposes geometry gradients for

standard vector graphics representations. In Result 1 (§ 4.1), we derive a corresponding shape derivative for level-sets in the variational setting. Motivated by the above limitations, we derive topological derivatives for hole and phase nucleation in Result 2 (§ 5.2). We show practical applications of image vectorization and text-to-vector graphics using an evolution equation that works with arbitrary loss functions. In § 5.1 we reason about the necessity of TDs in 3D. Result 3 (§ 5.2) discusses the construction of conic perturbations and the derivation for TD. We show improvements over previous methods that rely on SDs in terms of speed for visibility optimization (Figure 5), accuracy of recovery for complex topology (Figure 8) and also show an application of reconstructing shape ambigrams from a single image (Figure 9).

2. Related Work

In this section, we concisely situate our work with respect to literature on level-sets, differentiable rendering and topological derivatives. A more detailed discussion on literature relevant to our problem formulation, theory and experimental settings can be found within Sections 3, 4 and 5.

Level Sets We use the level-set method [38] for shape evolution. In the case of 3D, level-sets have been extensively used for shape reconstruction from range data [48], stereo images [11], multi-view images [12] and RGB-D data [34]. In 2D, they have been used for image segmentation as active contour models [7, 8, 21]. These methods derive equations of motion for curves and surfaces as a set of partial differential equations that minimize an optimization objective. Solem and Overgaard [45] provide a geometric viewpoint to analyze these methods. The first variational approach to level-set optimization was proposed by Zhao *et al*. [52] for multi-phase surface motion. More recently, level-sets as signed-distance functions have become a popular choice for generative modeling of 3D shapes [9, 39] and inverse rendering [18, 27, 36, 50].

Differentiable Rendering Differentiable renderers are designed to estimate the derivatives of a rendering integral with the primary focus on handling geometric discontinuities. One class of differentiable renderers handles primary-visibility discontinuity for triangle meshes [22, 26, 28, 41] with a blurring kernel — an approach that is fast but not physically accurate. Li *et al*. [24] propose edge sampling to correctly handle discontinuities and differentiate the full rendering equation [20], which was later extended for differentiable rasterization of vector graphics [25]. Subsequently, several works aim for numerical efficiency and accuracy [3, 17, 29, 51]. Work by Bangaru *et al*. [2] and Vicini *et al*. [46] is especially relevant to our work as they handle discontinuities for surfaces represented by signed-distance

functions. All such works measure image sensitivities with respect to visibility changes on silhouette boundaries, while we focus on meaasuring them in the shape's interior.

Topological Derivatives The standard definition of topological derivatives is given by Sokolowski and Zochowski [44]. Part of our notation comes from the asymptoptic analysis of TDs in [33]. They have been used in relatively low-dimensional problems in physics for electrical impedence tomography [14] and inverse scattering [6], or in structural engineering to discover optimal support structures [1]. In computer vision, the use of topological derivatives has been limited to a few problems in image processing [23]. The work by Burger *et al*. [5] is perhaps the closest to ours in terms of using the level-set method with SDs and TDs. Please refer to [37] for a recent survey on the usage of TDs.

3. Background and Problem Formulation

We consider the problem of inverse shape optimization from a set of images. Our focus is on recovering closed curves in 2D and closed surfaces in 3D. With Γ being the surface under consideration, we take a variational optimization approach to minimize an image functional,

$$\min_{\Gamma} I(\Gamma) = \int_{\mathcal{I}} g(\mathbf{u}) \, d\mathbf{u}. \tag{1}$$

The functional I is defined over an image plane $\mathcal{I}$ with $d\mathbf{u}$ as the pixel-area measure. The integrand g is evaluated at pixels $\mathbf{u}$ and can be any reasonable error function. For instance, $g = |\hat{L} - L|$ can evaluate the error between measured (L) and estimated radiance ($\hat{L}$).

We define Γ as a closed n-dimensional surface residing in $\mathbb{R}^{n+1}$, where $n = 1$ for planar curves and $n = 2$ for surfaces in 3D. We use a level-set function $\phi : \mathbb{R}^{n+1} \to \mathbb{R}$ to represent Γ as, $\Gamma := \{\mathbf{x} : \phi(\mathbf{x}) = 0\}$. We use the standard convention [38] to represent the interior of Γ as $\Omega_{\Gamma} = \{\mathbf{x} : \phi(\mathbf{x}) \leq 0\}$ and the exterior as $\mathbb{R}^{n+1} \setminus \Omega_{\Gamma}$.

To recover the optimal surface, we require a notion of gradient in the variational setting — for which we build on the framework of variational level-sets [45]. Consider the surface Γ to be a point on an m-dimensional manifold M of admissible surfaces. The functional I to be minimized can be considered a scalar function that maps $M \to \mathbb{R}$. We use the idea of *differentials* (and corresponding Gâteaux derivatives) from differential geometry [32]. We derive the gradient $\nabla_M I$ and optimize Γ with an initial value problem: $\dot{\Gamma}(t) = \nabla_M I(\Gamma(t))$, with Γ_0 as the surface at initialization. The optimal surface Γ^* can be then be recovered using a regular surface evolution of $t \mapsto \Gamma(t)$. By definition, the gradient flow ∇I is composed of only normal components as tangential flow fields keep Γ invariant [38]. We refer the reader to [45] for a more rigorous discussion on why ∇I

is a descent direction and reduces I. Intuitively, ∇I is a scalar speed in the direction of the normal n at each point $\mathbf{x} \in \Gamma$ such that I is reduced. The corresponding evolution equation for the underlying level-set function is,

$$\frac{\partial \phi}{\partial t} = -\nabla_M I |\nabla \phi(t)|. \tag{2}$$

In Sections 4 and 5 we derive evolution equations of this form to minimize I in the case of closed curves in 2D and surfaces in 3D respectively. Computing ∇I requires defining an inner product which operates on the tangent space $T_\Gamma M$ at Γ. For w and v normal velocity directions on $T_\Gamma M$, we can define $\langle \cdot, \cdot \rangle_\Gamma : T_\Gamma M \times T_\Gamma M \to \mathbb{R}$ as,

$$\langle w, v \rangle_\Gamma = \int_\Gamma w(\mathbf{x}) v(\mathbf{x}) \, d\sigma, \tag{3}$$

where $d\sigma$ is the surface measure. Consider $v \in T_\Gamma M$ as any normal velocity direction of the form $v = -\frac{\psi}{|\nabla \phi|}$. We can deform the surface Γ in this direction by updating the level-set function as $\phi^s = \phi + s\psi$, where s controls the level of deformation and ψ is the speed. The directional (Gâteaux) derivative of I in this direction v can then be defined as [45],

$$dI(\Gamma)v := \frac{d}{ds} I(\phi + s\psi) \Big|_{s=0}. \tag{4}$$

For *any* such normal velocities v in the tangent space, if we can reformulate (4) as,

$$dI(\Gamma)v = \frac{d}{ds} I(\phi + s\psi) \Big|_{s=0} \overset{?}{=} \langle w, v \rangle_\Gamma, \tag{5}$$

then w is the gradient $\nabla_M I$. We revisit this definition of Gâteaux derivative (5) and the evolution equation (2) in the subsequent sections to derive gradient flows using shape and topological derivatives.

