Eurographics Symposium on Geometry Processing 2026
M. Skouras and M. Livesu
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 45 (2026), Number 5

Circles of Confidence for Multi-Label Geometry Completion

Z. Wei[1], C. Hafner[2], A. Kalinov[2], P. Heiss-Synak[3], C. Wojtan[2]

[1]University of Southern California, USA [2]ISTA, Austria [3]Australian National University, Australia

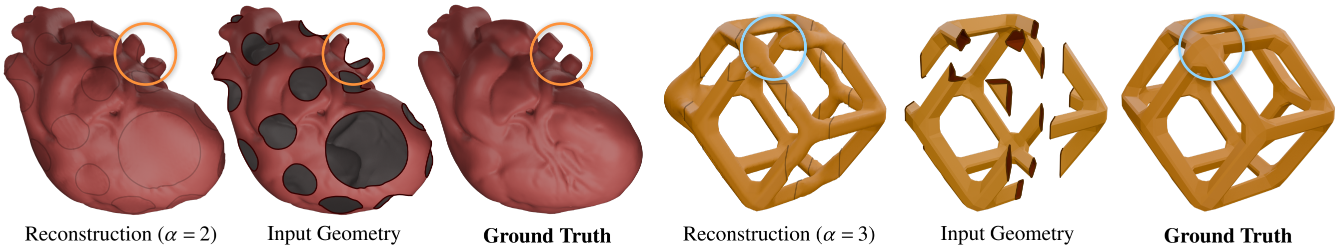


Figure 1: *Reconstruction of* Heart *and* Polyhedron *models with our method.*

Abstract

Inside–outside classification is widely used for geometry processing tasks such as surface reconstruction, geometry completion, and calculating signed distance fields. We introduce a new integral formulation of this problem, which assigns confidence scores that points are inside or outside, given incomplete boundary geometry. Even though our geometric construction does not appear in previous work, we show that it is unexpectedly linked to both the well-established generalized winding number (GWN) and pseudonormal methods for geometry completion, and it provably reduces to either one of them for specific values of a control parameter. The results obtained with our method frequently outperform screened Poisson surface reconstruction (PSR), GWN, and the pseudonormal method in terms of quality, and are at least on par with them on all of our examples. Unlike these methods, our algorithm naturally extends to the multi-label setting, in which regions with an arbitrary number of colors or physical materials can be reconstructed, and non-manifold features such as T-junctions may appear in the interface and boundary geometry.

CCS Concepts

• ***Computing methodologies* → *Shape modeling;* • *Theory of computation* → *Computational geometry;***

1. Introduction

This paper focuses on determining which regions of space are inside or outside of incomplete boundary geometry. While clean, watertight boundary data produces an explicit segmentation of space, broken surfaces and point clouds introduce uncertainty.

Some popular methods for inside–outside segmentation disambiguate this uncertainty by filling space with a scalar function whose zero-level set defines the location of the missing surface: Poisson surface reconstruction (PSR) [KBH06] solves an elliptic partial differential equation (PDE) to generate a smoothed indicator function from oriented point cloud data, and generalized winding numbers (GWN) [JKS13] specify a harmonic function based on the solid angle spanned by a surface. These techniques are extremely useful and influential for solving this problem, but they do exhibit some drawbacks. Due to the non-locality of both constructions, *every* part of the input geometry has a non-zero influence on the resulting surface, regardless of distance and occlusion. This can lead to unintuitive and potentially undesirable properties of the resulting surface, like unexpected bumps or topological connections that are sensitive to the orientation and noise of disconnected distant shapes, or the number and size of holes in the input.

Alternatively, approaches based on the pseudonormal [BA05] ignore corruptions caused by distant shapes by depending *only* on the single closest point on the input geometry. Surface reconstructions based on this local information are entirely robust to distant flaws in the input, but they lack enough geometric context to produce clean reconstructions, often resulting in geometry that is topologically correct but non-smooth and jagged.

Both these entirely local and highly global approaches have their merits, but it can be advantageous for different applications to exhibit behaviors somewhere in between these two extremes. The signed heat method [FC24], for example, smoothly interpolates between the pseudonormal and more global harmonic behavior by computing the solution to a heat equation, where the local/global trade-off is implicitly controlled by a diffusion time parameter. Because the solution comes from a smooth elliptic PDE, it is similar to global methods in that it can be sensitive to non-local geometric noise and connectivity.

This paper proposes a new scoring function based on *circles of confidence*, an integral formulation for aggregating evidence that a point is inside or outside, using local information from the input geometry. Our construction is geometrically elementary and uses only familiar concepts such as greatest empty circles, resulting in an explicit method with results that are easy to explain and interpret. The primary control parameter $\alpha > 0$ of our algorithm lets the user explore a continuous family of reconstructions, ranging from results with a large (but still locally supported) region of influence to the maximally localized behavior of the pseudonormal method in the limit $\alpha \to \infty$. The confidence score introduced in this manuscript is amenable to rigorous analysis, which allows us to describe the mathematical properties of the method for different values of α. In particular for $\alpha = 2$, our method admits a precise interpretation as a “GWN method with occlusion”, which lets it inherit favorable properties of GWN, such as good surface quality, while avoiding catastrophic failure cases due to global effects.

Finally, our technique naturally generalizes beyond inside–outside classification and supports the reconstruction of multiple regions with interior walls separating them. This is useful for processing vector graphics having areas with different fill colors, regions with different physical materials, or generating interfaces with non-manifold features such as T-junctions.

2. Related Work

Early works on **hole filling** targeted small mesh defects such as cracks and gaps, common artifacts of meshing CAD models at the time, and were based on explicit remeshing operations [BK97; BNK02]. Algorithms with strong robustness guarantees, such as always producing a closed manifold surface, were first developed by converting the input to a grid-based representation that simplifies topological queries [Ju04; BPK05].

Following these developments, **implicit methods** started gaining traction: Methods of this type are based on constructing a scalar function whose level sets can be used to complete missing geometry. The *pseudonormal method* defines a notion of signed distance to a mesh with holes, based on closest-point queries and a normal interpolation scheme [BA05]. Here, the zero-level set of the signed distance field provides the closing surface. Our algorithm shares similarities with this method, and even recovers its result exactly for a specific choice of parameters. Another recent work based on signed distances diffuses the normals of the available geometry to obtain an estimate of the signed distance gradient, an approach that also works on curved domains [FC24].

Other implicit methods estimate an **indicator function** of the mesh interior, and level sets of this function are used to fill holes in its boundary. Most of these methods are based either on *Poisson surface reconstruction* (PSR) or the *generalized winding number* (GWN) algorithm. The **PSR algorithm** [KBH06] is based on the observation that the distributional gradient of the indicator function is a normal-oriented delta distribution supported on the boundary. This distribution can be approximated by convolving the available normal samples with a smoothing kernel. Reconstruction of the indicator function χ is done by minimizing $\|\nabla\chi - n\|^2$, where n is the smoothed normal field. This influential idea has been extended with point interpolation constraints [KH13] and envelope constraints [KCRH20]; a stochastic version that offers answers to statistical queries such as confidence intervals [SJ22] and a learning-based extension [SJ23] have been developed recently.

The **GWN algorithm** [JKS13] uses the representation of the indicator function as an integral over the (solid) angle subtended by the domain boundary, often encoded by the faces of a surface mesh. This computation naturally extends to defective meshes with holes, in which case a smoothed indicator function is recovered. Extensions speed up the computation of the GWN and allow more general inputs such as point clouds [BDS*18], perform computations on curved domains [FGC23], increase robustness with regularized kernel functions [CMG24], and admit gradient-based optimization through smoothing [SWBH24]. Recent work has emphasized processing of curved rather than piecewise-linear input geometry and exploited analytical dimensionality reduction [SGW24; MB25]. Our geometric construction is different from GWN, but we show a surprising connection to the local behavior of our method.

Little work is available for filling holes in **non-manifold geometry**, such as meshes with T-junctions that represent the boundary and interfaces between several adjacent volumes. Algorithms that explicitly allow non-manifold input geometry usually try to remove these features and produce a manifold output [WLG03; DA21]. An exception is the mesh-repair step in a recent non-manifold surface tracking pipeline, where non-manifold features are reconstructed using a flood-fill algorithm [HKS*24]. Our algorithm naturally extends to the non-manifold setting, and is the first to combine this feature while producing results akin to established methods such as the GWN, PSR, and pseudonormal algorithms.

A more comprehensive overview of work on hole filling and mesh repair can be found in recent survey articles on surface reconstruction [ACK13; GXW18], and a survey on techniques for preparing geometry for 3d printing [LEM*17].

3. Method

We present a new algorithm for completing curves or surfaces from partially available boundary data, founded on the idea that each piece of the boundary geometry gives us some confidence that points to one side will lie inside the enclosed object, and points to the other side, outside. We first explain our *circles of confidence* method in simplified form and show that this yields a novel geometric interpretation of the generalized winding number (GWN) algorithm. Then we introduce extensions that let us control the qualitative behavior of the method and make it applicable to the multi-label setting, in which geometry with T-junctions can also be reconstructed. We explain and illustrate the method in $\mathbb{R}^2$, but an

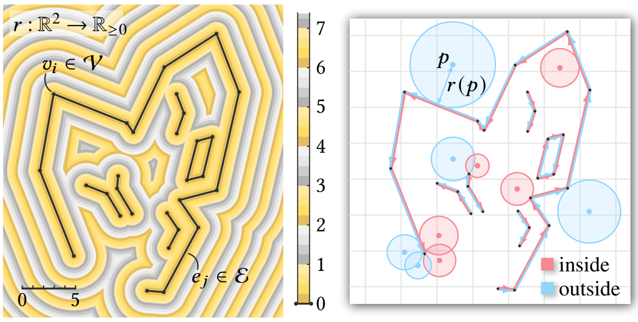


Figure 2: Left: *The input geometry consists of line segments forming polygons and polylines. The unsigned distance transform assigns to every point $p \in \mathbb{R}^2$ the distance $r(p)$ to the closest point in the geometry.* Right: *The distance value gives the radius of the largest empty ball centered at p. If a circle touches an input edge from the left, it receives the "inside" label* (red)*; otherwise, the "outside" label* (blue).

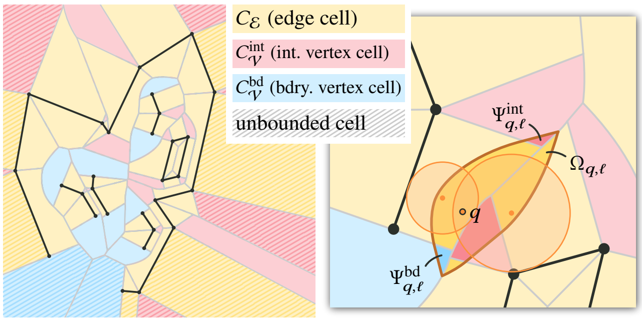


Figure 3: Left: *We partition $\mathbb{R}^2$ into the sets of points closest to edges (yellow), interior vertices (red), and boundary vertices (blue) of the input geometry. Vertices and edges on the convex hull of the geometry have unbounded cells (striped).* Right: *Given a query point $q \in \mathbb{R}^2$, we visualize the region of centers of circles that contain q. The saturated yellow, red, and blue parts of this region form the integration domains in Eq. 2, assuming all edges shown have label ℓ.*

adaptation to three dimensions is conceptually straight-forward as we show in Section 6.4.

