# DeepCurrents: Learning Implicit Representations of Shapes with Boundaries

David Palmer[*1]     Dmitriy Smirnov[*1]     Stephanie Wang[2]     Albert Chern[2]     Justin Solomon[1]

[1]Massachusetts Institute of Technology     [2]UC San Diego

## Abstract

*Recent techniques have been successful in reconstructing surfaces as level sets of learned functions (such as signed distance fields) parameterized by deep neural networks. Many of these methods, however, learn only closed surfaces and are unable to reconstruct shapes with boundary curves. We propose a hybrid shape representation that combines explicit boundary curves with implicit learned interiors. Using machinery from geometric measure theory, we parameterize currents using deep networks and use stochastic gradient descent to solve a minimal surface problem. By modifying the metric according to target geometry coming, e.g., from a mesh or point cloud, we can use this approach to represent arbitrary surfaces, learning implicitly defined shapes with explicitly defined boundary curves. We further demonstrate learning families of shapes jointly parameterized by boundary curves and latent codes.*

## 1. Introduction

Shape representation is a crucial component of geometry processing and learning algorithms. Depending on the target application, different representations have varying tradeoffs. Broadly, shape representations fall naturally into two classes: *Lagrangian* or *explicit*; *Eulerian* or *implicit*. In this work, we show how to use the theory of *currents* from geometric measure theory to design a flexible neural representation that combines favorable aspects from each category, representing the interiors of surfaces implicitly while maintaining an explicit representation of their boundaries.

Lagrangian representations encode a shape by giving coordinates of points or parameterizing regions of the shape. To represent a curve in a Lagrangian way, one might give coordinates of successive points along the curve. Analogously, to represent a surface in 3D, one might use a mesh, which assembles the surface out of simple patches. Lagrangian representations afford great precision but require predetermined combinatorial structures, making it difficult to represent families of shapes with varying topology.

In contrast, Eulerian representations encode a shape via a function on some background domain. For example, a surface might be encoded as the *level set* of a scalar function sampled on a regular grid. Level sets of signed distance fields (SDFs) form one popular implicit representation. Implicit functions naturally capture topological variation, but traditional implicit shape representations, in which the background geometry must be discretized with a fixed grid or mesh, waste resolution on regions far away from the level set of interest. Recent *neural implicit representations* alleviate this problem [7, 38, 46]. The universal approximation and differentiability properties of neural networks make them an appealing alternative to regular grid discretizations.

Neural implicit representations come with their own limitations. Like other implicit representations based on level sets, most neural implicit representations can only encode closed surfaces, which lack boundary curves. Boundaries are desirable as they can provide manipulation handles for controllable deformation, and common boundaries can be used to stitch together surfaces into a larger articulated surface.

In this paper, we describe a new way to encode neural implicit surfaces with boundaries, which can then be combined into more complex hybrid surfaces. The key to our representation is the theory of currents from geometric measure theory. In this theory, $k$-dimensional submanifolds are defined by their integration against differential $k$-forms, generalizing how distributions (0-currents) are defined by integration against smooth functions. Current spaces are complete normed linear spaces that make optimization over surfaces convenient, and the boundary operator also becomes linear on these spaces. Classically, currents were the key to solving Plateau's *minimal surface* problem by transforming it into *mass norm minimization*. We adopt the mass norm as the primary loss function encouraging our neural currents to converge to smooth surfaces.

We demonstrate our representation with three applications. We first demonstrate how it enables computing minimal surfaces efficiently through stochastic gradient descent. Then, by modifying the background metric used to define the mass norm, we reconstruct arbitrary surfaces from data. Finally, we demonstrate the flexibility of our representation by encoding families of surfaces with explicit boundary control.

---

[*]Authors contributed equally to this work.

**Contributions.** In summary, we

- propose a new neural implicit surface representation with explicit boundary curves;
- show how to use SGD on the mass norm to compute minimal surfaces;
- introduce a custom background metric and additional loss terms to represent surfaces from data; and
- describe a framework for learning families of surfaces parameterized by their boundaries along with a latent code.

## 2. Related Work

Our work takes classical ideas in minimal surface computation and brings them into the context of modern deep learning to form a new neural shape representation. Below, we summarize key prior works in these two areas.

### 2.1. Minimal Surface Computation

Most computational approaches to minimal surface generation use a mesh or grid representation of the surface. In the twentieth century, numerical minimal surface problems were discretized by finite difference methods on a grid, assuming the surfaces were function graphs [11, 17]. Grid-based methods were later adapted to triangle meshes [32, 66], allowing the generated surface to leave the space of function graphs [62]. Modern mesh-based minimal surface solvers use mean curvature flow [3, 15, 22], stretched grids [50], quasi-Newton iterations [49, 51], Voronoi tessellations [45], or curvature flows with a conformal constraint [12, 33]. These methods based on explicit surface representations are straightforward, but the optimization often suffers from local minima due to the non-convexity of the area functional and can even diverge if the initial mesh has the wrong topology [49, 62].

A different approach to the minimal surface problem is based on *geometric measure theory* (GMT), whose theoretical foundations were developed in the 1960s [25, 26, 42]. In this theory, curves and surfaces are represented implicitly by *currents* as dual to differential forms. Such representations have been used in geometry processing [5, 39, 40, 43] and medical imaging [6, 19, 20, 21, 28, 60]. In geometric measure theory, the minimal surface problem becomes the convex *minimal mass norm* problem (see Section 3.4). A discrete analog of the minimal mass problem on a graph is a linear program known as the optimal homologous chain problem [10, 16, 18, 56]. GMT-based discretization of the minimal surface problem in Euclidean space was pioneered by [47] and revisited by [4, 64].

### 2.2. Deep Learning for Shape Reconstruction

Using deep learning to produce 3D geometry has gained popularity in vision and graphics. Network architectures now can output many explicit shape representations, like voxel grids [13, 67, 71], point clouds [23, 68, 70], meshes [30, 44,

63], and parametric primitives [48, 52, 54, 55, 59]. While the content produced by these approaches is generally easy to render and manipulate, it is often restricted in topology and/or resolution, limiting expressiveness.

A different approach circumvents topology and resolution issues by representing 3D shapes *implicitly*, using functions parameterized by neural networks. In DeepSDF, Park et al. [46] learn a field that approximates signed distance to the target geometry, while Mescheder et al. [38] and Chen and Zhang [7] classify query points as being outside or inside a shape. Others further improve the results by proposing novel regularizers, loss functions, and training or rendering approaches [1, 29, 36, 57]. While these works achieve impressive levels of detail in surface reconstruction, they largely suffer from two drawbacks—lack of control and inability to represent open surfaces, i.e., those with boundary.

