

Defining, Contouring, and Visualizing Scalar Functions on Point-Sampled Surfaces

Giuseppe Patanè and Bianca Falcidieno

Istituto di Matematica Applicata e Tecnologie Informatiche

Consiglio Nazionale delle Ricerche

Via De Marini 6, 16149 Genova - Italy

Corresponding author e-mail: patane@ge.imati.cnr.it

Abstract

This paper addresses the definition, contouring, and visualization of scalar functions on unorganized point sets, which are sampled from a surface in 3D space; the proposed framework builds on moving least-squares techniques and implicit modeling. Given a scalar function $f : \mathcal{P} \rightarrow \mathbb{R}$, defined on a point set \mathcal{P} , the idea behind our approach is to exploit the local connectivity structure of the k -nearest neighbor graph of \mathcal{P} and mimic the contouring of scalar functions defined on triangle meshes. Moving least-squares and implicit modeling techniques are used to extend f from \mathcal{P} to the surface \mathcal{M} underlying \mathcal{P} . To this end, we compute an analytical approximation \tilde{f} of f that allows us to provide an exact differential analysis of \tilde{f} , draw its iso-contours, visualize its behavior on and around \mathcal{M} , and approximate its critical points. We also compare moving least-squares and implicit techniques for the definition of the scalar function underlying f and discuss their numerical stability and approximation accuracy. Finally, the proposed framework is a starting point to extend those processing techniques that build on the analysis of scalar functions on 2-manifold surfaces to point sets.

Key words: Approximation of surfaces and contours, nearest neighbor graph, graph algorithms, point-sampled geometry, topological and shape modeling, computational geometry and object modeling.

1. Introduction

Point-sampled surfaces, generated either by scanning real 3D objects with optical devices, or polygonizing implicit functions, or sampling parametric surfaces, are discrete models with a high number of points and the underlying surfaces can have an arbitrary genus and curvature. Point sets become a surface representation alternative to polygonal meshes, due to the simplicity of dealing with complex 3D shapes as point sets and using points as rendering primitives (40; 62; 66; 76). The lack of connectivity and the atomic definition of point sets provide a built-in multi-scale surface representation (61), thus avoiding to process the connectivity of polygonal meshes. Point sets are widely used for ray tracing (2), surface reconstruction (45; 73), sampling (3), simplification (60), spectral analysis (59), segmentation (10), progressive rendering and streaming (30).

Given a point set \mathcal{P} sampled from a 2D manifold surface, this paper addresses the definition, contouring, and visualization of an arbitrary scalar function $f : \mathcal{P} \rightarrow \mathbb{R}$, defined on \mathcal{P} , without converting \mathcal{P} to a simplicial representation. This aim is achieved by studying the level-sets and critical points of f , according to the case of scalar functions defined on 2-manifold surfaces.

The proposed framework builds on moving least-squares and implicit modeling techniques.

On the one hand, in point-based graphics the aforementioned problems have not been investigated; furthermore, the proposed solution provides a link between previous work on surface-based scalar functions and the ongoing research on point-sampled surfaces. On the other hand, the theoretical interest on those problems is accompanied by applications such as surface reconstruction from slices (12; 13; 44) and the design of high-level structures of \mathcal{P} (69) (e.g., Reeb graphs, Morse complexes), which are commonly used for shape comparison (18; 38; 19) and analysis (55; 56).

It is worth mentioning that scalar functions are extensively used in mathematical modeling, engineering simulation, scientific visualization, bio-medicine, and geographic information systems. In each of these research fields, a variety of phenomena is described by a large set of measurements: the resulting scientific data is usually given as a set of samples over a domain of interest and the measurements are represented as a scalar function. Finally, in shape modeling, computer graphics, and engineering, a large number of functions are generated also by restricting implicit maps from \mathbb{R}^3 to 3D shapes, solving

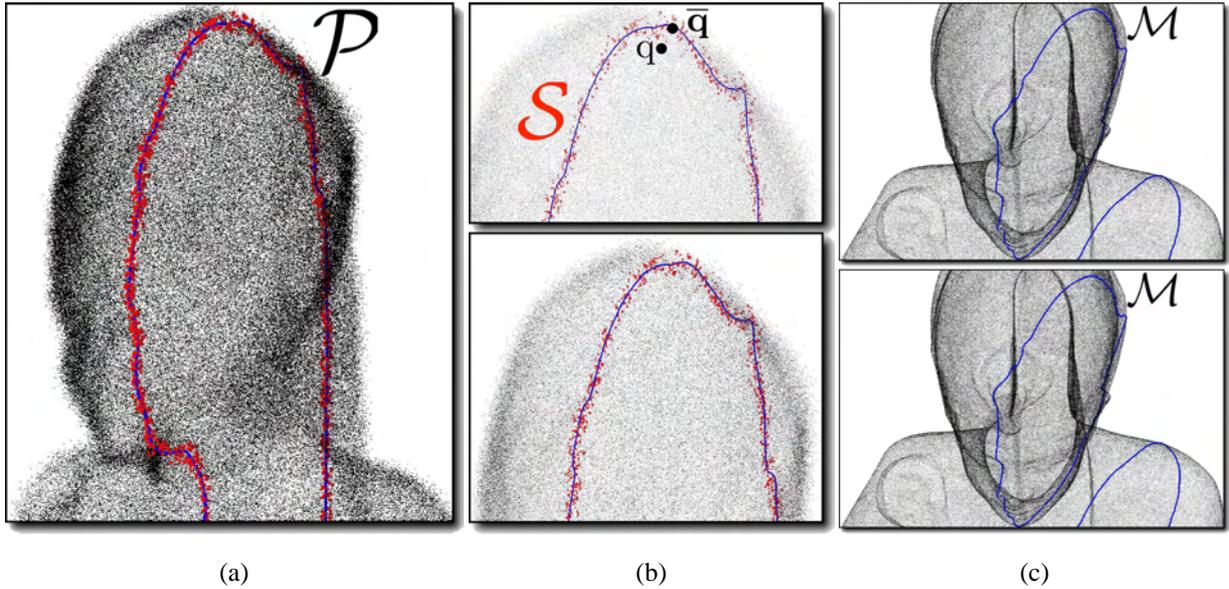


Fig. 1. Overview of the proposed approach. Given a scalar function $f: \mathcal{P} \rightarrow \mathbb{R}$, defined on a point set \mathcal{P} , the figure shows the main steps that compute the level-set $\gamma_\alpha := \{\mathbf{p} \in \mathcal{M} : \tilde{f}(\mathbf{p}) = \alpha\}$, where \mathcal{M} and \tilde{f} are the surface and the scalar function underlying \mathcal{P} and f respectively. (a) Computation of a point-wise approximation \mathcal{S} (red points) of γ_α by searching those points \mathbf{q} such that $f(\mathbf{q}) = \alpha$ on the edges of the k -nearest neighbor graph \mathcal{T} of \mathcal{P} ; to this end, we linearly or implicitly interpolate the f -values along the edges of \mathcal{T} . (b) Then, each point $\mathbf{q} \in \mathcal{S}$ is “projected” onto a point $\bar{\mathbf{q}} := \text{pr}(\mathbf{q})$ of γ_α and the set $\mathcal{C} := \{\bar{\mathbf{q}} := \text{pr}(\mathbf{q}) : \mathbf{q} \in \mathcal{S}\}$ (blue points) gives a discrete sampling of γ_α . (c) Identification of the subsets of \mathcal{C} that correspond to a sampling of the connected components of γ_α . Each sampled connected component is converted to a piecewise linear contour (blue curves).

differential equations related to simulation problems (e.g., the Laplace and heat equation (14; 52)) or decomposing the spectrum of data-dependent kernels (14).

Choosing a point set \mathcal{P} , hereinafter \mathcal{M} will generally be the surface underlying \mathcal{P} , i.e. any surface the points of \mathcal{P} belong or are close to. Our framework does not make assumptions on the way \mathcal{M} is built from \mathcal{P} : both moving least-squares (1; 3; 5; 6; 43) and implicit (51; 68; 70) techniques can be adopted. Indeed, we assume that \mathcal{M} is locally represented by the zero-set of an implicit function g ; that is, a point \mathbf{x} in a neighborhood $\mathcal{N}_{\mathbf{p}}$ of $\mathbf{p} \in \mathcal{P}$ belongs to \mathcal{M} if and only if $g(\mathbf{x}) = 0$. Furthermore, we admit boundary components, noise, outliers, and an irregular sampling density of \mathcal{P} .

First of all, the scalar function $f: \mathcal{P} \rightarrow \mathbb{R}$ is arbitrary and defined by its values on the points of \mathcal{P} . Then, we define the scalar function $\tilde{f}: \mathcal{M} \rightarrow \mathbb{R}$, underlying f , as a map that locally interpolates or approximates the values $\{f(\mathbf{p}_i)\}_{i=1}^n$. This smooth approximation \tilde{f} of f allows us to provide an exact differential analysis of \tilde{f} and visualize its behavior on and around \mathcal{M} . To define such a function, f is locally approximated by an implicit and smooth map $F: \mathbb{R}^3 \rightarrow \mathbb{R}$ such that $\tilde{f} := F|_{\mathcal{M}}$ is the restriction of F to \mathcal{M} . Among the several maps defined on \mathcal{P} , specific attention is devoted to the analysis of smooth functions, such as harmonic maps and Laplacian eigenfunctions (14; 15; 16; 41). These functions have been used in several applications that include shape comparison (64) and surface quadrangulation (27; 28).

Once \tilde{f} has been computed, we are ready to trace the level-sets of \tilde{f} on \mathcal{M} . The idea behind the contouring algorithm is to exploit the local connectivity structure of the k -nearest neighbor graph \mathcal{T} of \mathcal{P} and mimic the contouring of functions on

triangle meshes. We consider $(\mathcal{P}, \mathcal{T})$ as a rough approximation of the surface \mathcal{M} underlying \mathcal{P} and we show that it is precise enough to compute a set \mathcal{S} (Fig. 1(a)) of approximated samples of the level-set $\gamma_\alpha := \{\mathbf{p} \in \mathcal{M} : \tilde{f}(\mathbf{p}) = \alpha\}$. For each point $\mathbf{q} \in \mathcal{S}$, the volume-based approximation F of f and the implicit representation $\{\mathbf{x} : g(\mathbf{x}) = 0\}$ of \mathcal{M} in a neighborhood of \mathbf{q} are used to “project” \mathbf{q} (Fig. 1(b)) onto $\bar{\mathbf{q}} \in \gamma_\alpha$ by solving the following non-linear system

$$g(\mathbf{r}) = 0 \quad (\text{i.e., } \mathbf{r} \in \mathcal{M}), \quad F(\mathbf{r}) - \alpha = 0 \quad (\text{i.e., } \mathbf{r} \in \gamma_\alpha).$$

As starting point of the iterative solver of the aforementioned system, we choose $\mathbf{r}_0 := \mathbf{q}$. Finally, \mathcal{C} is converted to a family of piecewise linear curves, one for each connected component of γ_α (Fig. 1(c)).

Since the computation of the critical points of an arbitrary scalar function on a point-sampled surface is sensible to the local noise that affects \mathcal{P} and the function values, we introduce the concept of *fuzzy critical points*. More precisely, we associate to \mathcal{M} a probability measure $\pi: \mathcal{M} \rightarrow [0, 1]$ such that $\pi(\mathbf{p})$ is the probability that \mathbf{p} is a critical point of \tilde{f} . Then, the analysis of the level-sets of \tilde{f} improves the reliability of the classification of the critical points.

The choice of using the neighbor graph of \mathcal{P} is motivated by the analogy among its structure, the edge-based representation of polygonal models, and the classification of the critical points of scalar functions defined on triangulated surfaces (11). Furthermore, the modularity of \mathcal{T} with respect to k and σ avoids the need to recompute the neighbor graph when we decrease those parameters for approximating the normals of \mathcal{P} and the gradient field of f , or tracing the level-sets. The proposed approach also represents a starting point to extend those methods that build on the analysis of a scalar function on a 2-manifold surface to

point sets. Among them, we mention the surface reconstruction from slices (12; 13; 44), quadrilateral remeshing (27; 28), parameterization (57; 75), and shape comparison (18; 38; 64).

The paper is organized as follows. In Section 2, we briefly introduce the theoretical background and previous work on moving least-squares techniques, implicit modeling, and the definition of scalar functions on point sets. Section 3 discusses the computation of the map \tilde{f} underlying an arbitrary scalar function f defined on \mathcal{P} and the visualization of its behavior on and around the surface \mathcal{M} underlying \mathcal{P} . In Section 4, we introduce a contouring algorithm of \tilde{f} , which traces the level-sets without degenerate segments. In Section 5, we analyze the critical points of \tilde{f} and introduce the concept of fuzzy critical points for functions defined on point-sampled surfaces. In Section 6, we address the numerical stability of the proposed framework, its accuracy, and the choice of its main parameters. Finally, Section 7 concludes the paper.

2. Theoretical background and related work

In this section, we briefly review the main results on moving least-squares techniques, surface approximation with radial basis functions (Section 2.1), and the definition of scalar functions on point sets (Section 2.2). For more details, we refer the reader to the state-of-the-art reports (4; 37).

2.1. Approximation of point sets

Hereinafter, we focus our attention on the main methods used to compute the k -nearest neighbor graph, moving least-squares surfaces, and implicit approximations of point sets.

k -nearest neighbor graph. First of all, we introduce the k -nearest neighbor graph \mathcal{T} of a point set $\mathcal{P} := \{\mathbf{p}_i, i = 1, \dots, n\}$ of \mathbb{R}^3 . In \mathcal{T} , each point $\mathbf{p}_i \in \mathcal{P}$ is associated with the neighborhood $\mathcal{N}_{\mathbf{p}_i} := \{\mathbf{p}_{j_s}, s = 1, \dots, k\}$ that includes the k nearest points to \mathbf{p}_i , or those that fall inside the sphere of center \mathbf{p}_i and radius σ . In both cases, the proximity relations among the points of \mathcal{P} are computed with respect to the Euclidean distance. If we choose k instead of σ , then we set $\sigma := \max_{s=1, \dots, k} \{\|\mathbf{p}_i - \mathbf{p}_{j_s}\|_2\}$. To simplify the notation, once σ has been fixed we implicitly assume that $\mathcal{N}_{\mathbf{p}_i}$ contains $k := k(\sigma)$ points and we omit the dependence of k on σ . As described in (26; 50), the choice of σ can be adapted to the local sampling density $\varepsilon := \frac{k}{\pi\sigma^2}$ and the curvature of the surface underlying \mathcal{P} , thus improving the approximation of the normals to \mathcal{P} . The computation of \mathcal{T} requires $O(n \log n)$ -time (9; 17), where n is the number of input points, and the neighbor graph can be adapted to the surface sampling through the parameter k or the query radius σ . If an *a-priori* information on the surface is given, then k and σ are selected by the user; otherwise, they are deduced from the sampling density. Finally, we say that (i, j) is an edge of \mathcal{T} if and only if $\mathbf{p}_i \in \mathcal{N}_{\mathbf{p}_j}$ or $\mathbf{p}_j \in \mathcal{N}_{\mathbf{p}_i}$.

Moving least-squares surfaces. Given a point set \mathcal{P} , the moving least-squares (MLS, for short) surface \mathcal{M} implied by \mathcal{P} is

defined by a projection operator $\psi_{\mathcal{P}} : \mathbb{R}^3 \rightarrow \mathcal{M}$, which maps an arbitrary point \mathbf{p} onto \mathcal{M} (3; 5; 6; 43). In particular, each point of \mathcal{M} is a stationary point of $\psi_{\mathcal{P}}$; i.e., $\mathbf{p} \in \mathcal{M}$ implies $\psi_{\mathcal{P}}(\mathbf{p}) = \mathbf{p}$. In this paper, we use the explicit definition (5; 6) of the MLS surface in terms of the critical points of an energy function along lines induced by a vector field. More precisely, the energy function $e : \mathbb{R}^3 \times \mathbb{S}^2$ is defined as

$$e(\mathbf{x}, \mathbf{a}) := \sum_{i=1}^n |\langle \mathbf{p}_i - \mathbf{x}, \mathbf{a} \rangle|^2 \theta(\|\mathbf{x} - \mathbf{p}_i\|_2),$$

i.e., the sum of weighted squared distances of points in \mathcal{P} to the plane given by \mathbf{x} and the normal vector \mathbf{a} of the unit sphere \mathbb{S}^2 . Here, θ is a decreasing weighting function, e.g. $\theta(t) := \exp(-t^2/h^2)$, where h is a Gaussian scale parameter that defines the width of the kernel. A possible choice of h is $h := \sigma/\sqrt{3}$, where σ is the distance between \mathbf{p}_i and the points of its k -nearest neighborhood; for more details on the choice of h , we refer the reader to (26). Then, for each point \mathbf{x} the un-oriented normal field is defined as $\mathbf{n}(\mathbf{x}) := \arg \min_{\|\mathbf{a}\|_2=1} e(\mathbf{x}, \mathbf{a})$ and it is the unit eigenvector related to the smallest eigenvalue of the 3×3 symmetric covariance matrix $\mathbf{C} := (C_{ij})_{i,j=1}^3$ of the directions in \mathbf{x} . The entries C_{ij} of \mathbf{C} are defined as

$$C_{ij} := \sum_{l=1}^k (p_l^i - x^i)(p_l^j - x^j) \theta(\|\mathbf{x} - \mathbf{p}_l\|_2), \quad i, j = 1, 2, 3,$$

with $\mathbf{p}_l := (p_l^1, p_l^2, p_l^3)$ and $\mathbf{x} := (x^1, x^2, x^3)$. Successively, the normal vectors are coherently oriented by imposing that the angles between the normal at \mathbf{x} and at the points that fall in a neighborhood $\mathcal{N}_{\mathbf{x}}$ of \mathbf{x} are less than $\pi/2$. The plane $H_{\mathbf{x}} : \langle \mathbf{p} - \mathbf{x}, \mathbf{n}(\mathbf{x}) \rangle = 0$, $\mathbf{p} \in \mathbb{R}^3$, through \mathbf{x} and orthogonal to $\mathbf{n}(\mathbf{x})$ is called *reference plane* and is used to compute a local bivariate polynomial approximation of the surface. If the covariance matrix of the directions in \mathbf{x} is not singular, then the vector $\mathbf{n}(\mathbf{x})$ is uniquely defined. Under this assumption (74), the point \mathbf{x} belongs to \mathcal{M} and satisfies the implicit equation

$$g(\mathbf{x}) := \sum_{i=1}^n \left[1 - \frac{|\langle \mathbf{x} - \mathbf{p}_i, \mathbf{n}(\mathbf{x}) \rangle|^2}{h^2} \right] \langle \mathbf{x} - \mathbf{p}_i, \mathbf{n}(\mathbf{x}) \rangle \times \theta(\|\mathbf{x} - \mathbf{p}_i\|_2) = 0, \quad (1)$$

where θ is the Gaussian kernel. This expression of g has been used to compute the curvature of point-sampled surfaces and will be exploited to trace the level-sets of a scalar function defined on \mathcal{P} . Changing either the energy function or the vector field provides variants of the MLS surface. For instance, the RMLS variant (30) preserves sharp features of \mathcal{M} , which are commonly removed by the Gaussian weighting.

Previous work on MLS surfaces indirectly assumes that the normal of the approximating tangent frame is the surface normal. Since this assumption is generally not satisfied, (1) proposes a different expression of the implicit function g that allows us to exactly compute the surface normals. More precisely, the implicit function g that defines the surface \mathcal{M} as the zero-set $\{\mathbf{x} : g(\mathbf{x}) = 0\}$ is

$$g(\mathbf{x}) := \langle \mathbf{x} - \mathbf{b}(\mathbf{x}), \mathbf{n}(\mathbf{x}) \rangle, \quad (2)$$

where $\mathbf{n}(\mathbf{x})$ is the oriented normal at \mathbf{x} , which is computed as previously discussed, and $\mathbf{b}(\mathbf{x})$ is the weighted average of points at \mathbf{x} , i.e. $\mathbf{b}(\mathbf{x}) := \frac{\sum_{i=1}^n \theta(\|\mathbf{x} - \mathbf{p}_i\|_2) \mathbf{p}_i}{\sum_{i=1}^n \theta(\|\mathbf{x} - \mathbf{p}_i\|_2)}$.