3.1. Circles of Confidence

Let $\mathcal{V} = \{v_1, \dots, v_n\}$ be a collection of points in $\mathbb{R}^2$, and $\mathcal{E} = \{e_1, \dots, e_m\}$ with $e_i = (v_i^1, v_i^2) \in \mathcal{V} \times \mathcal{V}$ a collection of oriented edges. These edges may form polygons and open polylines, which are interpreted as partial boundaries of enclosed areas that we would like to complete. The direction of edges is important in that we assume a counterclockwise (CCW) orientation, or that an enclosed area always lies to the left of an oriented edge.

In the presence of uncertainty due to partially missing boundaries, each edge gives us some level of confidence that there is an enclosed area to the left, and empty space to the right. We propose a method to quantify this confidence by a metric that accounts for distance and visibility.

The starting point is the unsigned distance transform of the input geometry, which assigns a distance $r(p) \geq 0$ to every $p \in \mathbb{R}^2$, as shown in Fig. 2 *(left)*. A natural interpretation is that $r(p)$ gives the radius of the largest empty ball centered at p,

$$B_p = \{x \in \mathbb{R}^2 : \|x - p\| < r(p)\}.$$

Each ball will either be tangent to the interior of an edge $e \in \mathcal{E}$, or contain a vertex $v \in \mathcal{V}$ in its boundary. In the latter case, v may either be an *interior* vertex, which connects at least two edges, or a *boundary* vertex, which is an endpoint of an open polyline. This provides a partition $\mathbb{R}^2 = C_{\mathcal{E}} \cup C^{\text{int}}_{\mathcal{V}} \cup C^{\text{bd}}_{\mathcal{V}}$ into the set $C_{\mathcal{E}}$ of all $p \in \mathbb{R}^2$ such that B_p touches an edge, and the sets $C^{\text{int}}_{\mathcal{V}}$ and $C^{\text{bd}}_{\mathcal{V}}$ of all $p \in \mathbb{R}^2$ such that B_p touches an interior vertex or boundary vertex, respectively. These regions are visualized in Fig. 3 *(left)*.

For now, let us consider only points p in $C_{\mathcal{E}}$: If B_p touches an edge from the left, it represents evidence that points inside this ball belong to the interior of an enclosed region; the opposite is true if it touches an edge from the right. To model this, we introduce a set of two labels $\mathcal{L} = \{\text{inside}, \text{outside}\}$, and denote by $\ell(p) \in \mathcal{L}$ the label of ball B_p, depending on which side of an edge it touches, as shown in Fig. 2 *(right)*. If p is far way from its closest edge, we are less confident in the assignment of label $\ell(p)$. Thus we scale the confidence score of each ball with the reciprocal of its area, which is proportional to $1/r(p)^2$.

The final step is to evaluate the total confidence that a point $q \in \mathbb{R}^2$ has label $\ell \in \mathcal{L}$ by integrating the confidence contributed by each ball B_p that contains q and has label ℓ, or

$$c_\ell(q) = \int_{\Omega_{q,\ell}} \frac{1}{r(p)^2}\, dA(p), \tag{1}$$

with integration domain $\Omega_{q,\ell} = \{p \in C_{\mathcal{E}} : q \in B_p \text{ and } \ell(p) = \ell\}$. If $c_{\text{inside}}(q) > c_{\text{outside}}(q)$, then q is inside of the reconstructed enclosed area. The boundary of this area is composed exactly of the points q such that $c_{\text{inside}}(q) = c_{\text{outside}}(q)$. The integration domain $\Omega_{q,\ell}$ is visualized in Fig. 3 *(right)*.

3.2. Interpretation

To understand the behavior of the confidence metric defined by Eq. 1, we compute it in closed form for input geometry comprised of a single edge $e = (v_1, v_2)$ with $v_1 = (0,0)^t$ and $v_2 = (L,0)^t$. Given a point $p = (p_x, p_y)^t \in \mathbb{R}^2$, the largest empty ball centered at p will touch the interior of edge e if $0 < p_x < L$, so

$$C_{\mathcal{E}} = \{(p_x, p_y)^t \in \mathbb{R}^2 : 0 < p_x < L\}.$$

To compute the total confidence that an evaluation point $q = (q_x, q_y)^t \in \mathbb{R}^2$ is "inside", we have to determine the integration domain $\Omega_{q,\text{inside}}$. We see that $q \in B_p$ if and only if $\|q - p\| < r(p) = p_y$, which is equivalent to $p_y > \frac{q_y}{2} + \frac{1}{2q_y}(q_x - p_x)^2$; geometrically, this tells us that p lies above a parabola with focus q. This region is visualized in Fig. 4 *(right)*. We conclude

$$c_{\text{inside}}(q) = \int_0^L \int_{f(x)}^{\infty} \frac{1}{y^2}\, dy\, dx, \text{ with } f(x) = \frac{q_y}{2} + \frac{1}{2q_y}(q_x - x)^2.$$

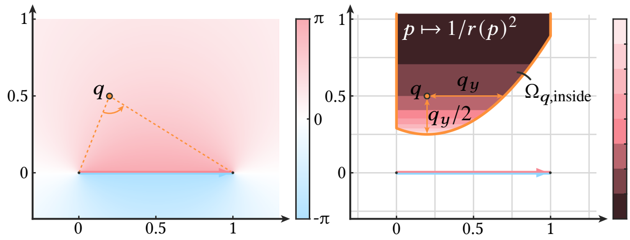


Figure 4: Left: *The generalized winding number (GWN) of an edge, evaluated at $q = (q_x, q_y)$, is the oriented angle subtended by the edge as seen from q.* Right: *The infinite region above the parabola contains the centers of all greatest empty balls that contain q and are tangent to the interior of the edge. Integrating the reciprocal square distance to the line segment over this region recovers the GWN evaluated at q.*

Evaluating this integral yields

$$c_{\text{inside}}(q) = 2 \cdot \left(\arctan \frac{q_x}{q_y} + \arctan \frac{L - q_x}{q_y} \right),$$

which is twice the *angle subtended by the edge as seen from q*.

This shows a surprising connection between our circle of confidence approach and the generalized winding number, visualized in Fig. 4 *(left)*: They agree exactly, given the design choices made thus far. This motivates us to investigate the possibilities of our method further by considering a wider range of parameter choices. We do this in the next section, by including circles that touch a vertex of the input geometry in the confidence evaluation, and by exploring a continuous family of confidence score functions.

For input geometry consisting of more edges, the result of our method will no longer match that of GWN globally, because we only consider greatest empty circles, and not all circles touching every edge. This makes our method robust to the failure cases that plague the GWN method—see Section 6 for examples—while retaining its excellent local qualities.

3.3. Full Model

In this section, we expand the simplified confidence evaluation function from Eq. 1 in three ways. First, it is natural to include the contribution from circles touching vertices of the input geometry, which entails adding integrals over (subsets of) $C_{\mathcal{V}}^{\text{int}}$ and $C_{\mathcal{V}}^{\text{bd}}$, the centers of circles touching interior and boundary vertices, respectively. These domains are visualized in red and blue in Fig. 3 *(left)*. Second, using a confidence score of $1/r^2$ is not the only possible metric, as any scoring function that decreases monotonically in r is potentially reasonable. We will explore the family of $1/r^\alpha$ confidence scores because it gives rise to a highly interpretable range of behaviors that interpolate between GWN-like behavior for $\alpha = 2$ and the pseudonormal method for $\alpha \to \infty$. Third, open polylines indicate missing boundary segments, so we introduce a term that decreases the confidence near boundary vertices, which are the endpoints of open polylines. We will first define the full model and then discuss its mathematical properties.

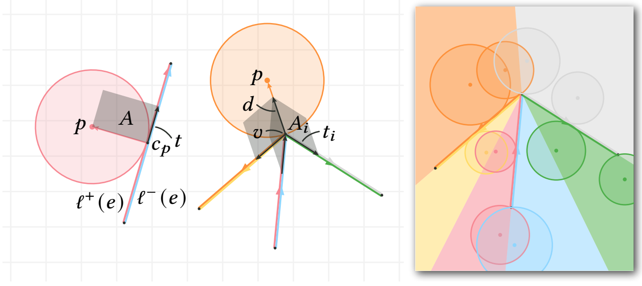


Figure 5: Left: *The sign of the oriented area spanned by the edge tangent t and the vector $p - c_p$ determines the label of the circle.* Center: *The label-giving edge for the circle is determined by the largest unsigned area $|A_i|$.* Right: *As a result, the interfaces between different labels are along edges and along angle bisectors of neighboring edges.*

Circle Labeling An inside–outside classification is often all that is needed for manifold geometry, but we allow a more general set of labels $\mathcal{L} = \{\ell_1, \ldots, \ell_k\}$ to support geometry with T-junctions as well. In this case, every enclosed region, as well as the "outside" region, may have its own label. This is furnished by equipping every oriented input edge e with two labels $\ell^+(e) \in \mathcal{L}$ and $\ell^-(e) \in \mathcal{L}$, which determine the area label to the left and right of the edge, respectively. These labels are transferred to circles of confidence as follows (see Fig. 5):

- Assume the greatest empty ball centered at p touches the interior of an input edge e at a point c_p, and let t be the (oriented) unit tangent of e. Then we set $\ell(p) := \ell^+(e)$ if $A := \det(t, p - c_p) > 0$, and $\ell(p) := \ell^-(e)$ otherwise.
- Assume the greatest empty ball centered at p touches an input vertex v. Let $t_1, \ldots, t_m$ denote the (oriented) unit tangents of all edges incident to v, and let $d := \frac{p-v}{\|p-v\|}$. Then compute $A_i := \det(t_i, d)$ for all i, and find $I := \arg\max_i |A_i|$. Set $\ell(p) := \ell^+(e_I)$ if $A_I > 0$, and $\ell(p) := \ell^-(e_I)$ otherwise.

This label assignment locally divides the region around every vertex into wedges, as shown in Fig. 5 *(right)*, with interfaces running along edges and angle bisectors.

Confidence Score Every circle will "splat" its label to all points in its interior, with a weight that we call its confidence score. For circles with centers in $C_{\mathcal{E}}$ or $C_{\mathcal{V}}^{\text{int}}$ (yellow and red regions in Fig. 3), we model this confidence score as $1/r^\alpha$, with $\alpha \geq 0$ a parameter exposed to the user.