Neural implicit learning methods typically overfit to a single target shape or learn a family of shapes parameterized by a high-dimensional latent space. While recent work has shown the possibility of adapting classical geometry processing algorithms to neural implicit geometries [69], applying targeted manipulations and deformations to learned shapes remains nontrivial. Several papers propose *hybrid representations*, combining the expressive power of neural implicit representations with the control afforded by explicit representations. Genova et al. [27] reconstruct shapes by learning multiple implicit representations arranged according to a learned template configuration. In DualSDF [31], manipulations can be applied to learned implicit shapes by making changes to corresponding explicit geometric primitives. BSP-Net [8] and CvxNet [14] restrict the class of learned implicit surfaces to half-spaces and convex hulls, respectively. [37] defines local implicit functions on point clouds, facilitating the transition between discrete points and smooth surfaces during training.

Because neural implicit shapes are typically level sets of learned functions, this limits the class of representable shapes to closed surfaces. Two notable exceptions are [9], which learns *unsigned* distance functions rather than SDFs, and [61], which maps an input point to its closest point on the target surface. Our DeepCurrents adopt a hybrid representation, which models boundaries explicitly and allows them to be used as handles for manipulation.

## 3. Preliminaries

Geometric measure theory is a vast field that we will not attempt to summarize here. For a comprehensive treatment, we refer the reader to [24, 35, 53]. We focus on the rudiments necessary to construct our optimization problem in dimensions two and three, eliding technical issues that arise in higher-dimensional ambient spaces.

The theory of currents is motivated by solving *Plateau's Problem*, the problem of finding the surface of minimal area

enclosed by a given boundary:

$$\arg\min_{\Sigma}\{A(\Sigma) : \partial\Sigma = \Gamma\}. \tag{1}$$

The problem (1) seeks a solution in the space of smooth embedded submanifolds with boundary, which lacks convenient properties such as convexity and compactness required for reasoning about optimization. In GMT, this space is *relaxed* to a space of *currents*, generalized submanifolds characterized by integration. Plateau's problem (1) is systematically translated into a problem over currents. The area functional becomes the convex *mass norm*, and $\partial$ becomes a linear operator constructed by dualizing the exterior derivative d. We describe this translation in detail below.

### 3.1. Currents

Currents are to submanifolds as distributions are to sets of points. Just as distributions are characterized by integration against functions, $k$-currents are characterized by integration against differential $k$-forms in the ambient space. For our purposes, that ambient space will be an open subset $U \subseteq \mathbb{R}^d$, $d \leq 3$. For now, we will assume the metric is Euclidean; see Section 4.2 for the generalization to Riemannian metrics. We will also assume that $U$ is bounded and contractible to elide various technical issues.

We denote the space of smooth $k$-forms with compact support in $U$ by $\Omega_c^k(U)$. Recall that a $k$-form $\zeta \in \Omega_c^k(U)$ smoothly assigns to each point $x \in U$ an element $\zeta_x \in \bigwedge^k T_x^* U$, the exterior power of the cotangent space at $x$. In Euclidean space, there is a canonical identification between covectors ($k = 1$) and vectors. A Riemannian metric provides a similar identification but requires more careful bookkeeping (see Section 4.2).

The space of $k$-currents

$$\mathcal{D}_k(U) = (\Omega_c^k(U))^* \tag{2}$$

is the *dual space* of (compactly-supported) $k$-forms, i.e., it consists of continuous linear functionals on $k$-forms. An element $T \in \mathcal{D}_k(U)$ is defined by its assignment of real values to $k$-forms:

$$\zeta \in \Omega_c^k(U) \mapsto T(\zeta) \in \mathbb{R}. \tag{3}$$

The following are two key examples of currents:

• A 0-current is simply a distribution, as

$$\mathcal{D}_0(U) = (\Omega_c^0(U))^* = (C_c^\infty(U))^* = \mathcal{D}(U). \tag{4}$$

• A submanifold $\Sigma \subset U$ of dimension $k$ can be viewed as a current $[\Sigma] \in \mathcal{D}_k(U)$ by integration against it:

$$[\Sigma](\zeta) := \int_\Sigma \zeta. \tag{5}$$

### 3.2. Boundary Operator

In generalizing the boundary operator from submanifolds to currents, we need to ensure that $\partial[\Sigma] = [\partial\Sigma]$. Stokes' Theorem tells us that

$$[\partial\Sigma](\zeta) = \int_{\partial\Sigma} \zeta = \int_\Sigma \mathrm{d}\zeta = [\Sigma](\mathrm{d}\zeta), \tag{6}$$

motivating the definition

$$\partial T(\zeta) := T(\mathrm{d}\zeta). \tag{7}$$

In words, we define $\partial$ as the adjoint of d.

### 3.3. Mass Norm

As we did for the boundary operator, we write the area functional in terms of integration against forms and then replace integration by current evaluation. This definition depends on a pointwise norm $|\cdot|$ on $k$-forms. As we are working in dimensions $d \leq 3$, it is sufficient to use the pointwise inner product norm, and we will not concern ourselves with complications that occur in higher dimensions.

If $\Sigma \subset U$ is a smooth $k$-submanifold with boundary, then its area satisfies

$$A(\Sigma) = \sup_{\zeta \in \Omega_c^k(U)} \left\{ \int_\Sigma \zeta : |\zeta_x| \leq 1 \; \forall x \in U \right\}. \tag{8}$$

So we define the *mass norm* of a current $T \in \mathcal{D}_k(U)$ as:

$$\mathbf{M}(T) := \sup_{\zeta \in \Omega_c^k(U)} \{ T(\zeta) : |\zeta_x| \leq 1 \; \forall x \in U \}. \tag{9}$$

### 3.4. Minimal Mass Problem

Applying the transformations above to the problem (1), one obtains a relaxation known as the *minimal mass problem*:

$$\min_{T \in \mathcal{D}_k(U)} \{ \mathbf{M}(T) : \partial T = \Gamma \}. \tag{10}$$

In classical GMT, the current $T$ is taken to be in the space $\mathcal{I}_k(U)$ of *integral $k$-currents*, which roughly means currents that look like integer linear combinations of Lipschitz surfaces. When optimizing over $\mathcal{I}_{d-1}(U)$ in ambient dimension $d \leq 7$, there is an optimal solution corresponding to a smooth submanifold (see [24] Theorem 5.4.15, [53] Theorem 5.8, [35] Theorem 3.10). More recent theory extends this result to optimization over general currents $\mathcal{D}_k(U)$ (see [4] Theorem 2, [53] Remark 5.2).

### 3.5. Representing Currents by Forms

For computational purposes, we follow [64] and optimize over $k$-currents represented by differential $(d-k)$-forms. This allows us to represent currents by neural networks.