Surface approximation with radial basis functions. Choosing a point $\mathbf{p}_i \in \mathcal{P}$, implicit modeling (20) provides an alternative approximation of \mathcal{M} in $\mathcal{N}_{\mathbf{p}_i}$ by defining an implicit function $g : \mathbb{R}^3 \rightarrow \mathbb{R}$ such that in $\mathcal{N}_{\mathbf{p}_i}$ the following relation holds: $\mathbf{x} \in \mathcal{M}$ if and only if $g(\mathbf{x}) = 0$. In this context, implicit approximation techniques (8; 29; 48; 58; 63) compute $g(\mathbf{x}) := \sum_{i=1}^k \alpha_i \varphi_i(\mathbf{x})$ as a linear combination of the basis elements $\mathcal{B} := \{\varphi_i(\mathbf{x}) := \varphi(\|\mathbf{x} - \mathbf{p}_i\|_2)\}_{i=1}^k$, where φ is the kernel function. Depending on the properties of φ and of the corresponding approximation scheme, we distinguish globally (21; 70) and compactly (51; 54; 71) supported radial basis functions (RBFs, for short), and the partition of unity (53; 72). Therefore, φ is not necessarily the Gaussian kernel; for instance, we can choose the bi-harmonic kernel $\varphi(t) := |t/h|^3$.

Assuming that $\mathcal{N}_{\mathbf{p}_i} := \{\mathbf{p}_{j_s}, s = 1, \dots, k\}$, in the following we compute g as done in (51; 68; 70). To avoid the trivial solution $g \equiv 0$, we add a positive- and negative-valued normal constraint at \mathbf{p}_{j_s} , close to the boundary constraint $g(\mathbf{p}_{j_s}) = 0$, and in the normal directions $\mathbf{n}(\mathbf{p}_{j_s})$ and $-\mathbf{n}(\mathbf{p}_{j_s})$, $s = 1, \dots, k$, respectively. Then, the function g is defined as a linear combination of $3k$ radial basis functions, whose centers belong to $\mathcal{N}_{\mathbf{p}_i} \cup \{\mathbf{p}_{j_s} \pm \delta \mathbf{n}(\mathbf{p}_{j_s})\}_{s=1}^k$, where δ is an off-set value proportional to the diagonal of the bounding box of \mathcal{P} . The coefficients of the combination are computed by imposing the interpolating conditions $g(\mathbf{p}_{j_s}) = 0$, $g(\mathbf{p}_{j_s} \pm \delta \mathbf{n}(\mathbf{p}_{j_s})) = \pm 1$, $s = 1, \dots, k$, and solving a $3k \times 3k$ linear system. For more details on the construction of g , we refer the reader to Section 3.2. Clearly, positive- and negative-valued constraints can be chosen in a subset of $\{\mathbf{p}_{j_s} \pm \delta \mathbf{n}(\mathbf{p}_{j_s})\}_{s=1}^k$, thus reducing the size of the corresponding linear system. For instance, a variation of this scheme is to center the basis elements only at the points $\{\mathbf{p}_{j_s}\}_{s=1}^k$ and impose that the resulting function g satisfies the aforementioned $3k$ conditions in a least-squares sense. In this case, we solve a $k \times k$ linear system and the least-squares formulation is preferable to the implicit interpolation in those cases where \mathcal{P} is affected by noise.

2.2. Geodesics, harmonic functions, and Laplacian eigenfunctions on point sets

Given a point set $\mathcal{P} := \{\mathbf{p}_i, i = 1, \dots, n\}$, there are roughly two distinct categories of functions defined on \mathcal{P} : *volume- and surface-based scalar functions*. In the former case, the values of f on \mathcal{P} are computed by sampling an implicit function $W : \mathcal{V} \supseteq \mathcal{P} \rightarrow \mathbb{R}$, defined on a region \mathcal{V} that contains \mathcal{P} . Main examples are the height function with respect to a given direction \mathbf{N} , i.e. $W(\mathbf{p}) := \langle \mathbf{p}, \mathbf{N} \rangle$; the Euclidean distance from a point \mathbf{b} , i.e. $W(\mathbf{p}) := \|\mathbf{p} - \mathbf{b}\|_2$; the distance from a reference plane, i.e. $W(\mathbf{p}) := |\langle \mathbf{p} - \mathbf{b}, \mathbf{N} \rangle|$. Therefore, we have that $f := W|_{\mathcal{P}}$. In the latter case, f is defined directly on \mathcal{P} ; in the following, we discuss the definition of the geodesic distance from a set of source points of \mathcal{P} , the harmonic functions, and

the Laplacian eigenfunctions. A generally low number of critical points, the regular variation of the shape of the level-sets, and a smooth transition among them (34; 49; 52) make those functions a natural choice as f for several applications such as quadrilateral remeshing (27) and shape matching (39).

Geodesic functions. Recent works (46; 65) on the computation of geodesics on a point set \mathcal{P} have enriched the class of scalar functions on \mathcal{P} with geodesics-based maps, previously defined on triangle meshes (38) and used for shape comparison (25; 47). For instance, in (65) piecewise linear approximations of geodesic paths on point-sampled surfaces are computed by minimizing an energy function, which takes into account both the geodesic path length and its closeness to the underlying surface. An alternative is to trace the shortest path among the nodes of an extended sphere-of-influence graph.

Harmonic functions and Laplacian eigenfunctions. To define harmonic scalar functions on a point set \mathcal{P} , we remind the relation between the Laplace Beltrami operator for differentiable functions on the 2-manifold surface \mathcal{M} and the heat flow problem $(\partial_t + \Delta)u(\mathbf{p}, t) = 0$. Here, $u(\mathbf{p}, 0) = f(\mathbf{p})$, $\mathbf{p} \in \mathcal{M}$, is the initial heat distribution. Then, on the point set \mathcal{P} the Laplacian matrix $\mathbf{L} := (L_{ij})_{i,j=1}^n$ is defined as (14; 16)

$$L_{ij} := \begin{cases} -1 & i = j, \\ a_{ij}/\alpha_i & \mathbf{p}_j \in \mathcal{N}_{\mathbf{p}_i}, \\ 0 & \text{else,} \end{cases} \quad \begin{cases} a_{ij} := \exp\left(-\frac{\|\mathbf{p}_i - \mathbf{p}_j\|_2^2}{h^2}\right), \\ \alpha_i := \sum_{j \in \mathcal{N}_{\mathbf{p}_i}} a_{ij}. \end{cases} \quad (3)$$

As shown in (16; 23), if the number of samples n tends to infinity and the kernel width h goes to zero then the eigenvectors of the Laplacian matrix tend to those of the Schroedinger operator $\Delta + E$, where E is a scalar potential that depends on the density of \mathcal{P} . We briefly remind that the vector \mathbf{h} , $\mathbf{h} \neq \mathbf{0}$, is an *eigenvector* of \mathbf{L} related to the *eigenvalue* λ if and only if $\mathbf{L}\mathbf{h} = \lambda\mathbf{h}$; in this case, \mathbf{h} is also called *eigenmap* on \mathcal{P} . To reduce the dependency of the Laplacian eigenmaps representation from the density of the data points, (41) suggested to normalize the Gaussian weights with an estimate of the point density and computed the Laplacian matrix with respect to these new weights. Therefore, the new Laplacian matrix that replaces (3) is built in two phases as follows

$$\tilde{L}_{ij} := \begin{cases} \frac{a_{ij}}{\alpha_i \alpha_j} & \mathbf{p}_j \in \mathcal{N}_{\mathbf{p}_i}, \\ 0 & \text{else,} \end{cases} \quad L_{ij} := \begin{cases} -1 & i = j, \\ \frac{\tilde{L}_{ij}}{\sum_{k \in \mathcal{N}_{\mathbf{p}_i}} \tilde{L}_{ik}} & \mathbf{p}_i \in \mathcal{N}_{\mathbf{p}_i}, \\ 0 & \text{else.} \end{cases}$$

In this case, in the limit of large sampled points and small scales the eigenvectors of the new Laplacian matrix converge to those of the Laplace-Beltrami operator on \mathcal{M} . Once \mathbf{L} has been built, the computation of the harmonic scalar function resembles the case of triangle meshes (28; 33; 52). An alternative discretization of the Laplacian matrix is described in (42). Choosing a set of boundary conditions $\mathcal{B} := \{f(\mathbf{p}_i) = a_i\}_{i \in \mathcal{I}}$, $\mathcal{I} \subseteq \{1, \dots, n\}$,

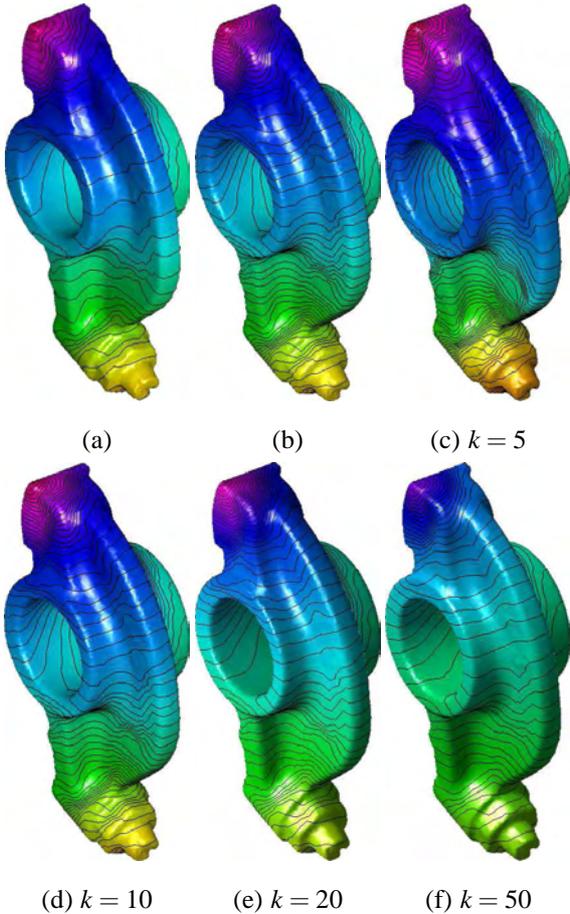


Fig. 2. Level-sets of harmonic scalar functions achieved by imposing the same boundary constraints and using (a) Gaussian (14) and (b) diffusion (41) weights. In both examples, we have chosen the size of the k -nearest neighborhood of each point as $k = 20$. (c-f) Level-sets of harmonic scalar functions with the same boundary constraints and a different k -neighborhood. In this example, the Laplacian matrix entries are the diffusion weights. We used one maximum and minimum as boundary constraints; note that the number of critical points, i.e. one maximum, one minimum, and two saddle points, remains constant in each discretization.

we solve the linear system $\mathbf{L}^* \mathbf{f}^* = \mathbf{b}$, where $\mathbf{f}^* := (f(\mathbf{p}_i))_{i \in \mathcal{I}^c}$ is the vector of unknowns, \mathcal{I}^c is the complementary set of \mathcal{I} , \mathbf{b} is a constant vector, and \mathbf{L}^* is achieved by canceling the i^{th} -row and i^{th} -column of \mathbf{L} , $i \in \mathcal{I}$ (Fig. 2(a,b)).

The *maximum principle* provides the main motivation to define harmonic functions on point sets. In fact, once we have fixed the Dirichlet boundary conditions this principle allows us to build functions with a minimal (i.e., one maximum, one minimum, and $2g$ saddles) or a pre-defined number of critical points. We briefly remind that the *critical points* of a smooth function $g : \mathcal{M} \rightarrow \mathbb{R}$, defined on a surface \mathcal{M} , are the solution of the equation $\nabla g(\mathbf{p}) = \mathbf{0}$, $\mathbf{p} \in \mathcal{M}$, and correspond to the *maxima*, *minima*, and *saddles* of g . For more details, we refer the reader to Section 5. In the case that all constrained minima are assigned the same global minimum value and all constrained maxima are assigned the same global maximum value, all the constraints will be guaranteed to be extrema in the resulting function. If there is not a predefined choice of the Dirichlet boundary conditions, then the Laplacian eigenfunctions pro-

vide an alternative to harmonic functions. Laplacian eigenfunctions still guarantee a low number of critical points, which are not clustered, and a smooth behavior of their level-sets on \mathcal{M} . In fact, the eigenvectors related to the smallest eigenvalues are smooth and slowly varying functions, while the eigenmaps corresponding to the largest eigenvalues show rapid oscillations and a higher number of critical points.

In Fig. 2, we have computed several harmonic functions by fixing the same Dirichlet boundary conditions and using a different k -nearest neighbor graph for the discretization of the Laplacian matrix. Tests in Fig. 2(a,b) show that the iso-contours calculated with the diffusion weights (41) are regularly distributed on the input surface and smoother than those provided by the Gaussian weights (14). As shown in Fig. 2(c-f), the level-sets related to the diffusion weights become smoother while increasing the parameter k .

3. Scalar functions on point-sampled surfaces

Given a scalar function f , we define the map $\tilde{f} : \mathcal{M} \rightarrow \mathbb{R}$, underlying $f : \mathcal{P} \rightarrow \mathbb{R}$, as the function that locally and smoothly interpolates or approximates the sampled values $\{f(\mathbf{p}_i), \mathbf{p}_i \in \mathcal{P}\}$. Note that if f is a volume-based scalar function and W is known (Section 2.2), then $\tilde{f} := W|_{\mathcal{M}}$ is already the function underlying f . However, the family of surface-based scalar functions includes important maps, such as geodesic, harmonic, and Laplacian functions, which are not associated with explicit underlying maps. For each of those functions, we use \tilde{f} to trace its level-sets on \mathcal{M} .

To build \tilde{f} , we discuss three methods: the first two approaches (Section 3.1) are based on the moving least-squares approach and the third one (Section 3.2) builds on implicit approximation. In all the aforementioned cases, the differential properties of \tilde{f} are derived analytically, thus allowing us to provide an exact differential analysis of \tilde{f} . Furthermore, the approximation scheme is local, i.e. the definition of $\tilde{f}(\mathbf{p})$, $\mathbf{p} \in \mathcal{M}$, is influenced only by those points of \mathcal{P} that belong to a neighborhood of \mathbf{p} . Finally, we analyze and compare the approximation errors of the f -values induced by the moving least-squares approximation and the implicit interpolation (Section 3.3). For the discussion of the numerical stability and approximation accuracy of the aforementioned methods, we refer the reader to Section 6.

3.1. Approximating scalar functions via moving least-squares modeling

Given $f : \mathcal{P} \rightarrow \mathbb{R}$, in the following we describe two methods for the computation of the scalar function \tilde{f} underlying f and based on the moving least squares approach. The *parameterized MLS approach* defines $\tilde{f}(\mathbf{p})$ through the approximation of the f -values in a neighborhood $\mathcal{N}_{\mathbf{p}}$ of \mathbf{p} and with respect to a local parameterization of f on a reference domain. The *explicit MLS approach* directly computes the \tilde{f} -values in $\mathcal{N}_{\mathbf{p}}$.

Parameterized MLS approach. Let $\mathcal{N}_{\mathbf{p}} := \{\mathbf{p}_{j_s}, s = 1, \dots, k\}$ be the k -nearest neighborhood of \mathbf{p} . Then, we define $\tilde{f}(\mathbf{p})$ by

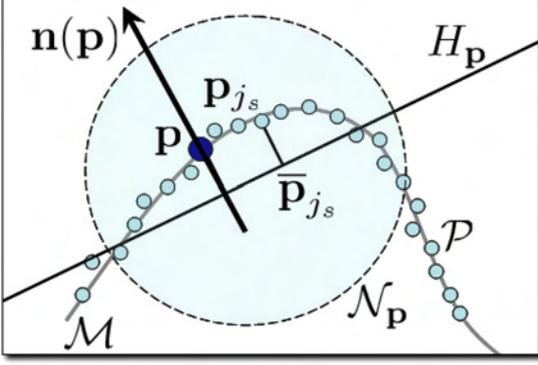


Fig. 3. Computation of the scalar function $\tilde{f}: \mathcal{M} \rightarrow \mathbb{R}$ underlying $f: \mathcal{P} \rightarrow \mathbb{R}$ in a neighborhood \mathcal{N}_p of a point \mathbf{p} with respect to the reference plane H_p at \mathbf{p} (c.f. Equation (4)). Here, \mathcal{M} is the surface underlying \mathcal{P} .

using the local reference domain H_p as follows. Indicating with $\bar{\mathbf{p}}_{j_s}$ the orthogonal projection of $\mathbf{p}_{j_s} \in \mathcal{N}_p$, $s = 1, \dots, k$, onto H_p , we compute a polynomial approximation r that minimizes the weighted least-squares error (Fig. 3)

$$E(\alpha_1, \dots, \alpha_m) := \sum_{s=1}^k |r(x_s, y_s) - f(\mathbf{p}_{j_s})|^2 \theta(\|\mathbf{p} - \mathbf{p}_{j_s}\|_2), \quad (4)$$

where (x_s, y_s) is the representation of $\bar{\mathbf{p}}_{j_s}$ in a local coordinate system in H_p . Since the functional (4) is quadratic and its unknowns are the m coefficients of the polynomial r , we get that the minimum of (4) is achieved by solving a $m \times m$ linear system. More precisely, we write the polynomial function in (4) as

$$r(x, y) := \sum_{l=1}^m \alpha_l b_l(x, y), \quad \alpha := (\alpha_l)_{l=1}^m,$$

where $\{b_l(x, y)\}_{l=1}^m$ is a basis of the linear space of the polynomials of degree at last d in the variables x and y ; then, $m = \frac{(d+1)(d+2)}{2}$. The minimum of the function (4) is the solution of the linear system

$$\begin{cases} \partial_{\alpha_j} E = 2 \sum_{s=1}^k (r(x_s, y_s) - f(\mathbf{p}_{j_s})) b_j(x_s, y_s) \theta(\|\mathbf{p} - \mathbf{p}_{j_s}\|_2) = 0, \\ j = 1, \dots, m, \end{cases}$$

with respect to the unknowns α_l , $l = 1, \dots, m$. A direct computation shows that the j^{th} equation can be written as

$$\sum_{l=1}^m \left(\sum_{s=1}^k b_l(x_s, y_s) b_j(x_s, y_s) \theta_s(\mathbf{p}) \right) \alpha_l = \sum_{s=1}^k f(\mathbf{p}_{j_s}) b_j(x_s, y_s) \theta_s(\mathbf{p}),$$

with $\theta_s(\mathbf{p}) := \theta(\|\mathbf{p} - \mathbf{p}_{j_s}\|_2)$, $s = 1, \dots, k$. This is equivalent to solve the $m \times m$ linear system

$$\mathbf{B} \Theta \mathbf{B}^T \alpha = \mathbf{B} \Theta \mathbf{f}, \quad (5)$$

where \mathbf{B}^T is the transpose of \mathbf{B} ,

$$\begin{cases} \mathbf{B} := (b_i(x_s, y_s))_{i=1, \dots, m}^{s=1, \dots, k} \in \mathbb{R}^{m \times k}, \\ \Theta := \text{diag}(\theta_1(\mathbf{p}), \dots, \theta_k(\mathbf{p})) \in \mathbb{R}^{k \times k}, \\ \mathbf{f} := (f(\mathbf{p}_{j_s}))_{s=1}^k \in \mathbb{R}^{k \times 1}, \end{cases}$$

and the coefficient matrix of (5) is symmetric and positive definite. As observed in (3), the choice of a low degree polynomial r

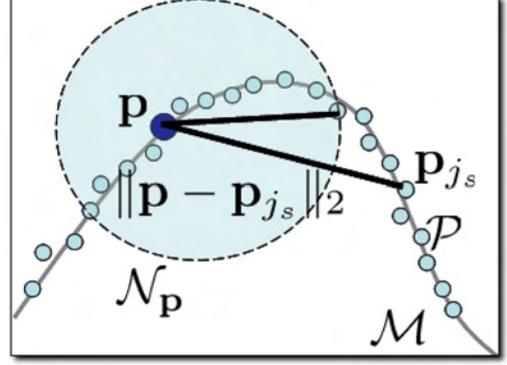


Fig. 4. Computation of the scalar function $\tilde{f}: \mathcal{M} \rightarrow \mathbb{R}$ underlying $f: \mathcal{P} \rightarrow \mathbb{R}$ in a neighborhood \mathcal{N}_p of a point \mathbf{p} with respect to the formulation (6). Here, \mathcal{M} is the surface underlying \mathcal{P} .

and its least-squares definition guarantee a fast approximation of f in \mathcal{N}_p and a smooth behavior without oscillations. For instance, if r is a polynomial of degree 3 or 4, then $m = 10$ or $m = 15$ respectively. If $f(\mathbf{p}_{j_s}) := \langle \mathbf{p}_{j_s} - \mathbf{p}, \mathbf{n}(\mathbf{p}) \rangle$ is the signed distance of \mathbf{p}_{j_s} from H_p , $s = 1, \dots, k$, then the function r is the local parameterization of the MLS surface \mathcal{M} with respect to the reference plane. However, in this paper f is arbitrary.

