Variational Implicit Point Set Surfaces

ZHIYANG HUANG, Washington University in St. Louis, USA
NATHAN CARR, Adobe Systems, USA
TAO JU, Washington University in St. Louis, USA

We propose a new method for reconstructing an implicit surface from an un-oriented point set. While existing methods often involve non-trivial heuristics and require additional constraints, such as normals or labelled points, we introduce a direct definition of the function from the points as the solution to a constrained quadratic optimization problem. The definition has a number of appealing features: it uses a single parameter (parameter-free for exact interpolation), applies to any dimensions, commutes with similarity transformations, and can be easily implemented without discretizing the space. More importantly, the use of a global smoothness energy allows our definition to be much more resilient to sampling imperfections than existing methods, making it particularly suited for sparse and non-uniform inputs.

CCS Concepts: • **Computing methodologies → Point-based models**; *Volumetric models.*

Additional Key Words and Phrases: surface reconstruction, implicit surfaces, point clouds, radial basis functions

ACM Reference Format:
Zhiyang Huang, Nathan Carr, and Tao Ju. 2019. Variational Implicit Point Set Surfaces. *ACM Trans. Graph.* 38, 4, Article 124 (July 2019), 14 pages. https://doi.org/10.1145/3306346.3322994

1 INTRODUCTION

Constructing a curve or surface that interpolates or approximates a given set of 2D or 3D points is one of the fundamental problems in geometric modeling. A common representation of the reconstructed output is the zero-level set of some smooth implicit function. This representation naturally ensures a smooth and closed manifold. In addition, an implicit function enables a range of applications such as boolean operations and collision detection.

While extensively studied [Berger et al. 2017], implicit modeling from points remains a difficult problem. A fundamental challenge is that the constant zero function, although meaningless for reconstruction, perfectly meets our goal: the function is smooth and its zero-level set (which includes the entire space) interpolates any input points. A common approach to avoid this trivial solution is to introduce additional constraints, such as normals at the input points or additional spatial locations with inside/outside labels. If such constraints are not available as part of the input or provided by the user, they will need to be inferred from the input data *prior to* reconstruction.

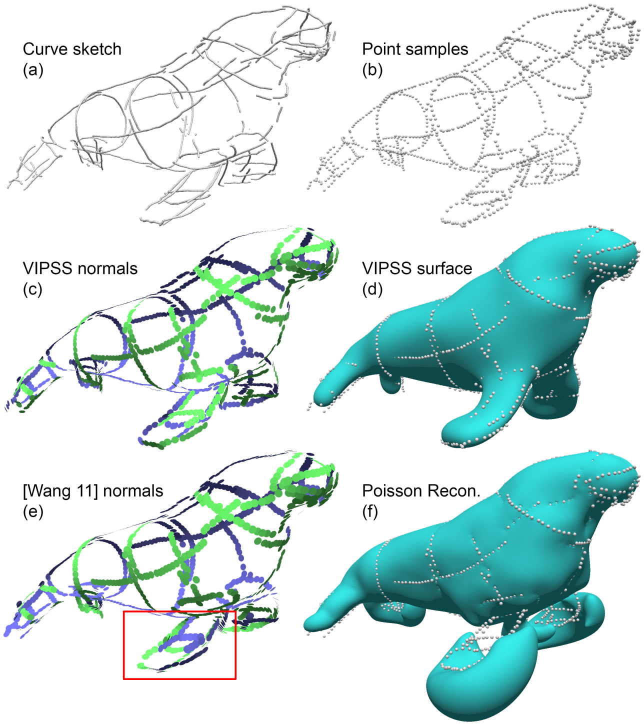

Fig. 1. Given sparse, non-uniform, noisy and un-oriented points (b) sampled from a set of unstructured 3D curves (a), our variational definition (VIPSS with $\lambda = 0.003$) simultaneously produces oriented normals (c) and a smooth approximating surface (d). The input is challenging for state-of-the-art normal estimation methods such as [Wang et al. 2011], which fails around sparsely sampled thin features (the flippers) (e). Incorrect normals lead to poor reconstructions using existing implicit methods such as Screened Poisson [Kazhdan and Hoppe 2013] (f, fitting weight $\alpha = 0.5$).

This two-stage paradigm - constraint estimation followed by reconstruction - has a number of drawbacks. The use of multiple disparate methods, each carrying its own set of parameters, makes parameter-tuning a challenging task and complicates the analysis (e.g., how the output changes with the input). More importantly, methods for constraint estimation are completely unaware of the quality of the reconstructed surface. Lacking any better guidance, current estimation methods (e.g., for normals) rely on local point neighborhoods, which are often unreliable when the points are sparse or non-uniformly distributed. Errors in constraint estimation, in turn, lead to poorly reconstructed surfaces (e.g., Figure 1 (e,f)).

We propose a direct definition of an implicit function from an un-oriented point set. Unlike the above-mentioned two-stage paradigm, our definition integrates constraint estimation and surface

Authors' addresses: Zhiyang Huang, Washington University in St. Louis, St. Louis, MO, USA; Nathan Carr, Adobe Systems, San Jose, CA, USA; Tao Ju, Washington University in St. Louis, St. Louis, MO, USA.

0730-0301/2019/7-ART124 $15.00
https://doi.org/10.1145/3306346.3322994

reconstruction within a single variational formulation. Specifically, *we seek an implicit function f minimizing some smoothness energy, such that the input points lie close to the zero-level set of f and their gradients of f have unit magnitude.* The rationale for constraining unit gradient is two fold. First, unlike prescribing normals, this constraint does not require a separate estimation step. Second, such constraint encourages the function to behave like a signed distance function around the zero-level set. By a judicious choice of the smoothness energy [Duchon 1977], we show that our definition can be expressed as a standard constrained quadratic optimization problem with closed-form coefficients. The variational problem can be solved using off-the-shelf optimization packages without the need to discretize the domain.

Our method has a number of advantages over the previous two-stage paradigm. First, as an explicit definition, we can analyze properties of the reconstruction as a function of the input points. In this paper, we show that the surface given by our definition reproduces linear geometry and commutes with similarity transformations (translation, rotation, uniform scaling). Second, our definition involves only a single parameter (λ) that controls the accuracy of approximation. If exact interpolation is desired, reconstruction is completely parameter-free for any point set by setting $\lambda = 0$. Third, we observed that the surfaces produced by our definition are much more resilient to sparse or non-uniform samples than existing methods. We owe such robustness to our integrated formulation and the chosen smoothness energy [Duchon 1977], which considers the global shape of the reconstructed implicit function instead of local neighborhood of points.

The main limitation of our method is its computational complexity, which scales cubically with the number of points. While we are actively exploring means to improve scalability, we demonstrate the usefulness of our method in one application (surfacing unstructured 3D sketches) which result in sparse, non-uniform, un-oriented samples that are challenging for existing reconstruction methods (Figure 1).