A $(d-k)$-form can be identified with a $k$-current $[\omega] \in \mathcal{D}_k(U)$ by defining

$$[\omega](\zeta) := \int_U \omega \wedge \zeta$$

for any $\zeta \in \Omega_c^k(U)$. With this identification, we have by Stokes' Theorem:

$$\partial[\omega](\zeta) = [\omega](\mathrm{d}\zeta) = \int_U \omega \wedge \mathrm{d}\zeta \qquad (11)$$
$$= (-1)^{d-k+1} \int_U \mathrm{d}\omega \wedge \zeta = \left[ (-1)^{d-k+1} \mathrm{d}\omega \right](\zeta).$$

The boundary constraint $\partial[\omega] = \Gamma$ thus becomes an exterior differential equation,

$$\mathrm{d}\omega = \delta_\Gamma, \qquad (12)$$

where $\delta_\Gamma$ is a singular $(d-k)$-form representing $\Gamma$.

Similarly, the mass norm of a $k$-current becomes the $L^1$ norm of a $(d-k)$-form:

$$\mathbf{M}([\omega]) = \|\omega\|_1 = \int_U |\omega(x)| \mathrm{dvol} \qquad (13)$$

where $\mathrm{dvol}$ is the volume form.

## 4. DeepCurrents

In the previous section, we described the relaxation of Plateau's minimal surface problem into a convex optimization problem over a space of currents, with the property that its optima include smooth surfaces, and we showed how to represent certain currents by differential forms. In the following section, we introduce our novel neural representation of currents and SGD mass minimization.

### 4.1. Neural Representation

The linear space of solutions to (12) can be parameterized using the Hodge decomposition as follows:

$$\omega = \mathrm{d}f + \alpha, \qquad (14)$$

where $\alpha \in \Omega^{d-k}(U)$ is any particular solution of (12), $f \in \Omega^{d-k-1}(U)$; we ignore the harmonic term as $U$ is contractible. For curves in $U \subset \mathbb{R}^2$ ($k = 1$, $d = 2$) and surfaces in $U \subset \mathbb{R}^3$ ($k = 2$, $d = 3$), $f$ will simply be a function on $U$, which we can represent by a neural network. It is convenient to use the (Euclidean) musical isomorphism $\sharp$ to encode our 1-form $\omega$ as the vector field $\omega^\sharp$. Intuitively, the vector field corresponding to a current points in the surface normal direction. Under this identification, $(\mathrm{d}f)^\sharp = \nabla f$, which can be computed by autodifferentiation.

As for $\alpha$, there is a particularly convenient choice known as the *Biot-Savart field*, which can be written in closed form when $\Gamma$ is a polygonal curve (see [65]):

$$\alpha^\sharp(x) := \int_\Gamma \frac{d\vec{\ell} \times \vec{r}}{|\vec{r}|^3} = \sum_i \frac{(\hat{t}_i \cdot (\hat{r}_i^1 - \hat{r}_i^0))(\hat{t}_i \times \vec{r}_i^0)}{|\hat{t}_i \times \vec{r}_i^0|^2}, \quad (15)$$

where $d\vec{\ell}$ denotes the vector arc measure on $\Gamma$, $\hat{t}_i$ is the unit tangent vector to the $i$th segment of $\Gamma$, $\vec{r}_i^0$ and $\vec{r}_i^1$ are, respectively, the vectors from the point $x$ to the initial and final vertices of segment $i$, and $\hat{r}_i^0$ and $\hat{r}_i^1$ are their normalized directions. In practice, we scale $\alpha$ by $10^{-3}$ to better match the normalization of our network weights; this only changes the mass minimization problem by a uniform scale. Figure 1(a) visualizes a Biot-Savart field in 2D.

Enacting the choices above, we can use neural networks to solve the minimal mass problem:

$$\underset{\theta}{\arg\min} \|\mathrm{d}f_\theta + \alpha_\Gamma\|_1$$
$$= \underset{\theta}{\arg\min} \, \mathbb{E}_{x \sim \mathcal{U}_U} \left[ \left| \nabla_x f_\theta(x) + \alpha_\Gamma^\sharp(x) \right| \right], \qquad (16)$$

where $f_\theta$ is a neural network with weights $\theta$, $\mathcal{U}_U$ is the uniform distribution over $U = [-1, 1]^d$. $\nabla$ is computed exactly via automatic differentiation, and $\alpha_\Gamma^\sharp$ is the boundary-dependent Biot-Savart field, computed in closed form. The expectation in (16) is approximated by uniform sampling over $U$, yielding a method to compute minimal surfaces via stochastic gradient descent (SGD).

In relation to previous discretizations of currents and the minimal surface problem like [64], we (a) represent $f$ via a neural network rather than a voxel grid; (b) evaluate $\alpha$ in closed-form; and (c) evaluate the mass norm as an expectation that is amenable to SGD. These key choices allow our minimal surfaces to achieve arbitrary resolution.

### 4.2. Modifying the Metric

Critical to the computer vision applications considered in this paper, we can use a background Riemannian metric to encode general surfaces that are not minimal under the Euclidean metric. The properties of mass norm minimization almost certainly carry over—in particular, the regularity of minima (see, e.g., [41]).

Let $g$ be a Riemannian metric given by

$$g(X, Y) = \langle AX, Y \rangle = \langle X, AY \rangle \quad \forall X, Y \in TU, \quad (17)$$

where $\langle \cdot, \cdot \rangle$ is the Euclidean inner product and $A$ is a smoothly varying symmetric positive definite linear map on the tangent bundle ($A_x : T_x U \to T_x U$). The Riemannian pointwise norm for a $k$-form $\zeta$ is given by
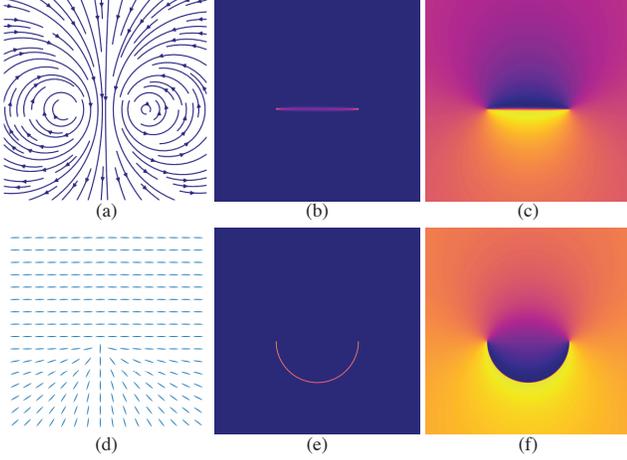
$$|\zeta_x|_g = |(A_x^{-1/2})^* \zeta_x|, \qquad (18)$$

Figure 1. Minimizing the mass norm $\|df + \alpha\|_1$ under the Euclidean metric in two dimensions yields a line segment connecting the two boundary points (b). With our custom data-dependent background metric, we can reconstruct the semicircle as a current (e). $\alpha$ is shown as a vector field (a) and the custom metric is depicted by oriented ellipsoids (d, not to scale). Corresponding functions $f$ are shown at right (c and f).

where $|\cdot|$ is the Euclidean pointwise norm and $B^*\zeta$ denotes the pullback form defined by $B^*\zeta(X_1, \ldots, X_k) = \zeta(BX_1, \ldots, BX_k)$. The $g$–mass norm is then:

$$\mathbf{M}_g([\omega]) = \|\omega\|_{1,g} = \int_U |(A^{-1/2})^*\omega|(\det A)^{d/2}\mathrm{dvol}. \quad (19)$$

The differential equation (12) and its solution (14) are topological and do not change.