This leaves circles with centers in $C_{\mathcal{V}}^{\text{bd}}$ (blue regions in Fig. 3), which touch an endpoint of an open polyline. The boundary of enclosed regions cannot be open, so geometry is necessarily missing near these endpoints, which reduces our confidence in the label of these circles. To model this, we expose a second user-controlled parameter $\beta \geq 0$ and use a confidence score $(1/r^\alpha) \cdot |\cos\theta|^\beta$, where θ is the angle between the edge normal and the vector from the endpoint of the edge to the circle center. The value of $\cos\theta$ equals A_i from the label computation above, so this induces little overhead. Fig. 6 shows the behavior of the confidence score near endpoints of open polylines for different values of β.

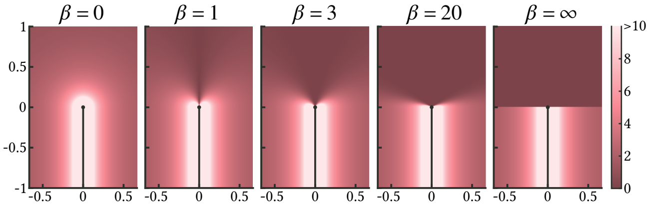


Figure 6: *The user-controlled parameter β determines the decrease in confidence near endpoints of open polylines. The plot shows the value of the confidence score $(1/r^\alpha)\cdot|\cos\theta|^\beta$ around these endpoints for $\alpha = 1$ and different values of β.*

Confidence Integration The final step is to evaluate the total confidence that a point $q \in \mathbb{R}^2$ has label ℓ by integrating the confidence score of each ball B_p that contains q and such that $\ell(p) = \ell$. In analogy to Eq. 1, this is achieved by

$$\begin{aligned} c_\ell(q) &= \int_{\Omega_{q,\ell}\cup\Psi^{\text{int}}_{q,\ell}} \frac{1}{r^\alpha}\,dA + \int_{\Psi^{\text{bd}}_{q,\ell}} \frac{1}{r^\alpha}\cdot|\cos\theta|^\beta\,dA, \quad \text{with} \\ \Omega_{q,\ell} &= \{p \in C_{\mathcal{E}} : q \in B_p \text{ and } \ell(p) = \ell\}, \qquad (2) \\ \Psi^{\text{int}}_{q,\ell} &= \{p \in C^{\text{int}}_{\mathcal{V}} : q \in B_p \text{ and } \ell(p) = \ell\}, \\ \Psi^{\text{bd}}_{q,\ell} &= \{p \in C^{\text{bd}}_{\mathcal{V}} : q \in B_p \text{ and } \ell(p) = \ell\}. \end{aligned}$$

Fig. 3 *(right)* illustrates the three domains, assuming that all contributions from the edges shown have label ℓ. Fig. 7 visualizes the total confidence c_ℓ for the different labels of the examples presented in Section 6. In the *tangram figure (top-right)*, it is particularly evident that the total confidence function is supported locally in the region close to the input edges carrying the corresponding label. This allows our method to strike a balance between depending only on a single point, like the pseudonormal method, and being completely global, like GWN and PSR. The final label of a point q is determined by the highest confidence score,

$$L(q) = \arg\max_{\ell\in\mathcal{L}} c_\ell(q), \qquad (3)$$

which provides the partition of $\mathbb{R}^2$ into regions enclosed by the input geometry. The reconstructed missing boundaries are given by the set of points where the maximum in Eq. 3 is simultaneously achieved by two labels $\ell \neq \tilde{\ell}$.

3.4. Analysis

The integrand in Eq. 2 has singularities at $r = 0$, and the integration domain will be unbounded for certain query points q. We must thus analyze carefully under which assumptions the total confidence c_ℓ takes finite values, and we also investigate limit behaviors for the parameters α and β.

3.4.1. Singularity Points

In $\mathbb{R}^2$, integrals of $1/r^\alpha$ will diverge for $\alpha \geq 2$ if the integration domain contains an open neighborhood of a singularity point, i.e., of a point p with $r(p) = 0$. We can show that this situation does not occur in Eq. 2:

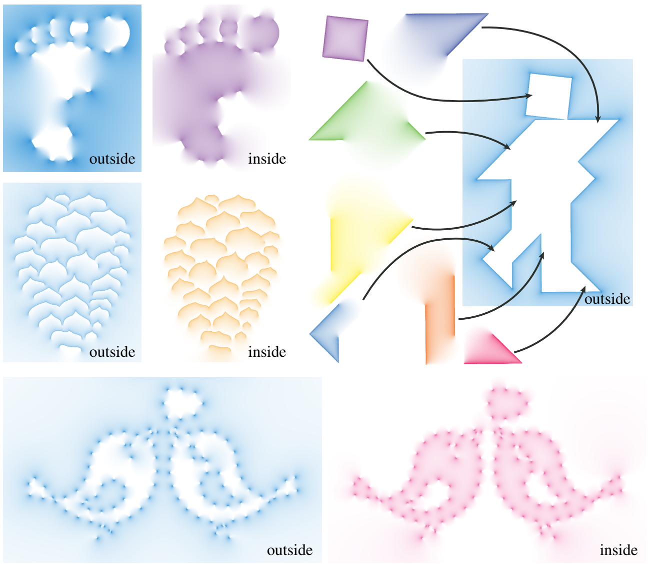


Figure 7: *Visualization of the total confidence c_ℓ per label ℓ, corresponding to the examples in Figs. 9–12. The "outside" confidence is visualized in light-blue.* Top-right: *The total confidence of the seven inside labels are shown in an exploded view.*

Proposition 1 For any $q \in \mathbb{R}^2$, the integration domains in Eq. 2 are separated from the input geometry by at least half the distance between q and the input geometry.

Proof Denote by $A \subset \mathbb{R}^2$ the union of all edges in the input geometry. It suffices to show that for any p contained in one of the integration domains in Eq. 2, we have $r(p) \geq \frac{1}{2}r(q)$, where $r(z) := \inf_{a\in A}\|z-a\|$. For any such p, it holds by assumption that $q \in B_p$, which implies $\|q-p\| \leq r(p)$. Using this, we compute

$$\begin{aligned} r(q) &= \inf_{a\in A}\|q-a\| = \inf_{a\in A}\|(q-p)+(p-a)\| \\ &\leq \|q-p\| + \inf_{a\in A}\|p-a\| \leq r(p) + r(p) = 2r(p), \end{aligned}$$

which implies $r(p) \geq \frac{1}{2}r(q)$. ☐

This shows that whenever q itself does not lie on the input geometry, the integration domains for the total confidence evaluation will not intersect the input geometry either, and thus contain no singularity points p with $r(p) = 0$.

3.4.2. Unbounded Domains

The integration domains associated with vertices and edges on the convex hull of the input geometry will be unbounded, as illustrated by the striped regions in Fig. 3. A domain associated with an edge is a *slab* (an "infinite rectangle"), and a domain associated with a vertex is a *wedge* (it contains an "infinite cone"). In some sense, wedges fill more space then slabs. Intuitively, an infinite cone with opening angle φ fills $\frac{\varphi}{2\pi}$ of $\mathbb{R}^2$; but a slab looks more and more like a ray when we zoom out, and fills almost no part of $\mathbb{R}^2$ at all. Integration also behaves differently for these types of domains: The integral of $1/r^\alpha$ over a slab will take finite values for any $\alpha > 1$, but over a wedge, only for $\alpha > 2$.

In general, it need not bother us if $c_\ell(q) = \infty$ for a single $\ell \in \mathcal{L}$, because then the final label of q is still well-defined by Eq. 3, and we have $L(q) = \ell$. But if $c_\ell(q) = \infty = c_{\tilde{\ell}}(q)$ with $\ell \neq \tilde{\ell}$ for all q in an open subset of $\mathbb{R}^2$, then there will be a region with multiple valid labels, which we would like to avoid.

For q inside the convex hull of the input geometry, the integration domains are always bounded, so $c_\ell(q)$ is finite regardless of the value of α, as shown by

Proposition 2 Let q be in the interior of the convex hull of the input geometry. Then the integration domains in Eq. 2 are bounded.

Proof The convex hull of the input geometry is a convex polygon whose vertices $v_1, \dots, v_n$ are also vertices of the input geometry. Assume p is in one of the domains defined in Eq. 2, so $\|q - p\| \leq r(p)$. It also holds that $r(p) \leq \|v_i - p\|$ for all $i = 1, \dots, n$ because $r(p)$ is the radius of the greatest empty ball centered at p, and this ball may not contain any input vertices in its interior. This implies $\|q - p\| \leq \|v_i - p\|$ for all $i = 1, \dots, n$.
Consider the Voronoi diagram of the sites $\{q, v_1, \dots, v_n\}$: By the preceding inequality, p is in the cell of site q. A cell in a Voronoi diagram is unbounded if and only if the site of this cell is on the boundary of the convex hull spanned by all sites. But q is in the interior of the convex hull of $\{q, v_1, \dots, v_n\}$ by assumption, so its cell is bounded. Because p was chosen arbitrarily from one of the integration domains, these domains are contained in the cell of q, so they are also bounded. ☐

For applications in which the labeling *outside* of the convex hull is important, there are two options: First, one can restrict the valid parameter range to $\alpha > 2$, so integrals take finite values even on all unbounded domains. Second, one can provide a labeling so that all unbounded integration domains carry the same label; then at most one label yields a total confidence of ∞. In many practical cases, this is automatic, because there is often a single "outside" label, which all edges on the convex hull carry on one side. The exact sufficient condition to guarantee label uniqueness outside the convex hull is provided (without proof) by

Proposition 3 Let v be a vertex on the convex hull of the input geometry, and E_1 and E_2 the edges of the convex hull boundary incident to v. Let e_1 and e_2 be the input edges incident to v such that the angle between E_i and e_i is minimal, for $i = 1, 2$. Denote by $\ell(e_i)$ the label of e_i which is *oriented towards*[†] E_i. Finally, denote by $\mathcal{L}_v \subset \{\ell(e_1), \ell(e_2)\}$ the set that contains $\ell(e_i)$ if and only if the angle between E_i and e_i is at most $\pi/2$. Then the labeling induced by Eqs. 2 and 3 is unique if the set $\bigcup_v \mathcal{L}_v$ has only one element.

This condition is easy to check computationally, so a warning can be issued to the user if it is not satisfied. Should this happen, we can still run our hole closing algorithm, but certain regions outside the convex hull will receive multiple labels.