Once r has been computed, we define the values of F at $\mathbf{q} \in \mathcal{N}_p$ as $F(\mathbf{q}) := r(\bar{\mathbf{q}})$, where $\bar{\mathbf{q}}$ is the orthogonal projection of \mathbf{q} on H_p . Then, in \mathcal{N}_p we set $\tilde{f}(\mathbf{q}) := F(\mathbf{q})$, $\mathbf{q} \in \mathcal{M}$. The following properties of F in (4) motivate our definition.

- *Locality*. The sum in (4) considers only those points of \mathcal{P} that belong to the k -nearest neighborhood of each point and that have been already computed to define the MLS surface underlying \mathcal{P} .
- *Least-squares property*. The definition of F guarantees the robustness of F to noisy f -values.
- *Continuity of f with respect to the sampling density*. Since the coefficients of r are the solutions of a linear system, which continuously depend on the input data, we have that if the kernel width h tends to zero then $\lim_{h \rightarrow 0} \tilde{f}(\mathbf{p}_i) = f(\mathbf{p}_i)$, $i = 1, \dots, n$.
- *Gradient field and higher order derivatives*. Once the polynomial function r has been computed, we can analytically evaluate its gradient vector and Hessian matrix by deriving r with respect to the two variables (x, y) in the local frame \mathcal{F} associated to H_p . Then, we apply to this vector and matrix the rotation that maps \mathcal{F} to the canonical reference frame in \mathbb{R}^3 .

Explicit MLS approach. An alternative approach is to define \tilde{f} directly, without using the reference plane (Fig. 4). More precisely, we consider the implicit function

$$F(\mathbf{p}) := \frac{\sum_{i=1}^n f(\mathbf{p}_i) \theta(\|\mathbf{p} - \mathbf{p}_i\|_2)}{\sum_{i=1}^n \theta(\|\mathbf{p} - \mathbf{p}_i\|_2)}, \quad \mathbf{p} \in \mathbb{R}^3, \quad (6)$$

and \tilde{f} is defined as $\tilde{f} := F|_{\mathcal{M}}$. Therefore, $F(\mathbf{p})$ is the weighted average of the f -values at a location \mathbf{p} and the influence of the noise on the approximation of f is smoothed by the Gaussian kernel. Note the analogy between the definition (6) of F and the weighted average $\mathbf{b}(\mathbf{x})$ of points at \mathbf{x} in (2), where the points \mathbf{p}_i

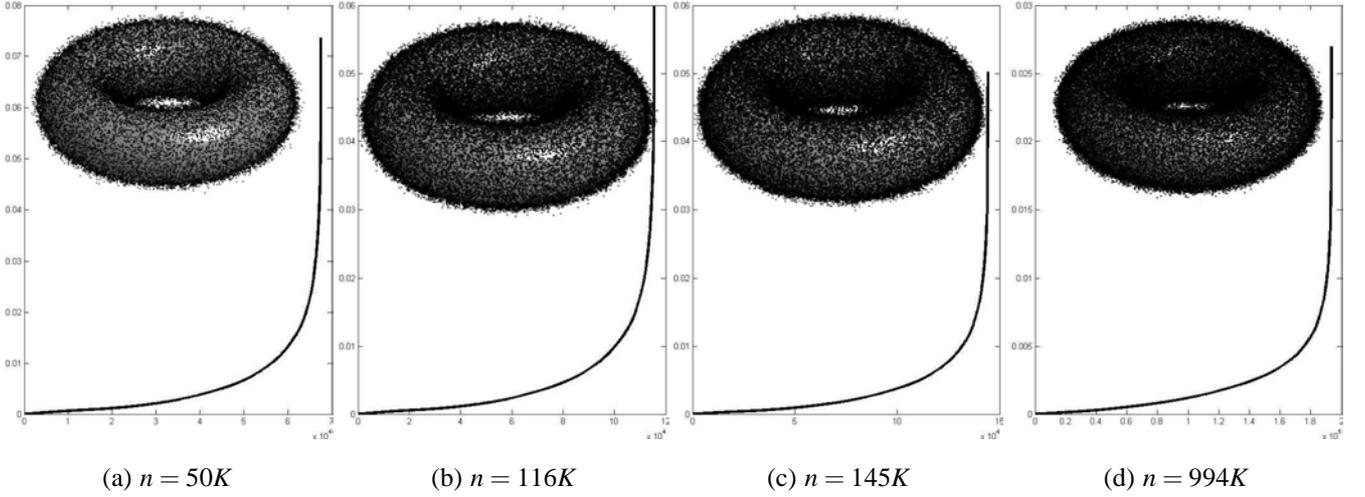


Fig. 5. (a-d) Given four noisy point sets \mathcal{P}_{noise} with a different number n of vertices and sampling density ε , we have evaluated the l_∞ -error between the input function f on \mathcal{P}_{noise} and its approximation $F|_{\mathcal{P}_{noise}}$. Here, F is given by (6) and σ is the kernel width. The noisy surface is defined as $\mathcal{P}_{noise} = \mathcal{P} + \mathcal{G}$, where \mathcal{P} is a set of points sampled on the torus surface and \mathcal{G} is the noise displacement, which has a normal distribution with mean zero and standard deviation one. The function f has been achieved by evaluating the function $W(\mathbf{p}) = \log(\|\mathbf{p}\|_2 + 1) + x^2 - y^2$, $\mathbf{p} := (x, y, z)$, at the points of \mathcal{P}_{noise} . Numerical values are reported in Table 1; see also Fig. 6.

Table 1

The table reports the parameters of the example shown in Fig. 5.

Test	#Vertices n	Kernel width σ	Sampling density ε^{-1}	l_∞ -error
Fig. 5(a)	50K	2.52×10^{-3}	7.93×10^{-5}	7.36×10^{-2}
-	90K	1.97×10^{-3}	6.21×10^{-5}	6.65×10^{-2}
Fig. 5(b)	116K	1.95×10^{-3}	6.32×10^{-5}	5.98×10^{-2}
Fig. 5(c)	145K	4.02×10^{-4}	7.11×10^{-5}	5.02×10^{-2}
-	160K	4.52×10^{-4}	6.15×10^{-5}	2.54×10^{-2}
Fig. 5(d)	994K	1.265×10^{-3}	3.97×10^{-5}	2.69×10^{-2}

in $\mathbf{b}(\mathbf{x})$ have been replaced by the f -values at these points. The following properties of F in (6) motivate our definition.

- *Linearity.* Let F (resp., H) be the scalar function underlying the map f (resp., h) on the point set \mathcal{P} and defined by (6); then, $\alpha F + \beta H$ is the function underlying $\alpha f + \beta h$, $\forall \alpha, \beta \in \mathbb{R}$.
- *Locality.* Since the weight function θ is close to zero in a certain distance from \mathbf{p} , the sum in (6) can be reduced to the indices of the points of \mathcal{P} that belong to the k -nearest neighborhood of \mathbf{p} .
- *Point-wise variation.* From the definition of \tilde{f} , it follows that the maximum variation between f and \tilde{f} on \mathcal{P} is bounded by the maximum of the f -value on \mathcal{P} , i.e.

$$\|\mathbf{f} - \tilde{\mathbf{f}}\|_\infty \leq \|\mathbf{f}\|_\infty, \quad \mathbf{f} := (f(\mathbf{p}_i))_{i=1}^n, \quad \tilde{\mathbf{f}} := (\tilde{f}(\mathbf{p}_i))_{i=1}^n,$$

where \mathbf{f} (resp., $\tilde{\mathbf{f}}$) is the vector of the f -values (resp., \tilde{f} -values). Here, the l_∞ -norm is defined as $\|\mathbf{f}\|_\infty := \max_{i=1, \dots, n} \{|f(\mathbf{p}_i)|\}$. Furthermore, we have that $\|\tilde{\mathbf{f}} - \mathbf{f}\|_\infty$ is bounded by the maximum variation of the f -values, i.e.

$$\begin{aligned} |\tilde{f}(\mathbf{p}_j) - f(\mathbf{p}_j)| &\leq \frac{\sum_{i=1}^n |f(\mathbf{p}_i) - f(\mathbf{p}_j)| \theta(\|\mathbf{p}_i - \mathbf{p}_j\|_2)}{\sum_{i=1}^n \theta(\|\mathbf{p}_i - \mathbf{p}_j\|_2)} \\ &\leq \max_{i \neq j} \{|f(\mathbf{p}_i) - f(\mathbf{p}_j)|\}. \end{aligned}$$

- *Bound on the global variation of F .* The F -values are bounded by the l_1 - and l_∞ -norm of \mathbf{f} ; in fact, we

have that $\frac{\theta(\|\mathbf{p} - \mathbf{p}_i\|_2)}{\sum_{j=1}^n \theta(\|\mathbf{p} - \mathbf{p}_j\|_2)} \leq 1$, $i = 1, \dots, n$, and therefore $\|F\|_\infty \leq \|\mathbf{f}\|_1$, $\|F\|_\infty \leq \|\mathbf{f}\|_\infty$. Here, the l_1 -norm is defined as $\|\mathbf{f}\|_1 := \sum_{i=1}^n |f(\mathbf{p}_i)|$.

- *Continuity of \tilde{f} with respect to the sampling density.* If the sampling density, and therefore the kernel width h , tends to zero, then the \tilde{f} -values at \mathcal{P} converge to the corresponding f -values, i.e. $\lim_{h \rightarrow 0} \tilde{f}(\mathbf{p}_i) = f(\mathbf{p}_i)$, $i = 1, \dots, n$.
- *Gradient field and higher order derivatives.* Deriving (6), we analytically compute the gradient field of F , which will be used in Section 5 to define the fuzzy critical points of \tilde{f} on \mathcal{M} . A direct computation shows that the following relation holds

$$\begin{aligned} \nabla F(\mathbf{p}) = & - \frac{2}{[h \sum_{i=1}^n \theta(\|\mathbf{p} - \mathbf{p}_i\|_2)]^2} \left[\left(\sum_{i=1}^n \theta(\|\mathbf{p} - \mathbf{p}_i\|_2) \right) \times \right. \\ & \left(\sum_{i=1}^n f(\mathbf{p}_i) \theta(\|\mathbf{p} - \mathbf{p}_i\|_2) (\mathbf{p} - \mathbf{p}_i) \right) - \\ & \left(\sum_{i=1}^n f(\mathbf{p}_i) \theta(\|\mathbf{p} - \mathbf{p}_i\|_2) \right) \times \\ & \left. \left(\sum_{i=1}^n \theta(\|\mathbf{p} - \mathbf{p}_i\|_2) (\mathbf{p} - \mathbf{p}_i) \right) \right], \quad \mathbf{p} \in \mathbb{R}^3. \end{aligned} \quad (7)$$

The least-squares property guarantees that F is robust to noise; the local property improves the efficiency of the computation of $F(\mathbf{p})$ by reducing the sum in (6) to the indices of the points that belong to the k -nearest neighborhood of \mathbf{p} . The continuity of \tilde{f} with respect to the sampling density implies that \tilde{f} converges to f on \mathcal{P} , $h \rightarrow 0$. Fig. 5, Fig. 6, and Table 1 confirm that the expression (6) is stable to noise; another example is given in Fig. 7. Even though both approaches are defined according to the projection procedure that is behind the moving least-squares methods, they do not guarantee that \tilde{f} interpolates the function values at the points of \mathcal{P} . Therefore, in the following section we discuss a local interpolation scheme,

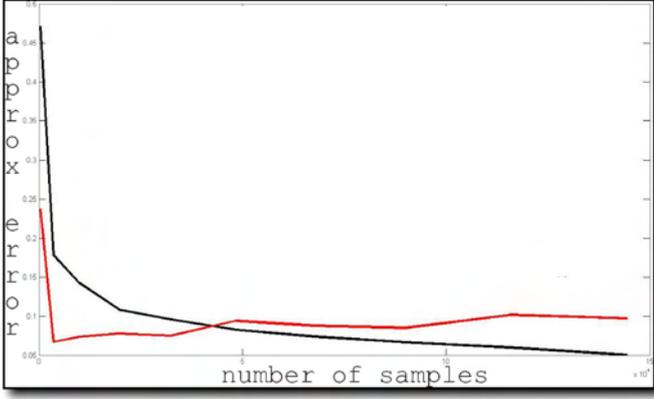


Fig. 6. With reference to Fig. 5 and Table 1, the variation of the l_∞ -error on the various point sets \mathcal{P}_{noise} and \mathcal{P} is shown by the black and red curves respectively. The behavior of the two curves is similar and the error between the input and the approximated f -values remains low while increasing the noise magnitude and the number of sampled points.

which is based on implicit modeling.

3.2. Approximating scalar functions with radial basis functions

In the following, we describe how f is locally approximated by an implicit smooth function $F: \mathbb{R}^3 \rightarrow \mathbb{R}$ on a neighborhood $\mathcal{N}_{\mathbf{p}} := \{\mathbf{p}_{j_1}, \dots, \mathbf{p}_{j_k}\}$ of $\mathbf{p} \in \mathbb{R}^3$, i.e.

$$F(\mathbf{p}_{j_s}) = f(\mathbf{p}_{j_s}), \quad s = 1, \dots, k. \quad (8)$$

Choosing a kernel function $\varphi: \mathbb{R}^+ \rightarrow \mathbb{R}$ (e.g., the Gaussian $\varphi(t) := \exp(-t/h)$ or bi-harmonic $\varphi(t) := |t/h|^3$ kernel), F is the implicit scalar function (63)

$$F(\mathbf{q}) := \sum_{i=1}^k \alpha_i \varphi_i(\mathbf{q}) + \eta(\mathbf{q}), \quad \eta(\mathbf{q}) := \beta_0 + \beta_1 x + \beta_2 y + \beta_3 z, \quad (9)$$

$\mathbf{q} := (x, y, z)$. Therefore, F is defined as a linear combination of the radial basis functions $\varphi_s(\mathbf{q}) := \varphi(\|\mathbf{q} - \mathbf{p}_{j_s}\|_2)$, centered at each point \mathbf{p}_{j_s} , $s = 1, \dots, k$, plus a first-degree polynomial $\eta(\mathbf{q})$. The linear function η avoids that the first term in (9) tries to fit f over regions of \mathcal{P} where f is linear or constant (e.g., at *plateaux*). Therefore, the coefficients in (9) that uniquely satisfy (8) are the solution of the following $(k+4) \times (k+4)$ square linear system

$$\begin{bmatrix} a_{11} & \dots & a_{1k} & 1 & p_{j_1}^x & p_{j_1}^y & p_{j_1}^z \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{k1} & \dots & a_{kk} & 1 & p_{j_k}^x & p_{j_k}^y & p_{j_k}^z \\ 1 & \dots & 1 & 0 & 0 & 0 & 0 \\ p_{j_1}^x & \dots & p_{j_k}^x & 0 & 0 & 0 & 0 \\ p_{j_1}^y & \dots & p_{j_k}^y & 0 & 0 & 0 & 0 \\ p_{j_1}^z & \dots & p_{j_k}^z & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_k \\ \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix} = \begin{bmatrix} f(\mathbf{p}_{j_1}) \\ \vdots \\ f(\mathbf{p}_{j_k}) \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad (10)$$

where $\mathbf{p}_{j_s} := (p_{j_s}^x, p_{j_s}^y, p_{j_s}^z)$, $s = 1, \dots, k$, and the entries of the non-singular matrix $\mathbf{A} := (a_{rs})_{r,s=1, \dots, k}$ are set equal to

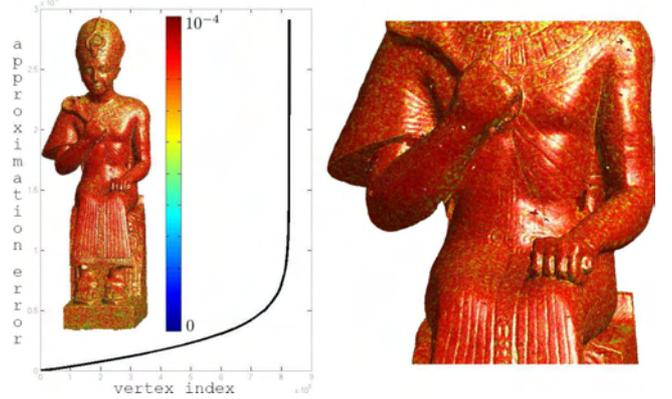


Fig. 7. The picture shows the increasing reordering of the l_∞ -error on a scanned point set \mathcal{P} with several holes and noise; the number of points is $n = 870K$. The corresponding color-map highlights that the approximation error between the input scalar function f and $F|_{\mathcal{P}}$ is equally distributed on \mathcal{P} ; F has been computed as (6).

$a_{rs} := \varphi(\|\mathbf{p}_{j_r} - \mathbf{p}_{j_s}\|_2)$, $r, s = 1, \dots, k$. The last four rows of the coefficient matrix in (10) correspond to the *natural additional constraints*

$$\sum_{s=1}^k \alpha_s p_{j_s}^x = 0, \quad \sum_{s=1}^k \alpha_s p_{j_s}^y = 0, \quad \sum_{s=1}^k \alpha_s p_{j_s}^z = 0.$$

These relations guarantee that the coefficient matrix in (10) is invertible; in fact, the $k \times k$ sub-matrix \mathbf{A} is conditionally positive-definite on the subspace of vectors that are orthogonal to the last four rows of the full matrix. Then, in $\mathcal{N}_{\mathbf{p}}$ we have that \tilde{f} is computed as $\tilde{f}(\mathbf{p}) := F(\mathbf{p})$, $\mathbf{p} \in \mathcal{M}$. We visualize the behavior of (\mathcal{P}, f) around \mathbf{p} by sampling F on a voxel grid centered at \mathbf{p} and extracting the iso-surface of F related to the iso-value $f(\mathbf{p})$, i.e. $\Sigma_{f(\mathbf{p})} := \{\mathbf{q} \in \mathbb{R}^3 : F(\mathbf{q}) = f(\mathbf{p})\}$. In analogy with the local approximation of smooth surfaces, we refer to $\Sigma_{f(\mathbf{p})}$ as *osculating paraboloid* of f related to $f(\mathbf{p})$ (Fig. 8(a-c)). Indeed, the approximation F is useful to make predictions about the phenomenon behavior measured by f on \mathcal{P} . The common way of analyzing the properties of f is to visualize the evolution of its level-sets and can be enhanced by adding also the visualization of the iso-surfaces of F in a neighborhood of \mathbf{p} . Deriving (9), we have that

$$\nabla F(\mathbf{p}) = \sum_{i=1}^n \alpha_i \frac{\varphi'(\|\mathbf{p} - \mathbf{p}_i\|_2)}{\|\mathbf{p} - \mathbf{p}_i\|_2} (\mathbf{p} - \mathbf{p}_i) + (\beta_1, \beta_2, \beta_3), \quad \mathbf{p} \in \mathbb{R}^3,$$

where $\varphi'(t) := -\frac{1}{h} \exp(-\frac{t}{h})$ (resp., $\varphi'(t) := \frac{3t^2}{h^3}$) if φ is the Gaussian (resp., bi-harmonic) kernel; therefore, we get

$$\nabla F(\mathbf{p}) = \begin{cases} \frac{1}{h} \sum_{i=1}^n \alpha_i \frac{\exp\left(\frac{\|\mathbf{p} - \mathbf{p}_i\|_2}{h}\right)}{h \|\mathbf{p} - \mathbf{p}_i\|_2} (\mathbf{p} - \mathbf{p}_i) + (\beta_1, \beta_2, \beta_3), & \text{if } \varphi \text{ is the Gaussian kernel,} \\ \frac{3}{h^3} \sum_{i=1}^n \alpha_i \|\mathbf{p} - \mathbf{p}_i\|_2 (\mathbf{p} - \mathbf{p}_i) + (\beta_1, \beta_2, \beta_3), & \text{if } \varphi \text{ is the bi-harmonic kernel.} \end{cases} \quad (11)$$