4. Curves in 2D

We derive equations of motion for closed planar curves subject to a rendering functional. These equations take the form of *shape derivatives* (§ 4.1) and *topological derivatives* (§ 4.2) and can be used to recover 2D vector shapes with respect to an error function. With this formulation, we show shape evolution for image vectorization (Figure 3) and generation of vector graphics from text prompts (Figure 4).

4.1. Shape Derivatives

Result 1 *Let I be an image functional of a closed curve Γ encoded as the 0-isocontour of a level-set function ϕ in $\mathbb{R}^2$. If the interior of the shape corresponds to a foreground scene function g_F and the exterior to a background function g_B then the shape derivative ∇I is,*
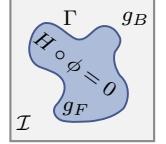
$$\nabla I = g_F - g_B.$$

We start by looking at the image integral (1) that integrates a scene function g over the image plane,

$$I(\Gamma) = \int_\mathcal{I} g(\mathbf{u}) \, d\mathbf{u} = \int_{\mathbb{R}^2} g(\mathbf{x}) \, d\mathbf{x}, \tag{6}$$

where g is any arbitrary function of color on the image plane. For 2D rasterization, the image plane coincides with $\mathbb{R}^2$ and hence we can change the integral domain as shown in (6). We also assume that g is band-limited to the support of the image *i.e.*, $g(\mathbf{u}) = 0$ for $\mathbf{u} \notin \mathcal{I}$. The closed curve Γ partitions the $\mathbb{R}^2$ plane into two regions as per Jordan's Curve Theorem [13]. We delineate the two regions with scene functions g_F and g_B for foreground and background respectively. We use a level-set function ϕ to synthesize a characteristic function $H \circ \phi$, where H is a standard heaviside function. By construction, our function evaluates to 0 in the foreground and 1 in the background. The characteristic function can be used to expand the integrand as,

$$I(\Gamma) = \int_{\mathbb{R}^2} g_F(\mathbf{x})[1 - H(\phi(\mathbf{x}))] + g_B(\mathbf{x}) H(\phi(\mathbf{x})) \, d\mathbf{x}.$$

We can now the derive the shape derivative ∇I using a perturbation of the form $\phi^s = \phi + s\psi$ that corresponds to velocity v. The Gâteaux derivative of I in this direction v is,

$$
\begin{aligned}
dI(\Gamma)v &= \frac{d}{ds} I(\phi + s\psi) \Big|_{s=0} \qquad \triangleleft \text{ From (4)} \\
&= \int_{\mathbb{R}^2} g_F \frac{d}{ds}[1 - H(\phi^s)] + g_B \frac{d}{ds} H(\phi^s) \, d\mathbf{x} \Big|_{s=0} \\
&= \int_{\mathbb{R}^2} -\delta(\phi)\psi g_F + \delta(\phi)\psi g_B \, d\mathbf{x} \\
&= \int_{\mathbb{R}^2} (g_F - g_B) \frac{-\psi}{|\nabla \phi|} \delta(\phi) |\nabla \phi| \, d\mathbf{x} \\
&= \int_\Gamma (g_F - g_B) v \, d\sigma = \langle g_F - g_B, v \rangle_\Gamma, \tag{7}
\end{aligned}
$$

where $d\sigma = \delta(\phi(\mathbf{x}))|\nabla \phi(\mathbf{x})| d\mathbf{x}$ is the surface measure [16, 45] and $v = \frac{-\psi}{|\nabla \phi|}$. Note the interchange of integral and differential operators due to Leibniz rule. Comparing the definitions in (5) and (7), we conclude that the shape derivative for functional I is,

$$\nabla I = g_F - g_B. \tag{8}$$

The shape derivative in (8) can be used to minimize I using the level-set evolution equation (2). As per the domain of the integral in (7), this derivative is defined only on the curve Γ. Li *et al*. [25] arrive at a similar result using differential calculus and numerically evolve 2D shapes by explicitly sampling the edges. As shown in Figure 3, SDs can fail to prompt crucial topology changes for image vectorization.
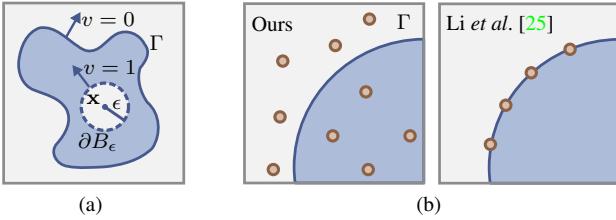
Figure 2. (a) We derive topological derivatives for 2D rendering functionals. An infinitesimal-hole of perturbation B_ϵ with radius $\epsilon \to 0$ is introduced at point $\mathbf{x}$. A Gâteaux derivative is computed in the direction of a normal velocity v such that $v = 1$ on ∂B_ϵ and 0 on the unperturbed surface Γ. (b) We find functional forms of shape and topological derivatives to be the same with the only difference being the domain. Numerically, this eliminates the need for explicitly sampling boundaries for shape evolution as in [25].

4.2. Topological Derivatives

The topological derivative measures sensitivities with respect to infinitesimally small perturbations in a shape's volume. We derive the topological derivative in the context of inverse rendering to inform where to nucleate holes and phases in a shape in order to reduce a rendering functional.

Result 2 *Consider a closed curve Γ with Ω_Γ as its interior. For a point $\mathbf{x} \in \mathbb{R}^2$ the topological derivative with respect to hole and phase nucleation is,*

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

We follow the standard definition of a topological derivative from Sokolowski and Zochowski [44]. We focus on hole nucleation in this section and defer the derivation for phase nucleation to the Supplementary. Let Ω_Γ denote the interior of a given curve Γ. For a point $\mathbf{u} \in \mathcal{I}$ on the image plane, we introduce a small circular hole $B_\epsilon(\mathbf{u})$ of radius ϵ. The topological derivative of the image functional I at a point $\mathbf{x} \in \Omega_\Gamma$ can then be defined as a scalar function,

$$D_\tau(\mathbf{x}, \Omega_\Gamma) \coloneqq \lim_{\epsilon \to 0} \frac{I(\Omega_\Gamma \setminus \overline{B_\epsilon(\mathbf{x})}) - I(\Omega_\Gamma)}{V(B_\epsilon)}, \quad (9)$$

where $I(\Omega_\Gamma)$ is the functional value with the unperturbed shape, $\overline{B_\epsilon}$ is the closure set of points inside the perturbation and $V(B_\epsilon) = \pi\epsilon^2$ is the area of the circle. The corresponding asymptotic expansion of this definition [33, 44] is,

$$I(\Omega_\Gamma \setminus \overline{B_\epsilon}) = I(\Omega_\Gamma) + V(B_\epsilon)D_\tau(\mathbf{x}, \Omega_\Gamma) + o(V(B_\epsilon)). \quad (10)$$