3.4.3. Limit Behaviors

The interesting special choices for $\alpha \geq 0$ are $\alpha = 2$ and $\alpha \to \infty$. The significance of $\alpha = 2$ through its connection to the GWN has already been discussed in Section 3.2. In the limit of $\alpha \to \infty$ and assuming $\beta = 0$, we exactly recover the result of the *pseudonormal method*, as shown by

Proposition 4 Given $q \in \mathbb{R}^2$, the final label $L(q)$ obtained by Eqs. 2 and 3 for $\alpha \to \infty$ and $\beta = 0$ is equal to $\ell(q)$, the label of the greatest empty circle centered at q. This is the same as the label of the input geometry at the point c_q closest to q.

Proof Denote by $\Phi_{q,\ell}$ the closure of the union of all three integration domains in Eq. 2. Then we have $c_\ell(q;\alpha) = \int_{\Phi_{q,\ell}} \left(\frac{1}{r}\right)^\alpha dA$. By Proposition 1, $\Phi_{q,\ell}$ contains no singularity points, so this integral exists at least for all $\alpha > 2$. We apply the limit relation for L^p-norms, which says that $\left(\int |f|^p\right)^{1/p} \to \operatorname{ess\,sup} |f|$ as $p \to \infty$. This gives

$$\lim_{\alpha\to\infty} (c_\ell(q;\alpha))^{1/\alpha} = \lim_{\alpha\to\infty} \left(\int_{\Phi_{q,\ell}} \left(\frac{1}{r}\right)^\alpha \right)^{1/\alpha} = \max_{\Phi_{q,\ell}} \frac{1}{r} = \frac{1}{\min_{\Phi_{q,\ell}} r}.$$

Here, we replaced "ess sup" by "max" because the function $1/r$ necessarily attains its maximum on (the boundary of) $\Phi_{q,\ell}$. This allows us to evaluate Eq. 3 by

$$\lim_{\alpha\to\infty} L(q;\alpha) = \lim_{\alpha\to\infty} \arg\max_{\ell\in\mathcal{L}} c_\ell(q;\alpha) = \lim_{\alpha\to\infty} \arg\max_{\ell\in\mathcal{L}} (c_\ell(q;\alpha))^{1/\alpha}$$

$$= \arg\max_{\ell\in\mathcal{L}} \lim_{\alpha\to\infty} (c_\ell(q;\alpha))^{1/\alpha} = \arg\max_{\ell\in\mathcal{L}} \frac{1}{\min_{\Phi_{q,\ell}} r} = \arg\min_{\ell\in\mathcal{L}} \min_{\Phi_{q,\ell}} r.$$

The second equality holds because the map $(\cdot)^{1/\alpha}$ is strictly monotonically increasing on $\mathbb{R}_{\geq 0}$ for all $\alpha > 0$, so applying it to the argument does not change the result of arg max. For the third equality, we tacitly assume that the final label is unique at q: Specifically, for $f_\ell(\alpha) := (c_\ell(q;\alpha))^{1/\alpha}$, we assume that the maximum over all $\{\lim_{\alpha\to\infty} f_\ell(\alpha)\}_{\ell\in\mathcal{L}}$ is attained by exactly one $\ell \in \mathcal{L}$. Then continuity of all f_ℓ implies interchangeability of the limit and arg max.
The last step is to find the label $\ell \in \mathcal{L}$ for which $\min_{\Phi_{q,\ell}} r$ is minimal. Proposition 1 gives a lower bound on the distance from $p \in \Phi_{q,\ell}$ to the input geometry, which is shown to be $\frac{1}{2} r(q)$. This bound is attained by the midpoint $m = \frac{1}{2}(q + c_q)$, which is an element of $\Phi_{q,\ell(m)}$, and thus $L(q) = \arg\min_{\ell\in\mathcal{L}} \min_{\Phi_{q,\ell}} r = \ell(m)$. Finally, m and q have the same closest point, c_q, on the input geometry, so $\ell(m) = \ell(q)$, and thus $L(q) = \ell(q)$. ☐

This shows that the result of our method for $\alpha \to \infty$ looks like the partition in Fig. 3 *(left)*, where each of the cells receives the label of the edge or vertex that it belongs to. The only difference is that cells belonging to vertices may be subdivided into two wedges according to the labeling rules visualized in Fig. 5 *(right)*.

The limit behavior of the parameter $\beta \geq 0$ is straightforward. The greater the value of β, the more the influence of the endpoints of open polylines is suppressed, as shown in Fig. 6. For $\beta = 0$, we will have $|\cos\theta|^\beta = 1$ almost everywhere, so this has the same effect as dropping the cosine term from Eq. 2. For $\beta \to \infty$, we have $|\cos\theta|^\beta \to 0$ almost everywhere, so this is the same as removing the integral over $\Psi^{\mathrm{bd}}_{q,\ell}$ from Eq. 2 altogether.

3.4.4. Interface Geometry

The interface curve between two reconstructed regions consists of points $q \in \mathbb{R}^2$ where the maximum of the total confidences $c_\ell(q)$ over $\ell \in \mathcal{L}$ is attained by two labels simultaneously. Generally, the

---

[†] In Fig. 5 *(right)*, the red label of the red-blue edge is "oriented towards" the yellow-orange edge, and the blue label is "oriented towards" the gray-green edge.

position of these interface curves depends on the nearby parts of the input geometry via Eq. 2, but its local behavior close to the geometry is amenable to analysis. In this section, we sacrifice rigor in favor of brevity and present heuristic arguments that can be made precise.

First, we note that the interface between labeled regions will only intersect the input geometry in points where the edge label changes.[‡] This happens at interior vertices having two consecutive incident edges with different edge labels pointing towards each other, e.g., the blue and green labels in Fig. 5 *(right)*. It also happens at boundary vertices incident to an edge with two different labels on each side.

The behavior of the interface near such points depends on the angle between the label-giving edges and on the value of $\alpha \geq 0$. In convex corners, with an angle less than π, such as the yellow-red or blue-green corners in Fig. 5 *(right)*, the tangent of the interface is guaranteed to bisect the angle for any $\alpha \geq 0$. In fact, the interface curve has a straight segment of finite length as it approaches the input geometry. In re-entrant corners with an angle greater than π, such as the orange-gray corner in Fig. 5 *(right)*, the tangent of the interface is only guaranteed to bisect the angle if $\alpha \geq 2$. The same holds at boundary vertices, which function as re-entrant corners with an angle of 2π in this context: The tangent of the interface only agrees with the direction of the open edge for $\alpha \geq 2$ in general, and may form a kink otherwise. Fig. 8 illustrates the arguments that lead to these conclusions, and Fig. 10 shows a concrete example.

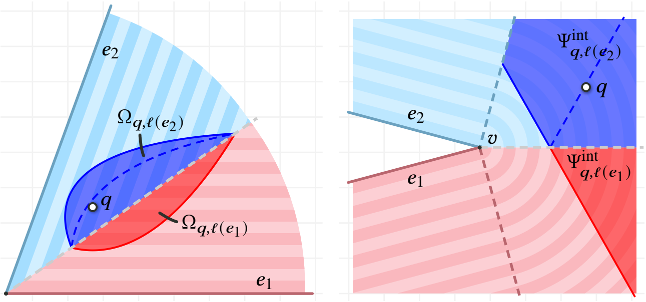


Figure 8: Left: *For q close enough to a convex corner, all greatest empty circles containing q touch either e_1 or e_2, and the only non-empty integration domains are as shown. For q above the angle bisector, the domain $\Omega_{q,\ell(e_2)}$ is a strict superset of (the mirror image of) $\Omega_{q,\ell(e_1)}$, leading to a greater confidence score for $\ell(e_2)$ due to symmetry of the distance field.* Right: *For q close to a re-entrant corner v, the integration domains induced by v look as shown. The difference in confidence between $\ell(e_2)$ and $\ell(e_1)$ due to v is the confidence contributed by the part of the blue region to the left of the dashed blue line, due to symmetry. For $\alpha \geq 2$, this confidence difference tends to $+\infty$ as q approaches v. Thus the confidence difference between these domains dominates the (finite) contribution of any other input geometry at short distances.*

4. Discretization

The total confidence $c_\ell(q)$ as defined by Eq. 2 cannot generally be evaluated in closed form.[§] To approximate the solution, we use numerical quadrature and discretize the integral as a finite sum over point samples on a regular grid. For simplicity of exposition, we analyze the case $\beta = 0$, so the right-hand side of Eq. 2 can be written as a single integral on $\Phi_{q,\ell}$, the union of all three integration domains. However, the argument applies to the original formulation with no change. We first define the discretization and then perform a convergence analysis.

4.1. Definition

Our scheme to approximate the total confidence has two steps. First, we compute $r(p)$ and $\ell(p)$ for all $p \in \mathbb{R}^2$ in a dense regular grid that covers the input geometry and a border region around it. Second, we evaluate an approximation of the integral in Eq. 2 as the finite sum over the contribution of all grid points p that fall inside the integration domain.

Step 1 Assume the input geometry is centered near the origin, and let $R > 0$ be the radius of the smallest bounding ball of all vertices and edges. Denote by $h > 0$ the grid cell size, and by $H > 0$ half the side length of the total area covered by the grid; in practice, we pick $H \approx \frac{3}{2}R$. Then define the set of all grid points by

$$\{p_1, \ldots, p_N\} := \{(ih, jh) \in \mathbb{R}^2 : i, j \in \mathbb{Z} \text{ and } -H \leq ih, jh \leq H\}.$$

This yields approximately $N \approx \frac{4}{h^2}H^2$ sample points. For each p_i, we compute the distance $r(p_i)$ to the closest point on the input geometry—a fast procedure for doing this is implemented in most geometry processing libraries. Finally, we evaluate $\ell(p_i)$ according to the rules in Section 3.3.

Step 2 Using the grid samples, we discretize $\int_{\Phi_{q,\ell}} 1/r^\alpha \, dA$ with the midpoint rule, which yields

$$c_\ell^{h,H}(q) = h^2 \sum_{i \in I_{q,\ell}} \frac{1}{r(p_i)^\alpha}, \quad \text{with} \tag{4}$$

$$I_{q,\ell} = \{i \in \{1, \ldots, N\} : \|q - p_i\| \leq r(p_i) \text{ and } \ell(p_i) = \ell\}.$$

Computationally, we evaluate this function by looping over all $p_1, \ldots, p_N$ and "splatting" their contributions $1/r(p_i)^\alpha$ to all query points within a distance of $r(p_i)$. Section 5 describes an efficient implementation of this procedure using a quadtree.

[‡] While this may seem obvious at first, a proof requires a careful argument about the rate at which the areas of the integration domains in Eq. 2 decay as q approaches an edge e of the input geometry. The area of the subdomain induced by e, similar to the epigraph of the parabola illustrated in Fig. 4 *(right)*, goes to zero with the square root of the distance between q and e, while the area of all other domains can be shown to decay strictly faster. Thus the area of the subdomain induced by e dominates at short distances, and boundedness of the integrand away from the input geometry is enough to conclude that $L(q)$ is constant in some (one-sided) neighborhood of e.