Contributions. We make the following technical contributions:

- We introduce a variational definition of an implicit surface directly from un-oriented points. The definition has a single parameter, applies to any dimensions, and does not need domain discretization or numerical integration (Section 3).
- We show several theoretical properties of the definition, including exact interpolation, reproducing linear geometry, and commuting with similarity transformations (Section 4).
- We propose an effective strategy for initializing the optimization (Section 5.1).

2 RELATED WORKS

2.1 Surface reconstruction from points

We review the most relevant methods for surface reconstruction from 3D point sets. For more in-depth and comprehensive discussions of these and other methods, we refer readers to the latest survey [Berger et al. 2017].

2.1.1 Combinatorial methods. One class of reconstruction algorithms directly produce a triangulated surface whose vertices are the input points. These methods are typically based on the Delaunay triangulation of the points or its dual, the Voronoi Diagram (see survey [Dey 2006]). Compared with implicit methods, the smoothness of the surfaces created by combinatorial methods is limited by the input sampling density. Also, a water-tight surface is not always guaranteed. Furthermore, as these methods usually work under the assumption that the input samples are dense enough with respect to the local shape, sparse or non-uniform sampling may lead to significantly degraded reconstruction quality (see Figure 11).

2.1.2 Implicit methods. We broadly classify implicit reconstruction methods into three types: ones that require "signed" input (e.g., oriented normals or additional labelled points), ones that perform signing in a post-process, and ones that do neither.

Requiring signed input. Most implicit methods require the input points to be equipped with oriented normals. The Poisson reconstruction method [Kazhdan et al. 2006] and its variants [Kazhdan and Hoppe 2013; Manson et al. 2008; Pan and Skala 2012; Taubin 2012] seek an "indicator function" that is 1 (resp. 0) in the interior (resp. exterior) of the shape and whose gradient near the shape's boundary agrees with the given normals. The implicit moving least squares (IMLS) method defines a local polynomial, such as a constant [Dey and Sun 2005; Kolluri 2008; Öztireli et al. 2009; Shen et al. 2004] or an algebraic sphere [Guennebaud and Gross 2007], at each spatial location that fits nearby samples and normals. Note that these methods are different from another group of methods (also called moving-least squares) that employ a projection operator [Alexa et al. 2003], whose result is generally not a level set of an implicit function [Amenta and Kil 2004]. IMLS using a constant polynomial effectively blends linear functions defined by the tangent planes at the input points. The same strategy has been used in several other methods with different choices of the blending weights [Barill et al. 2018; Boissonnat and Cazals 2002; Hoppe et al. 1992; Lu et al. 2018]. More generally, the partition-of-unity method [Ohtake et al. 2003a] blends polynomials that are fitted to groups of points.

Another tool for implicit reconstruction from signed input is radial basis functions (RBF). RBF interpolants for scattered data have been extensively studied in the literature [Buhmann 2003; Wendland 2004]. Unlike IMLS and partition-of-unity, an RBF interpolant is a linear combination of *fixed* radial basis kernels (typically centered at the input points) blended with a *fixed* set of weights chosen so that the given values at the input points are interpolated. To avoid the trivial interpolant of constant zero, most reconstruction methods that use RBF introduce additional spatial locations equipped with signed values [Carr et al. 2001; Dinh et al. 2002; Morse et al. 2001; Ohtake et al. 2003b; Samozino et al. 2006; Turk and O'Brien 2002; Walder et al. 2007]. These locations are often created by offseting along a sample's normal. However, care must be taken in determining the offset amount, particularly near thin features. Alternatively, some authors [Brazil et al. 2010; Ijiri et al. 2013; Liu et al. 2016] apply a Hermite variant of the RBF interpolant [Duchon 1977] directly to the input points and normals.

All methods in this class interpolate or closely approximate the normals at the input points. As a result, their reconstruction quality relies heavily on the accuracy of normal estimation. In this work, we use the Hermite RBF interpolant of [Duchon 1977] (which we call

Duchon's interpolant and will discuss in details in Section 3.2). However, our variational formulation removes the need for estimating the normals.

Signing in a post-process. A number of methods first create an unsigned distance function from the input points and then determine the sign afterwards. Various strategies were used for signing, including graph-cut [Hornung and Kobbelt 2006], ray-shooting [Mullen et al. 2010], watershed [Poranne et al. 2010], and energy minimization [Giraudot et al. 2013]. While these methods can be applied directly to un-oriented points, the heuristic nature of the signing step makes it difficult to analyze the results of these methods. Also, the two-stage pipeline involves several parameters that need to be carefully tuned.

Variational methods. Like our method, several other methods directly reconstruct a signed function from un-oriented points using some variational formulation. Zhao and colleagues [Lu et al. 2005; Zhao et al. 2001] employ the level-set method to minimize an energy that balances fitting error and surface curvature, resulting in a weighted minimal surface. After computing un-oriented normals at the samples, Alliez et al. [2007] seek an implicit function whose gradients best align with the un-oriented normals under the constraint that the weighted sum of biharmonic energy and fitting error has unit norm. Both Scholkopf et al. [2004] and Walder et al. [2005] propose un-constrained formulations using scalar (non-Hermite) RBF interpolants. Scholkopf et al. minimize a weighted sum of three energy terms. Walder et al. point out that Scholkopf's formulation leads to ill-shaped functions, and they replaced the last component of Scholkopf's objective by two more energy terms.

The above formulations (except for the level-set methods) all involve multiple parameters for balancing the various energy terms and/or constraints. Also, solving for these variational problems may require domain discretization (e.g., [Alliez et al. 2007] and level-set methods) or numerical integration (e.g., [Walder et al. 2005]), which introduces additional parameters as well as dependencies on the discretization structure or resolution. In contrast, our definition has a single parameter (which is fixed in the interpolation mode), and no discretization is needed for optimization. We also observed that our formulation tends to behave more robustly under sparse sampling (see Figure 11).

2.2 Normal estimation

As mentioned above, normal estimation is required by many implicit reconstruction methods. Existing estimation methods can be classified into two types, ones that separately estimate the direction and orientation (i.e., forward or backward) of normals, and ones that estimate both at once.

Estimation and orientation of un-oriented normals. A common strategy for estimating un-oriented normal directions is by fitting the local neighborhood of a point with a function. Linear functions are most common [Hoppe et al. 1992] (also known as the PCA method), but higher-order polynomials have also been used [Cazals and Pouget 2003; Guennebaud and Gross 2007]. It is well-known that the choice of the point neighborhood is crucial to the accuracy of estimation, and various proposals were made to deal robustly with noisy samples [Mitra et al. 2004; Pauly et al. 2003] and sharp features [Boulch and Marlet 2012; Li et al. 2010; Liu et al. 2015]. Another strategy is based on analyzing the shape of individual [Amenta and Bern 1999; Dey and Goswami 2004] or a group of [Alliez et al. 2007; Merigot et al. 2011] Voronoi cells. While many of these methods can successfully handle noisy samples, they tend to fail when the sampling rate is low or the pattern is not uniform (see Figure 8).