In summary, the Riemannian problem differs from the Euclidean one by a symmetric positive definite matrix $B_x$:

$$\begin{aligned}
&\arg\min_{\theta} \|\mathrm{d}f_\theta + \alpha_\Gamma\|_{1,g} \\
&= \arg\min_{\theta} \mathbb{E}_{x \sim \mathcal{U}_U}\left[\left|B_x(\nabla_x f_\theta(x) + \alpha_\Gamma^\sharp(x))\right|\right].
\end{aligned} \quad (20)$$

In the two-dimensional example depicted in Figure 1, minimizing the mass norm under the Euclidean metric yields a straight line segment (b). Changing the metric (d) yields a semicircle (e) instead. Corresponding density plots of $f$ are shown in (c) and (f), respectively.

### 4.3. Loss Functions

The main objective function optimized by our training procedures is the *current loss*, which follows from (20):

$$\mathcal{L}_{\mathrm{curr}}(\cdot) = \mathbb{E}_{x \sim \mathcal{U}_U}\left[\left|B_x(\nabla_x f_\theta(x) + \alpha_\Gamma^\sharp(x))\right|\right]. \quad (21)$$

We approximate the expectation by a sample average over a sample drawn from the uniform distribution on $U$.

For minimal surface computation (Section 5.1), we set $B_x = I$ for all $x \in U$. For surface reconstruction (Sections 5.2 and 5.3), we define

$$B_x = w_x(I - \hat{n}_{\mathrm{proj}_\Sigma(x)}\hat{n}_{\mathrm{proj}_\Sigma(x)}^\top), \quad (22)$$

where $\Sigma$ is the ground truth surface, $\mathrm{proj}_\Sigma(x)$ is the closest point on $\Sigma$ to $x$, and $\hat{n}_{\mathrm{proj}_\Sigma(x)}$ is its unit normal. This positive semidefinite matrix, corresponding to a degenerate Riemannian metric, penalizes the current's deviation from agreement with the surface's orientation. A patch aligned with $\Sigma$ (i.e., where $\nabla f + \alpha_\Gamma^\sharp \parallel \hat{n}$) costs nothing.

When evaluating Equation (20), for half of the samples in $U$, we set $w_x = 1$, and for the other half, we set

$$w_x = \exp\left(-\frac{1}{2\sigma^2}\|x - \mathrm{proj}_\Gamma(x)\|_2^2\right), \quad (23)$$

where $\mathrm{proj}_\Gamma(x)$ is the closest point on the boundary to $x$ under Euclidean distance, and $\sigma = 0.1$ in practice. We find empirically that adding this boundary weighting, where samples close to the prescribed boundary have a higher contribution to the current loss, with a Gaussian falloff, slightly improves our learned surfaces, particularly near the boundary. See Section 5.4 for an ablation study.

For surface reconstruction, we employ an additional loss term to guide our optimization. We define *surface loss* as:

$$\mathcal{L}_{\mathrm{surf}}(\cdot) = \mathbb{E}_{x,\epsilon}\left[(\delta - f(x - \varepsilon n_x) + f(x + \varepsilon n_x))^+\right], \quad (24)$$

where $x \sim \mathcal{U}_\Sigma$, the uniform measure on the target surface $\Sigma$, $n_x$ is the (oriented) surface normal vector at a point $x \in \Sigma$, $\varepsilon$ and $\delta$ are small threshold. In practice, we set $\delta = 0.01$, randomly pick $\varepsilon \sim \mathcal{U}_{[0.0199, 0.0201]}$, and approximate the expectation by sampling on $\Sigma$.

This hinge loss encourages the values of our learned function $f$ to differ by no less than a margin $\delta$ across the target surface. This objective may seem redundant as the metric (22) already encourages alignment to the target surface. In fact, the two are complementary—the surface loss encourages $f$ to jump near the target surface, while the current loss ensures that the bandwidth of the jump decreases. We find that using the surface loss term helps our models converge to better optima (see our ablation study in Section 5.4).

### 4.4. Network Architecture

We learn a single current $\mathrm{d}f_\theta + \alpha$, using a deep neural network to parameterize $f_\theta : \mathbb{R}^3 \to \mathbb{R}$. Given an input point $x \in \mathbb{R}^3$, we first project it onto a random Fourier feature (RFF) space, as in [58], to obtain $\hat{x} \in \mathbb{R}^{2048}$. Our RFF coefficients are 2048-dimensional and sampled from $\mathcal{N}(0, 4)$. We then decode the RFF vector to a scalar value $f_\theta(x)$ using an MLP $h_\theta$, which consists of three hidden layers, each with 256 units and softplus nonlinearities. This pipeline is illustrated in the top half of Figure 2.
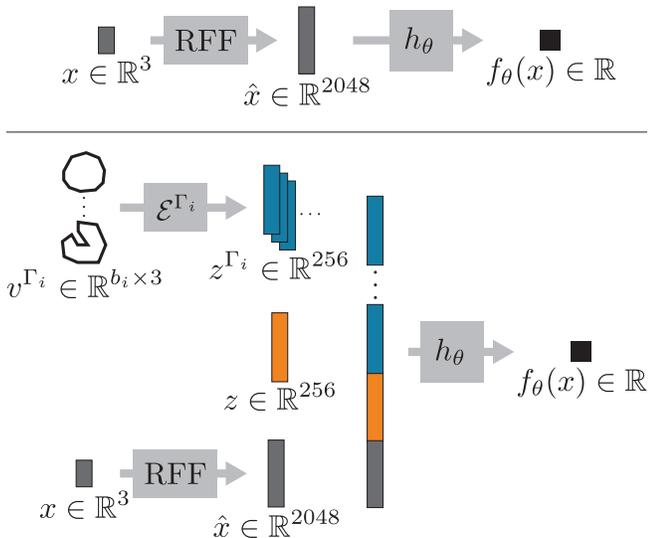
Figure 2. An overview of our network architectures for minimal surface optimization and single surface reconstruction (top) as well as shape space learning (bottom). An input point $x$ is first encoded using random Fourier features. These features are then optionally concatenated with latent codes corresponding to shape identity and boundary and finally decoded to a scalar output.