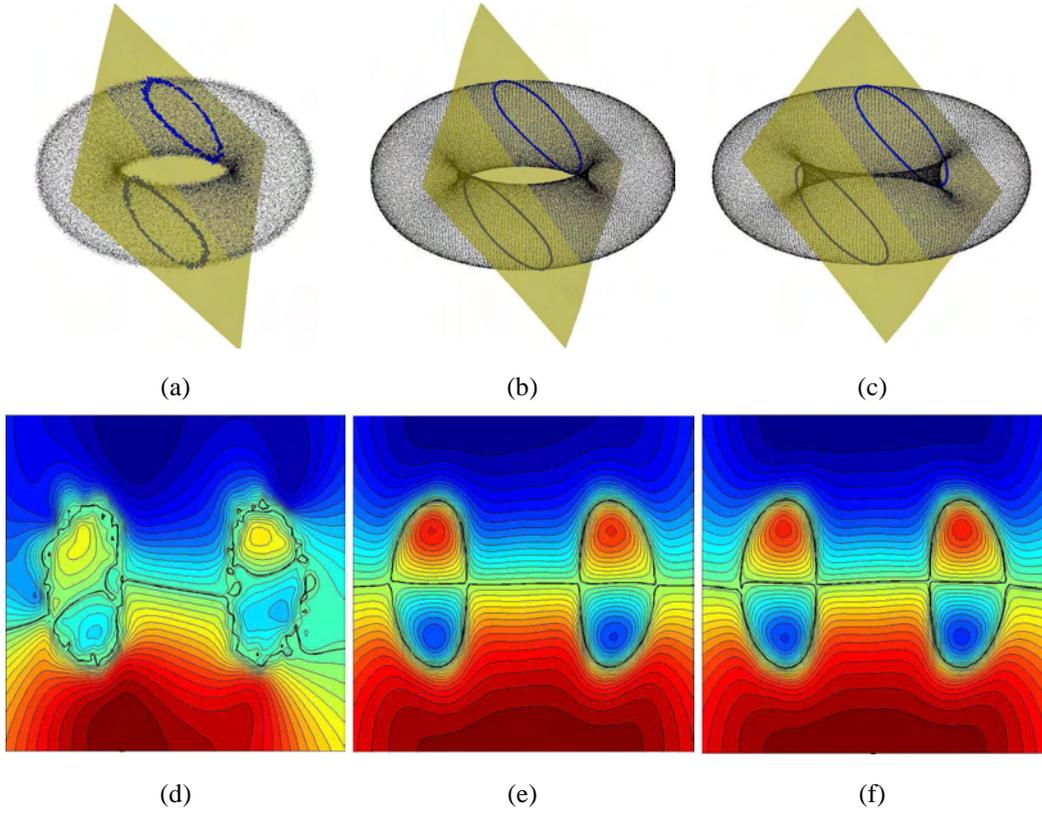


Fig. 8. (a-c) Point sets \mathcal{P} representing three noisy torii; the noise decreases from left to right. Choosing a scalar function f on \mathcal{P} and an iso-value α , we have computed the implicit approximation F , the osculating paraboloid $\Sigma_\alpha := F^{-1}(\alpha)$ (yellow surface), and the set \mathcal{S} of points that belong to Σ_α and to the edges of the k -nearest neighbor graph of \mathcal{P} . (d-f) Level-sets of the restriction of F to the least-squares plane μ that approximates \mathcal{S} . The black curves are the connected components of the contour $F|_\mu^{-1}(\alpha)$ and provide an approximation of the corresponding iso-contour on the surface underlying \mathcal{P} . In each case, the implicit approximation of f is smooth on and around \mathcal{P} ; if the noise magnitude is low (b,c), then $F|_\mu^{-1}(\alpha)$ is a smooth approximation of $f^{-1}(\alpha)$. See also Fig. 9.

The entries of the Hessian matrix are computed in an analogous way; i.e.,

$$\partial_{\mathbf{e}_j \mathbf{e}_j}^2 F(\mathbf{p}) = \sum_{i=1}^n \frac{\alpha_i}{\|\mathbf{p}_i - \mathbf{p}_j\|_2^2} \left[\varphi'(\|\mathbf{p} - \mathbf{p}_i\|_2) \|\mathbf{p} - \mathbf{p}_i\|_2 + \left(\mathbf{e}_j^T (\mathbf{p} - \mathbf{p}_i) \right)^2 \left(\varphi''(\|\mathbf{p} - \mathbf{p}_i\|_2) - \frac{\varphi'(\|\mathbf{p} - \mathbf{p}_i\|_2)}{\|\mathbf{p} - \mathbf{p}_i\|_2} \right) \right], \quad (12)$$

$$\partial_{\mathbf{e}_k \mathbf{e}_j}^2 F(\mathbf{p}) = \sum_{i=1}^n \frac{\alpha_i}{\|\mathbf{p}_i - \mathbf{p}_j\|_2} (\mathbf{e}_k^T (\mathbf{p} - \mathbf{p}_i)) (\mathbf{e}_j^T (\mathbf{p} - \mathbf{p}_i)) \times [\varphi''(\|\mathbf{p} - \mathbf{p}_i\|_2) - \varphi'(\|\mathbf{p} - \mathbf{p}_i\|_2)], \quad k \neq j, \quad (13)$$

where $\partial_{\mathbf{e}_i} F$ is the partial derivative of F with respect to the direction \mathbf{e}_i , $(\mathbf{e}_i)_j := \delta_{ij}$. Here, $\delta_{ij} = 0$ if $i \neq j$ and $\delta_{ii} = 1$, $i, j = 1, \dots, 3$.

Before discussing the properties of the implicit interpolation scheme, we remind that the *support* of $g : \mathbb{R}^3 \rightarrow \mathbb{R}$ is defined as $\text{supp}(g) := \overline{\{\mathbf{p} \in \mathbb{R}^3 : g(\mathbf{p}) \neq 0\}}$, i.e. the closure of the set where g is not null, and the function g has global support if $\text{supp}(g) = \mathbb{R}^3$. If φ has global support, then the corresponding coefficient matrix is full and the solution of the linear system (10) is computed in $O(k \log k)$ -time using specialized techniques such as fast multipole methods (36) for RBFs (21; 22). The choice of a locally-supported kernel re-

quires to solve a sparse linear system in $O(k \log k)$ -time (35) (Ch. 4, 12). Fig. 8(d-f) and Fig. 9(a-c) show the level-sets of the function F restricted to the least-squares plane that contains the level-sets in Fig. 8(a-c). For completeness, we recall that common sparse kernels are $\varphi(t) := (1-t)_+^4 (4t+1)$ (71) and the function $\varphi(t) := (1-t)_+^4 (4+16t+12t^2+3t^3)$ (51; 67), where $t := \frac{\|\mathbf{p} - \mathbf{p}_i\|_2}{h}$ and h is the kernel support. In this case, each kernel belongs to $C^2([0, 1])$ and the corresponding sparse matrix is built using the k -nearest neighbor graph of the input point set. If we select one of the aforementioned kernels, then we update the relations (11), (12), and (13) with the derivatives of the kernel function and the sum is related to the points that belong to the kernel support. Finally, the interpolation scheme with RBFs satisfies the properties listed in Section 3.1. Note that the level-sets in Fig. 9(b) are not nearly as smooth as the level-sets in Fig. 8(e). This is due to the fact that a local noise and a rough sampling density of \mathcal{P} might reduce the accuracy and smoothness of the MLS approximation of the scalar function underlying f with respect to the implicit scheme. We can overcome this problem by enlarging the width h of the kernel function or increasing the sampling density of \mathcal{P} through resampling techniques. In fact, a small value for h results in a fast decay of the Gaussian kernel and the approximation (6) is more local. Large values of h attenuate local oscillation of the f -values, guarantee that F in (6) is a global approximation of

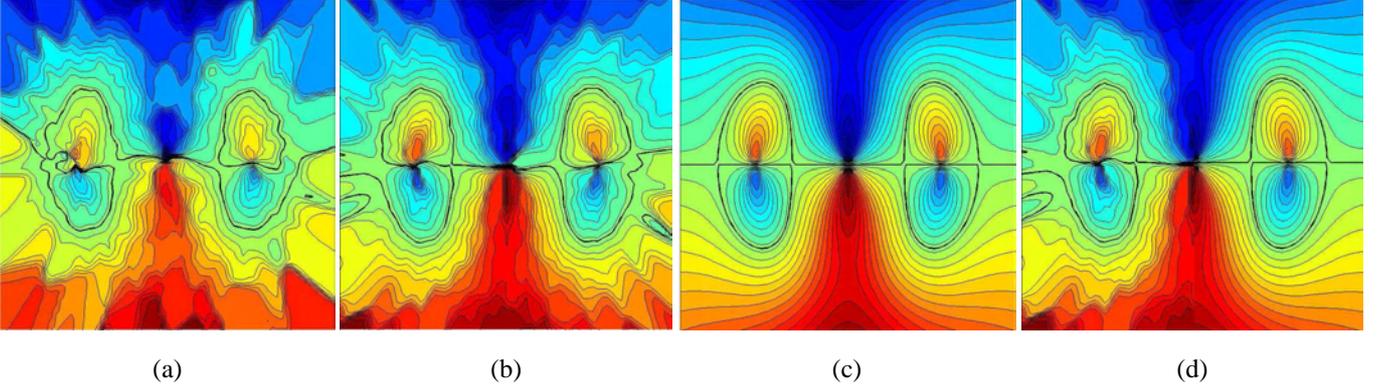


Fig. 9. (a-c) Level-sets of the moving least-squares approximation (6) of f related to the example shown in Fig. 8. Comparing Fig. 8(d-f) with (a-c), we see that the moving least-squares approximation F of f is also smooth. If we decrease the noise magnitude, then the iso-contours related to α have a similar shape. Since the MLS function underlying f does not interpolate the function values, the black curves in (a) are smoother than the corresponding contours in Fig. 8(d). (d) Comparing the left and right side of the image, we see that duplicating the width of the kernel in (b) we get smoother level-sets.

the f -values, and smooth sharp features of the corresponding level-sets. For instance, duplicating the width of the kernel in Fig. 9(b) we get smoother level-sets, which are shown on the right side of Fig. 9(d).

3.3. Moving least-squares versus implicit approximation of scalar functions

In this section, we derive two characterizations of the error between the moving least-squares approximation and the implicit interpolation. For the discussion on the main differences between the MLS and implicit approximation of scalar functions and the choice of the main parameters used by the proposed approaches, we refer the reader to Section 6.2.

Choosing $\mathbf{p}_i \in \mathcal{P}$, let us suppose that F has been computed by using the implicit approach, where as kernel φ we have chosen the decreasing weighting function θ of the moving least-squares scheme (i.e., the Gaussian kernel). Without loss of generality and for simplicity, we omit the linear term η in Equation (9). In this case, we have that the $k \times 1$ vector of the F -values at the points of $\mathcal{N}_{\mathbf{p}_i}$ is given by $(F(\mathbf{p}_{j_s}))_{s=1}^k = (f(\mathbf{p}_{j_s}))_{s=1}^k = \mathbf{b} = \mathbf{A}\boldsymbol{\alpha}$, where $\boldsymbol{\alpha} := (\alpha_i)_{i=1}^k$ is the solution of Equation (10). Let us now compute F by using the moving least-squares formulation in (6); here, we neglect the points that do not belong to $\mathcal{N}_{\mathbf{p}_i}$. In this case, we have that $\mathbf{b}_{MLS} := (F(\mathbf{p}_{j_s}))_{s=1}^k = \mathbf{D}^{-1}\mathbf{A}\mathbf{b}$, where $\mathbf{D} := \text{diag}(\theta_1, \dots, \theta_k)$ is the $k \times k$ diagonal matrix whose non-null entries are $\theta_i := \sum_{s=1}^k \theta(\|\mathbf{p}_i - \mathbf{p}_{j_s}\|_2)$, $i = 1, \dots, k$, \mathbf{A} is the $k \times k$ coefficient matrix in (10), and $\mathbf{b} := (f(\mathbf{p}_{j_s}))_{s=1}^k$ is the set of f -values on the points of $\mathcal{N}_{\mathbf{p}_i}$. From the identity

$$\mathbf{e} := \mathbf{b}_{MLS} - \mathbf{b} = \mathbf{D}^{-1}\mathbf{A}\mathbf{b} - \mathbf{b} = \mathbf{D}^{-1}(\mathbf{A} - \mathbf{D})\mathbf{b},$$

we have that $\mathbf{e} = \mathbf{0}$ if and only if $(\mathbf{A} - \mathbf{D})\mathbf{b} = \mathbf{0}$, i.e. \mathbf{b} belongs to the null-space of the matrix $(\mathbf{A} - \mathbf{D})$. Indeed, without additional overhead we can compute the approximation error as $e_{MLS} := \|\mathbf{e}\|_2 = \|(\mathbf{I} - \mathbf{D}^{-1}\mathbf{A})\mathbf{b}\|_2$ at each point of \mathcal{P} . In this case, we run the implicit scheme only in the neighborhood of those points of \mathcal{P} where e_{MLS} is greater than a given threshold.

4. Level-sets of scalar functions defined on point sets

From a general point of view, a contouring algorithm of a scalar function defined on a triangle mesh traces the level-set γ_α , related to the iso-value α , starting from a seed point $\mathbf{p} \in \gamma_\alpha$, evaluating the intersection points between γ_α and the edges of \mathcal{M} , and iterating the intersection search through the triangles adjacent to the intersected edges. Given an arbitrary scalar function $f : \mathcal{P} \rightarrow \mathbb{R}$ on a point set, we mimic this approach to trace the level-set $\gamma_\alpha := \{\mathbf{p} \in \mathcal{M} : \tilde{f}(\mathbf{p}) = \alpha\}$, which belongs to \mathcal{M} . The algorithm is summarized by the following steps:

- (i) computation of a point-wise approximation \mathcal{S} of γ_α by searching those points \mathbf{q} such that $f(\mathbf{q}) = \alpha$ on the edges of \mathcal{T} . To this end, we interpolate the f -values along the edges of \mathcal{T} with linear or radial basis functions. Roughly speaking, we say that γ_α intersects the edges of \mathcal{T} . Then, each point $\mathbf{q} \in \mathcal{S}$ is “projected” onto a point $\bar{\mathbf{q}} := \text{pr}(\mathbf{q})$ of γ_α and the set $\mathcal{C} := \{\bar{\mathbf{q}} := \text{pr}(\mathbf{q}) : \mathbf{q} \in \mathcal{S}\}$ gives a discrete sampling of γ_α (Section 4.1). This generalized projection is defined by a non-linear system whose equations represent the surface \mathcal{M} and the scalar function \tilde{f} underlying \mathcal{P} and f respectively (c.f., Equation (14));
- (ii) identification of the subsets of \mathcal{C} that correspond to a sampling of the connected components of γ_α . Then, each sampled connected component is converted to a piecewise linear contour (Section 4.2);
- (iii) coding of the evolution of the family $\{\gamma_{\alpha_i} := \tilde{f}^{-1}(\alpha_i)\}_i$ of level-sets of (\mathcal{M}, \tilde{f}) (Section 4.3). Fig. 10 and Fig. 11 give an overview of the method on a 2D test.

4.1. Sampling level-sets

As first step of the contouring algorithm, we evaluate the set \mathcal{S} of the intersection points between γ_α and the edges of \mathcal{T} . To compute \mathcal{S} , we consider the set \mathcal{E} of the edges of \mathcal{T} along which the scalar function f assumes values of opposite signs, i.e.

$$\mathcal{E} := \{(i, j) \in \mathcal{T} : (f(\mathbf{p}_i) \leq \alpha \leq f(\mathbf{p}_j)) \vee (f(\mathbf{p}_j) \leq \alpha \leq f(\mathbf{p}_i))\}.$$

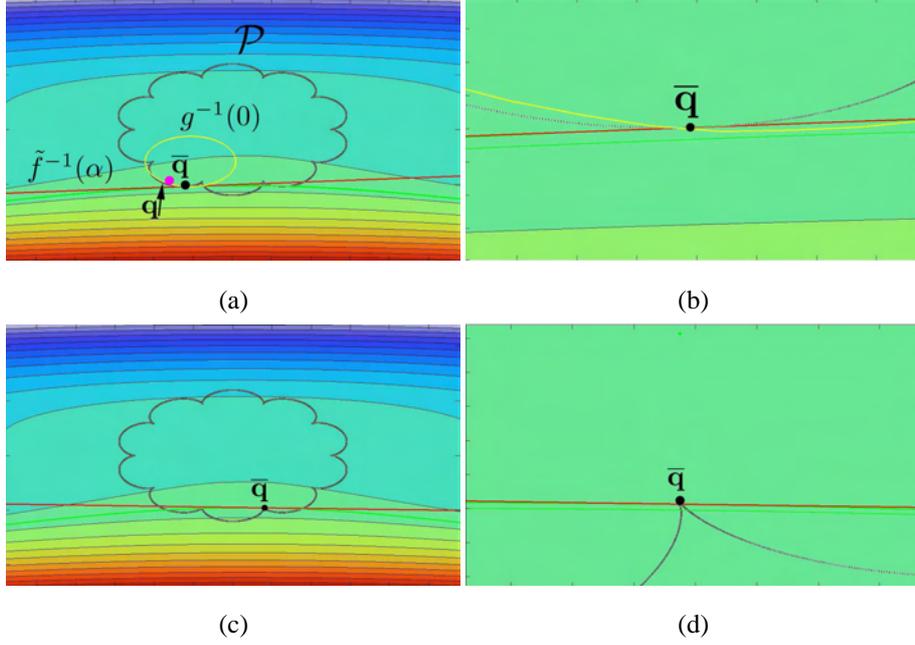


Fig. 10. This example shows the idea behind the iso-contouring technique. (a) Given the point set \mathcal{P} , we consider an input scalar function $f: \mathcal{P} \rightarrow \mathbb{R}$ whose iso-contours are represented by black lines. (b) Choosing an iso-value α , we compute a point \mathbf{q} that belongs to an edge of the k -nearest neighbor graph \mathcal{T} of \mathcal{P} and such that $f(\mathbf{q}) = \alpha$; here, f has been linearly interpolated along the edges of \mathcal{T} . In a neighborhood $\mathcal{N}_{\mathbf{q}}$ of \mathbf{q} , we approximate the input curve as the zero-set of the implicit function g ; the corresponding contour $g^{-1}(0)$ is shown in yellow. In a similar way, the red curve shows the level-set $\tilde{f}^{-1}(\alpha)$, where \tilde{f} is the scalar function underlying f in $\mathcal{N}_{\mathbf{q}}$ computed using the MLS approach described in Section 4.1. Therefore, the point $\bar{\mathbf{q}} \in \mathcal{M}$ such that $\tilde{f}(\bar{\mathbf{q}}) = \alpha$ is the intersection point between the red and yellow curve. The point \mathbf{q} is the starting point used to compute $\bar{\mathbf{q}}$ as solution of the non-linear system (14). (c,d) Analogous case related to a spike point of \mathcal{P} . See also Fig. 11.

Choosing a couple $(i, j) \in \mathcal{E}$, we have that γ_α intersects the edge $\mathbf{p}_i\mathbf{p}_j$ at \mathbf{q} ; to calculate \mathbf{q} , we can proceed in two different ways.

The simplest approach is to assume that f linearly varies along the edge $\mathbf{p}_i\mathbf{p}_j$, $(i, j) \in \mathcal{E}$; then, under the hypothesis that $f(\mathbf{p}_i) \leq \alpha \leq f(\mathbf{p}_j)$, we have

$$\begin{cases} \mathbf{q}(t) := t\mathbf{p}_i + (1-t)\mathbf{p}_j, t \in [0, 1], \\ f(\mathbf{q}(t)) := tf(\mathbf{p}_i) + (1-t)f(\mathbf{p}_j), \end{cases}$$

and therefore $\mathbf{q} = \mathbf{q}(\bar{t})$ with $\bar{t} := \frac{\alpha - f(\mathbf{p}_i)}{f(\mathbf{p}_j) - f(\mathbf{p}_i)}$ (Fig. 12). Changing i with j , an analogous discussion applies to the case $f(\mathbf{p}_j) \leq \alpha \leq f(\mathbf{p}_i)$; indeed, the linear interpolation separately treats each edge together with the related intersection point. In this case, we have that $\mathcal{S} := \{\mathbf{q}(\bar{t}), (i, j) \in \mathcal{E}\}$. A more precise approach uses the MLS techniques described in Section 3.1 and 3.2 to compute an approximation of f on the whole set \mathcal{E} .