We compute the Gâteaux derivative of this expression in the direction of a normal velocity v defined as shown in Figure 2 (a). Intuitively, the defined velocity increases the size of the perturbation by a constant and keeps the curve Γ as it is. Formally, $v = 0$ on the unperturbed curve Γ and $v = 1$ at the boundary ∂B_ϵ of the perturbation hole, such that it smoothly
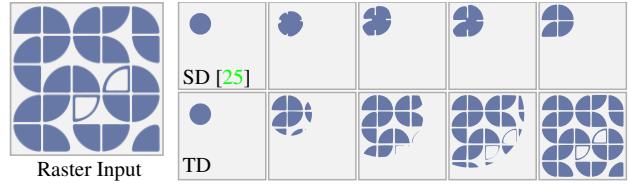
Figure 3. **Reconstruction of vector images from raster input.** Shape derivatives (analogous to curve gradients in [25]) evolve closed curves with visibility changes only on the boundaries. These gradients are sparse and the optimization can plateau at a local minimum (*top*). Topological derivatives can reconstruct a vector image (*bottom*) from a raster input (*left*) with adaptive topological changes.

goes to 0 outside a small neighborhood of $\mathbf{x}$. Assuming $\mathbf{x} \notin \Gamma$ and since velocity v is 0 on the unperturbed curve, the shape derivative (8) in the direction v is 0:

$$dI(\Omega_\Gamma)v = \int_\Gamma (g_F - g_B)v \, d\sigma = 0. \quad (11)$$

From (9), (10) and (11) we can rewrite the TD as:

$$\begin{aligned} D_\tau(\mathbf{x}, \Omega_\Gamma) &= \lim_{\epsilon \to 0} \frac{1}{V'(B_\epsilon)} dI(\Omega_\Gamma \setminus \overline{B_\epsilon})v \\ &= \lim_{\epsilon \to 0} \frac{1}{V'(B_\epsilon)} \int_{\Gamma \cup \partial B_\epsilon} (g_F - g_B)v \, d\sigma. \end{aligned} \quad (12)$$

Note that with the introduction of a hole, the perturbed shape is comprised of surface elements from the original domain Γ in addition to the boundary ∂B_ϵ of the hole — that is, $\hat{\Gamma} = \Gamma \cup \partial B_\epsilon$ where $\hat{\Gamma}$ is the perturbed curve (see Figure 2 (a)). The denominator of (12) is a result of the fact that $dV(\epsilon)v = V'(\epsilon) = 2\pi\epsilon$ as $v = 1$ on the hole's boundary. Further simplification yields:

$$\begin{aligned} D_\tau(\mathbf{x}, \Omega_\Gamma) &= \lim_{\epsilon \to 0} \frac{1}{2\pi\epsilon} \int_\Gamma (g_F - g_B)v \, d\sigma + \\ &\qquad \int_{\partial B_\epsilon} (g_F - g_B)v \, d\sigma \\ &= \lim_{\epsilon \to 0} \frac{1}{2\pi\epsilon} \int_{\partial B_\epsilon} (g_F - g_B) \, d\sigma \\ &= g_F(\mathbf{x}) - g_B(\mathbf{x}). \end{aligned} \quad (13)$$

Equation 13 is the topological derivative of the image functional I at a point $\mathbf{x}$. Intuitively, at a given point in the interior of the shape, if the error with respect to the background color is lower than the foreground, formation of a hole can be prompted by increasing the value of $\phi(\mathbf{x})$. We note that D_τ is defined in $\mathbb{R}^2 \setminus \Gamma$. This is in stark contrast with the shape derivatives from Result 1 that encouraged visibility changes only on the curve. By analogy, we also derive topological derivatives for phase nucleation and arrive
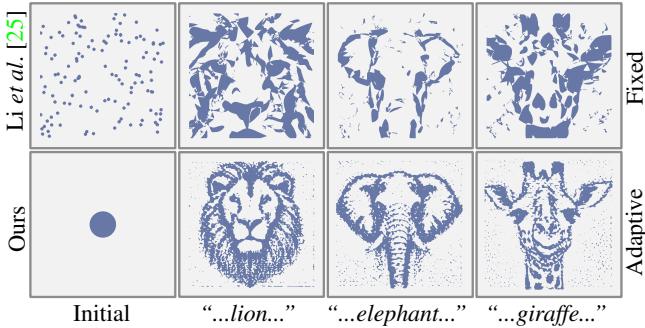
Figure 4. **Adaptive generation of detailed vector graphics.** Our formulation for surface evolution can be used to induce topological changes subject to any differentiable loss functions for images. Here we show an example of generating vector graphics from a text prompt using the score-distillation-sampling loss from [40]. (*top*) Existing methods use differentiable vectorization from [25] and are forced to work with fixed shape topologies. (*bottom*) Conversely, topological derivatives can be used to adaptively generate detailed vector graphics from a simple initialization (*bottom-left*). [1]

at the same result, *i.e.* $D_\tau(\mathbf{x}, \mathbb{R}^2 \setminus \Omega_\Gamma) = g_F(\mathbf{x}) - g_B(\mathbf{x})$, where $\mathbf{x}$ is a point in the exterior of the given shape. The full derivation is in the Supplementary.

Level-Set Evolution We can now formulate the evolution equation in (2) using both shape and topological derivatives:

$$\frac{\partial \phi}{\partial t} = -\left[\nabla I(\mathbf{x}) + D_\tau(\mathbf{x}) \right] |\nabla \phi|. \quad (14)$$

The SD ∇I is defined on Γ and is 0 elsewhere. The TD D_τ is defined on $\mathbb{R}^2 \setminus \Gamma$. As shown in Figure 3, we can use the evolution equation for image-based reconstruction of vector graphics from raster input. When only the first term ∇I is used, the optimization can plateau at a local minimum. The second term is critical for recovering the target geometry, particularly when the target has a differing geometric structure from the initialization. We show more results in this context along with the implementation details in the Supplementary.