Additionally, we propose a boundary-conditioned autodecoder architecture for learning families of currents (see Figure 2, bottom). We initialize a latent code $z_j \sim \mathcal{N}(0, 0.1)$ for each mesh, and we encode the mesh boundary geometry using a boundary encoder $\mathcal{E}^\Gamma$. For shapes that have more than one boundary, we use a separate encoder for each boundary loop to obtain a set of boundary latent codes $z^{\Gamma_i}$. We then concatenate the latent codes along with the RFFs $[z_j \mid z_j^{\Gamma_1} \mid \ldots \mid z_j^{\Gamma_B} \mid \hat{x}]$ and pass this vector through a decoder $h_\theta$ as in the overfitting setting above.

Our boundary encoder inputs boundary vertices $v^\Gamma \in \mathbb{R}^{b \times 3}$. The encoder is a network with three 1-dimensional convolutional layers with stride 1 and circular boundary conditions. The first layer uses a kernel of size 5 while the latter two use kernels of size 3. Each layer has 256 channels, and we use ReLU after each layer except the last. After the convolutions, we take the mean across all the boundary vertices to obtain the boundary latent code. Circular convolutions combined with mean pooling ensure that our encoder is invariant to cyclic permutations of the vertices, which correspond to the same boundary geometry.

## 5. Experimental Results

We evaluate DeepCurrents experimentally by demonstrating results on minimal surface computation, overfitting for single surface reconstruction, and shape space learning and interpolation. We also show an ablation study to validate our main design choices. All of our models are trained on a single NVIDIA GeForce RTX 3090 GPU using Adam [34].
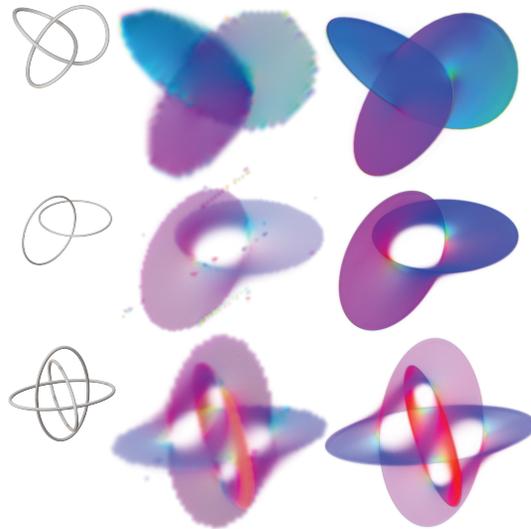


Figure 3. Minimal currents computed via [64] on a $90 \times 90 \times 90$ grid (middle) display prominent grid artifacts, especially near the boundary. In contrast, with a similar total number of parameters (725,249 weights), our DeepCurrents achieve higher effective resolution (right). Boundaries (left) are the trefoil knot (top), Hopf link (middle), and Borromean rings (bottom).

### 5.1. Minimal Surfaces

We use our method to compute minimal surfaces for three boundary configurations. We train each model for $10^5$ iterations ($\sim 12$ minutes) with a learning rate of 0.0005, sampling 4096 points from the ambient space at each step and reducing the learning rate by a factor of 0.6 every 10,000 steps. We only optimize $\mathcal{L}_{\text{curr}}$ with the Euclidean metric in these examples—we do not use $\mathcal{L}_{\text{surf}}$ or boundary weighting.

Figure 3 compares our results to those from [64], which uses a voxel grid; the colors represent local current orientation, which corresponds to the surface normal direction. For fair comparison, we choose the grid size to approximate our number of trainable parameters ($90^3 \approx 725,249$). While our learned currents adhere well to the smooth input boundaries, the currents of [64] show significant grid artifacts. Thus, our representation exhibits greater capacity to encode high-resolution surfaces with the same number of parameters

### 5.2. Surface Reconstruction

We perform surface reconstruction using DeepCurrents by overfitting to several segmented parts of models from the FAUST human body dataset [2]. We preprocess the data by splitting the mesh according to the provided segmentations, rigidly aligning all the models within each segmentation class, and rescaling them to fit into $[-0.5, 0.5]^3$.

We train each model for 10,000 iterations ($\sim 4$ minutes) with an initial learning rate of 0.001, decayed by a factor of 0.6 every 2000 iterations. We sample 4000 random points from the ambient space (to compute $\mathcal{L}_{\text{curr}}$) and 4000 points
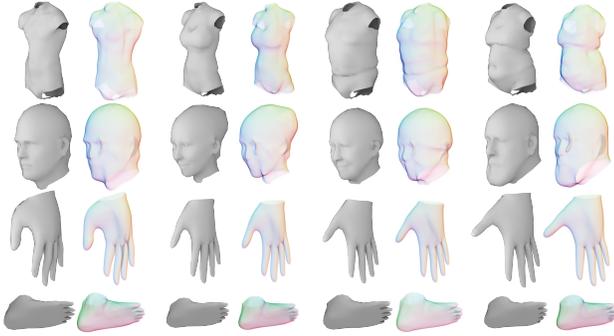
Figure 4. Human body surface reconstructions. We overfit Deep-Currents models to reconstruct several torso, head, hand, and foot meshes. We show a volume rendering of each learned current (right) next to the ground truth mesh (left).

| Model | UCD ([61]) | UCD (ours) |
|-------|-----------|------------|
| head  | 0.0049    | **0.0010** |
| hand  | 0.0045    | **0.0011** |
| torso | 0.0049    | **0.00092** |
| foot  | 0.0055    | **0.00092** |

Table 1. Quantitative comparison of unidirectional Chamfer distance to [61] on single surface reconstruction of random models from each of four shape categories.

from the mesh surface (to compute $\mathcal{L}_{\text{surf}}$) at each step.

We show results on torsos, heads, hands, and feet from randomly chosen models in Figure 4. Our currents faithfully reconstruct the target geometry.

Additionally, we compare quantitatively to [61] in Table 1. We train their model for the same amount of time as ours on randomly picked models. Because their model predicts the closest point on the target surface given any input point, we use this to compute unidirectional Chamfer distance (i.e., $\mathbb{E}_{y \sim \mathcal{U}_\Sigma}[\text{dist}_{\Sigma^*}(y)]$, where $\mathcal{U}_\Sigma$ is the uniform distribution on the ground truth mesh, and $\text{dist}_{\Sigma^*}$ is Euclidean distance to the learned surface).