Orienting the un-oriented normals can be treated as a combinatorial optimization problem. The choice of energy varies from simple consistency among neighboring normals [Hoppe et al. 1992] to more sophisticated ones that better handle thin features [Huang et al. 2009; König and Gumhold 2009; Xie et al. 2003]. The energy can be minimized by a minimal spanning tree [Hoppe et al. 1992] or a global solver [Schertler et al. 2017]. However, if the un-oriented normal has a wrong direction (e.g., orthogonal to the true normal), it cannot be corrected by the orientation step.

Direct estimation of oriented normals. Wang et al. [2011] proposed a variational method that estimates both the direction and orientation of normals in a single step. They formulate a quadratic optimization problem that minimizes the weighted sum of two energy terms on each pair of nearby normals, a consistency energy similar to [Hoppe et al. 1992] and an orthogonality energy. In our experiments, we found this method outperforms the two-step methods mentioned above for sparse and non-uniform samples, but tuning its parameters can be challenging (see Section 6.2 and Figure 9). While our definition solves a similar quadratic optimization problem, our objective captures the global regularity of the implicit function, which leads to more robust results while removing the need for parameter tuning.

Recently, deep learning has been employed to infer oriented normals from point clouds [Boulch and Marlet 2016; Guerrero et al. 2018]. These methods can produce impressive results on densely sampled points contaminated with noise. However, we found that they are less successful on sparse samples (see Figure 9).

3 DEFINITION

We introduce our definition of the point set surface in this section. We first give a general definition for any choice of smoothness energy, and then specialize it to a particular energy that leads to a simple and computable definition.

Notations. In this paper, scalar values are italicized (e.g., x), vector values are bold (e.g., $\mathbf{x}$), and matrices are capitalized (e.g., M). All vectors are assumed to be column vectors. We use D^i to denote taking the i-th derivative (or gradient, if the variable is a vector), and $D = D^1$. For a two variable function $f(\cdot,\cdot)$, we use $D^{i,j}$ to denote taking the i-th partial derivative of the first variable and j-th partial derivative of the second variable. $\|\cdot\|$ denotes the L_2 norm.

3.1 A general definition

Given a set of points $\mathbf{x}_i \in \mathbb{R}^d$ $(i = 1, \ldots, n)$, we wish to define a smooth implicit function $f(\mathbf{x})$ whose zero-level set is as close to $\mathbf{x}_i$ as possible. As mentioned earlier, to avoid the trivial solution $f \equiv 0$, we need an additional constraint. To this end, we note that existing implicit modeling methods either look for a true signed distance function, or an "indicator function" that is close to a signed distance function near the shape boundary. In the same spirit, we require that the gradient of f should have *unit magnitude* at each input point. Unlike signed constraints, such as normals or labelled points, the unit-gradient constraint does not need to be provided or estimated *a priori*. Note that the same constraint has been used previously for fitting implicit primitives such as spheres [Guennebaud and Gross 2007; Pratt 1987].

We arrive at the following variational definition of an implicit point set surface:

Definition 3.1. The *variational implicit point set surface* (VIPSS) of points $\mathbf{x}_i \in \mathbb{R}^d$, for a chosen energy E and approximation parameter λ, is the zero-level set of a function $f \in \mathbb{R}^d$ that

$$\begin{aligned} \text{Minimizes:} \quad & \textstyle\sum_i f(\mathbf{x}_i)^2 + \lambda\, E(f) \\ \text{Subject to:} \quad & \|Df(\mathbf{x}_i)\| = 1, \;\; \forall i \end{aligned} \tag{1}$$

Here, λ balances the two goals of data fitting and smoothness. A larger λ leads to a smoother surface at the cost of a less accurate approximation of the input points.

A key ingredient of this definition is the energy E, which needs to be appropriately chosen for the definition to be meaningful and practical. On one hand, lower E values should correspond to a smoother zero-level set of f. This requires an energy of at least 2nd-order, so that linear functions (whose level sets are hyperplanes, which are perfectly smooth) have zero energy. On the other hand, the energy should make the variational problem computable, and ideally not requiring any discretization of the domain.

We describe one class of energy E that makes Definition 3.1 computation-friendly without discretization. Suppose we are given an additional set of Hermite data $\mathbf{s} = \{s_i\}$ and $\mathbf{g} = \{\mathbf{g}_i\}$, so that each point $\mathbf{x}_i$ is equipped with a scalar s_i and a (possibly un-normalized) vector $\mathbf{g}_i$. Let $f_{\mathbf{s},\mathbf{g}}$ be the interpolating function with minimal energy,

$$f_{\mathbf{s},\mathbf{g}} = \arg\min_f \{E(f) |\, f(\mathbf{x}_i) = s_i,\; Df(\mathbf{x}_i) = \mathbf{g}_i, \forall i\}$$

The solution to the constrained optimization problem of (1) is therefore the interpolant $f_{\mathbf{s},\mathbf{g}}$ for the Hermite data $\{\mathbf{s}, \mathbf{g}\}$ that

$$\begin{aligned} \text{Minimizes:} \quad & \mathbf{s}^T\mathbf{s} + \lambda\, E(f_{\mathbf{s},\mathbf{g}}) \\ \text{Subject to:} \quad & \mathbf{g}_i{}^T\mathbf{g}_i = 1, \;\; \forall i \end{aligned} \tag{2}$$

Conceptually, while $f_{\mathbf{s},\mathbf{g}}$ smoothly interpolates a *given* Hermite data, we look for the *best* Hermite data (under the unit-vector constraint) that results in the smoothest interpolant $f_{\mathbf{s},\mathbf{g}}$. If energy E is chosen such that the energy-minimizing Hermite interpolant $f_{\mathbf{s},\mathbf{g}}$, as well as its energy $E(f_{\mathbf{s},\mathbf{g}})$, can be expressed in closed-form by $\mathbf{s}$ and $\mathbf{g}$, then (2) becomes a constrained optimization problem on a finite variable set $(\mathbf{s}, \mathbf{g})$.

Next, we review one choice of E that has the above characteristics, namely being (at least) 2nd-order and that the energy-minimizing Hermite interpolant has closed form (Section 3.2). With this choice, we will show that Definition 3.1 becomes a constrained quadratic optimization problem (Section 3.3).

3.2 Duchon's energy

Our formulation builds upon the Hermite interpolant first developed by Duchon [1977] using radial basis functions (some authors have called it the *Hermite RBF interpolant*). We briefly review its definition and properties, and refer the reader to more in-depth discussions in [Wendland 2004].