Eliminating Edge Sampling Comparing Result 1 and Result 2, we find that the topological and shape derivatives are exactly the same except for the domain on which they are defined. This simplifies the evolution equation in (14) with a single term for the entirety of $\mathbb{R}^2$,

$$\frac{\partial \phi}{\partial t} = (g_B - g_F)|\nabla \phi|. \quad (15)$$

In practice, for numerical optimization, we find that this simplification eliminates the need for explicitly sampling

---

[1]Text-prompt: "Frontal face of `animal`. Minimal line drawing. Trending on artstation. Plain white background. Black and white."
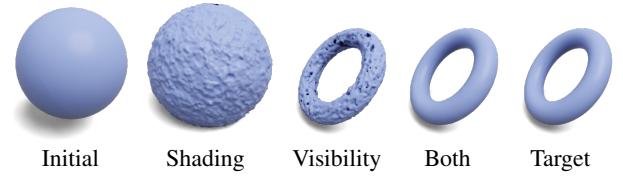

Figure 5. **Visibility gradients drive topological changes.** Given a set of multi-view images, we recover a genus-1 shape from a sphere. When shading gradients are used the optimization makes only local updates. Using visibility gradients the topology and the silhouettes of the recovery match the target. Both the terms are required for optimal recovery.

boundaries for shape evolution as done in Li *et al.*'s [25] method for differentiable vector graphics (Figure 2 (b)). For complex shapes, with a large set of paths, this leads to faster optimization. We provide details in the Supplementary.

Generative Vector Graphics Using the chain-rule, the evolution equation can be adapted for problems beyond just reconstruction. Given an aribitrary differentiable loss function $\mathcal{L}$, we can evolve the shapes with topological changes such that $\mathcal{L}$ is minimized by the following evolution:

$$\frac{\partial \phi}{\partial t} = \frac{\partial \mathcal{L}}{\partial I}(g_B - g_F)|\nabla \phi|. \quad (16)$$

We show an application of generating 2D vector graphics from text-prompts with a text-to-image diffusion model [43]. The loss function $\mathcal{L}$ is the score-distillation-sampling loss from [40]. Previous methods for text-to-svg use the differentiable rasterizer from [25]. Such methods require a fixed number of shapes starting from the intialization and cannot adaptively change the topology in a differentiable manner. Using topological derivatives, we show highly-detailed generation of vector graphics from a relatively simple initialization of a disk (Figure 4). More examples and implementation details are in the Supplementary.

5. Surfaces in 3D

Similar to the 2D case, our goal is to minimize the image functional I defined in (1). Comparing it to the rendering equation [20], I can depend on scene parameters such as the geometry, material and lighting in the scene. We focus on geometry optimization and take the material and lighting parameters as given. The function g is discontinuous in terms of the geometry parameters and hence naively differentiating I using automatic differentiation is erroneous [24].

5.1. Background on Shading and Visibility

To address discontinuities with respect to visibility changes, Gargallo *et al.* [12] derive the shape derivative
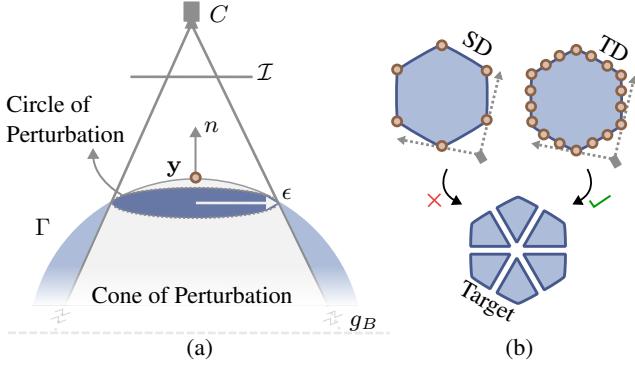
Figure 6. (a) We derive topological derivative at a point $\mathbf{y}$ on a surface Γ using a vanishing cone of perturbation originating at the camera C. (b) Visibility terms from shape derivatives (SD) are evaluated only at the apparent contours. In flatland, these points resemble the extreme points of the convex set of the shape. A target shape with the same convex boounday as the initial shape cannot be recovered using the visibility terms in SDs. TDs evaluate visibility changes on the entire surface and can resolve such ambiguities.

of I to evolve a level-set function ϕ as,

$$dI(\Gamma) = \underbrace{-\nabla g \cdot \frac{\mathbf{x}}{\mathbf{x}_z^3} \chi}_{\text{Shading}} + \underbrace{(g - g_B)\frac{\mathbf{x} \cdot \nabla(n \cdot \mathbf{x})}{\mathbf{x}_z^3}\delta(n \cdot \mathbf{x})\,\chi}_{\text{Visibility}},$$

(17)

where χ denotes the visibility, n the normal and $\mathbf{x}_z$ the z coordinate. The first term here is similar to shading gradients in more recent differentiable renderers [17, 22, 24, 41], which can be used to make local updates in the interior of the visible surface. The second term is enabled only at the silhouette boundary as per $\delta(n \cdot \mathbf{x})$, where δ is a dirac distribution function. The difference $g - g_B$ moves the visible contours of the surface based on the foreground (g) and background (g_B) radiance. This visibility term is similar to shape derivatives in the 2D case (8), with additional factors to account for the distance from the camera, visibility and shape curvature.

Need for Topological Derivatives For recovering geometry with complex topology, we find the visibility gradients to be critical in terms of recovering the overall structure. We empirically show this in Figure 5, where a genus-1 shape is optimized from a sphere using the two gradient forms. When only shading gradients are used, the optimization fails to evolve the topology; with visibility gradients in isolation we can retrieve the correct topology but not the details; when both are used together, we recover the target shape. Despite this significance, the optimization signal obtained from visibility gradients is quite sparse. As evident from (17), these gradients are evaluated only at the extremeties of the shape and cannot induce visibility changes in the interior of the shape. In flatland, this means that visibility gradients

account for changes only at the extreme points of the convex set covering a shape (as shown in Figure 6 (b)). This makes the optimization susceptible to unexpected local minima as topological changes are less likely in regions away from the surface. Our goal is to resolve this by enabling topological changes in the interior regions using gradients that measure visibility changes on the *entire* surface.