[§] The integration domains are bounded by parabolic arcs and line segments, and the integrand takes the form $|\cos\theta|^\beta / \|x - x_0\|^\alpha$. Even in the simplest case where $\beta = 0$ and the integration domain is a rectangle, special functions such as the hypergeometric function are needed to evaluate Eq. 2, except for specific values of α such as integers.

4.2. Analysis

To show that Eq. 4 is an acceptable approximation, we would like to verify that the discretized total confidence $c_\ell^{h,H}(q)$ converges to the true solution $c_\ell(q)$ as $h \to 0$ and $H \to \infty$. For $q \in \mathbb{R}^2$ such that the integration domains in Eq. 2 are bounded—see Section 3.4 for a discussion on unbounded domains—there are two sources of error to consider:

1. Away from the boundary of $\Phi_{q,\ell}$, the integral of $1/r(p)^\alpha$ over a square of side length h centered at p_i is approximated by $h^2/r(p_i)^\alpha$. This incurs the *approximation error* typical for quadrature methods.
2. For samples p_i near the boundary of $\Phi_{q,\ell}$, the square of side length h centered at p_i may not be fully contained in $\Phi_{q,\ell}$. Thus we effectively approximate the integration domain as the union of squares with centers contained in $\Phi_{q,\ell}$, incurring a *domain error*.

The approximation error approaches zero in the limit $h \to 0$ based on the observation that $r(p)$ is bounded from below by $\frac{1}{2}r(q)$ according to Proposition 1. Thus the integrand has bounded first derivative in the integration domain, and the standard error estimates for numerical quadrature apply.

To conclude that the domain error also approaches zero as $h \to 0$, we must show that the total area of the grid cells intersecting the boundary of the domain approaches zero as $h \to 0$. This is a consequence of the fact that the boundary curve (interpreted as a subset of $\mathbb{R}^2$) has Lebesgue measure zero because it is composed of a finite number of linear and parabolic segments.

For unbounded domains, we have another source of error:

3. The part of the integration domain outside of the grid bounds $[-H,H] \times [-H,H]$ is *truncated* by the discretization.

In each edge cell (see Fig. 3), $r(p)$ a is linear function, so we can bound the integral over the ignored part of the domain by

$$L\int_H^\infty 1/r^\alpha dr = L(\alpha-1)H^{1-\alpha},$$

for $\alpha > 1$, where $L > 0$ is the length of the edge. Similarly, in a vertex cell, $r(p)$ is a radial function, and the integral over the ignored part of the domain can be bounded by

$$\varphi\int_H^\infty (1/r^\alpha)\, r\, dr = \varphi(\alpha-2)H^{2-\alpha},$$

for $\alpha > 2$, where φ is the opening angle of an infinite cone containing the cell. We conclude that for $\alpha > 2$, which is the range in which all integrals take finite values, the truncation error approaches zero as $H \to \infty$. The convergence speed will be higher for greater values of α, because then more of the mass of $1/r^\alpha$ is concentrated close to the input geometry.

5. Implementation

A naive implementation of the quadrature scheme defined in Section 4.1 on a grid with $N = n^2$ samples has a worst-case complexity of $O(n^4)$. This is because we evaluate the greatest empty circles centered at n^2 grid points, and then splat the confidence score to all grid points in their interior; a circle that covers much of the domain will contain $O(n^2)$ such points.

We can reduce this complexity to $O(n^3)$ without further approximation by using a quadtree data structure. Level zero of the quadtree stores a value per grid point (i,j) with $i,j \in \{0,\dots,n-1\}$. Level k contains aggregated nodes indexed by (I,J) with $I,J \in \{0,\dots,\lceil n^{-k}\rceil - 1\}$, and the value stored at node (I,J) will eventually be added to the values at all grid points with indices

$$(i,j) \in \{2^k I,\dots,2^k(I+1)-1\} \times \{2^k J,\dots,2^k(J+1)-1\}.$$

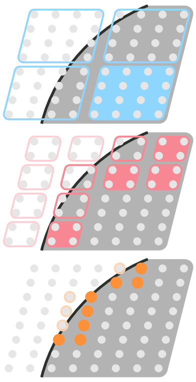
Thus a node at level k covers $2^k \times 2^k$ grid points, whose bounding box (bbox) is a square with side length $2^k - 1$. The implementation proceeds in two steps:

Step 1 At each grid point, we splat the confidence score to the quadtree nodes contained in the greatest empty circle centered at this point, using aggregated nodes whenever possible. To do this for a circle with radius R, we start at the highest level K such that a node could be fully contained in the circle, i.e., K is the greatest integer such that a square of side length $2^K - 1$ fits in a circle of radius R. Then we push each node in level K whose bbox intersects the bbox of the circle onto a stack.

From then on, we pop nodes from the stack until empty:

(a) if the bbox of the node is fully contained in the circle, then the confidence score is added to the value stored at this node;
(b) if the bbox of the node intersects the circle partially, then its four child nodes are added to the stack;
(c) if the bbox of the node does not intersect the circle, then no action is performed.

Nodes at level zero contain individual grid points, so case (b) is excluded. The inset below illustrates the splatting procedure onto the levels of a quadtree from top to bottom: Colored nodes are traversed during the procedure, and cases (a), (b), and (c) correspond to nodes with fill-in color, border color, and desatured border color, respectively. The number of quadtree cells visited for one circle has complexity $O(R)$, which is bounded from above by $O(n)$. After processing all n^2 circles, this leads to a worst-case complexity of $O(n^3)$.



Step 2 Finally, we aggregate the values of all levels in the quadtree by summation. To do this, we iteratively add the value stored at a node to the values of all four children, starting from the highest level of the quadtree. In the end, the values in level zero contain exactly the discretized total confidence samples defined in Eq. 4.

6. Results

We hope to convince the reader that our method for geometry completion compares favorably with existing methods for inside–outside classification, while also extending to the multi-label (or non-manifold) setting. Our examples show the effect of varying the

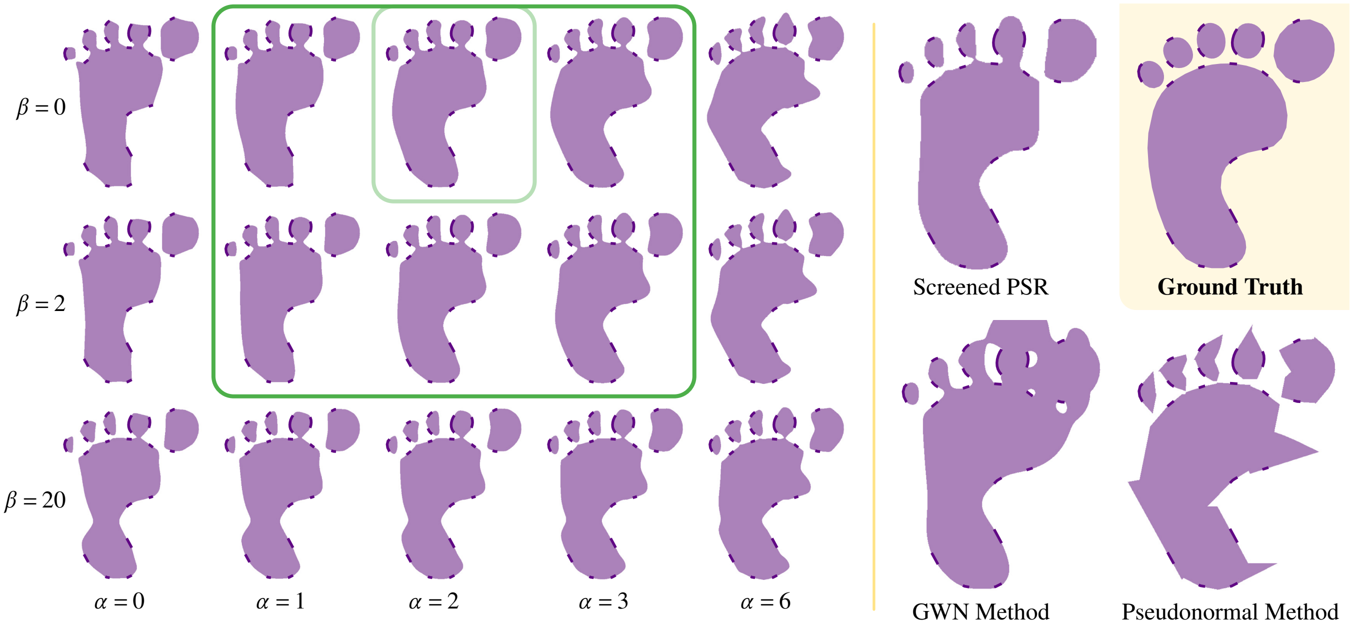


Figure 9: *Reconstruction of a* Footprint *clipart from a very sparse set of input boundary segments (dark purple) by different methods.* Left: *Our method achieves good results for the parameter range in the green box; we find $\alpha=2$ and $\beta=0$ to give the best reconstruction for this example.* Right: *Results obtained by three other methods, and the ground truth geometry.*

parameters $\alpha,\beta \geq 0$ and demonstrate the behavior of the circle-of-confidence method for inputs from which boundaries are missing to varying degrees. For reconstruction tasks with only two labels, we compare our results to those obtained by the GWN [JKS13] and screened PSR [KH13] methods, as well as the PN (pseudonormal) method [BA05], which is equivalent to setting $\alpha=\infty$ in our method. For a fair comparison with screened PSR, we run the method with a dense point sampling of the input geometry with exact oriented normals, and a narrow smoothing kernel on a high-resolution grid, unless stated otherwise. We go into substantial detail only for the first two examples; the remaining results serve to help build intuition on how our method behaves in different situations, and to illustrate its strengths and limitations.

Footprint In Fig. 9, we show reconstruction results of a footprint clipart from which a very sparse set of boundary segments has been extracted to serve as input geometry. Our method *(left)* tends to work best in the parameter range $\alpha=1\ldots3$ and $\beta=0\ldots2$ for most examples, including this one, where $(\alpha,\beta)=(2,0)$ gives the most convincing result. For $\alpha=0$ and $\alpha=1$, we observe that the reconstructed interface has sharp corners at the endpoints of some input edges, as discussed in Section 3.4.4; for $\alpha\geq 2$, the interface is C^1 everywhere. For $\alpha=6$, the result is starting to form bulges which will eventually grow into the sharp corners characteristic of the pseudonormal method if α is increased further.

Fig. 9 *(right)* shows the original clipart and the result of three other reconstruction methods. The pseudonormal method has no parameters, and the sharp corners in the result cannot be avoided, so it is less suitable for organic shapes. The GWN method is generally sensitive to missing geometry, and the default choice of thresholding the generalized winding-number function at the 0-level set produces catastrophic results (not shown) for this example. For the GWN reconstruction plotted in the figure, we hand-picked the threshold that gave the least objectionable result during visual inspection. Screened PSR gives a good result that is qualitatively similar to ours, while perhaps straightening the sides of the foot and extending the second toe a bit too much. This method is sensitive to boundary effects, i.e., closeness of the geometry to the domain boundary on which the Poisson equation is solved, so it was necessary to enlarge the domain beyond the area shown in the figure to achieve high quality.