To interpolate Hermite data $\{s_i, \mathbf{g}_i\}$ at a set of locations $\mathbf{x}_i$ in $\mathbb{R}^d$, Duchon's interpolant consists of a linear combination of radial basis kernels $\phi(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|^3$ (known as the triharmonic kernel for $d = 3$), their derivatives, and other low-degree terms:

$$f_{\mathbf{s},\mathbf{g}}(\mathbf{x}) = \sum_i a_i \phi(\mathbf{x}, \mathbf{x}_i) + \sum_i \mathbf{b}_i^T D^{0,1}\phi(\mathbf{x}, \mathbf{x}_i) + \mathbf{c}^T\mathbf{x} + d \tag{3}$$

where $a_i \in \mathbb{R}, \mathbf{b}_i \in \mathbb{R}^d, \mathbf{c} \in \mathbb{R}^d, d \in R$ are constants determined by $\{s_i, \mathbf{g}_i\}$. Specifically, these constants need to satisfy the interpolatory conditions ($f_{\mathbf{s},\mathbf{g}}(\mathbf{x}_i) = s_i, Df_{\mathbf{s},\mathbf{g}}(\mathbf{x}_i) = \mathbf{g}_i$ for all i) and additional orthogonality conditions including $\sum_i a_i = 0$ and $\sum_i a_i\mathbf{x}_i + \sum_i \mathbf{b}_i = 0$ to ensure the existence of a unique solution. All of these conditions can be expressed by a system of linear equations

$$A \begin{pmatrix} \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \\ d \end{pmatrix} = \begin{pmatrix} \mathbf{s} \\ \mathbf{g} \\ \mathbf{0} \\ 0 \end{pmatrix} \tag{4}$$

where $\mathbf{a} = \{a_i\}$, $\mathbf{b}$ is the flattened array of $\{\mathbf{b}_i\}$ of length $d \times n$, and $\mathbf{g}$ is the flattened array of $\mathbf{g}_i$ of length $d \times n$. The coefficient matrix A of this system, called the *interpolation matrix*, has the form

$$A = \begin{pmatrix} M & N \\ N^T & \mathbf{0} \end{pmatrix}, \;\; M = \begin{pmatrix} M_{00} & M_{01} \\ M_{01}^T & M_{11} \end{pmatrix}, \;\; N = \begin{pmatrix} N_0 & \mathbf{1} \\ N_1 & \mathbf{0} \end{pmatrix} \tag{5}$$

Here, the matrices M_{00}, M_{01}, M_{11} have dimensions $n \times n$, $n \times dn$, and $dn \times dn$ respectively. Each (i, j)-th entry (or block) of matrix $M_{\alpha\beta}$ where $\alpha, \beta \in \{0, 1\}$ is the differential $D^{\alpha,\beta}\phi(\mathbf{x}_i, \mathbf{x}_j)$. Note that both M_{00}, M_{11} are symmetric matrices. N_0 has dimension $n \times d$ and its i-th row is $\mathbf{x}_i^T$. N_1 has dimension $dn \times d$ and consists of n identity matrices of dimension $d \times d$.

Assuming that the points $\mathbf{x}_i$ are pairwise disjoint, the matrix A is always invertible and hence the constants $\mathbf{a}, \mathbf{b}, \mathbf{c}, d$ satisfying (4) uniquely exist. In one dimension ($d = 1$), the resulting interpolant $f_{\mathbf{s},\mathbf{g}}$ coincides with the ordinary piecewise cubic Hermite interpolation. In this sense, Duchon's interpolant can be considered as a generalization of cubic Hermite interpolation to arbitrary dimensions. Figure 2 (a) gives examples of the interpolant in 1D and 2D (with $s_i = 0$ and $\mathbf{g}_i$ set to a constant for all i).

A key feature of Duchon's interpolant is that it is optimal with respect to a differential energy, which we call *Duchon's energy*. In one dimension ($d = 1$), Duchon's energy of a function f is the same as the classical thin plate energy ($\int_{\mathbb{R}} \|D^2 f(x)\|^2 dx$). In higher dimensions, the energy assumes a similar 2nd-order form but on the Fourier transform of f (we refer the reader to [Duchon 1977] for the precise form). It can be shown that, among all functions in a semi-Hilbert space that all interpolate the same Hermite data, the interpolant in (3) has the minimal Duchon's energy. This energy is in fact the semi-norm in this function space, which is equipped with a semi-inner product defined by the radial basis kernels ϕ and their

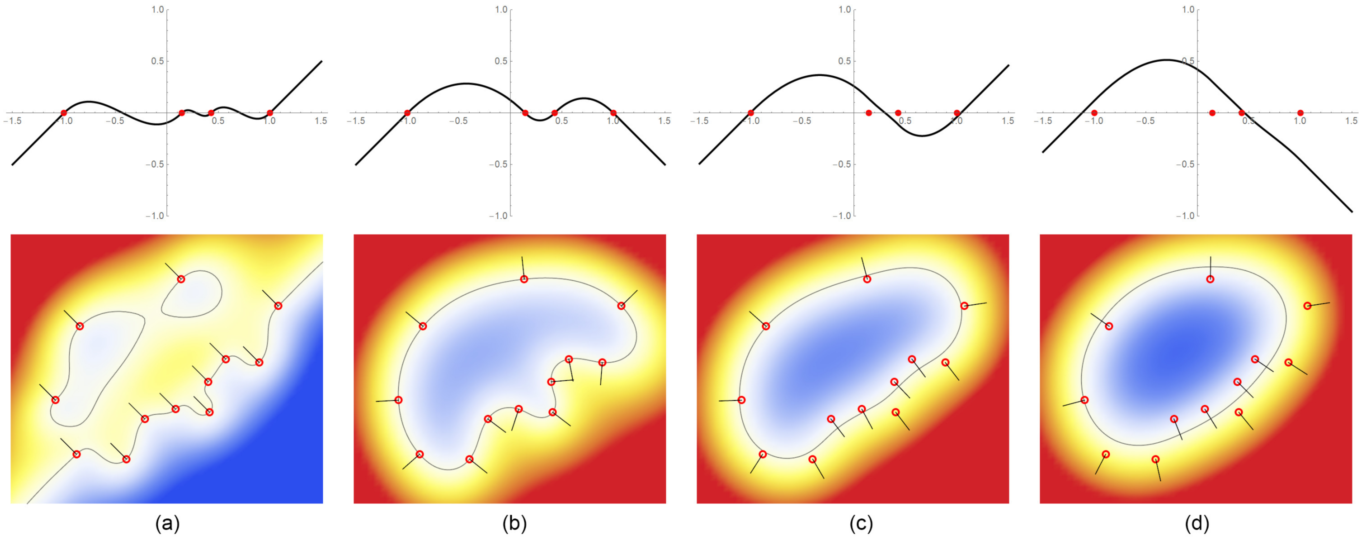

Fig. 2. Examples of Duchon's interpolants $f_{\mathbf{s},\mathbf{g}}$ that interpolate scattered points in 1D (top, red dots) and 2D (bottom, red circles) for different choices of the Hermite data $\{\mathbf{s}, \mathbf{g}\}$. In (a), $s_i = 0$ and $\mathbf{g}_i$ is a constant vector at each point. In (b,c,d), $\{\mathbf{s}, \mathbf{g}\}$ are obtained by our variational formulation (8) with $\lambda = 0, 0.1, 1.0$ respectively. The zero-level set in (b,c,d) (black curves) is the VIPSS at the respective λ.

derivatives. The semi-norm (or Duchon's energy) of the interpolant in (3) has the form

$$E(f_{\mathbf{s},\mathbf{g}}) = \begin{pmatrix} \mathbf{a}^T & \mathbf{b}^T \end{pmatrix} M \begin{pmatrix} \mathbf{a} \\ \mathbf{b} \end{pmatrix} \tag{6}$$

3.3 Definition using Duchon's energy

We now specialize the VIPSS definition (3.1) to the case where E is chosen as Duchon's energy. That is, we look for the scalars $\mathbf{s}$ and unit vectors $\mathbf{g}$ that minimize the energy objective in (2) where $E(f_{\mathbf{s},\mathbf{g}})$ is defined in (6).