5.2. Topological Derivatives

Result 3 *Let I be an image functional of a closed and connected surface Γ in $\mathbb{R}^3$. The functional I integrates a scene function g for the surface and g_B for the background. Then the topological derivative of I at a point $\mathbf{y} \in \Gamma$ with respect to an infinitesimal conical perturbation from the origin through $\mathbf{y}$ is,*

$$D_\tau(\mathbf{y}, \Gamma) = (g(\mathbf{y}) - g_B(\mathbf{y}))\frac{\kappa \mathbf{y}^t \mathbf{y}}{\mathbf{y}_z^3}.$$

In case of surfaces in $\mathbb{R}^3$, there is a large class of infinitesimal perturbations that change the configuration of the shape. In this work we focus on perturbations that influence the genus of the shape. We begin by considering a point $\mathbf{y}$ on a surface Γ at which want to estimate the topological derivative. Without loss of generality, we will assume that the surface Γ is placed in between the camera at the origin and a background scene. A curved circle of planar radius ϵ is placed on the surface with $\mathbf{y}$ as the center (Figure 6 (a)). We term the corresponding planar circle as the circle of perturbation. Starting from the origin, we can find an elliptic cone that inscribes this circle at an angle that is consistent with the normal at $\mathbf{y}$. We use this cone to define our perturbed shape as the difference of the enclosed volume of the original shape and the intersection of the original shape with the cone of perturbation. If Ω_Γ is the interior of Γ and Ω_ϵ of the perturbation cone, then $\Omega_{\hat{\Gamma}(\epsilon)} = \Omega_\Gamma - \Omega_\Gamma \cap \Omega_\epsilon$ is the volume enclosed by the perturbed surface $\hat{\Gamma}$. By construction, we know that as $\epsilon \to 0$, the perturbed surface $\hat{\Gamma}(\epsilon) \to \Gamma$. Beyond just the perturbation, we are interested in its resultant effect on the image. The image plane intersects with the cone of perturbation and inscribes an ellipse. Intuitively, as we increase the radius of the circle of perturbation, the area of the ellipse on the image plane increases and the image accumulates more light from the background.

In the given construction, we now define the topological derivative of I at $\mathbf{y}$ as,

$$D_\tau(\mathbf{y}, \Gamma) := \lim_{\epsilon \to 0} \frac{I(\hat{\Gamma}) - I(\Gamma)}{V(\epsilon)}. \tag{18}$$

Note that this is a slightly different notion of the topological derivative as compared to the standard form [44] which introduces vanishing balls B_ϵ at the points of perturbation. Our deviation from this norm is out of necessity. The error
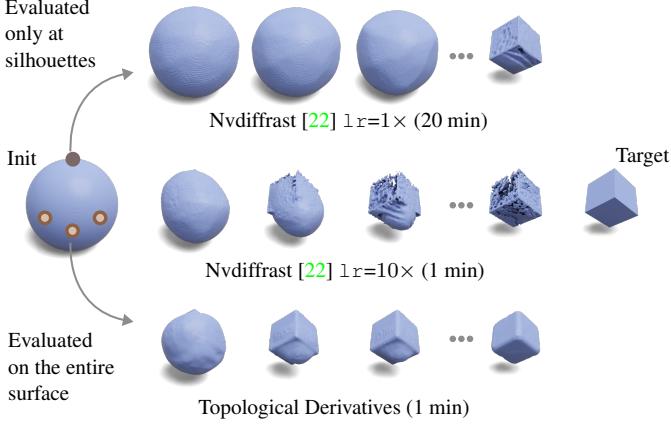
Figure 7. **Large steps in visibility optimization.** Standard differentiable renderers evaluate visibility gradients only at the silhouette. Topological derivatives estimate sensitives with respect to visibility changes on the entire surface. This enables a more robust way to take large steps in inverse rendering. We compare topological derivatives (*bottom*) with visibility gradients (*top*) from Nvdiffrast [22] for multi-view reconstruction. (*middle*) We observe that silhouette gradients are unstable at higher learning rate (lr).

functionals that we consider for inverse rendering operate only on the projection of the shape and hence any perturbations of the standard form in the shape's interior will have no effect on the error. The term $V(\epsilon)$ in the standard form is the Lebesgue measure of the ball B_ϵ, although, in our case, we choose $V(\epsilon) = \pi\epsilon^2$ as the area of the planar circle. This choice will lend us a limit (18) that exists and is finite.

Similar to the 2D case (§ 4.2), we define a normal velocity v such that $v = 1$ on the newly formed visible contour around the circle of perturbation, and $v = 0$ on the unperturbed surface. Hence, the Gâteaux derivative of the image functional $I(\Gamma)$ in the direction v is,

$$\langle dI(\Gamma), v \rangle_\Gamma = \int_\Gamma dI(\Gamma)v \, d\sigma = 0, \qquad (19)$$

where dI is defined as in (17). Analogous to the case of TDs in 2D (10), by using asymptotic expansion we can rework the definition for 3D in (18) as,

$$D_\tau(\mathbf{y}, \Gamma) = \lim_{\epsilon \to 0} \frac{1}{V'(\epsilon)} dI(\hat{\Gamma})v. \qquad (20)$$

Intuitively, as $\epsilon \to 0$, this definition measures the rate of change in the image of a perturbed shape with respect to the rate of increase in the area of the circle of perturbation. Using the shape derivative defined in (17) we can replace the

$dI(\hat{\Gamma})$ with,

$$D_\tau(\mathbf{y}, \Gamma) = \lim_{\epsilon \to 0} \frac{1}{V'(\epsilon)} \int_{\hat{\Gamma}} \left[-\nabla g \cdot \frac{\mathbf{x}}{\mathbf{x}_z^3} \chi \right] v +$$
$$\left[(g - g_B) \frac{\mathbf{x} \cdot \nabla(n \cdot \mathbf{x})}{\mathbf{x}_z^3} \delta(n \cdot \mathbf{x})\chi \right] v \, d\sigma. \quad (21)$$

We simplify this integral with two assumptions. First, in the small neighborhood of $\mathbf{y}$, as $\epsilon \to 0$, the radiance function g is constant. This eliminates the first term in the integrand as $\nabla g = 0$. Next, we assume constant curvature. This results in $\mathbf{x} \cdot \nabla(n \cdot \mathbf{x}) = \kappa \mathbf{x}^t \mathbf{x}$ and curvature [2] κ can be taken out of the integral sign as,

$$D_\tau(\mathbf{y}, \Gamma) = \lim_{\epsilon \to 0} \frac{\kappa}{V'(\epsilon)} \int_{\hat{\Gamma}} (g - g_B)\frac{\mathbf{x}^t\mathbf{x}}{\mathbf{x}_z^3}\delta(n \cdot \mathbf{x})\chi v d\sigma. \quad (22)$$

The term $\delta(n \cdot \mathbf{x})\chi$ constrains the domain for the integral to the set of apparent contours that are visible from the camera. The choice of v further restricts this domain to only the circle of perturbation (which we denote by $\partial\circ$) as by definition, $v = 0$ for all other contours:

$$D_\tau(\mathbf{y}, \Gamma) = \lim_{\epsilon \to 0} \frac{\kappa}{V'(\epsilon)} \int_{\partial\circ} (g - g_B)\frac{\mathbf{x}^t\mathbf{x}}{\mathbf{x}_z^3} \, d\sigma$$
$$= (g(\mathbf{y}) - g_B(\mathbf{y}))\frac{\kappa\mathbf{y}^t\mathbf{y}}{\mathbf{y}_z^3}. \quad (23)$$

This is the topological derivative of functional I at a visible point $\mathbf{y}$ on the surface. Unlike the visibility gradients by Gargallo *et al.* [12] in (17) and other differentiable renderers [17, 22, 24, 41], the topological derivative in (23) can prompt visibility changes on the entire surface.