Computationally, the GWN and pseudonormal methods are the fastest by far, and produce a result in less than one second. For a grid resolution of 850×1000, our method takes 9 seconds to produce a reconstruction. In an equal-time comparison, the PSR method can be run on a 550×650 grid. An advantage of our method is that running it for multiple parameter sets, like the 15 combinations shown in Fig. 9, is only marginally slower than computing one, even in our single-threaded implementation. This is because most time is spent on traversing the quadtree, see Section 5, and very little time on summing the contributions of each circle. Computing all 15 images takes only 30% more time than computing a single one, which makes it attractive to explore the solution space.

Tangram Figure Fig. 10 *(top)* shows a multi-label reconstruction example, in which all interior interface edges are deleted, and only the silhouette of the clipart serves as input to our algorithm. There are no open edges, so the value of β has no effect. Varying α demonstrates the behavior near corners discussed in Section 3.4.4: In convex corners (orange circle), the interface bisects the corner angle

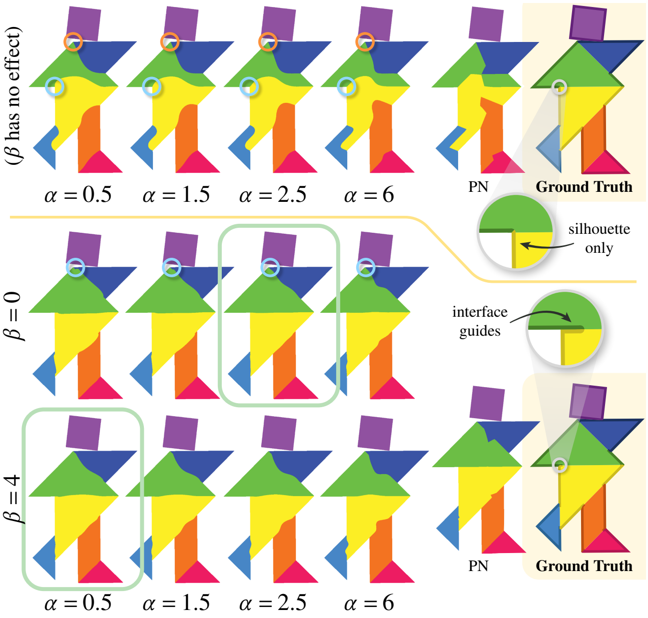


Figure 10: *Reconstruction of multi-label interfaces in a* Tangram figure. Top: *The silhouette of the figure serves as input geometry. In convex corners (orange circle), the interface bisects the corner angle for all $\alpha \geq 0$, but in re-entrant corners (cyan circle), only for all $\alpha \geq 2$.* Bottom: *At every label transition, we add to the input geometry a very short edge pointing in the desired interface direction, which improves the fidelity of the reconstruction. At the endpoints of these open edges (cyan circles), the interface is C^1 for $\alpha \geq 2$.*

for all values of α; in re-entrant corners (cyan circle), bisection is only guaranteed for $\alpha \geq 2$. Overall, no parameter setting achieves a reconstruction close to the ground truth given only the silhouette.

An interesting feature of our algorithm is that it can make use of extra information about interface tangent directions. To show this, we add a very short interior edge to every label transition on the silhouette; this edge acts as an interface guide and points in the tangent direction. Every greatest empty circle that would usually touch the label transition vertex will now touch the interface guide instead; this causes label resolution to be consistent with the intended interface direction. As a result, the reconstruction quality is strongly improved, and near-linear interfaces emerge for some parameter choices (green boxes) without any priors on linearity.

We feel that adding interface directions to the input data is a natural choice in the non-manifold/multi-label setting, similar to how normal orientations usually accompany point samples for manifold reconstruction with PSR. Instead of explicitly adding short edges to serve as interface guides, one could also modify the circle labeling rules illustrated in Fig. 4 *(right)* to take into account the interface direction; this is equivalent to letting the length of the interface guide go to zero, and setting $\beta = 0$.

Other Examples Figs. 11 and 12 show three more reconstruction examples. In Fig. 11 *(left)* we reconstruct an illustration of a *pinecone* after removing all downward-facing boundary edges, similar to geometry that might be obtained by 3d-scanning an object from one side. This type of bias in the input geometry is a catastrophic failure case for the GWN method because it leads to a global gradient in the GWN field, and level sets do not meaningfully capture local features.

In Fig. 11 *(right)*, we reconstruct a *hot air balloon* after randomly removing every input edge with a 50% probability. Screened PSR is difficult to tune for this example: The reconstructed stripes of the balloon are lumpy, and in the basket area, where the feature size is very small, the reconstructed contour does not follow the input edges faithfully. The GWN method fails catastrophically on this example, similar to the pine cone (not shown).

Fig. 12 shows reconstructions of *lovebirds* based on a sparse set of uniformly distributed point samples with oriented normals, which is the typical input for screened PSR. To run our method on this type of input, we replace every point sample with a short edge (length = 0.01 × sample distance) orthogonal to the input normal. Here our method yields results similar to those of the GWN method, both of which retain more definition than the result of screened PSR. For the GWN method to work here, the level set parameter needs to be adapted precisely to the sampling density. This is only possible for a uniform sample distribution; otherwise, artifacts like those in Fig. 9 are unavoidable.

6.1. Comparison with VHRBF

Apart from core geometry completion methods like PSR and GWN, we may also investigate implicit methods based on *variational Hermite radial basis functions* (VHRBF), which have traditionally been used in the context of sketch-based modeling [BMS*10; HCJ19]. Similar to PSR, these methods are based on fitting a function so its 0-level set passes through the sample points, and its gradient aligns with the oriented normal at the sample locations. We apply the method suggested by Brazil et al. [BMS*10], which is based on the Hermite interpolation scheme by Duchon [Duc77], to some of our examples in Fig. 14.

The method performs well on the *footprint* example, but struggles to varying degrees with the others. Especially on the *pinecone*, the Hermite basis is not suitable to reproduce the pointy features, because the interpolation constraints force stationary points to appear in the 0-level set of the VHRBF implicit function that coincide with these features. This creates severe artifacts in the reconstruction derived from the 0-level set. The method fares a little better on the *lovebirds* and the *balloon*, except that it merges the bird beaks with the bottom-most heart, and creates high-frequency artifacts in the region left of the balloon burners. Since the method has no parameters, these issues are difficult to circumvent using VHRBF interpolation, but they may be alleviated by adding a regularization term [HCJ19], which leads to approximation of the sample points instead of interpolation.

6.2. Extending Other Methods to Multi-Label Inputs

Our method extends to the multi-label setting by computing one confidence function per label, and then combining them with the

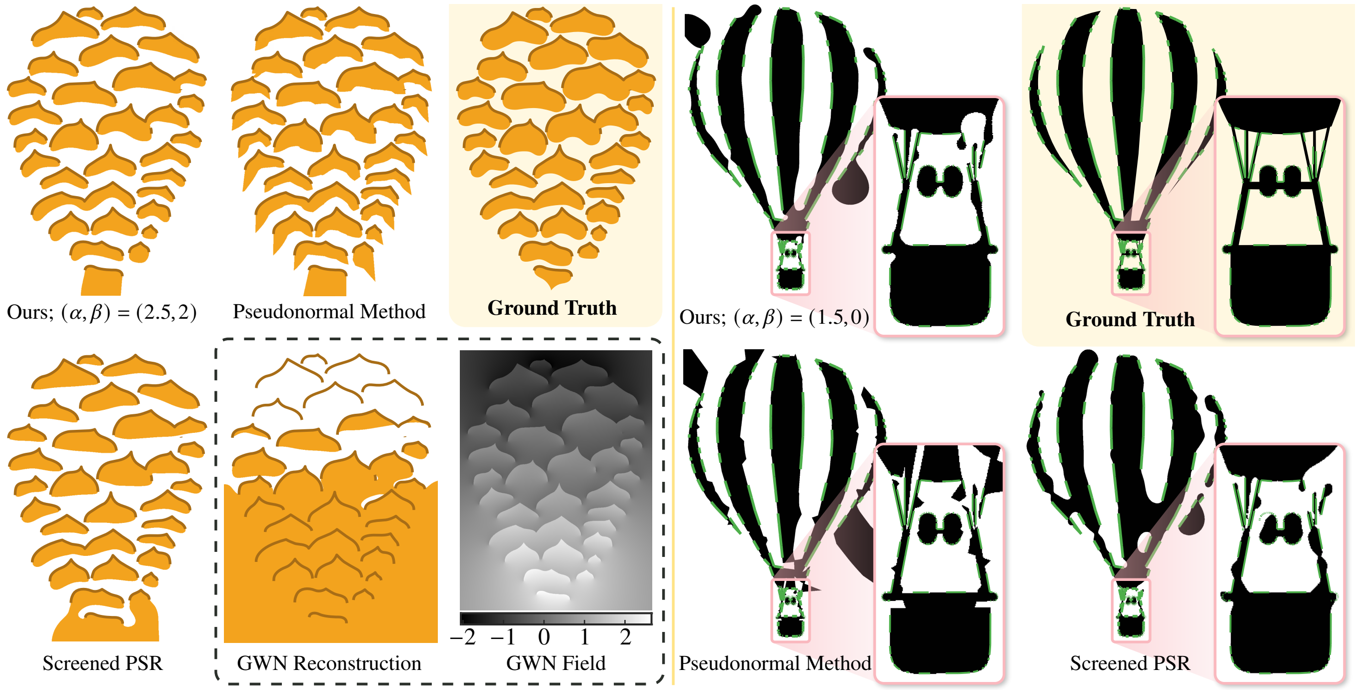


Figure 11: Left: *Reconstruction of a* Pinecone *clipart after removing downward-facing boundary edges.* Right: *Reconstruction of a* Hot air balloon *clipart after randomly removing 50% of the boundary edges. Remaining input edges are shown in green.*

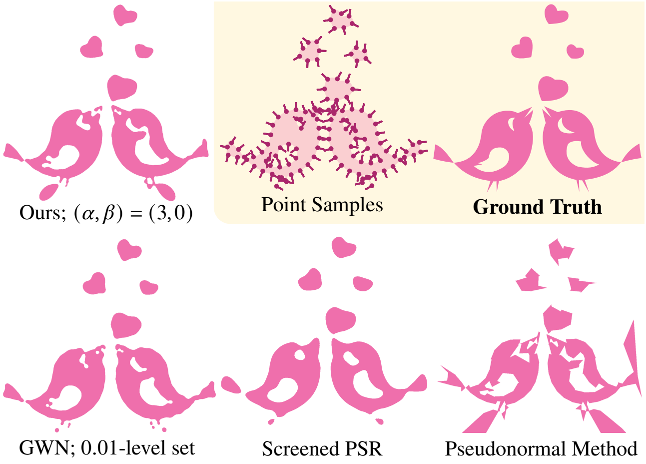


Figure 12: *Reconstruction of* Lovebirds *clipart from sparse point samples with oriented normals.*