For the implicit interpolation scheme, we build the function $F: \mathbb{R}^3 \rightarrow \mathbb{R}$ that interpolates the f -values at the vertices of the edges in \mathcal{E} . Indeed, the function F satisfies the conditions

$$F(\mathbf{p}_i) = f(\mathbf{p}_i), \quad F(\mathbf{p}_j) = f(\mathbf{p}_j), \quad \forall (i, j) \in \mathcal{E},$$

and it is computed by centering the basis functions in (8) at each vertex of the edges in \mathcal{E} . Otherwise, F is computed according to Equation (6). In both cases, \mathcal{S} is the set of intersection points between the iso-surface $\Sigma_\alpha := \{\mathbf{p} \in \mathbb{R}^3 : F(\mathbf{p}) = \alpha\}$ related to α and each edge $\mathbf{p}_i\mathbf{p}_j$, $(i, j) \in \mathcal{E}$. The intersection point on $\mathbf{p}_i\mathbf{p}_j$ is $\mathbf{q} = \bar{t}\mathbf{p}_i + (1-\bar{t})\mathbf{p}_j$ and the parameter \bar{t} is the solution of the non-linear equation $F(t\mathbf{p}_i + (1-t)\mathbf{p}_j) = \alpha$

in $[0, 1]$. The value \bar{t} is calculated by using the Gauss-Newton or Leberg-Marquard iterative algorithm (24), whose starting point $t_0 := \frac{\alpha - f(\mathbf{p}_i)}{f(\mathbf{p}_j) - f(\mathbf{p}_i)}$ is the solution of the linear problem.

Once the point \mathbf{q} has been computed with one of the aforementioned approaches, the projection $\bar{\mathbf{q}} := \text{pr}(\mathbf{q})$ of \mathbf{q} on γ_α is the solution of a system of two non-linear equations, which involve the approximation F of f (i.e., $F(\mathbf{q}) - \alpha = 0$, $\mathbf{q} \in \mathcal{N}_{\mathbf{p}}$) and the representation of \mathcal{M} (i.e., $g(\mathbf{q}) = 0$, $\mathbf{q} \in \mathcal{M}$) in $\mathcal{N}_{\mathbf{p}}$. Therefore, the point $\bar{\mathbf{q}} \in \gamma_\alpha$ is computed by solving the non-linear system

$$\mathbf{r} \in \gamma_\alpha \iff \begin{cases} F(\mathbf{r}) - \alpha = 0, \\ g(\mathbf{r}) = 0, \end{cases} \quad (\text{i.e., } \mathbf{r} \in \mathcal{M}); \quad (14)$$

in this case, the starting point of the iterative scheme is set equal to $\mathbf{r}_0 := \mathbf{q}$. Regardless the complexity of the functions $F(\mathbf{r})$ and $g(\mathbf{r})$, (14) is efficiently solved using the iterative algorithms previously mentioned; implementations of these methods are available in several software packages (7). The gradient vector (c.f., Equations (7), (12)) and Hessian matrix (c.f., Equations (11), (13)) of both F and g are analytically computed and used in the iterative scheme, thus improving the convergence to the solution of (14). Our experiments have shown that from five to ten steps are usually enough to approximate $\bar{\mathbf{q}}$ with an error of order 10^{-7} . Examples of iso-contour sampling and reconstruction are given in Fig. 13 and Fig. 14. In the following, the term projection of \mathbf{q} onto γ_α will refer to the operator that maps \mathbf{q} to the solution $\bar{\mathbf{q}} \in \gamma_\alpha$ of (14).

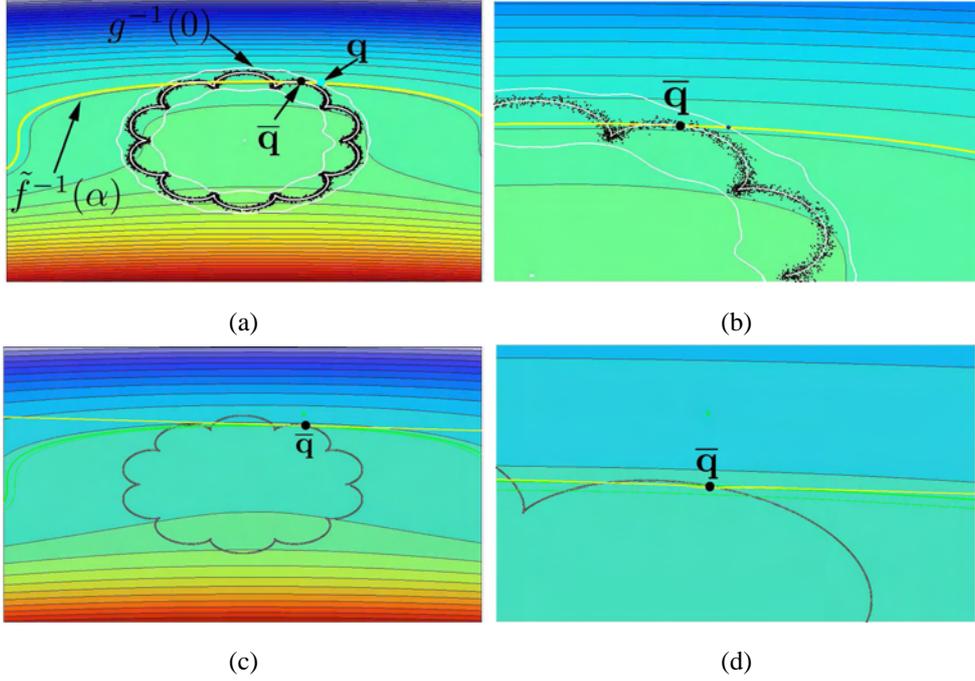


Fig. 11. With reference to Fig. 10, the pictures (a,b) show the search of a point $\bar{\mathbf{q}}$ such that $\tilde{f}(\bar{\mathbf{q}}) = \alpha$ on a noisy point set. The functions F and \tilde{f} have been computed by using the moving least-squares technique described in Section 3.1. In (c,d), we consider the same example on a noisy point set; here, F and \tilde{f} have been computed using the implicit interpolation scheme described in Section 3.2.

Let us now suppose that we have chosen two close iso-values α_1, α_2 and that from two distinct edges $e_1, e_2 \in \mathcal{T}$ we got the same point \mathbf{q} . This means that using the linear interpolation of the f -values on e_1 and e_2 gives the same point \mathbf{q} such that $f(\mathbf{q}) = \alpha_1, f(\mathbf{q}) = \alpha_2$. If $\alpha_1 \neq \alpha_2$, then the value $f(\mathbf{q})$ is inconsistent; however, this inconsistency disappears when we project \mathbf{q} onto γ_{α_1} and γ_{α_2} through (14). In fact, choosing α_1 (resp., α_2) we project \mathbf{q} onto γ_{α_1} (resp., γ_{α_2}) and compute the point $\bar{\mathbf{q}}_1 \in \mathcal{M}$ (resp., $\bar{\mathbf{q}}_2 \in \mathcal{M}$) such that $F(\bar{\mathbf{q}}_1) = \alpha_1$ (resp., $F(\bar{\mathbf{q}}_2) = \alpha_2$). If $\bar{\mathbf{q}} := \bar{\mathbf{q}}_1 = \bar{\mathbf{q}}_2$, then from (14) it follows that $\alpha_1 = F(\bar{\mathbf{q}}_1) = F(\bar{\mathbf{q}}_2) = \alpha_2, \gamma_{\alpha_1} = \gamma_{\alpha_2}$, and therefore $\tilde{f}(\bar{\mathbf{q}}_1) = \tilde{f}(\bar{\mathbf{q}}_2)$. This relation implies that we do not have an inconsistent definition of \tilde{f} at $\bar{\mathbf{q}}$.

4.2. Counting and approximating the connected components of a sampled level-set

To approximate each sampled iso-contour with a piecewise linear curve, we identify the points of \mathcal{C} that belong to the same connected component. To this end, we select a point $\mathbf{p} \in \mathcal{C}$ and recursively define the connected component Γ associated to \mathbf{p} as the set of points Γ such that: $\mathbf{p} \in \Gamma$ and \mathbf{q} is added to Γ if and only if there exists $\mathbf{r} \in \Gamma$ and $\|\mathbf{q} - \mathbf{r}\|_2 \leq \delta$. Here, the threshold δ is proportional to the averaged sampling step ε on \mathcal{P} ; in our implementation, we have chosen $\delta := 2\varepsilon$ and this value becomes the size of the holes that we can recover (Fig. 14-16(a)). For converting \mathcal{C} to a piecewise linear approximation of γ_α , we use the method presented in (32). More precisely, the sampled points $\{\mathbf{q}_i\}_{i=1}^r \subseteq \mathcal{C}$ of a connected component Γ of γ_α are mapped into the parameterization values $\{t_{j(i)}\}_{i=1}^r \subseteq \mathbb{R}$ and the permutation j gives an ordering of the points in \mathcal{C} that is used

to compute a piecewise linear interpolation curve of Γ . This approach simply requires to solve a $r \times r$ sparse linear system based on convex combinations. Once the piecewise linear curve that interpolates the points of Γ has been calculated, we can use the parameterization values to compute a least-squares approximation of Γ (35) (Ch. 5). This step provides smooth iso-contours and allows us to increase/decrease the sampling rate of γ_α , independently of that of \mathcal{P} and \mathcal{C} (Fig. 15). If \mathcal{M} has holes, which usually are introduced by the scanning process, then the least-squares approximation is useful to recover the shape of the iso-contours in those regions of \mathcal{M} where points are missed. In fact, small connected components with respect to δ are joined in a unique contour (Fig. 13(a,b)). The analysis of self-intersecting iso-contours is described in Section 5.

4.3. Evolution of the iso-contours and computational cost

First of all, we assume that the function values $\{f(\mathbf{p}_i)\}_{i=1}^n$ are sorted from the smallest to the largest value in $O(n \log n)$ -time using a reordering technique. Choosing m iso-values $\{\alpha_i\}_{i=1}^m, \alpha_i < \alpha_j, i < j$, we extract the set of the corresponding iso-contours $\{\gamma_{\alpha_i} := \tilde{f}^{-1}(\alpha_i)\}_{i=1}^m$. To trace γ_{α_i} , we use the increasing reordering of the function values to identify the s edges of \mathcal{T} intersected by γ_{α_i} , and compute the coarse sampling \mathcal{S} of γ_{α_i} . This step takes $O(s \log n)$ -time. If f is linearly interpolated along the set \mathcal{E} of the edges of \mathcal{T} intersected by γ_α , then the computation of the set \mathcal{S} of intersection points between γ_α and \mathcal{T} takes $O(s)$ -time. Using the implicit approximation of the f -values along the edges of \mathcal{T} takes $O(sk \log k)$ to compute \mathcal{S} . The projection of \mathcal{S} onto \mathcal{C} , the counting of its connected components, and the piecewise linear approxima-

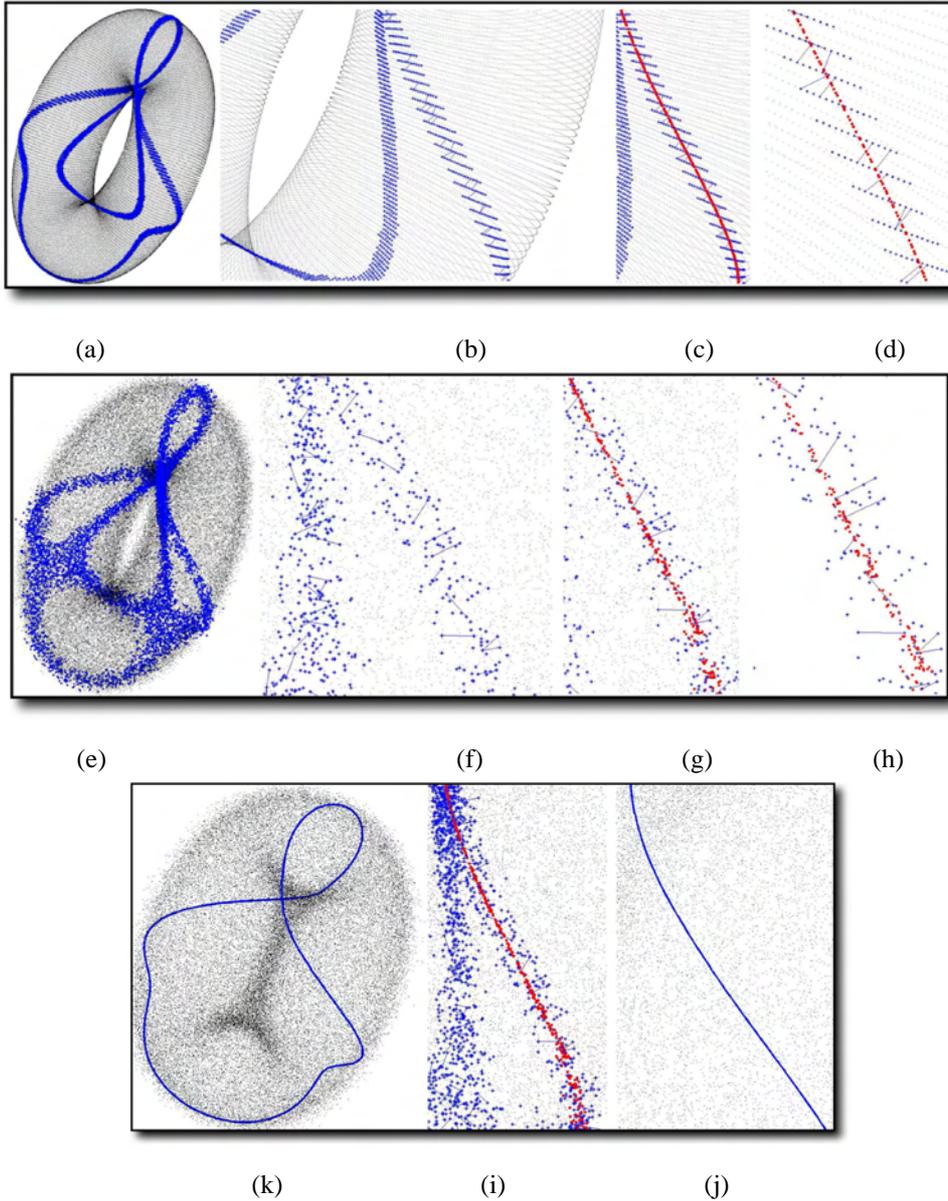


Fig. 12. (a) Smooth point set \mathcal{P} and set \mathcal{E} of the edges of the k -nearest neighborhood graph of \mathcal{P} intersected by the iso-contour γ_α ; f has been linearly interpolated. Bold blue points are the vertices of the (blue) edges of the k -nearest neighborhood intersected by γ_α . (a-d) Point-wise approximation \mathcal{S} of γ_α (red points); $\mathbf{q} \in \mathcal{S}$ if and only if $f(\mathbf{q}) = \alpha$ and $\mathbf{q} \in e$, $e \in \mathcal{E}$. (d) Closest view on (c). (e-h) These pictures show the same steps in (a-d) for a noisy point set. (k-j) Contour reconstruction related to the example in (e-h).

tion take $O(s + s \log s)$ -time. Additional information such as the number of connected components, the osculating paraboloid, and the classification of the fuzzy critical and regular points (Section 5) can be stored. Boundary components of the surface underlying \mathcal{P} do not affect the steps of the contouring algorithm; in fact, the k -nearest neighbor graph does not make distinctions between boundary and internal edges. If $F : \mathbb{R}^3 \rightarrow \mathbb{R}$ is an implicit field, then the proposed iso-contouring algorithm can also be used to evaluate the intersection between the surface underlying \mathcal{P} and the iso-surface $\Sigma_\alpha := F^{-1}(\alpha)$. Algorithm 1 and Table 2 summarize the main steps of the iso-contouring method and the corresponding computational costs.

Our tests have shown that linearly interpolating the f -values along the edges of \mathcal{T} provides an initial sampling \mathcal{S} of γ_α

such that the iterative scheme, which solves (14) with a starting point in \mathcal{S} , is always convergent. To compute \mathcal{S} , we use the MLS formulation or the implicit approach only for tracing the level-sets at saddle points or in case of irregularly sampled point sets. If the input point set is noisy, then we apply the moving least-squares approach (6) to compute the scalar function \tilde{f} underlying f . Otherwise, we apply the implicit interpolation scheme (9). In fact, a higher accuracy in the computation of \mathcal{S} is crucial to solve local ambiguities related to both \mathcal{M} and \tilde{f} . In our implementation, the function g is built using (1) (c.f., Equation (2)); this choice is motivated by the global support of g and the reliable approximation of the normals to \mathcal{M} . Firstly, the global support of g avoids that the iterative method converges to a point $\mathbf{p} \notin \text{supp}(g)$ (i.e., $g(\mathbf{p}) = 0$)

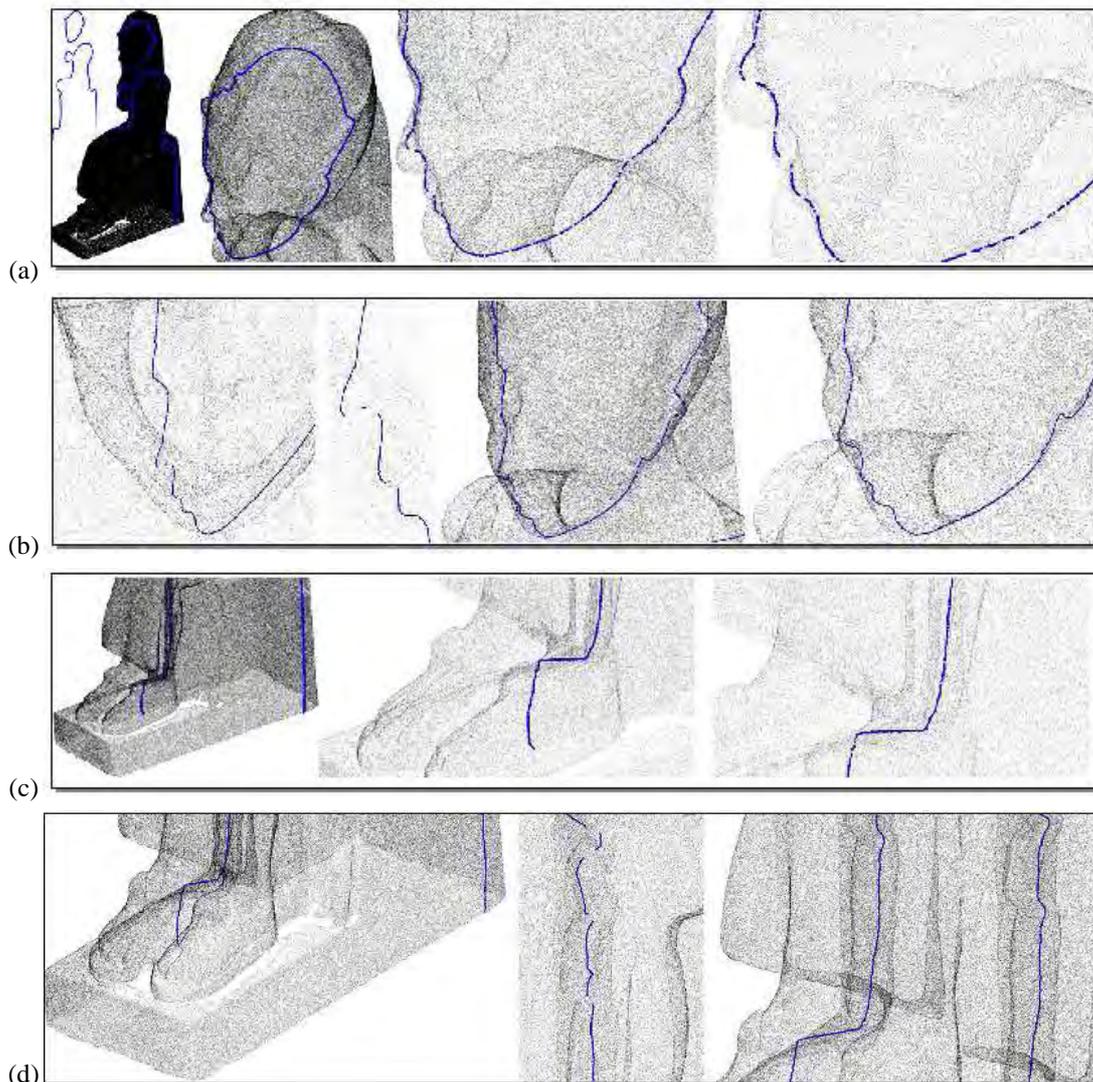


Fig. 13. Input point set containing holes and noise. In (a,c), the blue curves represent the sampled iso-contours and their piecewise linear approximations are given in (b,d). As shown in (a,b) and described in Section 4.2, the piecewise linear approximation can be used to recover the shape of the iso-contour where holes occur. See also Fig. 14 and Fig. 15.

such that $\tilde{f}(\mathbf{p}) \neq \alpha$. However, this situation did not happen in our tests, even considering holes and noise in the point set. It can be artificially generated by choosing a starting point of the iterative scheme that is far from the real solution, thus compromising its convergence. Secondly, as shown in (26) the method proposed in (1) provides a robust estimation of the surface normals, thus allowing us a reliable definition of the fuzzy critical points.