We do the same for our method by meshing our learned current: We compute the average value $s$ of $f$ over a boundary curve; this makes sense even if there are multiple boundary curves, assuming our surface has one connected component. Then, we extract a mesh of the level set $f^{-1}(s)$ using marching cubes. This level set is generically a closed surface containing our represented surface with boundary $\Sigma^*$ as a subset. We extract a mesh of $\Sigma^*$ by removing vertices $x$ for which $|\nabla_x f_\theta(x) + \alpha_\Gamma^\sharp(x)| < \delta$. In practice, we use $\delta = 5 \times 10^{-3}$.

Our method consistently achieves better quality reconstructions than [61].

## 5.3. Latent Space Learning

We use our boundary-conditioned autodecoder (Section 4.4) to learn a disentangled representation that can interpolate in a high-dimensional learned latent space capturing shape identity while having explicit control over boundary geometry. We associate each mesh in our dataset with a random latent code (a trainable parameter), and, to disambiguate shape identity from boundary geometry, we perform random transformations. These transformations change the boundary shape while preserving the latent code.

At each iteration, we perform random augmentations to the target meshes: we rotate each mesh by sampling a value in $[-10°, 10°]$ for each Euler angle, we rescale each boundary loop of the mesh by a random factor between 0.85 and 1.15 along each of its two principal directions, we propagate these transformations to the entire mesh using harmonic skinning weights, and we shift the mesh by a random offset between $-0.05$ and $0.05$ in each dimension.

We train a model for each shape category for 300,000 iterations (about 10 hours) with an initial learning rate of 0.0004, decayed by a factor of 0.5 every 60,000 iterations. At each step, we sample a random batch of 8 meshes and sample 4000 points from each mesh.

In Figure 5, we pick two models from each shape category and independently interpolate between their boundaries and latent identities. Our model disentangles high-level pose and style while respecting the prescribed geometry.

## 5.4. Ablation Study

We validate some of our key design choices. In Figure 6, we overfit five models to the same hand mesh. While our full model achieves a sharp reconstruction, removing boundary weighting from our current loss metric yields a fuzzier surface around the boundary. Changing the softplus activation functions in $h_\theta$ to ReLUs makes the entire learned surface significantly less sharp, which we conjecture is due to ReLU's zero second derivative when optimizing currents $df + \alpha$. Removing the surface loss term from our optimization fails to recover much of the target surface, supporting our claim that surface loss significantly helps convergence. Finally, foregoing the projection of the input points onto random Fourier features prevents the model from learning.

## 6. Conclusion

By adopting tools from geometric measure theory, we have constructed a neural implicit representation for surfaces with boundary. Our SGD approach to mass norm minimization enables computing minimal surfaces with arbitrary resolution, in contrast to previous work that represents currents on a fixed-resolution grid. In addition, by constructing a background metric, we can engineer a mass minimization problem to encode an arbitrary surface. Combining this con-
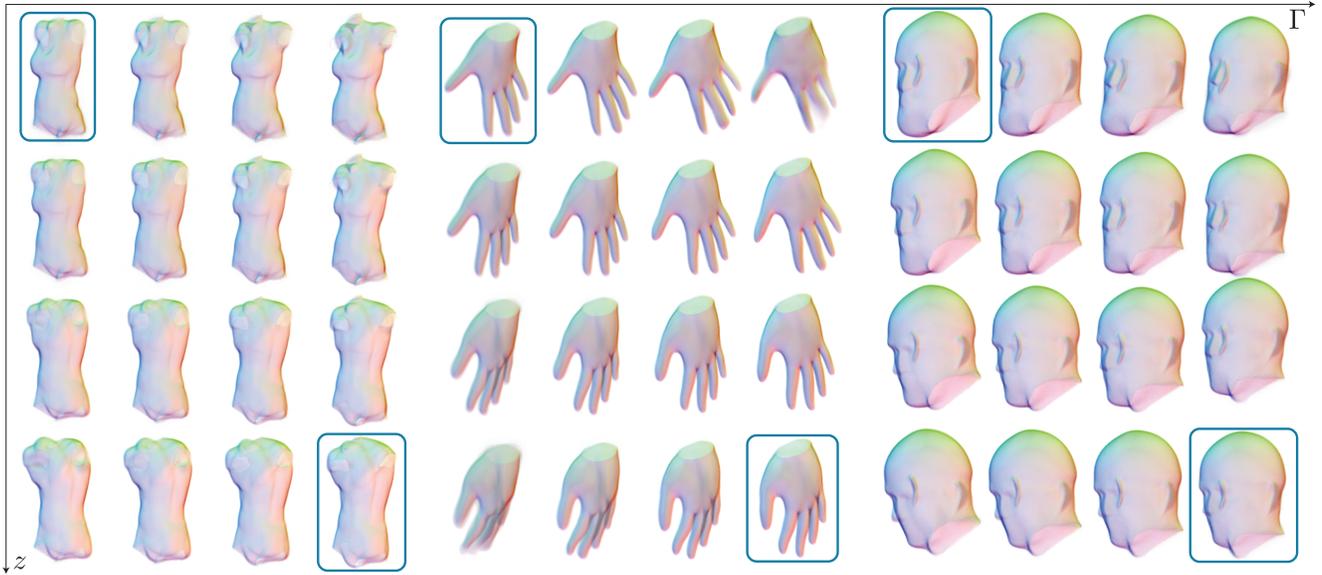
Figure 5. Interpolations of DeepCurrents in latent and boundary space. For each category, we pick two meshes from the training set (shown with a blue border) and interpolate linearly between the two boundaries (horizontal axis) as well as the two latent codes (vertical axis). The latent space interpolation yields a smooth transition between the two meshes while obeying the prescribed boundary interpolants.
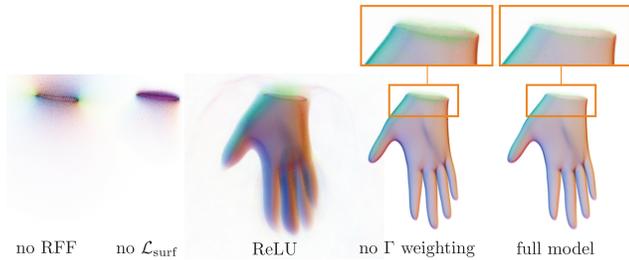


Figure 6. Ablation study. From left to right: reconstruction results on a hand without random Fourier features, without our surface loss term, without boundary weighting, and our full model.

struction with the expressive power of neural representations, we can encode whole families of surfaces.

We see DeepCurrents as a key tool for building flexible neural surface representations. Stitching together DeepCurrents along their boundaries would produce a hybrid surface representation where the explicit boundary curves provide "handles" for user control. Such a representation would be applicable where the target surface is decomposed into parts. Unlike, say, a mesh decomposition, a DeepCurrent decomposition would not require the parts to have simple shapes or even to be simply connected.