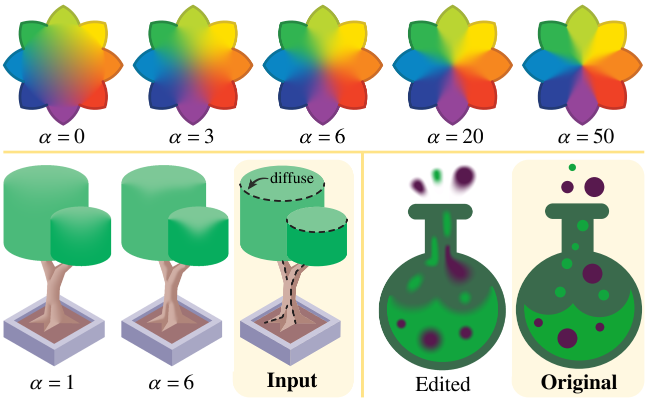


Figure 13: *Color blending with the total confidence score. Our method can mimic "diffusion curves" and additionally control the sharpness of color transitions with α.*

arg max operator, see Eq. 3. Since other methods also perform classification by thresholding a function derived from the input geometry, one may wonder if they trivially extend to multi-label inputs in the same way. To test this idea, we partition the input geometry into sets of oriented edges sharing the same label, and compute one function for each, adapting the convention that a positive sign indicates being inside of a labeled region. Finally, every point receives the label corresponding to the function with the greatest value.

Fig. 15 shows that this has merit for GWN, but fails for the Screened PSR and VHRBF methods. The opening-angle indicator function computed by GWN naturally has the property that it decays far from edges with the corresponding label. This is helpful in avoiding spurious resurfacing of distant labels and leads to a satisfactory result. However, the multi-label GWN variant will suffer from the same catastrophic failure cases as the GWN inside–outside classifier, as shown in Figs. 9 and 11. The approximate indicator functions derived from PSR have a maximum value on the

Figure 14: *Reconstruction examples with the VHRBF method [BMS*10], using the same input data as Figs. 9–11.*

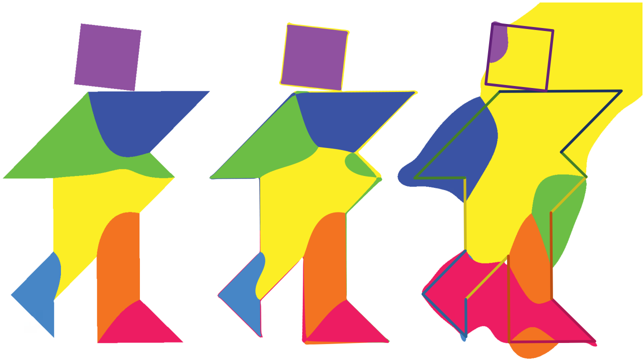


Figure 15: *Attempts at extending other inside–outside classification methods to the multi-label setting.* Right: *Input edges shown for comparison with the classification.*

order of +1, but this value is not tightly controlled; this is by design, because only the 0-level set matters in the traditional inside–outside setting. As a result, the classification obtained by combining multiple PSR indicator functions depends on which function happens to attain a higher maximum. This yields an unintuitive classification in which the yellow and blue regions are adjacent. Moreover, artifacts appear near the silhouette of the figure, because the functions corresponding to the two labels of the edge are both approximately zero there (owing to the screening term), so a different label with a strictly positive function value will bleed through. Finally, the values of VHRBF implicit functions have no magnitude control at all, so combining them via the max operator yields a nonsensical result.

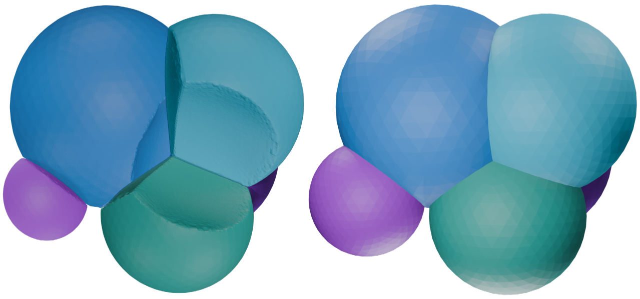
6.3. Color Blending

The confidence scores $c_\ell(q)$ of the labels $\ell \in \mathcal{L}$, which we usually use to determine the final label at a point q via Eq. 3, can also be used to visualize the "probability" of different labels. To do this, we associate each label ℓ with a color u_ℓ, and compute the final color at q via

$$u(q) = \frac{\sum_{\ell \in \mathcal{L}} c_\ell(q) \cdot u_\ell}{\sum_{\ell \in \mathcal{L}} c_\ell(q)},$$

which is well-defined because $c_{\ell(q)}(q) > 0$ everywhere, so the denominator never vanishes.

This achieves an effect similar to *diffusion curves* [OBW*08], and we show a few examples of this in Fig. 13. One qualitative difference between our method and the original diffusion curves is that colors have local support, even within an enclosed region. For example, in Fig. 13 *(top)*, each petal of the *flower* clipart retains the color of its boundary in a region with positive area. This is because all greatest empty circles that contain a point close to the tip of a petal touch the boundary of the same petal; this implies that there is only one label with a non-zero confidence score in this region. Another advantage of our method is that the parameter α provides an easy way to control the sharpness of color transitions. Setting $\alpha = 0$ causes maximal color mixing, and increasing α separates the colors gradually; in the limit $\alpha \longrightarrow \infty$, the result of the pseudonormal method is recovered, as in Fig. 10.



$\alpha = 2$ (cut-away view) **Input**

$\alpha = 2$ (cut-away view) **Input**

Figure 16: *Reconstruction of* Bubble *and* ABC letter *models.*

Fig. 13 *(bottom)* shows two more examples of color blending. On the *left*, we mark some edges of a *tree* clipart as diffuse, which is effectively done by removing them from the set of input edges. This causes colors between adjacent regions to mix. On the *right*, we edit a clipart of a *poison flask* by specifying additional regions in which colors are blended to gain more control over the amount and direction of blurring.

6.4. 3D Prototype

The labeling model proposed in Section 3 extends to $\mathbb{R}^3$ in a straight-forward manner. As input geometry, we assume a triangle mesh equipped with a double-sided face labeling, which is used to label greatest empty spheres. This is done as in Section 3.3, except that $\det(t, \cdot)$ is replaced by $\langle n, \cdot \rangle$, with n the oriented face normal. This gives a labeling analogous to Fig. 5 *(right)* when the greatest empty sphere centered at a point p touches an edge or vertex of the input geometry. In case the sphere touches a boundary of the input geometry, we distinguish touching a boundary edge—in which case the angle θ is defined analogous to the two-dimensional setting—or a boundary vertex v. We implement the latter case by computing angles θ_1, θ_2 based on the two boundary edges incident to the vertex, and choosing $\cos\theta$ as the weighted mean of $\cos\theta_1$ and $\cos\theta_2$ with weights $w_i = |\langle d, t_i\rangle|^{-1}$, $i = 1, 2$, where $d = (v-p)/\|v-p\|$ and t_i are the edge tangent vectors.

The confidence integration and final label selection in Eqs. 2 and 3 remain the same. The analysis done in Section 3.4 remains valid in three dimensions, except that the critical exponent is $\alpha = 3$ instead of $\alpha = 2$. This is because the integral of $1/r^\alpha$ over the unbounded region associated with a vertex on the convex hull of the input geometry is finite only for $\alpha > 3$ in $\mathbb{R}^3$. Similarly, the critical exponents associated with edge and face regions are $\alpha = 2$ and $\alpha = 1$, respectively.

We implemented the algorithm presented in Section 5 for three dimensions using an octree in place of a quadtree. This achieves a complexity of $O(n^5)$ on a regular grid with $N = n^3$ sample points.

Fig. 1 shows two examples of manifold surface completion using meshes from the Thingi10K dataset after manually removing parts of the original model. For the *heart* model, we reconstruct missing disks, leading to a smooth appearance and a surprisingly faithful reconstruction of the two veins (orange circles). On the *polyhedron* model, we test the capability of our method to reconstruct surface parts with different genera. This attempt restores the original topology of the model apart from one difference in a region with a lot of missing geometry (blue circles).

In Fig. 16 we show examples of multi-label surface completion. The *bubble cluster* model is a union of five intersecting spheres having different labels, with no internal geometry present in the input mesh. Our reconstruction generates internal walls between the spheres, and features T-junctions between three labels and an interior point where four labels coalesce with equal confidence. The *ABC letter* model is an arrangement of three meshes with different labels that intersect in irregularly shaped regions. To generate the input, we remove all internal geometry, so we are left with only the boundary of the union of the three volumes. Our reconstruction plausibly partitions the interior volume into separate regions corresponding to the three letters.

7. Limitations

The 3d prototype of our algorithm has several limitations that are not observed in two dimensions. Firstly, the complexity of our algorithm in 3d enables computation on a grid with $N = 200^3$ samples within about 10 minutes, so it is far from interactive, and does not scale well to higher resolutions. This makes the current implementation ill-suited to reconstructing geometry with very fine details and is practical only for meshes with a complexity similar to that of the *heart* example shown in Fig. 1 *(left)*. For increased scalability, it will be necessary to introduce adaptive sampling near the reconstructed surface, for example by refining the octree locally; this is not something we explore in this manuscript.

The result of our method is a per-point label assignment, or an inside–outside classification in the manifold setting. In two dimensions, this label assignment can easily be visualized by coloring pixels according to labels. In 3d, visualizations are usually obtained by extracting the interface surfaces between regions and using them in a mesh-based rendering pipeline. Producing a clean surface from a per-point labeling is a challenging problem in its own right, and especially for the non-manifold setting, an active area of research. To generate the images in Fig. 16, we surface each labeled region separately by using the standard marching-cubes algorithm to extract the 0-level set of the function

$$f_\ell(q) = c_\ell(q) - \max_{\tilde{\ell}\neq\ell} c_{\tilde{\ell}}(q),$$

which produces the boundary of the region where the label $\ell \in \mathcal{L}$ has the highest confidence. This results in clean meshes for each region, but they remain separate as can be observed by zooming into the T-junction in Fig. 16 (top-left), and have no non-manifold connections. To obtain a single mesh with the correct connectivity between adjacent region, one can employ multi-label variants of marching cubes, or more sophisticated algorithms for meshing implicit arrangements such as that by Du et al. [DZCJ22].