5. Critical points of scalar functions defined on point sets

Given a C^1 function $f: \mathcal{M} \rightarrow \mathbb{R}$ defined on a smooth 2-manifold surface \mathcal{M} , the *critical points* of f are defined as those points $\mathbf{p} \in \mathcal{M}$ such that $\nabla f(\mathbf{p}) = \mathbf{0}$ and they correspond to the maxima, minima, and saddles of f . For polyhedral surfaces, the method described in (11) classifies a vertex according to the values of f on its neighborhood. If \mathcal{M} is a triangle mesh, then the vertex \mathbf{p} is a *maximum* or *minimum* if its function

Table 2

Computational cost of the main steps of the iso-contouring algorithm; n is the number of vertices of the input point set and s is the number of intersected edges between \mathcal{T} and the level-set γ_α .

Task	Computational cost
k -nearest neighbor graph	$O(n \log n)$
Computation of \mathcal{S}	$O(s)$
Computation of \mathcal{C}	$O(s \log s)$
Counting connected components	$O(s)$
Piecewise linear approximation of γ_α	$O(s \log s)$

value is higher or lower than those on its 1-*star*, respectively. We briefly remind that the 1-*star* of a vertex \mathbf{p}_i is defined as the set of vertices incident to \mathbf{p}_i ; i.e., $\{\mathbf{p}_j : (\mathbf{p}_i, \mathbf{p}_j) \text{ edge}\}$. If two or more iso-curves related to the same iso-value α share a vertex \mathbf{p} , then \mathbf{p} is a *saddle*. Those points that do not fall in the previous classification are defined as *regular*.

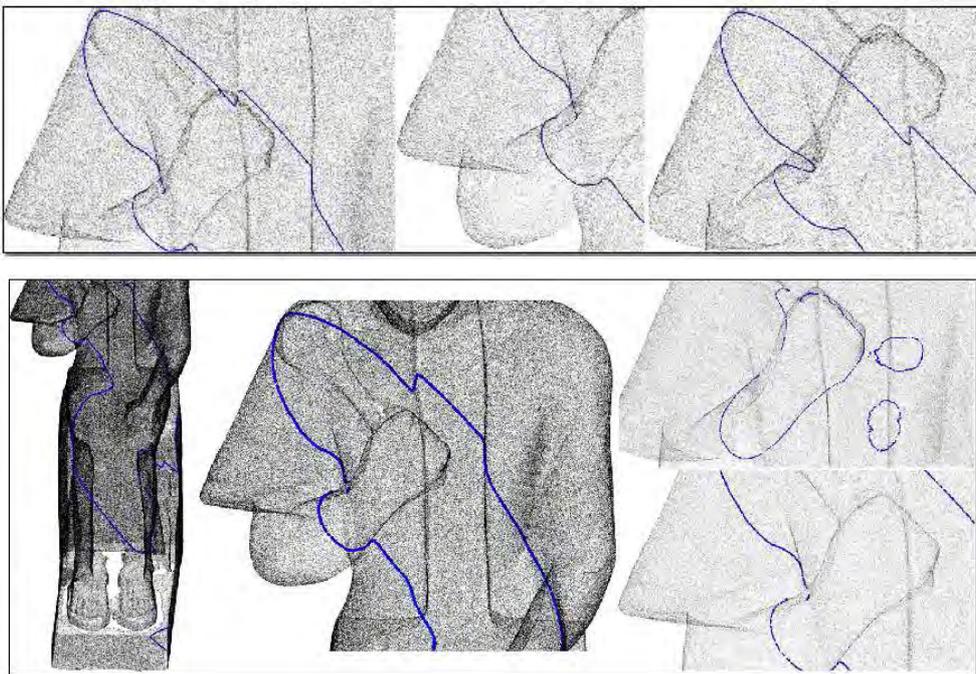


Fig. 14. Contour sampling and reconstruction achieved by projecting \mathcal{S} on the surface underlying \mathcal{P} . Another example is shown in Fig. 15.

In an analogous way, given a point set \mathcal{P} and a scalar function f defined on \mathcal{P} we want to classify its points as regular or critical; therefore, we know the coordinates of the candidate critical points. This choice resembles the case of scalar functions defined on triangulated surfaces; here, the critical points are searched among the vertices of the input triangle mesh. A first attempt to extend this definition to a scalar function defined on a point set is to replace the 1-star of \mathbf{p} with its k -nearest neighborhood. Since the choice of k or σ is not fixed *a-priori* but deduced from the point distribution, the resulting classification of \mathbf{p} will be affected by a different value of k and σ . For instance, choosing k and indicating with $\mathcal{N}_{\mathbf{p}}^{(k)}$ the k -neighborhood of \mathbf{p} it might happen that a point \mathbf{p} classified as maximum (i.e., $f(\mathbf{p}) \geq f(\mathbf{p}_i)$, $\mathbf{p}_i \in \mathcal{N}_{\mathbf{p}}^{(k)}$) becomes a regular point if $f(\mathbf{p}) < f(\mathbf{q})$ at the new point \mathbf{q} of the $(k+1)$ -neighborhood $\mathcal{N}_{\mathbf{p}}^{(k+1)} \supseteq \mathcal{N}_{\mathbf{p}}^{(k)}$ of \mathbf{p} . Furthermore, those issues are crucial at saddle points, where the changes of the sign of f give the multiplicity of the saddle. It follows that we need a characterization of the critical points that takes into account the behavior of f in a neighborhood of any point without being affected by the discretization of $\mathcal{N}_{\mathbf{p}}$. In Section 5.1, we derive a geometric characterization of the critical points of a scalar function defined on a smooth surface. In Section 5.2 and 5.3, we discuss how this characterization can be applied to functions defined on point sets.

5.1. Equivalent definitions of critical points

Let us consider a scalar function $F : \mathbb{R}^3 \rightarrow \mathbb{R}$, $F \in C^1(\mathbb{R}^3)$, and its restriction $f := F|_{\mathcal{M}}$ to a smooth surface \mathcal{M} ; in the following, we make explicit the notion of critical point of f

on \mathcal{M} . Let $\mathbf{r}(u, v) := (x(u, v), y(u, v), z(u, v))$ be a local parameterization of \mathcal{M} around the point $\mathbf{p} = \mathbf{r}(u_0, v_0) \in \mathcal{M}$, where $(u, v) \in \Omega \subseteq \mathbb{R}^2$, Ω is an open disc, and (u_0, v_0) belongs to Ω . Then, the values of f on a neighborhood of \mathbf{p} are given by $w(u, v) := F(\mathbf{r}(u, v))$ and \mathbf{p} is a critical point of (\mathcal{M}, f) if and only if $\nabla w(u_0, v_0) = \mathbf{0}$. Using the derivation formula of composite functions, the following relations hold

$$\begin{cases} \partial_u w(u_0, v_0) = \langle \nabla F(\mathbf{p}), \partial_u \mathbf{r}(u_0, v_0) \rangle = 0, \\ \partial_v w(u_0, v_0) = \langle \nabla F(\mathbf{p}), \partial_v \mathbf{r}(u_0, v_0) \rangle = 0. \end{cases}$$

Since the normal vector to the surface \mathcal{M} at \mathbf{p} is given by $\mathbf{n}(\mathbf{p}) = \partial_u \mathbf{r}(u_0, v_0) \wedge \partial_v \mathbf{r}(u_0, v_0)$, we get that \mathbf{p} is a critical point of f if and only if $\nabla F(\mathbf{p})$ is parallel to $\mathbf{n}(\mathbf{p})$. This relation does not directly depend on the neighborhood of \mathbf{p} and expresses the local dependence between the gradient field ∇F , related to the differential properties of F , and the normal field defined by the geometry of \mathcal{M} . Two main examples are the height function and the Euclidean distance from a point. For the height function, once the direction \mathbf{N} has been fixed we have that $\nabla F \equiv \mathbf{N}$ and $\mathbf{p} \in \mathcal{M}$ is a critical point of \tilde{f} if and only if \mathbf{N} is parallel to the normal $\mathbf{n}(\mathbf{p})$ at \mathbf{p} . A similar discussion applies when F is the Euclidean distance from a point; in this case, $\nabla F(\mathbf{p}) = 2(\mathbf{p} - \mathbf{b})$ and $\mathbf{p} \in \mathcal{P}$ is critical if and only if we have that $\langle \mathbf{p} - \mathbf{b}, \mathbf{n}(\mathbf{p}) \rangle = 0$. For a point set \mathcal{P} , the normal vector at each point $\mathbf{p} \in \mathcal{P}$ is calculated using the eigensystem of the covariance matrix of a local neighborhood of \mathbf{p} (Section 3.1). According to the discussion in Section 3, we consider as F the implicit function that interpolates the values of f in a neighborhood $\mathcal{N}_{\mathbf{p}}$ of \mathbf{p} or one of the moving least-squares approximations. Then, we compute $\nabla F(\mathbf{p})$ in linear time as shown in Equation (7) and (11). The local definition of critical points also guarantees that increasing or decreasing k or σ will

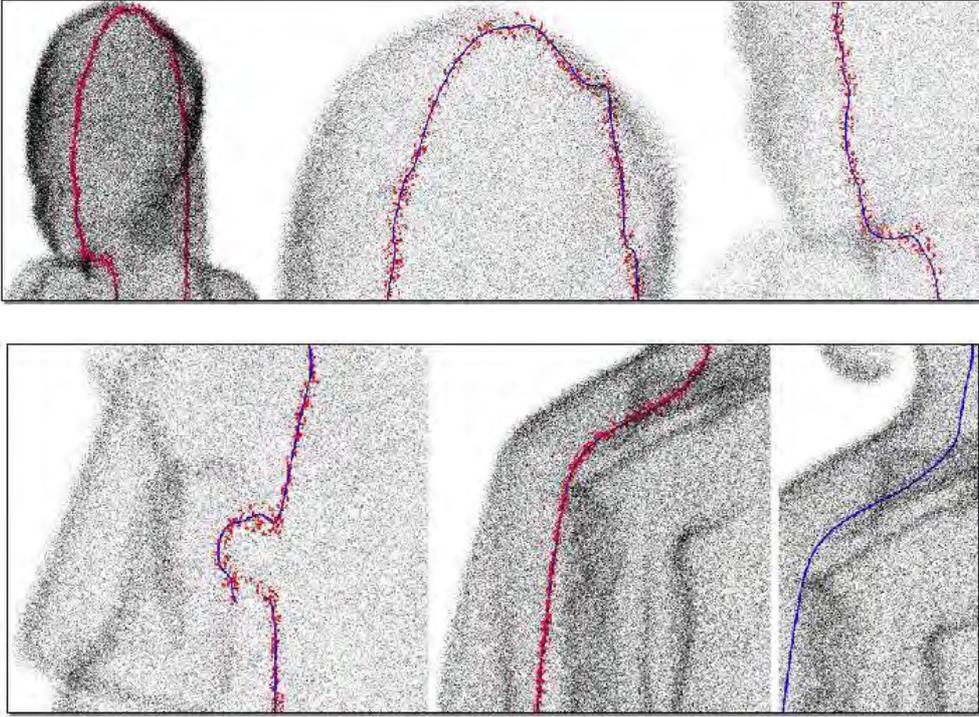


Fig. 15. Noisy point set \mathcal{P} ; the red points show the initial sampling \mathcal{S} of the chosen level-sets and the blue curves are the reconstructed iso-contours achieved by projecting \mathcal{S} on the surface underlying \mathcal{P} .

not affect the classification of \mathbf{p} .

We are now ready to clarify the relation among the critical points and the evolution of the iso-contours. In Section 5.2, these properties will be used to classify the critical points of scalar functions defined on point sets. Assuming that $f(\mathbf{p}_i) < f(\mathbf{p}_j)$, $i < j$, the topology of the level-sets remains the same as long as t belongs to the open interval $(f(\mathbf{p}_i), f(\mathbf{p}_{i+1}))$. If \mathbf{p}_{i+1} is a maximum or a minimum, then the level-set $f^{-1}(t)$, $t \in [f(\mathbf{p}_{i+1}) - \varepsilon, f(\mathbf{p}_i)]$ (resp., $t \in (f(\mathbf{p}_i), f(\mathbf{p}_{i+1}) + \varepsilon]$, $\varepsilon > 0$, degenerates to \mathbf{p}_i or develops a set of homeomorphic contours. If \mathbf{p}_{i+1} is a saddle point, then two or more contours cycles of the level-sets $f^{-1}(t)$, $t \in (f(\mathbf{p}_{i+1}) - \varepsilon, f(\mathbf{p}_{i+1}) + \varepsilon)$, are joined into a new cycle or an existing cycle is split into two or more cycles.

5.2. Critical point analysis and related iso-contours

Indicating with $\text{Im}(f)$ the minimal interval containing the image of f , in most of the cases an arbitrary value $\alpha \in \text{Im}(f)$ will be regular, i.e. $\tilde{f}^{-1}(\alpha)$ does not contain a critical point of \tilde{f} on \mathcal{M} . In fact, under mild conditions on the regularity of \tilde{f} and \mathcal{M} the set of critical points of \tilde{f} is finite and therefore it has null measure (49). In the following, we introduce the concept of fuzzy critical point and discuss how we handle self-intersecting or degenerate iso-contours, which are generated when α is close or equal to a critical iso-value. Once the gradient field of F and the normal $\mathbf{n}(\mathbf{p})$ at each point $\mathbf{p} \in \mathcal{P}$ have been calculated, we consider the angle $\omega_{\mathbf{p}}$ between $\nabla F(\mathbf{p})$ and $\mathbf{n}(\mathbf{p})$. Theoretically, \mathbf{p} is a critical point of \tilde{f} if and only if $\nabla F(\mathbf{p})$ is parallel to $\mathbf{n}(\mathbf{p})$. Computing the angle $\omega_{\mathbf{p}}$ and con-

sidering \mathbf{p} as critical if $\omega_{\mathbf{p}}$ is equal or close to zero works in few situations and generally fails in those cases where we deal with an irregularly sampled point set and a noisy scalar function. Therefore, we introduce the concept of *fuzzy critical* and *fuzzy regular point* of a scalar function defined on a point set. Here, the term fuzzy means that we associate to (\mathcal{M}, \tilde{f}) a probability measure $\pi : \mathcal{M} \rightarrow [0, 1]$ such that $\pi(\mathbf{p})$ is the probability that $\mathbf{p} \in \mathcal{M}$ is a critical point of \tilde{f} . We define this function as

$$\pi(\mathbf{p}) := |\langle \nabla F(\mathbf{p}), \mathbf{n}(\mathbf{p}) \rangle| = |\cos \omega_{\mathbf{p}}|,$$

where we assume that both the gradient field and the normal vector have unit length (Fig. 16(b)). Therefore, $\mathbf{p} \in \mathcal{M}$ is called *fuzzy critical point* if $\pi(\mathbf{p}) \approx 1$ and *regular* otherwise; if fuzzy critical points are close to each others, then we consider as representative point the one with the highest π -value.

Once the values $\{\pi(\mathbf{p})\}_{\mathbf{p} \in \mathcal{P}}$ have been computed, the threshold ε , which is used to verify if $(1 - \pi(\mathbf{p})) \leq \varepsilon$, is set by considering the variation $|\pi(\mathbf{p}) - \pi(\mathbf{q})|$, $\mathbf{q} \in \mathcal{N}_{\mathbf{p}}$, of the probability values at \mathbf{p} and at the points of its 1-neighborhood. Note that considering $\mathbf{p} \in \mathcal{P}$ as a critical point if $\pi(\mathbf{p}) \equiv 1$ does not cover all the possible cases; in fact, it might happen that $\max_{\mathbf{p} \in \mathcal{P}} \{\pi(\mathbf{p})\}$ is lower than one. Finally, we stress that also for a scalar function f defined on triangulated surfaces the concept of critical point is derived in an approximate way by using the f -values along the mesh edges and without computing the gradient field of f . Since \tilde{f} is smooth, we expect that two points \mathbf{p}, \mathbf{q} in $\mathcal{N}_{\mathbf{r}}$ whose probability values are close to one will have close \tilde{f} -values. Therefore, a more precise characterization of \tilde{f} in $\mathcal{N}_{\mathbf{r}}$ is achieved by comparing the related iso-contours $\tilde{f}^{-1}(\tilde{f}(\mathbf{p}))$ and $\tilde{f}^{-1}(\tilde{f}(\mathbf{q}))$.

Algorithm 1. The pseudocode summarizes the main steps of the routine *iso-contouring*($\mathcal{P}, f, \mathcal{A}$) used to draw the iso-contours of an arbitrary scalar function f , defined on a point set \mathcal{P} , and related to the iso-values in \mathcal{A} .

Require: A point set \mathcal{P} , a scalar function $f: \mathcal{P} \rightarrow \mathbb{R}$, and a set of iso-values $\mathcal{A} := \{\alpha_i\}_{i=1}^m$.
Ensure: The set of piecewise linear contours $\{\gamma_{\alpha_i} := \tilde{f}^{-1}(\alpha_i)\}_{i=1}^m$ on \mathcal{M} , where \tilde{f} and \mathcal{M} are the function and the surface underlying f and \mathcal{P} respectively.

- 1: Compute the k -nearest neighbor graph \mathcal{T} of \mathcal{P} .
- 2: Sort the function values $\{f(\mathbf{p}_i), \mathbf{p}_i \in \mathcal{P}\}$.
- 3: **for** $\alpha_s \in \mathcal{A}$ **do**
- 4: $\mathcal{C} := \emptyset$.
- 5: Find

$$\mathcal{E} := \{(i, j) \in \mathcal{T} : (f(\mathbf{p}_i) \leq \alpha_s \leq f(\mathbf{p}_j)) \mid (f(\mathbf{p}_j) \leq \alpha_s \leq f(\mathbf{p}_i))\}$$
- 6: **for** $e := (i, j) \in \mathcal{E}$ **do**
- 7: compute

$$\mathbf{q} := \bar{t}\mathbf{p}_i + (1 - \bar{t})\mathbf{p}_j, \quad \bar{t} := \frac{\alpha_s - f(\mathbf{p}_i)}{f(\mathbf{p}_j) - f(\mathbf{p}_i)};$$
- 8: compute g as (2) and F as (6) in $\mathcal{N}_{\mathbf{q}}$;
- 9: compute the solution $\bar{\mathbf{q}} \in \gamma_{\alpha_s}$ of

$$\begin{cases} F(\mathbf{r}) - \alpha_s = 0, \\ g(\mathbf{r}) = 0, \end{cases} \quad \text{with initial point } \mathbf{r}_0 := \mathbf{q};$$
- 10: update the sampling \mathcal{C} of γ_{α_s} as $\mathcal{C} := \mathcal{C} \cup \{\bar{\mathbf{q}}\}$.
- 11: **end for**
- 12: Compute the subset $\{\Gamma_j\}_j$ of \mathcal{C} that corresponds to a connected component of γ_{α_s} .
- 13: Convert each Γ_j to a piecewise linear curve $\gamma_{\alpha_s}^{(j)}$.
- 14: Draw/Store $\{\gamma_{\alpha_s}^{(j)}\}_j$.
- 15: **end for**

To classify each fuzzy critical point as maximum, minimum, or saddle, we consider the values of \tilde{f} and the shape of the corresponding iso-contour γ_{α} . More precisely, note that when the iso-value α is equal or close to a local maximum or minimum value $f(\mathbf{p}_i)$, then the set \mathcal{E} of the edges of \mathcal{T} intersected by γ_{α} contains none or few intersected edges. Therefore, if γ_{α} is empty or falls inside the sphere $S(\mathbf{p}_i, \delta)$ of center \mathbf{p}_i and radius $\delta := 2\varepsilon$, with ε sampling density, then \mathbf{p}_i is classified as maximum (resp., minimum) if $\tilde{f}(\mathbf{p}_i) > \tilde{f}(\mathbf{q})$ (resp., $\tilde{f}(\mathbf{p}_i) < \tilde{f}(\mathbf{q})$), where $\mathbf{q} \in \mathcal{P}$, $\mathbf{q} \neq \mathbf{p}_i$, is the closest point to \mathbf{p}_i . Fig. 17 shows an example of computation of critical points based on ground-truth.