Level-Set Evolution We can finally use topological derivatives (23) and shape derivatives (17) together to evolve a level-set function ϕ using the evolution equation from (2):

$$\frac{\partial\phi}{\partial t} = -dI(\Gamma)|\nabla\phi| - D_\tau(\mathbf{x}, \Gamma)|\nabla\phi|. \qquad (24)$$

Although, the shape derivative term $(dI(\Gamma))$ is valid only for primary visibility. We can lift this assumption by replacing the shape derivatives with gradients from a differentiable path tracer for triangle meshes [17, 24] as follows [30, 42],

$$\frac{\partial\phi}{\partial t} = -\frac{\partial I}{\partial\mathbf{x}} \cdot n - D_\tau(\mathbf{x}, \Gamma)|\nabla\phi|. \qquad (25)$$

This replacement enables higher-order shading and visibility gradients while being practical in terms of availability of multiple mesh-based differentiable renderers that are 1) fast [22, 41], 2) have user-friendly APIs [17, 22, 24, 41], and 3) can handle complex light transport effects [17, 24].

[2] More details on the relevance of κ are in the Supplementary.
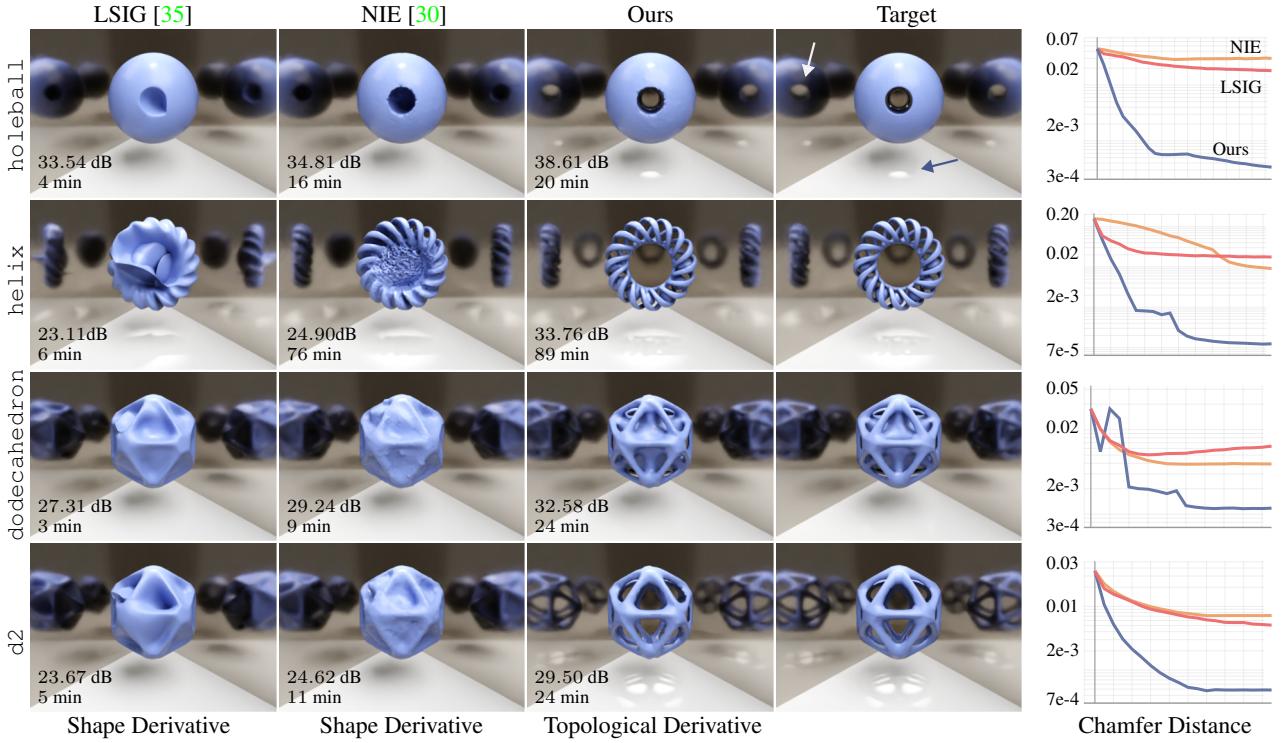
Figure 8. **Inverse rendering of geometry with complex topology.** Given a set of images, we cover complex shapes from a spherical initialization. We compare our method that uses topological derivatives to a differentiable renderer [22] that uses visibility gradients defined on the silhouette. We regularize the gradients from [22] with two recent methods. LSIG [22] uses a triangle mesh representation and can make topological changes using remeshing [4]. NIE [30] uses the same gradients with a level-set function. These approaches struggle with prompting visibility changes in the interior and can get stuck in a local minimum. Convergence plots of chamfer distance are on the right. We also provide PSNR and optimization time for each scene. Our method, as expected, requires a slight time-overhead over NIE to estimate the TD. We place the recovered shapes in front of two glossy surfaces to highlight the quality of recovery.

Large Steps in Visibility Optimization Unlike the visibility gradients in present differentiable renderers, the topological derivative from Result 3 is defined for the interior of the visible shape. For shape recovery from multi-view images, there is the immediate benefit that, for each point on the surface, we can obtain visibility sensitivities defined from several viewing angles. Conversely, with standard shape derivatives, visibility gradients are defined for points only when they appear at the silhouette. The resulting optimization signal is quite sparse in practice — and in cases with complex lighting and materials, it is noisy. We find that topological derivatives provide a more robust signal for visibility changes and enable faster optimization. In Figure 7 we show a comparison with visibility gradients from nvdiffrast [22] for inverse shape recovery from multi-view images.

Multi-View Reconstruction We validate the proposed theory using synthetic 3D shapes of complex topology. Given a set of images with known material and environment lighting, we use the level-set evolution in (25) to minimize reprojection error. We show qualitative and quantitative results in Figure 8 with nvdiffrast [22]. Since [22] is de-

fined for triangle meshes, we regularize the gradients for smoother optimization using LSIG [35] and enable topological changes with NIE [30]. Theoretically, our method becomes an extension of NIE in the form of an additional term in the level-set PDE (25) for topological derivatives. Starting from a genus-0 sphere, we find that shape derivatives are insufficient to prompt visibility changes in the shape's interior. The holeball example, which requires puncturing three axial holes through the sphere, is a prime illustration of this problem. The regions of hole nucleation are within the silhouette of the shape and hence other methods require shading gradients to prompt the visibility changes — which, as discussed earlier in Figure 5, are unreliable for topology evolution. We find that by using topological derivatives we can successfully recover the shown examples as they can evaluate visibility changes in such regions. We provide more examples of shape recovery in the Supplementary.