While Duchon's energy in (6) is expressed by the coefficients $\mathbf{a}, \mathbf{b}$ of the interpolant, we need to transform it to an expression on the unknown Hermite data $\{\mathbf{s}, \mathbf{g}\}$. This is made possible by observing that the two sets of quantities are linearly related, as indicated by (4), via the interpolation matrix A. We first write the inverse of A as

$$A^{-1} = \begin{pmatrix} J & K \\ K^T & L \end{pmatrix}, \quad J = \begin{pmatrix} J_{00} & J_{01} \\ J_{01}^T & J_{11} \end{pmatrix} \tag{7}$$

where matrices $J, K, J_{00}, J_{01}, J_{11}$ have the respective sizes as M, N, M_{00}, M_{01}, M_{11} in (5), and J, L, J_{00}, J_{11} are symmetric. We now present our main result:

Proposition 3.2. *The VIPSS of points $\mathbf{x}_i$, where E is Duchon's energy, is the zero-level set of $f_{\mathbf{s},\mathbf{g}}$ defined in (3) for Hermite data $\{\mathbf{s}, \mathbf{g}\}$, such that* $\mathbf{g}$

$$\begin{aligned} &\textit{Minimizes:} && \mathbf{g}^T H \mathbf{g} \\ &\textit{Subject to:} && \mathbf{g}_i{}^T \mathbf{g}_i = 1, \quad \forall i \end{aligned} \tag{8}$$

where

$$H = J_{11} - \lambda J_{01}^T (I + \lambda J_{00})^{-1} J_{01} \tag{9}$$

and $\mathbf{s}$ *is obtained from* $\mathbf{g}$ *by*

$$\mathbf{s} = -\lambda (I + \lambda J_{00})^{-1} J_{01}\, \mathbf{g} \tag{10}$$

Proof. Using notations in (7), we have the linear relation

$$\begin{pmatrix} \mathbf{a} \\ \mathbf{b} \end{pmatrix} = J \begin{pmatrix} \mathbf{s} \\ \mathbf{g} \end{pmatrix} \tag{11}$$

Substituting into (6) yields an expression of energy in terms of $\{\mathbf{s}, \mathbf{g}\}$,

$$E(f_{\mathbf{s},\mathbf{g}}) = \begin{pmatrix} \mathbf{s}^T & \mathbf{g}^T \end{pmatrix} J\, M\, J \begin{pmatrix} \mathbf{s} \\ \mathbf{g} \end{pmatrix} \tag{12}$$

To simplify this expression further, note that by $A^{-1}A = I$ we have $J\,M + K\,N^T = I$ and $J\,N = \mathbf{0}$. Therefore,

$$\begin{aligned} J\,M\,J &= (I - K\,N^T)\,J \\ &= J - K\,N^T\,J \\ &= J - K\,(J\,N)^T \\ &= J \end{aligned}$$

which leads to

$$E(f_{\mathbf{s},\mathbf{g}}) = \begin{pmatrix} \mathbf{s}^T & \mathbf{g}^T \end{pmatrix} J \begin{pmatrix} \mathbf{s} \\ \mathbf{g} \end{pmatrix} \tag{13}$$

Hence the objective in (2) is a quadratic function on $\{\mathbf{s}, \mathbf{g}\}$,

$$\mathbf{s}^T \mathbf{s} + \lambda \begin{pmatrix} \mathbf{s}^T & \mathbf{g}^T \end{pmatrix} J \begin{pmatrix} \mathbf{s} \\ \mathbf{g} \end{pmatrix} \tag{14}$$

For a given $\mathbf{g}$, (14) is minimized when $\mathbf{s}$ is given by (10), and the minimum has the form $\lambda \mathbf{g}^T H \mathbf{g}$ where H is defined in (9). By dropping the constant λ, we have proven the proposition. □

We add two technical notes here. First, matrix $I + \lambda J_{00}$ is invertible for generic values of λ, as long as $-1/\lambda$ is not an eigenvalue of J_{00}. Second, H is a positivie semi-definite matrix, since $\lambda \mathbf{g}^T H \mathbf{g}$ is the sum of $\mathbf{s}^T \mathbf{s}$ and Duchon's energy of $f_{\mathbf{s},\mathbf{g}}$, both of which are non-negative for any choice of $\mathbf{g}$.

One may observe that Proposition 3.2 transforms the problem of finding an implicit function (1) to that of finding unit vectors (normals) $\mathbf{g}$ at the input points (8). From this perspective, our definition offers another way of estimating normals from un-oriented points. The key distinction between our formulation and existing normal estimation approaches is that, while the latter is guided by the local shape of point samples, ours is guided by the global shape of the reconstructed implicit function (via minimizing Duchon's energy). As we shall see in the experimental results, the coupling of normal estimation with reconstruction allows our method to better deal with imperfect sampling than previous methods.

Figure 2 (b,c,d) give examples of the VIPSS specialized to Duchon's energy with varying values of λ $(0, 0.1, 1)$ in both 1D and 2D. Observe that the VIPSS in 2D is able to interpolate or approximate sparse and non-uniformly distributed points. Increasing λ results in smoother curves that deviate further from the input. At $\lambda = 0$, the VIPSS exactly interpolates the points (a property that will be discussed in the next section).

4 PROPERTIES

We show that the VIPSS satisfies a few basic properties that are desirable for surface reconstruction.

4.1 Exact interpolation

It is easy to see that the VIPSS interpolates all the input points when $\lambda = 0$. In this case, $\mathbf{s} = \mathbf{0}$ by (10), which implies that the Duchon's interpolant $f_{\mathbf{s},\mathbf{g}}$ is precisely zero at each $\mathbf{x}_i$. Note that, since the fitting is exact, the optimization problem (1) in the general definition of VIPSS reduces to a *parameter-free* form:

$$\begin{aligned} \text{Minimizes:} \quad & E(f) \\ \text{Subject to:} \quad & \|Df(\mathbf{x}_i)\| = 1, \; f(\mathbf{x}_i) = 0, \;\; \forall i \end{aligned}$$

4.2 Linear reproduction

Since Duchon's energy is 2nd-order, the VIPSS reproduces linear geometry. Specifically, suppose that $\mathbf{x}_i$ span a $(d-1)$-dimensional hyperplane in $\mathbb{R}^d$ (e.g., a line in 2D or a plane in 3D). Such hyperplane can be defined as the zero-level set of some linear function f, which has vanishing objective in (1). By choosing the f with a unit gradient, we have found a solution to the variational problem of (1), whose zero-level set (the VIPSS) is the hyperplane.