If α is close to the value of f at a saddle point $\mathbf{p}_i \in \mathcal{P}$, with $f(\mathbf{p}_i) = \alpha$, then the iso-contour γ_{α} is self-intersecting at this point and the conversion of \mathcal{C} to a piecewise linear contour fails. Therefore, we need to remove from \mathcal{C} a neighborhood of \mathbf{p}_i , where the self-intersection of γ_{α} occurs. In this case, we consider \mathcal{C} and select the sampled connected component Γ that contains \mathbf{p}_i by applying the procedure described in Section 4.2. To approximate Γ with a piecewise linear contour, we first extract the two sub-loops which belong to Γ and that share the point \mathbf{p}_i . To this end, we center at \mathbf{p}_i a sphere $S(\mathbf{p}_i, \delta)$ of radius δ and remove the set Γ^* of points that fall inside $S(\mathbf{p}_i, \delta)$, thus generating two connected components Γ_1 and Γ_2 such that

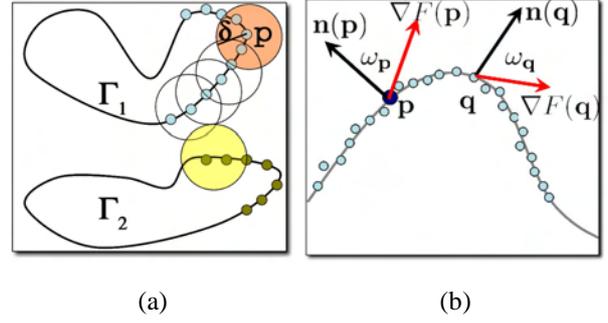


Fig. 16. (a) Identification of the connected components of a sampled level-set. (b) The angle between $\nabla F(\mathbf{p})$ and $\mathbf{n}(\mathbf{p})$ is used to measure the probability that the point \mathbf{p} is critical.

$\Gamma = \Gamma_1 \cup \Gamma_2 \cup \Gamma^*$ (Fig. 18(a)). For each component Γ_s , $s = 1, 2$, we update the shape of Γ around \mathbf{p}_i by selecting the points \mathbf{p}_1 and \mathbf{p}_2 of Γ_s closest to \mathbf{p}_i , such that $\|\mathbf{p}_1 - \mathbf{p}_2\|_2 > \delta$, and adding to Γ_s a set of points sampled on the segments $\mathbf{p}_1\mathbf{p}_s$ and $\mathbf{p}_2\mathbf{p}_s$. These new samples recover the geometry of the iso-contour around the saddle where we removed Γ^* (Fig. 18(b)). Then, the updated components Γ_1 and Γ_2 are converted to piecewise linear curves as previously described (Fig. 18(c)). Another example is shown in Fig. 19.

Our tests have shown that the set $\mathcal{C}_{\tilde{f}}$ of the fuzzy critical points of \tilde{f} always contains the set \mathcal{C}_f of the critical points of f computed using the triangle mesh connectivity, which is assumed to be *a-priori* known, and according to the method described in (11). Dealing with smooth surfaces and scalar functions generally reduces the number of fuzzy critical points and therefore the gap between $\mathcal{C}_{\tilde{f}}$ and \mathcal{C}_f . A high sampling density increases the reliability of the computation of the normal $\mathbf{n}(\mathbf{p})$ and the gradient $\nabla F(\mathbf{p})$ but generates clustered fuzzy critical points. In fact, a higher sampling density corresponds to a larger number of gradient vectors $\nabla F(\mathbf{q})$, whose directions are closely aligned with $\nabla F(\mathbf{q})$, $\mathbf{q} \in \mathcal{N}_{\mathbf{p}}$, in a neighborhood $\mathcal{N}_{\mathbf{p}}$ of \mathbf{p} . Therefore, we get closer values of $\pi(\mathbf{q})$, $\mathbf{q} \in \mathcal{N}_{\mathbf{p}}$. A relevant amount of clustered fuzzy critical points are removed by analyzing the shape of the corresponding level-sets, applying clustered techniques, or selecting among the critical points of the same cluster the one with the highest π -value. Finally, note that in the case of bordered surfaces the classification of the fuzzy critical and regular points fails. In fact, it is not able to identify the points of \mathcal{P} that belong to the boundary components and to provide a good approximation of the gradient field of f at those points. Furthermore, the evaluation of the normals on the boundary points is not trivial and we face similar limitations for the definition of the critical points of scalar functions defined on bordered triangle meshes.

5.3. Refined classification of the critical points

Degenerate situations are related to close critical points that belong to the same k -nearest neighborhood and have the same f -values. The presence of noise in \mathcal{P} and f , as well as the different approximations of the underlying surface and scalar function, motivate our choice to replace the “exact” definition

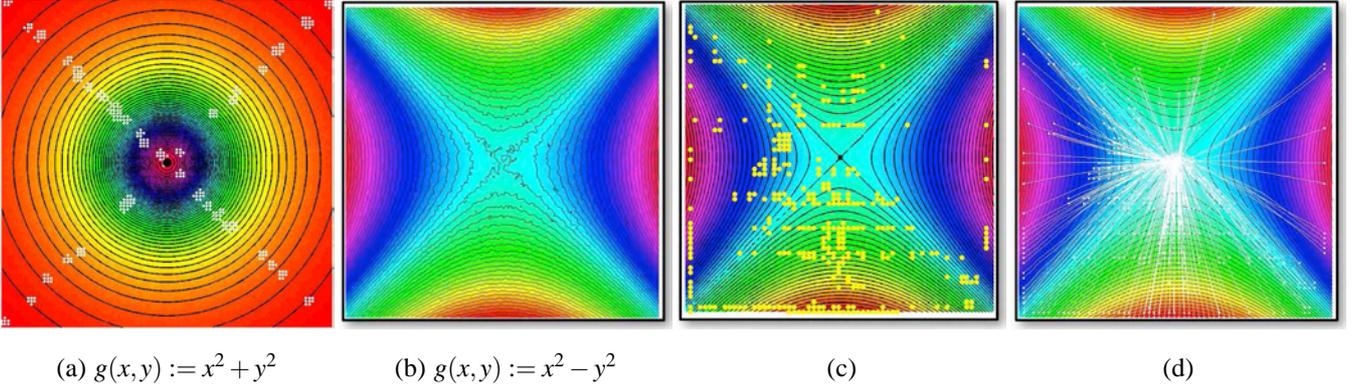


Fig. 17. (a) Level-sets of the implicit approximation F associated to the sampling $f: \mathcal{P} \rightarrow \mathbb{R}$ of the function $g(x,y)$ on the unit square centered at the origin $\mathbf{0} := (0,0)$, which is the unique critical point of g . Level-sets of (b) a noisy function f and (c) its explicit MLS approximation. In (a,c,d), the dots have been randomly selected and used as starting guesses of the iterative scheme that computes the critical point (black dot) of F , with an error lower than 10^{-5} . (d) Paths that join the initial guesses and the computed critical point.

of critical points with the concept of fuzzy critical points. The probability measure π allows us to handle degenerate and non-degenerate cases in the same way. However, we can further refine the previous classification of the critical points as follows. Let us suppose that a point $\mathbf{p} \in \mathcal{C}$ has been classified as critical; we improve its position by solving the following system of non-linear equations

$$\begin{cases} \langle \nabla F(\mathbf{q}), \mathbf{n}(\mathbf{q}) \rangle = \mathbf{0} & (\text{i.e., } \mathbf{q} \text{ critical point}), \\ g(\mathbf{q}) = 0 & (\text{i.e., } \mathbf{q} \in \mathcal{M}); \end{cases} \quad (15)$$

to this end, we use as starting point of the iterative scheme $\mathbf{q} := \mathbf{p}$. The solution $\bar{\mathbf{p}}$ of (15) is considered the “true” critical point and this procedure is applied to each point of \mathcal{C} . Therefore, the classification of the critical points and degenerate cases can be improved by comparing the related iso-contours $\tilde{f}^{-1}(\tilde{f}(\mathbf{p}))$ and $\tilde{f}^{-1}(\tilde{f}(\mathbf{q}))$ or computing the “true” critical point through (15).

6. Discussion

The stability of the definition of the MLS surfaces (1), (2) and of the iterative solver of (10) have been discussed extensively in state-of-the-art works (4), (35) (Ch. 5). Therefore, in the following (Section 6.1) we focus our attention on the numerical stability and accuracy of the approximation schemes presented in Section 3. Then (Section 6.2), we discuss the choice of the main parameters used by the iso-contouring algorithm.

6.1. Numerical stability and approximation accuracy

Let us suppose that we perturb each function value $f(\mathbf{p}_i)$ and consider $f(\mathbf{p}_i) + e_i$, $e_i \in \mathbb{R}$, $i = 1, \dots, n$; then, we want to analyze the discrepancy between the approximation F and F_e related to the sets $\{f(\mathbf{p}_i)\}_{i=1}^n$ and $\{f(\mathbf{p}_i) + \varepsilon_i\}_{i=1}^n$ respectively. The functions F and F_e , as well as $\tilde{f} := F|_{\mathcal{M}}$ and $\tilde{f}_e := F_e|_{\mathcal{M}}$, will be computed according to the schemes described in Section 3.

Parameterized MLS formulation. We focus our attention on the neighborhood $\mathcal{N}_{\mathbf{p}}$ of \mathbf{p} and we assume that $\mathcal{N}_{\mathbf{p}} := \{\mathbf{p}_1, \dots, \mathbf{p}_k\}$; then, we define the following $k \times 1$ vectors

$$\mathbf{F} := (F(\mathbf{p}_j))_{j=1}^k, \quad \mathbf{F}_e := (F_e(\mathbf{p}_j))_{j=1}^k, \quad \mathbf{e} := (e_j)_{j=1}^k.$$

Indicating with r and r_e the polynomial that approximates the real and perturbed f -values in $\mathcal{N}_{\mathbf{p}}$ (Section 3.1), we have that

$$\begin{aligned} |r(x,y) - r_e(x,y)| &= \left| \sum_{i=1}^m (\alpha_i - \beta_i) b_i(x,y) \right| \\ &\leq C \|\alpha - \beta\|_1 \\ &\leq C_{\text{new}} \|(\mathbf{B}\mathbf{O}\mathbf{B}^T)^{-1} \mathbf{e}\|_2 \\ &\leq C_{\text{new}} \frac{\|\mathbf{e}\|_2}{\min_{i=1, \dots, m} \{\lambda_i(\mathbf{B}\mathbf{O}\mathbf{B}^T)\}}, \quad \forall (x,y), \end{aligned} \quad (16)$$

where $\lambda_i(X)$ is the i^{th} -eigenvalue of X . In particular, we have that $\lim_{\|\mathbf{e}\|_2 \rightarrow 0} \|r(x,y) - r_e(x,y)\|_2 = 0$ and therefore $\lim_{\|\mathbf{e}\|_2 \rightarrow 0} \tilde{f}_e(\mathbf{p}_i) = f(\mathbf{p}_i)$, $i = 1, \dots, k$.

Explicit MLS formulation. Assuming that F is computed as in (6), we have that

$$F_e(\mathbf{p}) = \frac{\sum_{i=1}^n (f(\mathbf{p}_i) + e_i) \theta(\|\mathbf{p} - \mathbf{p}_i\|_2)}{\sum_{i=1}^n \theta(\|\mathbf{p} - \mathbf{p}_i\|_2)}.$$

From the definition, it follows that if $\|\mathbf{e}\|_\infty \rightarrow 0$ then $F_e(\mathbf{p}) \rightarrow F(\mathbf{p})$; in particular, $\lim_{\|\mathbf{e}\|_\infty \rightarrow 0} \tilde{f}_e(\mathbf{p}_i) = f(\mathbf{p}_i)$, $i = 1, \dots, n$. Furthermore,

$$\begin{aligned} F_e(\mathbf{p}) &= \frac{\sum_{i=1}^n f(\mathbf{p}_i) \theta(\|\mathbf{p} - \mathbf{p}_i\|_2)}{\sum_{i=1}^n \theta(\|\mathbf{p} - \mathbf{p}_i\|_2)} + \frac{\sum_{i=1}^n e_i \theta(\|\mathbf{p} - \mathbf{p}_i\|_2)}{\sum_{i=1}^n \theta(\|\mathbf{p} - \mathbf{p}_i\|_2)} \\ &= F(\mathbf{p}) + \frac{\sum_{i=1}^n e_i \theta(\|\mathbf{p} - \mathbf{p}_i\|_2)}{\sum_{i=1}^n \theta(\|\mathbf{p} - \mathbf{p}_i\|_2)} \\ &\leq F(\mathbf{p}) + \|\mathbf{e}\|_\infty, \quad \forall \mathbf{p}, \end{aligned}$$

with $\mathbf{e} := (e_i)_{i=1}^n$. Therefore, $|F_e(\mathbf{p}) - F(\mathbf{p})| \leq \|\mathbf{e}\|_\infty$, $\forall \mathbf{p}$. Using the following matrix notation (see also Section 3.1 and 3.3)

$$\begin{aligned} \mathbf{F} &:= (F(\mathbf{p}_i))_{i=1}^n, & \mathbf{F}_e &:= (F_e(\mathbf{p}_i))_{i=1}^n, \\ \mathbf{D} &:= \text{diag}(\theta_1, \dots, \theta_n), & \theta_i &:= \sum_{j=1}^n \theta(\|\mathbf{p}_i - \mathbf{p}_j\|_2), \end{aligned}$$

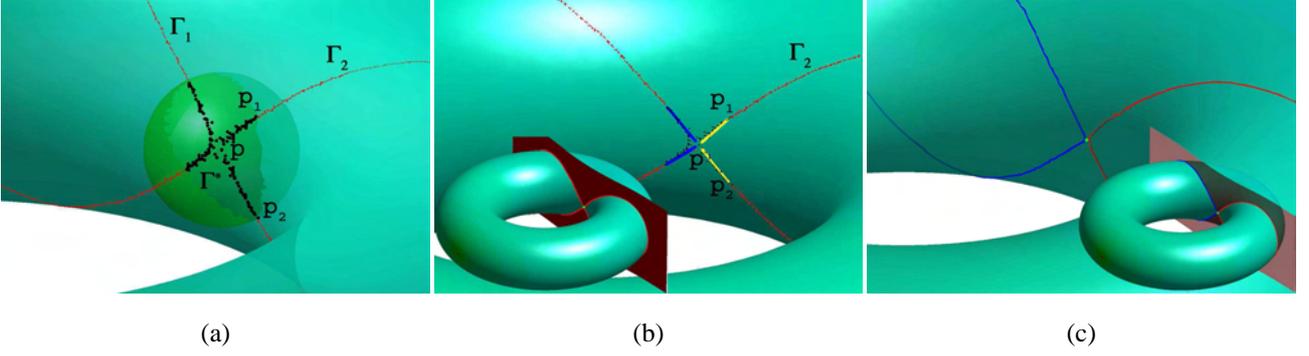


Fig. 18. (a) Sampled points on the iso-contour Γ related to a saddle point \mathbf{p} and identification of the two loops Γ_1 and Γ_2 of Γ , (b) smoothing of the shape of Γ around \mathbf{p} , (c) iso-contour reconstruction.

we have that

$$\mathbf{A} := (\theta(\|\mathbf{p}_i - \mathbf{p}_j\|))_{i,j=1}^n, \quad \mathbf{F} = \mathbf{D}^{-1}\mathbf{A}\mathbf{f}; \quad \mathbf{F}_e = \mathbf{D}^{-1}\mathbf{A}(\mathbf{f} + \mathbf{e}).$$

Then, we estimate the global variation of this approximation as

$$\begin{aligned} \|\mathbf{F}_e - \mathbf{F}\|_2 &= \|\mathbf{D}^{-1}\mathbf{A}\mathbf{e}\|_2 \\ &\leq \|\mathbf{D}^{-1}\|_2 \|\mathbf{A}\|_2 \|\mathbf{e}\|_2 \\ &\leq \frac{\|\mathbf{A}\|_2 \|\mathbf{e}\|_2}{\min_{i=1,\dots,n} \{\theta_i\}}. \end{aligned}$$

Approximation of scalar functions with radial basis functions.

Assuming that F has been computed as in (9) and neglecting the linear term, the following relations hold

$$F(\mathbf{p}) = \sum_{i=1}^k \alpha_i \varphi_i(\mathbf{p}), \quad F(\mathbf{p}_i) = f(\mathbf{p}_i), \quad i = 1, \dots, k,$$

and the vector $\alpha := (\alpha_i)_{i=1}^k$ is the solution of the linear system $\mathbf{A}\alpha = \mathbf{f}$ in (10), with $\mathbf{A} := (\varphi_i(\mathbf{p}_j))_{i,j=1}^k$. Analogously, the perturbed function values are associated to the implicit function

$$F_e(\mathbf{p}) = \sum_{i=1}^k \beta_i \varphi_i(\mathbf{p}), \quad F_e(\mathbf{p}_i) = f(\mathbf{p}_i) + e_i, \quad i = 1, \dots, k,$$

and the vector $\beta := (\beta_i)_{i=1}^k$ is the solution of the linear system $\mathbf{A}\beta = \mathbf{f} + \mathbf{e}$. Therefore, using the property of bounded variation of the basis functions $|\varphi_i(\mathbf{p})| \leq C, \forall \mathbf{p}, i = 1, \dots, k, C \in \mathbb{R}$ constant, (e.g., for the Gaussian kernel, $C = 1$) and the inequality $\|\mathbf{e}\|_1 \leq \sqrt{k}\|\mathbf{e}\|_2$ we have that

$$\begin{aligned} |F_e(\mathbf{p}) - F(\mathbf{p})| &= \left| \sum_{i=1}^k (\alpha_i - \beta_i) \varphi_i(\mathbf{p}) \right| \\ &\leq C \sum_{i=1}^k |\alpha_i - \beta_i| \\ &= C \|\alpha - \beta\|_1 \leq C \|\mathbf{A}^{-1}\mathbf{e}\|_1 \quad (17) \\ &\leq C\sqrt{k} \|\mathbf{A}^{-1}\mathbf{e}\|_2 \\ &\leq C\sqrt{k} \|\mathbf{A}^{-1}\|_2 \|\mathbf{e}\|_2 \\ &= C\sqrt{k} \frac{\|\mathbf{e}\|_2}{\min_{i=1,\dots,k} \{\lambda_i(\mathbf{A})\}}. \end{aligned}$$

If $\|\mathbf{e}\| \rightarrow 0$, then $F_e(\mathbf{p}) \rightarrow F(\mathbf{p}), \forall \mathbf{p}$. Using the matrix formulation used in Section 3.2, it follows that

$$\mathbf{F} = \mathbf{A}^{-1}\mathbf{f}, \quad \mathbf{F}_e = \mathbf{A}^{-1}(\mathbf{f} + \mathbf{e});$$

therefore, we estimate the global variation of this approximation as

$$\|\mathbf{F}_e - \mathbf{F}\|_2 = \|\mathbf{A}^{-1}\mathbf{e}\|_2 \leq \|\mathbf{A}^{-1}\|_2 \|\mathbf{e}\|_2 = \lambda_{\min}^{-1}(\mathbf{A}) \|\mathbf{e}\|_2;$$

and $\lim_{\|\mathbf{e}\|_2 \rightarrow 0} \|\mathbf{F} - \mathbf{F}_e\|_2 = 0$. Finally, note the analogy between (16) and (17); in both cases, the function \tilde{f}_e underlying f_e is built as a linear combination of a different set of basis functions and each of them is associated to a different coefficient matrix of the corresponding linear system.

6.2. Choice of the parameters and approximation schemes

Under the assumption that \mathcal{P} is highly noisy, the least-squares approach provides a function underlying f that is smoother than the one related to the implicit interpolation scheme. Furthermore, the MLS approach does not require to solve a linear system. In those cases where we need a high accuracy in the approximation of the f -values and the differential properties of f (e.g., gradient field, Hessian matrix), we prefer the implicit interpolating scheme (Section 3.2) to the MLS methods (Section 3.1). Note that the interpolation scheme implicitly assumes that the f -values are not highly noisy. This is the case of scalar functions that are the solution of differential equations such as the Laplace equation with Dirichlet boundary conditions and the Laplace eigenproblem. As shown in Fig. 20, a higher sampling density improves the smoothness of the level-sets; however, a low density does not affect the iso-contour sampling and reconstruction. This is mainly due to the definition of each sample of a given iso-contour as the solution of a system of implicit equations, which smoothly approximate both the surface and the function values in those regions where the information is low or partially missed. Furthermore, the sampling density can be changed by either up-sampling or re-sampling the point set. Also, smoothing techniques can be used to create a new point set that is free of outliers and locally smooth (4; 31), thus improving the regularity of \tilde{f} . Table 3 summarizes the main properties and the computational cost of the MLS and implicit scheme described Section 3.