5.2.1 Secondary Visibility

We extend our formulation for TD to a rendering integral with secondary light bounces. We project a point **u** on the im-
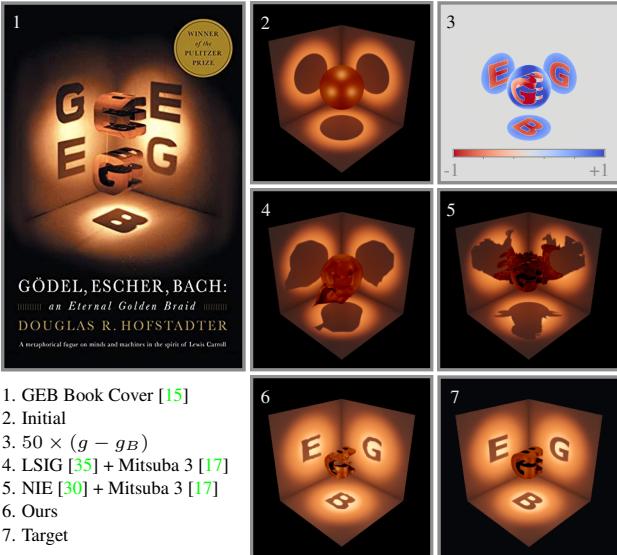
Figure 9. **Shape ambigram from a single image.** We reproduce the famous shape ambigram from the cover of *GEB* by Hofstader (1). Given a single image (7), we aim to recover a shape with complex shadow projections from a relatively simple shape (2). A differentiable path tracer like Mitsuba 3 [17] uses a form of shape derivative and struggles with visibility changes necessary in the interior regions. We regularize the gradients from [17] using two recent methods. One which works with triangle meshes [35] (4) and the other with level-sets [30] (5). Our method uses topological derivatives for secondary visibility and can successfully recover the shadows (6). In (3) we visualize the term $g - g_B$ at intialization.

age plane to a shading point in the scene by $\mathbf{x} = \pi^{-1}(\mathbf{u})$ (see the inline figure). We can integrate a scene function over the hemispherical domain around this shading point $\mathbf{x}$ and relay the differentials back to $\mathbf{u}$. We follow a similar construction as in the case of primary visibility (Result 3) and estimate TD at a point $\mathbf{x}' \in \Gamma$ acting as an occluder for the light path $\mathbf{u} \to \mathbf{x} \to \mathbf{x}'$. Let $g(\mathbf{x}')$ be the scene function along this path with the unperturbed surface and $g_B(\mathbf{x}')$ the scene function after the conical perturbation. The TD of the shading functional around $\mathbf{x}$ with respect to a perturbation at $\mathbf{x}'$ is,

$$D_\tau(\mathbf{x}', \Gamma) = (g(\mathbf{x}') - g_B(\mathbf{x}')) \frac{\kappa(\mathbf{x}' - \mathbf{x})^t(\mathbf{x}' - \mathbf{x})}{(\mathbf{x}' - \mathbf{x})_z^3}. \quad (26)$$

We provide the full derivation in the Supplementary. A surface evolution equation including a TD of this form can perforate through shapes along the secondary segments of light paths. To demonstrate the feasibility of this approach we propose a task of recovering 3D shape ambigrams from a single image. Inspired by the popular cover of *Gödel, Escher, Bach* [15], we construct a scene with an ambigram of letters G, E and B placed inside a box with shadow projections
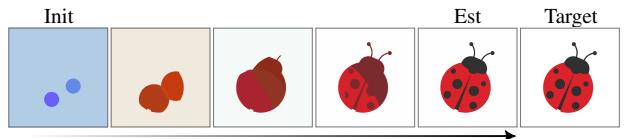
of the letters on different planes. As shown in Figure 9, differentiable path tracers such as Mitsuba 3 [17] cannot recover this shape with a genus-0 intialization. By adding a TD term to (25), we can deform the sphere such that more light can be accumulated in the regions with shadows.

6. Discussion

As shown in the experiments (Figures 4, 8 and 9), topological derivatives (TDs) can be used to prompt visibility changes away from shape contours. The form of these derivatives resemble occupancy and segmentation loss functions [36, 50] that are frequently employed in shape reconstruction methods. Roughly put, topological derivative can be interpreted as a *soft* notion of occupancy loss and can be used when a segmentation mask of the final geometry is unavailable.

In comparison to volumetric-rendering based methods [31] the advantage of using TDs is efficient sampling and the ability to use more complex materials and light transport effects. As evident from the synthetic nature of the experiments, however, there is still a gap between the reconstruction quality achieved using TDs and volumetric methods. We anticipate a few significant challenges, that if resolved could reduce the gap. *First*, balancing the shape and topological derivative terms in Eq. 25 for shape evolution in 3D is not straightforward. TDs are not always necessary, especially when the optimization has reached a point where the topology of the shape is the same as that of the target. *Second*, the assumption of constant curvature (in Eq. 23) could lead to unexpected behavior and its effect needs a thorough inquiry. *Third*, we do not derive TDs for phase nucleation in 3D. In 2D, a sample in the pixel space corresponds to a single sample on the canvas. As a result, the term $(g - g_B)$ in Equation 13 relates to a single point in the shape's exterior. For 3D, in empty space, a pixel-sample corresponds to a ray in the direction of that sample. This makes the problem of phase nucleation significantly more challenging.

Finally, joint optimization of geometry and other parameters such color, material properties and lighting is not trivial and poses additional challenges. As a preliminary step, below, we show an experiment with the task of recovering both geometry and color for 2D vectorization.

References

[1] Samuel Amstutz and Antonio A Novotny. Topological optimization of structures subject to von mises stress constraints. *Structural and Multidisciplinary Optimization*, 41(3):407, 2010. 2

[2] Sai Bangaru, Michael Gharbi, Tzu-Mao Li, Fujun Luan, Kalyan Sunkavalli, Milos Hasan, Sai Bi, Zexiang Xu, Gilbert Bernstein, and Fredo Durand. Differentiable rendering of neural sdfs through reparameterization. In *ACM SIGGRAPH Asia 2022 Conference Proceedings*, 2022. 1, 2

[3] Sai Bangaru, Tzu-Mao Li, and Frédo Durand. Unbiased warped-area sampling for differentiable rendering. *ACM Trans. Graph.*, 39(6):245:1–245:18, 2020. 2

[4] Mario Botsch and Leif Kobbelt. A remeshing approach to multiresolution modeling. In *Proceedings of Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 185–192, 2004. 8

[5] Martin Burger, Benjamin Hackl, and Wolfgang Ring. Incorporating topological derivatives into level set methods. In *Journal of Computational Physics*, 2004. 2