4.3 Commutativity with similarity transformations

Ideally, a reconstruction method should be invariant to the change of the coordinate system. In other words, the reconstruction operator should *commute* with similarity transformations (e.g., translation, rotation, and uniform scaling): reconstructing from the transformed points should be equivalent to transforming the reconstruction from the original points.

Commutativity to isometry (translations and rotations) is a direct consequence of the invariance of Duchon's energy to isometry. Consider a set of transformed points $\tilde{\mathbf{x}}_i = T(\mathbf{x}_i)$ where T is an isometry. For any function f, the objective in (1) with respect to the original points $\mathbf{x}_i$ is the same as the objective of the transformed function $\tilde{f}(\mathbf{x}) = f(T^{-1}(\mathbf{x}))$ with respect to the transformed points $\tilde{\mathbf{x}}_i$. Since gradient magnitudes are preserved under isometry, we conclude that, if f is the solution to (1) for $\mathbf{x}_i$, $\tilde{f}$ must be the solution for $\tilde{\mathbf{x}}_i$.

Duchon's energy is not invariant to uniform scaling, but is multiplied by some power of the scale. To ensure that the VIPSS commutes with scaling, the value of λ needs to be properly scaled with the input points. As the next proposition shows, λ should scale *cubically* with the data size.

Proposition 4.1. *Let f be the solution to (1) using Duchon's energy for a given point set $\mathbf{x}_i$ and λ, and $w > 0$. Then $\tilde{f}(\mathbf{x}) = wf(\mathbf{x}/w)$ is the solution for points $\tilde{\mathbf{x}}_i = w\mathbf{x}_i$ $(i = 1, \ldots, n)$ and $\tilde{\lambda} = w^3\lambda$.*

Proof. In the following, we use symbol ˜ for quantities involving the transformed points $\tilde{\mathbf{x}}_i$. We first note that

$$\begin{aligned} \phi(w\mathbf{x}, w\mathbf{y}) &= w^3\phi(\mathbf{x}, \mathbf{y}) \\ D^{0,1}\phi(w\mathbf{x}, w\mathbf{y}) &= w^2 D^{0,1}\phi(\mathbf{x}, \mathbf{y}) \\ D^{1,1}\phi(w\mathbf{x}, w\mathbf{y}) &= wD^{1,1}\phi(\mathbf{x}, \mathbf{y}) \end{aligned}$$

Hence matrix $\tilde{A}$ in (4) for $\tilde{\mathbf{x}}_i$ is related to A for $\mathbf{x}_i$ by

$$\tilde{A} = w^{-3} \; W \; A \; W$$

where W is a diagonal matrix whose diagonal consists of n of w^3, dn of w^2, d of w, and a single 1, in order. The inverse W^{-1} is also a diagonal matrix, whose diagonal consists of n of w^{-3}, dn of w^{-2}, d of w^{-1}, and a single 1, in order. Thus we have the following relation between the inverses, $\tilde{A}^{-1}$ and A^{-1},

$$\tilde{A}^{-1} = w^3 \; W^{-1} \; A^{-1} \; W^{-1}$$

and the relations between the sub-matrices,

$$\tilde{J}_{00} = w^{-3} \; J_{00}, \quad \tilde{J}_{01} = w^{-2} \; J_{01}, \quad \tilde{J}_{11} = w^{-1} \; J_{11}$$

Substituting the above into (9) and noting that $\tilde{\lambda} = w^3\lambda$ yields

$$\tilde{H} = w^{-1}H$$

Therefore, if $\mathbf{g}$ minimizes $\mathbf{g}^T H\mathbf{g}$, then it also minimizes $\mathbf{g}^T\tilde{H}\mathbf{g}$. For notational consistency, we denote $\tilde{\mathbf{g}} = \mathbf{g}$. By (10), we have $\tilde{\mathbf{s}} = w\mathbf{s}$. As a result,

$$\begin{pmatrix} \tilde{\mathbf{a}} \\ \tilde{\mathbf{b}} \\ \tilde{\mathbf{c}} \\ \tilde{d} \end{pmatrix} = \tilde{A}^{-1} \begin{pmatrix} \tilde{\mathbf{s}} \\ \tilde{\mathbf{g}} \\ \mathbf{0} \\ 0 \end{pmatrix} = w^{-3}W^{-1}A^{-1}W^{-1} \begin{pmatrix} w\mathbf{s} \\ \mathbf{g} \\ \mathbf{0} \\ 0 \end{pmatrix} = \begin{pmatrix} w^{-2}\mathbf{a} \\ w^{-1}\mathbf{b} \\ \mathbf{c} \\ wd \end{pmatrix}$$

Substituting the above into the definition of Duchon's interpolant (3) yields $f_{\tilde{\mathbf{s}},\tilde{\mathbf{g}}}(\mathbf{x}) = wf_{\mathbf{s},\mathbf{g}}(\mathbf{x}/w)$. This proves the proposition, because $f_{\mathbf{s},\mathbf{g}}$ and $f_{\tilde{\mathbf{s}},\tilde{\mathbf{g}}}$ are the solutions to (1) for inputs $\{\mathbf{x}, \lambda\}$ and $\{\tilde{\mathbf{x}}, \tilde{\lambda}\}$, respectively, due to Proposition 3.2. □

The above proposition implies that the VIPSS of the scaled points (zero-level set of $\tilde{f}$) is the VIPSS of the original points (zero-level set of f) scaled by the same factor w. In summary, the VIPSS undergoes the same similarity transformation with the input data, as long as parameter λ is multiplied by the cubic power of the scaling factor whenever uniform scaling is involved.

5 IMPLEMENTATION

Reconstructing the VIPSS from a point set involves four steps:

(1) Computing matrix H. This involves constructing the interpolation matrix A by (4), computing its inverse A^{-1}, and computing H from sub-matrices of A^{-1} by (9).

(2) Optimizing vectors $\mathbf{g}$ by (8).

(3) Constructing Duchon's interpolant $f_{\mathbf{s},\mathbf{g}}$. This involves recovering interpolated values $\mathbf{s}$ from $\mathbf{g}$ by (10) and then the constants $\mathbf{a}, \mathbf{b}, \mathbf{c}, d$ in $f_{\mathbf{s},\mathbf{g}}$ by $A^{-1} \cdot \{\mathbf{s}^T, \mathbf{g}^T, \mathbf{0}^T, 0\}^T$.

(4) Surfacing the zero-level set of $f_{\mathbf{s},\mathbf{g}}$.