Another direction for future work would be to investigate other loss functions and optimization problems that can be expressed in the language of currents. For example, the convex problems studied in [39] could be optimized using a neural representation and SGD. One could also compute minimal currents in spaces such as the rotation groups $\mathrm{SO}(d)$ or special Euclidean groups $\mathrm{SE}(d)$. Mass minimization in

this context could provide a useful prior for reconstruction of shapes that come with an orientation or frame field, or it could exploit the Gauss map to encode a smoothness prior.

Another extension of our method would be to support periodic minimal surfaces, i.e., replacing the domain $[-1, 1]^3$ by the torus $\mathbb{T}^3$. This would require a modification of our explicit $\alpha$ and the evaluation of $\mathrm{d}f$ at the boundary.

While our latent space model often produces high-quality interpolants, they are not explicitly regularized to encourage them to look like surfaces. This sometimes yields fuzzy results (see Figure 5, top right among the hands). Future work could design loss terms to ensure interpolants remain minimal with respect to some metric—analogously to the eikonal regularization for SDFs in [29].

## Acknowledgements

# References

[1] M. Atzmon and Y. Lipman. SAL: Sign agnostic learning of shapes from raw data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[2] F. Bogo, J. Romero, M. Loper, and M. J. Black. FAUST: Dataset and evaluation for 3D mesh registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3794–3801, 2014. 6

[3] K. A. Brakke. The surface evolver. *Experimental Mathematics*, 1(2):141–165, 1992. 2

[4] H. Brezis and P. Mironescu. The Plateau problem from the perspective of optimal transport. *Comptes Rendus Mathematique*, 357(7):597–612, 2019. 2, 3

[5] B. Buet, G. P. Leonardi, and S. Masnou. Discretization and approximation of surfaces using varifolds. *Geometric Flows*, 3(1):28–56, 2018. 2

[6] N. Charon and A. Trouvvé. Functional currents: a new mathematical tool to model and analyse functional shapes. *Journal of Mathematical Imaging and Vision*, 48(3):413–431, 2014. 2

[7] Z. Chen and H. Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5939–5948, 2019. 1, 2

[8] Z. Chen, A. Tagliasacchi, and H. Zhang. BSP-Net: Generating compact meshes via binary space partitioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[9] J. Chibane, A. Mir, and G. Pons-Moll. Neural unsigned distance fields for implicit function learning. In *Advances in Neural Information Processing Systems*, 2020-12. 2

[10] D. Cohen-Steiner, A. Lieutier, and J. Vuillamy. Lexicographic optimal homologous chains and applications to point cloud triangulations. In *Proceedings of the 36th International Symposium on Computational Geometry*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020. 2

[11] P. Concus. Numerical solution of the minimal surface equation. *Mathematics of Computation*, 21(99):340–350, 1967. 2

[12] K. Crane, U. Pinkall, and P. Schröder. Spin transformations of discrete surfaces. *ACM Transactions on Graphics*, 30(4):1–10, 2011. 2

[13] J. Delanoy, M. Aubry, P. Isola, A. A. Efros, and A. Bousseau. 3D sketching using multi-view deep volumetric prediction. *Conference on Computer Graphics and Interactive Techniques*, 1(1):1–22, 2018. 2

[14] B. Deng, K. Genova, S. Yazdani, S. Bouaziz, G. Hinton, and A. Tagliasacchi. CvxNet: Learnable convex decomposition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020-06. 2

[15] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, pages 317–324, 1999. 2

[16] T. K. Dey, A. N. Hirani, and B. Krishnamoorthy. Optimal homologous cycles, total unimodularity, and linear programming. *SIAM Journal on Computing*, 40(4):1026–1044, 2011. 2

[17] J. Douglas. A method of numerical solution of the problem of Plateau. *Annals of Mathematics*, pages 180–188, 1927. 2

[18] N. M. Dunfield and A. N. Hirani. The least spanning area of a knot and the optimal bounding chain problem. In *Proceedings of the 27th Annual Symposium on Computational Geometry*, pages 135–144, 2011. 2

[19] S. Durrleman, X. Pennec, A. Trouvé, P. Thompson, and N. Ayache. Inferring brain variability from diffeomorphic deformations of currents: an integrative approach. *Medical Image Analysis*, 12(5):626–637, 2008. 2

[20] S. Durrleman, X. Pennec, A. Trouvé, and N. Ayache. Statistical models of sets of curves and surfaces based on currents. *Medical Image Analysis*, 13(5):793–808, 2009. ISSN 1361-8415. 2

[21] S. Durrleman, P. Fillard, X. Pennec, A. Trouvé, and N. Ayache. Registration, atlas estimation and variability analysis of white matter fiber bundles modeled as currents. *NeuroImage*, 55(3):1073–1090, 2011. ISSN 1053-8119. 2

[22] G. Dziuk. An algorithm for evolutionary surfaces. *Numerische Mathematik*, 58(1):603–611, 1990. 2

[23] H. Fan, H. Su, and L. J. Guibas. A point set generation network for 3D object reconstruction from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, page 6, 2017. 2

[24] H. Federer. *Geometric Measure Theory*. Springer, 1996. ISBN 978-3-540-60656-7. 2, 3

[25] H. Federer and W. H. Fleming. Normal and integral currents. *Annals of Mathematics*, pages 458–520, 1960. 2

[26] W. H. Fleming. Geometric measure theory at brown in the 1960s, 2015. 2

[27] K. Genova, F. Cole, A. Sud, A. Sarna, and T. Funkhouser. Local deep implicit functions for 3D shape. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4857–4866, 2020. 2

[28] J. Glaunes, A. Trouvé, and L. Younes. Diffeomorphic matching of distributions: A new approach for unlabelled point-sets and sub-manifolds matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2. IEEE, 2004. 2

[29] A. Gropp, L. Yariv, N. Haim, M. Atzmon, and Y. Lipman. Implicit geometric regularization for learning shapes. In *Proceedings of Machine Learning and Systems*, pages 3569–3579, 2020. 2, 8

[30] R. Hanocka, G. Metzer, R. Giryes, and D. Cohen-Or. Point2Mesh: A self-prior for deformable meshes. *ACM Transactions on Graphics*, 39(4), 2020. ISSN 0730-0301. 2

[31] Z. Hao, H. Averbuch-Elor, N. Snavely, and S. Belongie. DualSDF: Semantic shape manipulation using a two-level representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7631–7641, 2020. 2

[32] M. Hinata, M. Shimasaki, and T. Kiyono. Numerical solution of Plateau's problem by a finite element method. *Mathematics of Computation*, 28(125):45–60, 1974. 2