Another challenge for surfacing is that $c_\ell(q)$ diverges as q approaches the input surface, which leads to poor reproduction of the input geometry when using marching-cubes directly on $f_\ell(q)$. A simple fix is to replace the sample value $f_\ell(q_i)$ with

$$\tilde{f}_\ell(q_i) := \operatorname{sgn}(f_\ell(q_i)) \cdot r(q_i),$$

for all samples q_i that are closer to the input geometry than the diameter of one marching-cubes cell. This leaves the sign of the sample—and thus the topology of the reconstructed surface—unchanged, but replaces the magnitude with the unsigned distance to the input geometry, improving reproduction quality. This is only done very close to the input mesh, so the reconstructed interfaces away from the input geometry are not affected.

8. Conclusion

We introduce a new method for geometry completion which is the first algorithm that naturally extends to the multi-label (or non-manifold) setting. In two dimensions, we compare this method to established techniques such as screened PSR, the GWN method, and the pseudonormal method. On all of our examples, our method is at least on par with previous work, and produces better results on many of them. In three dimensions, we demonstrate the potential of our method, but leave several possible improvements, especially relating to performance and surface extraction, as avenues for future work.

References

[ACK13] ATTENE, MARCO, CAMPEN, MARCEL, and KOBBELT, LEIF. "Polygon mesh repairing: An application perspective". *ACM Comput. Surv.* 45.2 (Mar. 2013). ISSN: 0360-0300. DOI: `10.1145/2431211.2431214`. URL: `https://doi.org/10.1145/2431211.2431214` 2.

[BA05] BAERENTZEN, J.A. and AANAES, H. "Signed distance computation using the angle weighted pseudonormal". *IEEE Transactions on Visualization and Computer Graphics* 11.3 (2005), 243–253. DOI: `10.1109/TVCG.2005.49` 1, 2, 9.

[BDS*18] Barill, Gavin, Dickson, Neil G., Schmidt, Ryan, et al. “Fast winding numbers for soups and clouds”. *ACM Trans. Graph.* 37.4 (July 2018). ISSN: 0730-0301. DOI: 10.1145/3197517.3201337. URL: https://doi.org/10.1145/3197517.3201337 2.

[BK97] Barequet, Gill and Kumar, Subodh. “Repairing CAD models”. *Proceedings of the 8th Conference on Visualization ’97*. VIS ’97. Phoenix, Arizona, USA: IEEE Computer Society Press, 1997, 363–ff. ISBN: 1581130112 2.

[BMS*10] Brazil, E. Vital, Macedo, I., Sousa, M. Costa, et al. “Sketching Variational Hermite-RBF implicits”. *Proceedings of the Seventh Sketch-Based Interfaces and Modeling Symposium*. SBIM ’10. Annecy, France: Eurographics Association, 2010, 1–8. ISBN: 9783905674255 10, 12.

[BNK02] Borodin, Pavel, Novotni, Marcin, and Klein, Reinhard. “Progressive Gap Closing for MeshRepairing”. *Advances in Modelling, Animation and Rendering*. Ed. by Vince, John and Earnshaw, Rae. London: Springer London, 2002, 201–213. ISBN: 978-1-4471-0103-1 2.

[BPK05] Bischoff, Stephan, Pavic, Darko, and Kobbelt, Leif. “Automatic restoration of polygon models”. *ACM Trans. Graph.* 24.4 (Oct. 2005), 1332–1352. ISSN: 0730-0301. DOI: 10.1145/1095878.1095883. URL: https://doi.org/10.1145/1095878.1095883 2.

[CMG24] Chen, Hanyu, Miller, Bailey, and Gkioulekas, Ioannis. “3D Reconstruction with Fast Dipole Sums”. *ACM Trans. Graph.* 43.6 (Nov. 2024). ISSN: 0730-0301. DOI: 10.1145/3687914. URL: https://doi.org/10.1145/3687914 2.

[DA21] Diazzi, Lorenzo and Attene, Marco. “Convex polyhedral meshing for robust solid modeling”. *ACM Trans. Graph.* 40.6 (Dec. 2021). ISSN: 0730-0301. DOI: 10.1145/3478513.3480564. URL: https://doi.org/10.1145/3478513.3480564 2.

[Duc77] Duchon, Jean. “Splines minimizing rotation-invariant seminorms in Sobolev spaces”. *Constructive Theory of Functions of Several Variables*. Ed. by Schempp, Walter and Zeller, Karl. Berlin, Heidelberg: Springer Berlin Heidelberg, 1977, 85–100. ISBN: 978-3-540-37496-1 10.

[DZCJ22] Du, Xingyi, Zhou, Qingnan, Carr, Nathan, and Ju, Tao. “Robust computation of implicit surface networks for piecewise linear functions”. *ACM Trans. Graph.* 41.4 (July 2022). ISSN: 0730-0301. DOI: 10.1145/3528223.3530176. URL: https://doi.org/10.1145/3528223.3530176 13.

[FC24] Feng, Nicole and Crane, Keenan. “A Heat Method for Generalized Signed Distance”. *ACM Trans. Graph.* 43.4 (July 2024). ISSN: 0730-0301. DOI: 10.1145/3658220. URL: https://doi.org/10.1145/3658220 2.

[FGC23] Feng, Nicole, Gillespie, Mark, and Crane, Keenan. “Winding Numbers on Discrete Surfaces”. *ACM Trans. Graph.* 42.4 (July 2023). ISSN: 0730-0301. DOI: 10.1145/3592401. URL: https://doi.org/10.1145/3592401 2.

[GXW18] Guo, Xiaoyuan, Xiao, Jun, and Wang, Ying. “A survey on algorithms of hole filling in 3D surface reconstruction”. *The Visual Computer* 34.1 (2018), 93–103 2.

[HCJ19] Huang, Zhiyang, Carr, Nathan, and Ju, Tao. “Variational implicit point set surfaces”. *ACM Trans. Graph.* 38.4 (July 2019). ISSN: 0730-0301. DOI: 10.1145/3306346.3322994. URL: https://doi.org/10.1145/3306346.3322994 10.

[HKS*24] Heiss-Synak, Peter, Kalinov, Aleksei, Strugaru, Malina, et al. “Multi-Material Mesh-Based Surface Tracking with Implicit Topology Changes”. *ACM Trans. Graph.* 43.4 (Sept. 2024) 2.

[JKS13] Jacobson, Alec, Kavan, Ladislav, and Sorkine-Hornung, Olga. “Robust Inside-Outside Segmentation using Generalized Winding Numbers”. *ACM Transactions on Graphics (proceedings of ACM SIGGRAPH)* 32.4 (2013), 33:1–33:12 1, 2, 9.

[Ju04] Ju, Tao. “Robust repair of polygonal models”. *ACM Trans. Graph.* 23.3 (Aug. 2004), 888–895. ISSN: 0730-0301. DOI: 10.1145/1015706.1015815. URL: https://doi.org/10.1145/1015706.1015815 2.

[KBH06] Kazhdan, Michael, Bolitho, Matthew, and Hoppe, Hugues. “Poisson surface reconstruction”. *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*. SGP ’06. Cagliari, Sardinia, Italy: Eurographics Association, 2006, 61–70. ISBN: 3905673363 1, 2.

[KCRH20] Kazhdan, Misha, Chuang, Ming, Rusinkiewicz, Szymon, and Hoppe, Hugues. “Poisson Surface Reconstruction with Envelope Constraints”. *Computer Graphics Forum* 39.5 (2020), 173–182. DOI: https://doi.org/10.1111/cgf.14077. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.14077. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.14077 2.

[KH13] Kazhdan, Michael and Hoppe, Hugues. “Screened poisson surface reconstruction”. *ACM Trans. Graph.* 32.3 (July 2013). ISSN: 0730-0301. DOI: 10.1145/2487228.2487237. URL: https://doi.org/10.1145/2487228.2487237 2, 9.

[LEM*17] Livesu, Marco, Ellero, Stefano, Martínez, Jonís, et al. “From 3D models to 3D prints: an overview of the processing pipeline”. *Comput. Graph. Forum* 36.2 (May 2017), 537–564. ISSN: 0167-7055 2.

[MB25] Martens, C. and Bessmeltsev, M. “One-Shot Method for Computing Generalized Winding Numbers”. *Computer Graphics Forum* 44.5 (2025), e70194. DOI: https://doi.org/10.1111/cgf.70194. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.70194. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.70194 2.

[OBW*08] Orzan, Alexandrina, Bousseau, Adrien, Winnemöller, Holger, et al. “Diffusion curves: a vector representation for smooth-shaded images”. *ACM SIGGRAPH 2008 Papers*. SIGGRAPH ’08. Los Angeles, California: Association for Computing Machinery, 2008. ISBN: 9781450301121. DOI: 10.1145/1399504.1360691. URL: https://doi.org/10.1145/1399504.1360691 12.

[SGW24] Spainhour, Jacob, Gunderman, David, and Weiss, Kenneth. “Robust Containment Queries over Collections of Rational Parametric Curves via Generalized Winding Numbers”. *ACM Trans. Graph.* 43.4 (July 2024). ISSN: 0730-0301. DOI: 10.1145/3658228. URL: https://doi.org/10.1145/3658228 2.

[SJ22] Sellán, Silvia and Jacobson, Alec. “Stochastic Poisson Surface Reconstruction”. *ACM Trans. Graph.* 41.6 (Nov. 2022). ISSN: 0730-0301. DOI: 10.1145/3550454.3555441. URL: https://doi.org/10.1145/3550454.3555441 2.

[SJ23] Sellán, Silvia and Jacobson, Alec. “Neural Stochastic Poisson Surface Reconstruction”. *SIGGRAPH Asia 2023 Conference Papers*. SA ’23. Sydney, NSW, Australia: Association for Computing Machinery, 2023. ISBN: 9798400703157. DOI: 10.1145/3610548.3618162. URL: https://doi.org/10.1145/3610548.3618162 2.

[SWBH24] Sun, Haoran, Wang, Jingkai, Bao, Hujun, and Huang, Jin. “GauWN: Gaussian-smoothed Winding Number and its Derivatives”. *SIGGRAPH Asia 2024 Conference Papers*. SA ’24. Tokyo, Japan: Association for Computing Machinery, 2024. ISBN: 9798400711312. DOI: 10.1145/3680528.3687569. URL: https://doi.org/10.1145/3680528.3687569 2.

[WLG03] Wagner, Marc, Labsik, Ulf, and Greiner, Günther. “Repairing Non-Manifold Triangle Meshes using Simulated Annealing”. *International Journal of Shape Modeling* 09.02 (2003), 137–153. DOI: 10.1142/S0218654303000085. eprint: https://doi.org/10.1142/S0218654303000085. URL: https://doi.org/10.1142/S0218654303000085 2.