Steps (1,3) can be done using standard linear algebra packages (we use Armadillo). To solve the constrained optimization problem of (8), we first convert it into an unconstrained problem by representing each $\mathbf{g}_i$ using two spherical angles and then solve it by the L-BFGS method (we use NLopt and its default parameters, with 3000 maximum iterations). For step (4), one may use any available method that polygonalizes level sets of implicit functions. Since the input points are usually close to the zero-level surface, we use a tracing-based marching cubes method [Bloomenthal 1994] at a fixed grid resolution (100^3 to 200^3 in our experiments) that starts from a data point. More advanced meshing methods, such as [Boissonnat and Oudot 2005], can be applied to produce surfaces with better triangle shapes.

As with many non-linear optimization problems, the quality of the solution depends heavily on the quality of the initialization. In the following (Section 5.1), we describe a practically effective method for initializing our optimization problem. We end this section with a complexity analysis (Section 5.2).

5.1 Initializing the optimization

Optimization formulations like ours (8) commonly appear in the literature on computing direction fields [Vaxman et al. 2016]. A typical initialization strategy is relaxing the per-vector unit-norm constraint (${\mathbf{g}_i}^T \mathbf{g}_i = 1$) to constraining the total norm of all vectors to be one ($\mathbf{g}^T \mathbf{g} = 1$). By the Rayleigh Quotient Theorem, the minimal value of $\mathbf{g}^T H \mathbf{g}$ under the relaxed constraint is achieved when $\mathbf{g}$ is the eigenvector of H with the smallest eigenvalue (denoted by e_H). This eigenvector, after normalization, is then used as the initial solution to start the optimization under the original, per-vector unit-norm constraint [Huang and Ju 2016].

However, the spectral initialization strategy often fails for our problem. In these failure cases, there is a significant variation among the norms of individual vectors $\|\mathbf{g}_i\|$ derived from e_H. Those vectors with extremely small norms tend to be less accurate, and such vectors often form clusters with "flipped" orientations that are difficult to be corrected by optimization. We illustrate such a case in 2D in Figure 3 (a,b). Note that the initial vectors associated with the lower points in (a) have very small lengths, and their orientations are opposite to those vectors associated with the upper points (see zoom-in). Optimizing from this initialization leads to a high-energy local minimum shown in (b).

To find more stable initial vectors, we observed that the norms of individual vectors $\|\mathbf{g}_i\|$ derived from e_H tend to be more uniformly distributed, both in length and in direction, as λ increases. This can be conceptually explained by the fact that a larger λ leads to a smoother interpolant, whose gradients at the input points are more similar with each other. We can also provide a more rigorous argument in the limiting case of $\lambda \longrightarrow \infty$:

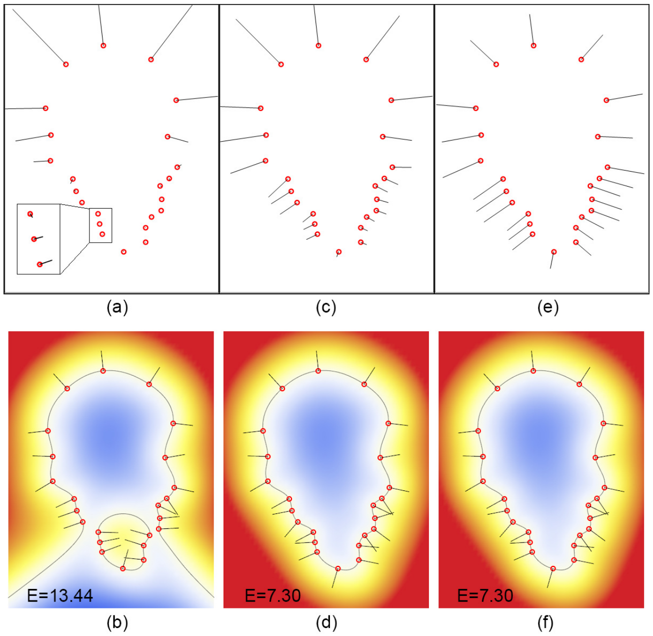

Fig. 3. Initial vectors generated using the spectral method with $\lambda = 0$ (a) contains vectors with very small magnitudes and flipped orientations (see insert, vectors shown with 20x scaling), which leads to a high-energy result after optimization (b). Initial vectors generated with $\lambda = 0.01$ (c) and 0.1 (e) are more uniform, and they lead to the same low-energy result (d,f) after optimization with $\lambda = 0$.

Proposition 5.1. *As $\lambda \longrightarrow \infty$, any $\mathbf{g} = \{\mathbf{g}_i\}$ where $\mathbf{g}_i = \mathbf{g}_j$ for all pairs i, j is an eigenvector of H with zero eigenvalue.*

Proof. By $JN = \mathbf{0}$, where J is defined in (7) and N in (5), we have

$$\begin{aligned} J_{00}\ N_0 + J_{01}\ N_1 &= \mathbf{0} \\ J_{01}^T\ N_0 + J_{11}\ N_1 &= \mathbf{0} \end{aligned}$$

We then have

$$\begin{aligned} H\ N_1 &= J_{11} N_1 - J_{01}^T (I/\lambda + J_{00})^{-1} J_{01} N_1 \\ &= -J_{01}^T N_0 + J_{01}^T (I/\lambda + J_{00})^{-1} J_{00} N_0 \end{aligned}$$

As $\lambda \longrightarrow \infty$, $(I/\lambda + J_{00})^{-1} J_{00} \longrightarrow I$. Therefore

$$H\ N_1 \longrightarrow -J_{01}^T N_0 + J_{01}^T N_0 = 0$$

As a result, the columns of N_1 are eigenvectors of H with zero eigenvalues as $\lambda \longrightarrow \infty$. Since the $\mathbf{g}$ in the proposition can be expressed as a linear combination of the columns of N_1, it is also an eigenvector of H with zero eigenvalue. □

In other words, the output of spectral initialization tends toward a constant vector field as $\lambda \longrightarrow \infty$. Hence it can be expected that, as λ increases, vectors produced by spectral initialization become increasingly more uniform. We illustrate this behavior in Figure 3