[33] M. Kazhdan, J. Solomon, and M. Ben-Chen. Can mean-curvature flow be made non-singular? In *Computer Graphics Forum*, volume 31, pages 1745–1754. Wiley Online Library, 2012. 2

[34] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2014-12. 6

[35] U. Lang. Introduction to geometric measure theory, 2004-04. 2, 3

[36] Y. Lipman. Phase transitions, distance functions, and implicit neural representations. In *Proceedings of the 38th International Conference on Machine Learning*, 2021. 2

[37] S.-L. Liu, H.-X. Guo, H. Pan, P. Wang, X. Tong, and Y. Liu. Deep implicit moving least-squares functions for 3D reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[38] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger. Occupancy networks: Learning 3D reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2

[39] T. Mollenhoff and D. Cremers. Lifting vectorial variational problems: a natural formulation based on geometric measure theory and discrete exterior calculus. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11117–11126, 2019. 2, 8

[40] T. Möllenhoff and D. Cremers. Flat metric minimization with applications in generative modeling. In *International Conference on Machine Learning*, pages 4626–4635. PMLR, 2019. 2

[41] F. Morgan. Regularity of isoperimetric hypersurfaces in Riemannian manifolds. *Transactions of the American Mathematical Society*, 355(12):5041–5052, 2003. 4

[42] F. Morgan. *Geometric measure theory: a beginner's guide*. Academic Press, 2016. 2

[43] P. Mullen, A. McKenzie, Y. Tong, and M. Desbrun. A variational approach to Eulerian geometry processing. *ACM Transactions on Graphics*, 26(3), 2007. 2

[44] C. Nash, Y. Ganin, S. A. Eslami, and P. Battaglia. PolyGen: An autoregressive generative model of 3D meshes. In *Proceedings of the 37th International Conference on Machine Learning*, pages 7220–7229. PMLR, 2020. 2

[45] H. Pan, Y.-K. Choi, Y. Liu, W. Hu, Q. Du, K. Polthier, C. Zhang, and W. Wang. Robust modeling of constant mean curvature surfaces. *ACM Transactions on Graphics*, 31(4): 1–11, 2012. 2

[46] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019-06. 1, 2

[47] H. R. Parks and J. T. Pitts. Computing least area hypersurfaces spanning arbitrary boundaries. *SIAM Journal on Scientific Computing*, 18(3):886–917, 1997. 2

[48] D. Paschalidou, A. O. Ulusoy, and A. Geiger. Superquadrics revisited: Learning 3D shape parsing beyond cuboids. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[49] U. Pinkall and K. Polthier. Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics*, 2 (1):15–36, 1993. 2

[50] E. Popov. On some variational formulations for minimum surface. *Transactions of the Canadian Society for Mechanical Engineering*, 20(4):391–400, 1996. 2

[51] H. Schumacher and M. Wardetzky. Variational convergence of discrete minimal surfaces. *Numerische Mathematik*, 141 (1):173–213, 2019. 2

[52] G. Sharma, D. Liu, S. Maji, E. Kalogerakis, S. Chaudhuri, and R. Měch. ParSeNet: A parametric surface fitting network for 3D point clouds. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 261–276. Springer, 2020. 2

[53] L. Simon. Introduction to geometric measure theory. *Tsinghua Lectures*, 2014. 2, 3

[54] D. Smirnov, M. Fisher, V. G. Kim, R. Zhang, and J. Solomon. Deep parametric shape predictions using distance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[55] D. Smirnov, M. Bessmeltsev, and J. Solomon. Learning manifold patch-based representations of man-made shapes. In *International Conference on Learning Representations (ICLR)*, 2021. 2

[56] J. M. Sullivan. A crystalline approximation theorem for hypersurfaces. *Ann Arbor, Thesis (Ph.D.), Princeton University*, 1990. 2

[57] T. Takikawa, J. Litalien, K. Yin, K. Kreis, C. Loop, D. Nowrouzezahrai, A. Jacobson, M. McGuire, and S. Fidler. Neural geometric level of detail: Real-time rendering with implicit 3D shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[58] M. Tancik, P. P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. T. Barron, and R. Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In *Advances in Neural Information Processing Systems*, 2020. 5

[59] S. Tulsiani, H. Su, L. J. Guibas, A. A. Efros, and J. Malik. Learning shape abstractions by assembling volumetric primitives. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, 2017. 2

[60] M. Vaillant and J. Glaunes. Surface matching via currents. In *Biennial International Conference on Information Processing in Medical Imaging*, pages 381–392. Springer, 2005. 2

[61] R. Venkatesh, T. Karmali, S. Sharma, A. Ghosh, R. V. Babu, L. A. Jeni, and M. Singh. Deep implicit surface point prediction networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12653–12662, 2021-10. 2, 7

[62] H.-J. Wagner. A contribution to the numerical approximation of minimal surfaces. *Computing*, 19(1):35–58, 1977. 2

[63] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, and Y.-G. Jiang. Pixel2Mesh: Generating 3D mesh models from single RGB images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 2

[64] S. Wang and A. Chern. Computing minimal surfaces with differential forms. *ACM Transactions on Graphics*, 40(4): 1–14, 2021. 2, 3, 4, 6

[65] S. Weißmann and U. Pinkall. Real-time interactive simulation of smoke using discrete integrable vortex filaments. In H. Prautzsch, A. Schmitt, J. Bender, and M. Teschner, editors, *Workshop in Virtual Reality Interactions and Physical Simulation "VRIPHYS" (2009)*. The Eurographics Association, 2009. ISBN 978-3-905673-73-9. 4

[66] W. L. Wilson. On discrete Dirichlet and Plateau problems. *Numerische Mathematik*, 3(1):359–373, 1961. 2

[67] J. Wu, C. Zhang, X. Zhang, Z. Zhang, W. T. Freeman, and J. B. Tenenbaum. Learning 3D shape priors for shape completion and reconstruction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 3, 2018. 2

[68] G. Yang, X. Huang, Z. Hao, M.-Y. Liu, S. Belongie, and B. Hariharan. PointFlow: 3D point cloud generation with continuous normalizing flows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4541–4550, 2019. 2

[69] G. Yang, S. Belongie, B. Hariharan, and V. Koltun. Geometry processing with neural fields. In *Advances in Neural Information Processing Systems*, 2021. 2

[70] K. Yin, H. Huang, D. Cohen-Or, and H. Zhang. P2P-NET: Bidirectional point displacement net for shape transform. *ACM Transactions on Graphics*, 37(4):152:1–152:13, 2018. 2

[71] X. Zhang, Z. Zhang, C. Zhang, J. B. Tenenbaum, W. T. Freeman, and J. Wu. Learning to reconstruct shapes from unseen classes. In *Advances in Neural Information Processing Systems*, 2018. 2