In case of point sets with a low sampling density, implicit modeling techniques for surface and function approximation are preferred to MLS techniques. In all the other cases, both the implicit and MLS approaches provide satisfactory results;

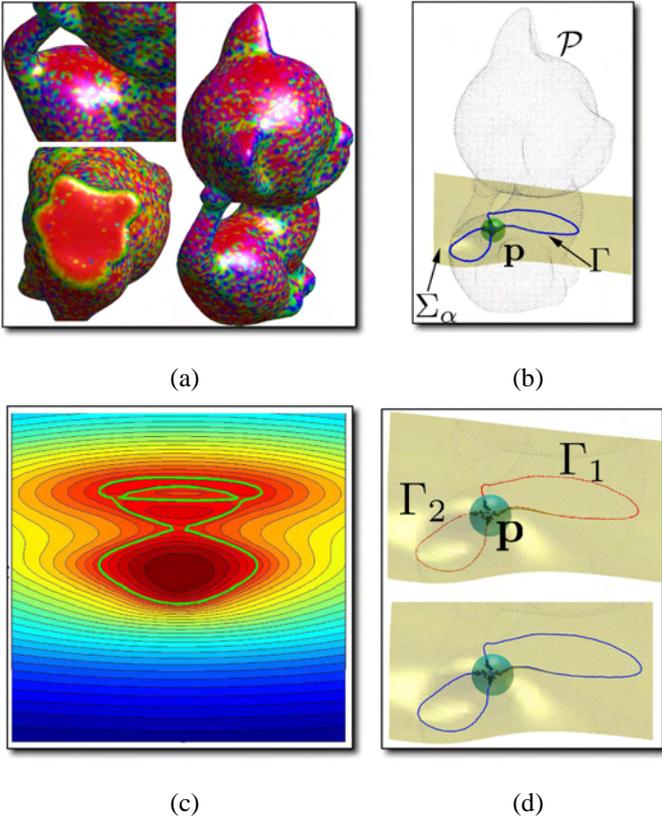


Fig. 19. (a) Distribution of the probability π on the input shape: the red regions located on the body and bottom part include the points that have been classified as fuzzy critical points. (b) Sampled iso-contour Γ at a saddle \mathbf{p} and osculating paraboloid (yellow surface). The sphere centered at \mathbf{p} is used to subdivide Γ into two sub-loops Γ_1 and Γ_2 . (c) Iso-contours of the scalar function \tilde{f} underlying the input map f ; \tilde{f} has been sampled on a square grid belonging to the least-squares plane that approximates Γ . (d) The two sampled components of Γ are updated around \mathbf{p} and converted to piecewise linear curves. The input point set has been sub-sampled to better visualize the iso-contour shape.

furthermore, as shown in Section 3.3 in the limit of large samples they provide the same results. The sampling density of \mathcal{P} , which is computed according to the description in Section 2, is the main parameter of the proposed approach and it controls the computation of the surface \mathcal{M} underlying \mathcal{P} as well as the definition of the steps of the iso-contouring algorithm. These steps are the level-set sampling, the counting of the number of connected components of each level-set, and the conversion of the iso-contours to piecewise linear curves. Indeed, it can be easily estimated and improved by using re-sampling techniques until a satisfactory rate is reached.

7. Conclusions and future work

This paper has studied the definition, analysis, and contouring of scalar functions on point-sampled surfaces. Given an arbitrary map f on a point set \mathcal{P} , the definition of the function \tilde{f} underlying f and the contouring algorithm of \tilde{f} build on the local connectivity structure of the k -nearest neighbor graph of \mathcal{P} . The analytical definition of \tilde{f} allowed us to provide an exact differential analysis of \tilde{f} and visualize its behavior

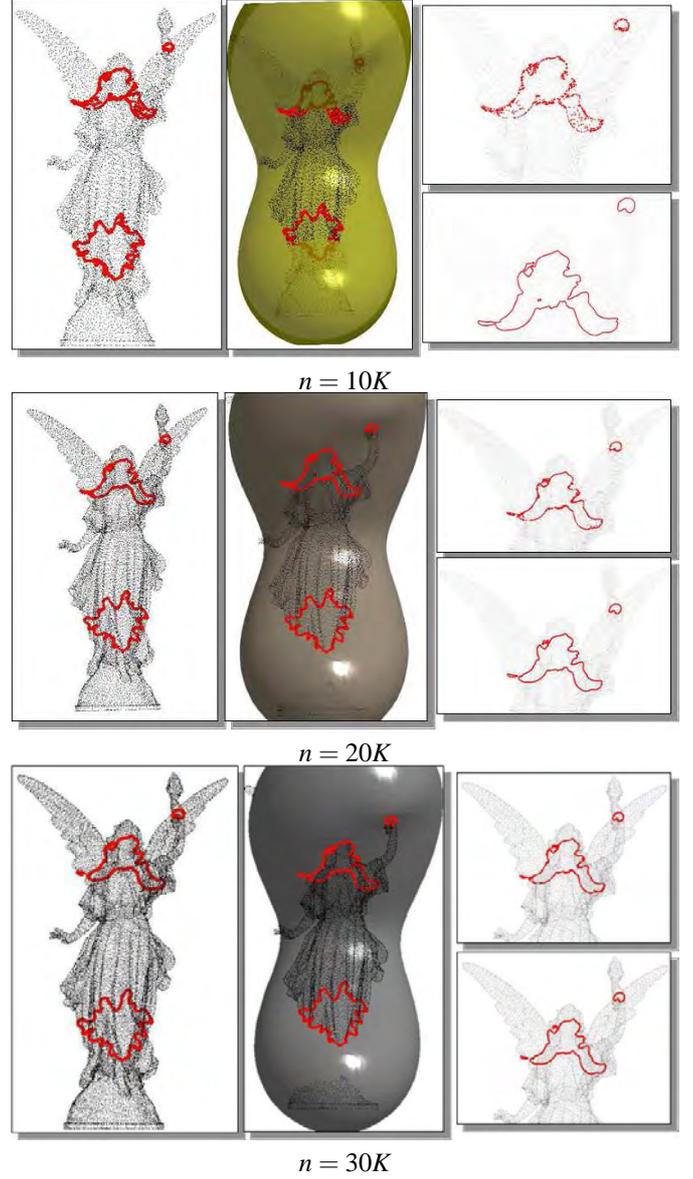


Fig. 20. Robustness of the iso-contouring algorithm with respect to a different surface sampling. (Left) Level-set samples, (middle) osculating paraboloid, and (right) piecewise linear reconstruction.

on and around the surface underlying \mathcal{P} . Since the stability of the critical point depends on the local noise that affects \mathcal{P} and the f -values, we have introduced the concept of fuzzy critical points. In this context, the analysis of the shape of the level-sets is intended to increase the reliability of the critical point classification. It is worth mentioning that the extracted level-sets can be used to visualize the behavior of f and compute a triangulated approximation of \mathcal{P} by applying surface reconstruction techniques from contour slices (12; 13). As future work, we plan to generalize the proposed approximation schemes to 3D scalar functions and use the iso-contouring algorithm for surface reconstruction and abstraction.

Acknowledgements. Special thanks are given to the anonymous reviewers for their numerous comments and precise sug-

Table 3

This table summarizes the main properties and the computational cost of the interpolation and approximation scheme of f . A full circle \bullet indicates that the corresponding property is satisfied; otherwise, an empty circle \circ appears.

Property	\tilde{f} w.r.t. H_p	MLS \tilde{f} (6)	\tilde{f} implicit (9)
Linearity	\circ	\bullet	\bullet
Least-squares definition	\bullet	\bullet	\bullet (interp)
Contin. w.r.t. surf. sampl.	\bullet	\bullet	\bullet
Higher order derivatives	\bullet	\bullet	\bullet
Stability w.r.t. noise	\bullet	\bullet	\bullet/\circ (high noise)
Computational cost	$O(k^3)$	$O(1)$	$O(k \log k)$

gestions, which helped us to improve the quality and contribution of this paper. The activities of Giuseppe Patanè have been partially funded by the GNCS-INdAM ‘‘F. Severi’’, ‘‘Young Researcher Programme’’. This work has been partially supported by FOCUS K3D Coordinating Action; models are courtesy of the AIM@SHAPE Repository and the Stanford 3D Scanning Repository.

References

- [1] A. Adamson and M. Alexa. Approximating and intersecting surfaces from points. In *ACM Symposium on Geometry Processing*, pages 230–239, 2003.
- [2] A. Adamson and M. Alexa. Ray tracing point set surfaces. In *IEEE Shape Modeling International*, pages 272–282, 2003.
- [3] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silva. Point set surfaces. In *IEEE Visualization*, pages 21–28, 2001.
- [4] M. Alexa, M. Gross, M. Pauly, H. Pfister, M. Stamminger, and M. Zwicker. Point-based computer graphics. In *ACM Siggraph Course Notes*, 2004.
- [5] N. Amenta and Y. Kil. The domain of a point set surface. *Symposium on Point-Based Graphics*, pages 139–148, 2004.
- [6] N. Amenta and Y. J. Kil. Defining point-set surfaces. In *ACM Siggraph*, pages 264–270, 2004.
- [7] E. Anderson and J. Dongarra. Implementation guide for LAPACK. LAPACK Working Note 18, Department of Computer Science, University of Tennessee, Knoxville, 1990.
- [8] N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68:337–404, 1950.
- [9] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *Journal of the ACM*, 45(6):891–923, 1998.
- [10] M. Attene and G. Patanè. Hierarchical structure recovery of point-sampled surfaces. *Computer Graphics Forum*, (29):1905–1920, 2010.
- [11] T. Banchoff. Critical points and curvature for embedded polyhedra. *Journal of Differential Geometry*, 1:245–256, 1967.
- [12] G. Barequet, D. Shapiro, and A. Tal. Multilevel sensitive reconstruction of polyhedral surfaces from parallel slices. *The Visual Computer*, 16(2):116–133, 2000.
- [13] G. Barequet and A. Vaxman. Nonlinear interpolation between slices. In *ACM Symposium on Solid and Physical Modeling*, pages 97–107, 2007.
- [14] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.
- [15] M. Belkin and P. Niyogi. Towards a theoretical foundation for laplacian based manifold methods. *Journal of Computer and System Sciences*, In press, 2007.
- [16] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434, 2006.
- [17] J. L. Bentley. Multidimensional binary search trees used for associative searching. *Communication of the ACM*, 18(9):509–517, 1975.
- [18] S. Biasotti, S. Marini, M. Mortara, G. Patanè, M. Spagnuolo, and B. Falcidieno. 3D shape matching through topological structures. In *Discrete Geometry and Computer Imagery*, pages 194–203, 2003.
- [19] S. Biasotti, S. Marini, M. Spagnuolo, and B. Falcidieno. Sub-part correspondence by structural descriptors of 3D shapes. *Computer-Aided Design*, 38(9):1002–1019, 2006.
- [20] J. Bloomenthal and B. Wyvill, editors. *Introduction to Implicit Surfaces*. Morgan Kaufmann Publishers Inc., 1997.
- [21] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans. Reconstruction and representation of 3D objects with radial basis functions. In *ACM Siggraph*, pages 67–76, 2001.
- [22] J. B. Cherrie, R. K. Beatson, and G. N. Newsam. Fast evaluation of radial basis functions: Methods for generalized multiquadrics in \mathbb{R}^n . *SIAM Journal Scientific Computation*, 23(5):1549–1571, 2002.
- [23] R. R. Coifman and S. Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5–30, 2006.
- [24] T. F. Coleman and Y. Li. An interior trust region approach for nonlinear minimization subject to bounds. Technical report, Ithaca, NY, USA, 1993.
- [25] T. Darom, M. R. Ruggeri, D. Saube, and N. Kiryati. Processing of textured surfaces represented as surfel sets: representation, compression and geodesic paths. In *IEEE Int. Conference on Image Processing*, pages 605–608, 2005.
- [26] T. K. Dey and J. Sun. An adaptive MLS surface for reconstruction with guarantees. In *ACM Symposium on Geometry Processing*, pages 43–52, 2005.
- [27] S. Dong, P.-T. Bremer, M. Garland, V. Pascucci, and J. C. Hart. Spectral surface quadrangulation. In *ACM Siggraph*, pages 1057–1066, 2006.
- [28] S. Dong, S. Kircher, and M. Garland. Harmonic functions for quadrilateral remeshing of arbitrary manifolds. *Computer Aided Geometric Design*, 22(5):392–423, 2005.
- [29] N. Dyn, D. Levin, and S. Rippa. Numerical procedures for surface fitting of scattered data by radial functions.

- SIAM Journal on Scientific and Statistical Computation*, 7(2):639–659, 1986.
- [30] S. Fleishman, D. Cohen-Or, M. Alexa, and C. T. Silva. Progressive point set surfaces. *ACM Transactions on Graphics*, 22(4):997–1011, 2003.
- [31] S. Fleishman, D. Cohen-Or, and C. T. Silva. Robust moving least-squares fitting with sharp features. *ACM Transactions on Graphics*, 24(3):544–552, 2005.
- [32] M. S. Floater. Analysis of curve reconstruction by meshless parameterization. *Numerical Algorithms*, (32):87–98, 2003.
- [33] M. S. Floater and K. Hormann. Surface parameterization: a tutorial and survey. In *Advances in Multiresolution for Geometric Modelling*, pages 157–186. 2005.
- [34] A. Fomenko and T. L. Kunii. *Topological Modelling for Visualization*. Springer Verlag, 1997.
- [35] G. Golub and G. VanLoan. *Matrix Computations*. John Hopkins University Press, 2nd. edition, 1989.
- [36] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *Journal of Computational Physics*, 73(2):325–348, 1987.
- [37] M. Gross, H. Pfister, M. Alexa, M. Pauly, M. Stamminger, and M. Zwicker. Point-based computer graphics. *Eurographics Tutorial*, 2002.
- [38] M. Hilaga, Y. Shinagawa, T. Kohmura, and T. L. Kunii. Topology matching for fully automatic similarity estimation of 3D shapes. In *ACM Siggraph*, pages 203–212, 2001.
- [39] V. Jain and H. Zhang. Robust 3D shape correspondence in the spectral domain. In *IEEE Shape Modeling and Applications*, page 19, 2006.
- [40] A. Kalaiah and A. Varshney. Modeling and rendering of points with local geometry. *IEEE Transactions on Visualization and Computer Graphics*, 9(1):30–42, 2003.
- [41] S. Lafon, Y. Keller, and R. R. Coifman. Data fusion and multicue data matching by diffusion maps. *IEEE Transactions Pattern Analysis Machine Intelligence*, 28(11):1784–1797, 2006.
- [42] C. Lange and K. Polthier. Anisotropic smoothing of point sets. *Computer Aided Geometric Design*, 22(7):680–692, 2005.
- [43] D. Levin. Mesh-independent surface interpolation. *Geometric Modeling for Scientific Visualization*, 2003.
- [44] L. Liu, C. Bajaj, J. Deasy, D. A. Low, and T. Ju. Surface reconstruction from non-parallel curve networks. *Computer Graphics Forum*, 27(2):155–163, 2008.
- [45] B. Mederos, L. Velho, and L. H. de Figueiredo. Moving least squares multiresolution surface approximation. *SibGrapi*, pages 19–26, 2003.
- [46] F. Memoli and G. Sapiro. Distance functions and geodesics on submanifolds of \mathbb{R}^d and point clouds. *SIAM Journal of Applied Mathematics*, 65(4):1227–1260, 2005.
- [47] F. Memoli and G. Sapiro. A theoretical and computational framework for isometry invariant recognition of point cloud data. *Foundations of Computational Mathematics*, 5(3):313–347, 2005.
- [48] C. A. Micchelli. Interpolation of scattered data: Distance matrices and conditionally positive definite functions. *Constructive Approximation*, 2:11–22, 1986.
- [49] J. Milnor. *Morse Theory*, volume 51 of *Annals of mathematics studies*. Princeton University Press, 1963.
- [50] N. J. Mitra and A. Nguyen. Estimating surface normals in noisy point cloud data. In *Symposium on Computational Geometry*, pages 322–328, 2003.
- [51] B. S. Morse, T. S. Yoo, D. T. Chen, P. Rheingans, and K. R. Subramanian. Interpolating implicit surfaces from scattered surface data using compactly supported radial basis functions. In *IEEE Shape Modeling and Applications*, pages 89–98, 2001.
- [52] X. Ni, M. Garland, and J. C. Hart. Fair morse functions for extracting the topological structure of a surface mesh. In *ACM Siggraph*, pages 613–622, 2004.
- [53] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, and H.-P. Seidel. Multi-level partition of unity implicits. *ACM Siggraph*, 22(3):463–470, 2003.
- [54] Y. Ohtake, A. Belyaev, and H.-P. Seidel. 3D scattered data interpolation and approximation with multilevel compactly supported RBFs. *Graphical Models*, 67(3):150–165, 2005.
- [55] V. Pascucci and K. Cole-McLaughlin. Efficient computation of the topology of level sets. In *IEEE Visualization*, pages 187–194, 2002.
- [56] V. Pascucci and K. Cole-McLaughlin. Parallel computation of the topology of level sets. *Algorithmica*, 38(1):249–268, 2003.
- [57] G. Patanè, M. Spagnuolo, and B. Falcidieno. Para-graph: graph-based parameterization of triangle meshes with arbitrary genus. *Computer Graphics Forum*, 23(4):783–797, 2004.
- [58] G. Patanè, M. Spagnuolo, and B. Falcidieno. Topology- and error-driven extension of scalar functions from surfaces to volumes. *ACM Transactions on Graphics*, 29(1):1–20, 2009.
- [59] M. Pauly and M. Gross. Spectral processing of point-sampled geometry. In *ACM Siggraph*, pages 379–386, 2001.
- [60] M. Pauly, M. Gross, and L. P. Kobbelt. Efficient simplification of point-sampled surfaces. In *IEEE Visualization*, pages 163–170, 2002.
- [61] M. Pauly, L. P. Kobbelt, and M. Gross. Point-based multiscale surface representation. *ACM Transactions on Graphics*, 25(2):177–193, 2006.
- [62] H. Pfister, M. Zwicker, J. van Baar, and M. Gross. Surfels: surface elements as rendering primitives. In *ACM Siggraph*, pages 335–342, 2000.
- [63] T. Poggio and F. Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, 78(9):1481–1497, 1990.
- [64] M. Reuter, F.-E. Wolter, and N. Peinecke. Laplace-Beltrami spectra as Shape-DNA of surfaces and solids. *Computer-Aided Design*, 38(4):342–366, 2006.
- [65] M. R. Ruggeri, T. Darom, D. Saube, and N. Kiryati. Approximating geodesics on point set surfaces. In *Symposium on Point-based Graphics 2006*, pages 85–94, 2006.

- [66] S. Rusinkiewicz and M. Levoy. Qsplat: a multiresolution point rendering system for large meshes. In *ACM Siggraph*, pages 343–352, 2000.
- [67] B. Schölkopf, J. Giesen, and S. Spalinger. Kernel methods for implicit surface modeling. In *Advances in Neural Information Processing Systems 17*, pages 1193–1200. MIT Press, 2005.
- [68] C. Shen, J. F. O’Brien, and J. R. Shewchuk. Interpolating and approximating implicit surfaces from polygon soup. In *ACM Siggraph 2005 Courses*, page 204, 2005.
- [69] A. Tagliasacchi, H. Zhang, and D. Cohen-Or. Curve skeleton extraction from incomplete point cloud. *ACM Transactions on Graphics*, 28(3):1–9, 2009.
- [70] G. Turk and J. F. O’Brien. Modelling with implicit surfaces that interpolate. *ACM Siggraph*, 21(4):855–873, 2002.
- [71] H. Wendland. Real piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Advances in Computational Mathematics*, 4(4):389–396, 1995.
- [72] H. Xie, K. T. McDonnell, and H. Qin. Surface reconstruction of noisy and defective data sets. In *IEEE Visualization*, pages 259–266, 2004.
- [73] H. Xie, J. Wang, J. Hua, H. Qin, and A. Kaufman. Piecewise C^1 continuous surface reconstruction of noisy point clouds via local implicit quadric regression. In *IEEE Visualization*, page 13, 2003.
- [74] P. Yang and X. Qian. Direct computing of surface curvatures for point-set surfaces. In *Symposium on Point-based Graphics*, pages 605–608, 2007.
- [75] E. Zhang, K. Mischaikow, and G. Turk. Feature-based surface parameterization and texture mapping. *ACM Transactions on Graphics*, 24(1):1–27, 2005.
- [76] M. Zwicker, H. Pfister, J. van Baar, and M. Gross. Ewa splatting. *IEEE Transactions on Visualization and Computer Graphics*, 8(3):223–238, 2002.