[6] Ana Carpio and Maria Luisa Rapún. *Topological Derivatives for Shape Reconstruction*, volume 1943 of *Lecture Notes in Mathematics*. 2008. 1, 2

[7] Vicent Caselles, Ron Kimmel, and Guillermo Sapiro. Geodesic active contours. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pages 694–699. IEEE, 1995. 2

[8] Tony F Chan and Luminita A Vese. Active contours without edges. *IEEE Transactions on image processing*, 10(2):266–277, 2001. 2

[9] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[10] Amaël Delaunoy and Emmanuel Prados. Gradient Flows for Optimizing Triangular Mesh-based Surfaces: Applications to 3D Reconstruction Problems Dealing with Visibility. In *International Journal of Computer Vision*, 2011. 1

[11] Olivier Faugeras and Renaud Keriven. *Variational Principles, Surface Evolution, PDE's, Level Set Methods and the Stereo Problem*. IEEE, 2002. 1, 2

[12] Pau Gargallo, Emmanuel Prados, and Peter Sturm. Minimizing the Reprojection Error in Surface Reconstruction from Images. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2007. 1, 2, 5, 7

[13] Thomas C Hales. Jordan's proof of the jordan curve theorem. *Studies in logic, grammar and rhetoric*, 10(23):45–60, 2007. 3

[14] M Hintermuller and Antoine Laurain. Electrical impedance tomography: from topology to shape. *Control and Cybernetics*, 37(4):913–933, 2008. 2

[15] Douglas Hofstader. *Gödel, Escher, Bach: An Eternal Golden Braid*. 1979. 9

[16] Lars Hörmander. *The analysis of linear partial differential operators I: Distribution theory and Fourier analysis*. Springer, 2015. 3

[17] 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. 1, 2, 6, 7, 9

[18] Yue Jiang, Dantong Ji, Zhizhong Han, and Matthias Zwicker. Sdfdiff: Differentiable rendering of signed distance fields for 3d shape optimization. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2

[19] Hailin Jin. *Variational Methods for Shape Reconstruction in Computer Vision*. Washington University in St. Louis, 2003. 1

[20] James T Kajiya. The rendering equation. In *Proceedings of the 13th annual conference on Computer Graphics and Interactive Techniques*, pages 143–150, 1986. 2, 5

[21] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International journal of computer vision*, 1(4):321–331, 1988. 2

[22] 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. 1, 2, 6, 7, 8

[23] Ignacio Larrabide, RA Feijóo, AA Novotny, and EA Taroco. Topological derivative: a tool for image processing. *Computers & Structures*, 86(13-14):1386–1403, 2008. 2

[24] Tzu-Mao Li, Miika Aittala, Frédo Durand, and Jaakko Lehtinen. Differentiable monte carlo ray tracing through edge sampling. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, 37(6):222:1–222:11, 2018. 1, 2, 5, 6, 7

[25] Tzu-Mao Li, Michal Lukáč, Michaël Gharbi, and Jonathan Ragan-Kelley. Differentiable vector graphics rasterization for editing and learning. 39(6):1–15. 1, 2, 3, 4, 5

[26] Shichen Liu, Tianye Li, Weikai Chen, and Hao Li. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2019. 2

[27] Shaohui Liu, Yinda Zhang, Songyou Peng, Boxin Shi, Marc Pollefeys, and Zhaopeng Cui. Dist: Rendering deep implicit signed distance function with differentiable sphere tracing. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2

[28] Matthew M Loper and Michael J Black. Opendr: An approximate differentiable renderer. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VII 13*, pages 154–169. Springer, 2014. 2

[29] Guillaume Loubet, Nicolas Holzschuch, and Wenzel Jakob. Reparameterizing discontinuous integrands for differentiable rendering. *ACM Transactions on Graphics (TOG)*, 38(6):1–14, 2019. 2

[30] 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. 1, 7, 8, 9

[31] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 9

[32] John Milnor and David W Weaver. *Topology from the differentiable viewpoint*, volume 21. Princeton university press, 1997. 2

[33] Serguei A Nazarov and Jan Sokołowski. Asymptotic analysis of shape functionals. *Journal de Mathématiques pures et appliquées*, 82(2):125–196, 2003. 2, 4

[34] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE international symposium on mixed and augmented reality*, pages 127–136. Ieee, 2011. 2

[35] Baptiste Nicolet, Alec Jacobson, and Wenzel Jakob. Large steps in inverse rendering of geometry. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)*, 40(6), Dec. 2021. 8, 9

[36] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3504–3515, 2020. 2, 9

[37] Antonio André Novotny, Jan Sokołowski, and Antoni Żochowski. *Applications of the topological derivative method*. Springer, 2019. 2

[38] Stanley Osher and James A Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of computational physics*, 1988. 1, 2

[39] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[40] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 5

[41] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv:2007.08501*, 2020. 1, 2, 6, 7

[42] Edoardo Remelli, Artem Lukoianov, Stephan Richter, Benoit Guillard, Timur Bagautdinov, Pierre Baque, and Pascal Fua. Meshsdf: Differentiable iso-surface extraction. In *Advances in Neural Information Processing Systems*, pages 22468–22478. Curran Associates, Inc., 2020. 7

[43] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, 2022. 5

[44] Jan Sokolowski and Antoni Zochowski. On the topological derivative in shape optimization. *SIAM journal on control and optimization*, 37(4):1251–1272, 1999. 1, 2, 4, 6

[45] Jan Erik Solem and Niels Chr. Overgaard. A Geometric Formulation of Gradient Descent for Variational Problems with Moving Surfaces. In *Scale Space and PDE Methods in Computer Vision*, volume 3459 of *Lecture Notes in Computer Science*, pages 419–430. 2005. 1, 2, 3

[46] Delio Vicini, Sébastien Speierer, and Wenzel Jakob. Differentiable signed distance function rendering. *ACM Transactions on Graphics (TOG)*, 41(4):1–18, 2022. 1, 2

[47] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021. 1

[48] Ross T Whitaker. A level-set approach to 3d reconstruction from range data. *International journal of computer vision*, 29(3):203–231, 1998. 2

[49] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. In *Advances in Neural Information Processing Systems*, volume 34, 2021. 1

[50] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. In *Advances in Neural Information Processing Systems*, volume 33, 2020. 2, 9

[51] Cheng Zhang, Bailey Miller, Kai Yan, Ioannis Gkioulekas, and Shuang Zhao. Path-space differentiable rendering. *ACM Trans. Graph.*, 39(4):143:1–143:19, 2020. 2

[52] Hong-Kai Zhao, Tony Chan, Barry Merriman, and Stanley Osher. A variational level set approach to multiphase motion. *Journal of computational physics*, 127(1):179–195, 1996. 1, 2