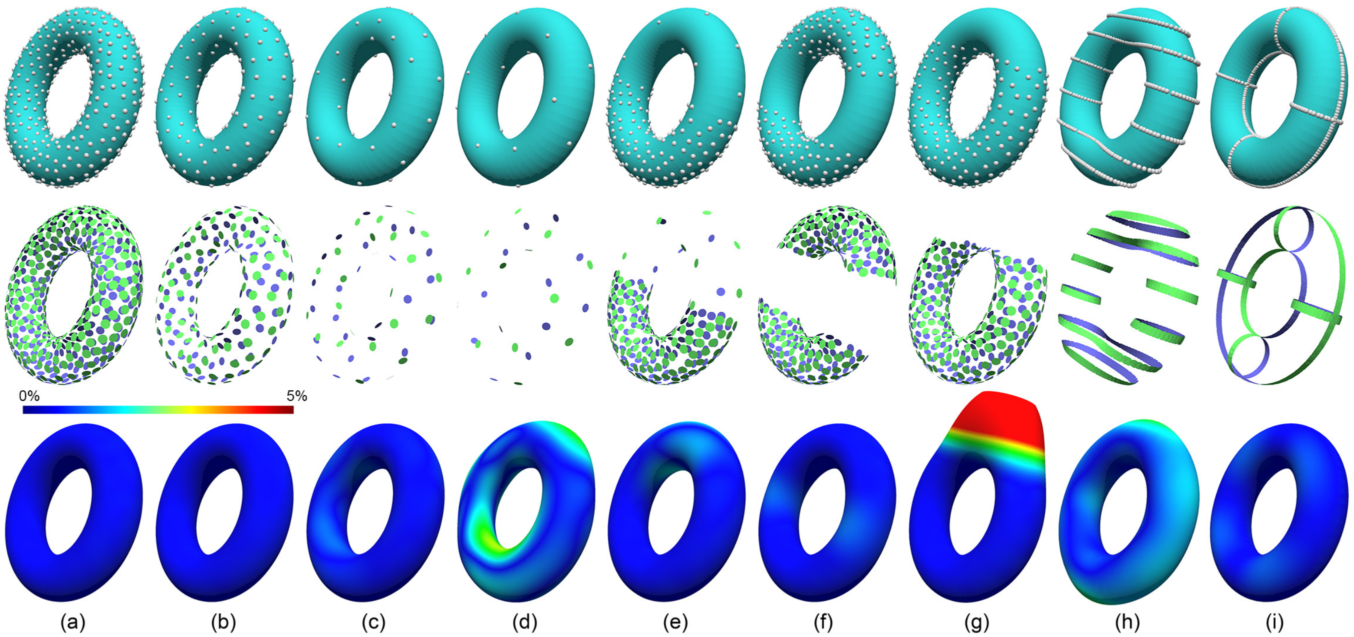

Fig. 4. Top row: sampling a torus surface with decreasing density (a,b,c,d with 500, 200, 50, 25 points respectively), varying sampling density (e), missing samples (f,g), and along 1-dimensional curves (h,i). Middle row: optimized vectors **g** visualized as oriented disks (green/blue: front/back side). Bottom row: the VIPSS ($\lambda = 0$) colored by distance from the original torus surface (see color bar; the percentages are of the largest dimension of the shape).

(c,e) for two different values of λ (0.01, 0.1) on the same point set as (a). Using either set of vectors as the initial solution, optimizing (8) with $\lambda = 0$ successfully reaches the same low-energy solution shown in (d,f).

Guided by the observation, we propose to compute not one, but multiple candidate initializations corresponding to different values of λ. We pre-define a set of "offset" values $\{\lambda_1, \ldots, \lambda_k\}$. Let λ_0 be the parameter chosen by the user for the VIPSS. For each λ_i, we construct the matrix H using $\lambda = \lambda_0 + \lambda_i$ and compute its eigenvector e_H. This gives us k initial solutions. We then optimize (with $\lambda = \lambda_0$) for k times, each time starting from one of the initial solutions, and take the optimized result with the least energy. We use the offset values $\{0, 0.001, 0.01, 0.1, 1\}$ in our tests, assuming the data is scaled to fit in a $2 \times 2 \times 2$ cube.

5.2 Complexity analysis

The asymptotic complexity of running time, with respect to the dimensionality d and number of input points n, of the four steps described at the beginning of Section 5 are as follows:

1. $O(d^3n^3)$: $O(d^3n^2)$ for constructing A, $O(d^3n^3)$ for inverting A, and $O(n^3 + d^2n^3)$ for constructing H.
2. $O(d^3n^3)$: $O(d^3n^3)$ for computing the eigenvector of H with the smallest eigenvalue, and $O(ld^2n^2)$ for gradient-descent optimization, where l is the number of descent iterations (which is typically much smaller compared to n).
3. $O(d^2n^2)$: $O(dn^2)$ for computing **s**, and $O(d^2n^2)$ for recovering the constants **a**, **b**, **c**, d.
4. $O(mdn)$ where m is the number of points at which the surfacing algorithm evaluates the interpolant.

It is clear from the analysis that the bottlenecks of the algorithm are the inversion of matrix A and finding the eigenvector of matrix H, both taking $O(d^3n^3)$ time. Note that A is a dense matrix due to the global nature of the triharmonic basis. While we have observed that H often contains a large amount of close-to-zero entries, particularly for uniformly sampled points and a small value of λ, the sparsity tends to decrease with the increase of non-uniformity in sampling and noise level (which necessitates larger values of λ).

6 EXPERIMENTS

We show experimental results of our method in 3D under varying sampling conditions and compare with relevant methods for normal estimation and surface reconstruction. We end this section with a performance report and an application. In our examples, uniform sampling from existing surfaces is generated using the Poisson-disk sampling method [Corsini et al. 2012] implemented in MeshLab.

6.1 Results

We first evaluate our method under varying sampling densities and patterns using a synthetic Torus surface (level set of a degree-4 polynomial). We fix $\lambda = 0$ to perform exact interpolation. As seen in Figure 4, our method can reconstruct an almost perfect torus from as few as 50 points (c), and a close approximation from 25 points (d). Also, our method is robust under different types of non-uniform sampling patterns, such as varying density (e), missing samples (f,g), and highly anisotropic sampling along curves (h,i), unless the samples are too ambiguous for inferring the shape (e.g., in (g)).

We next test our method on samples from more complex 3D surfaces (Figures 5, 6) and wireframes (Figure 6). Again, we fix

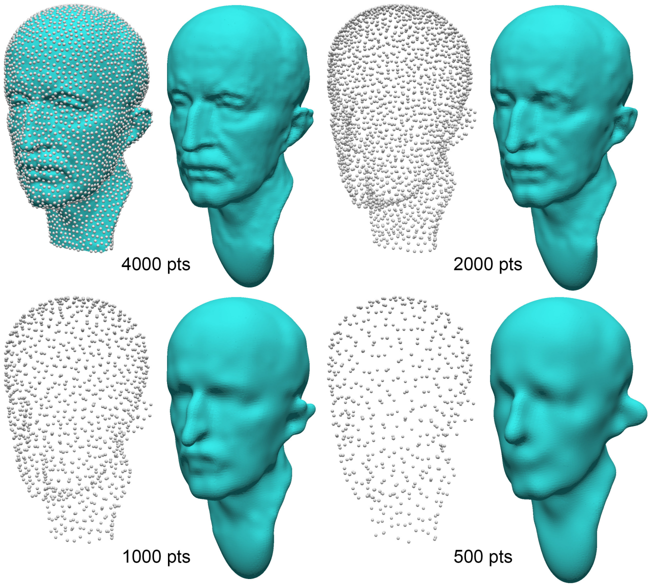

Fig. 5. VIPSS ($\lambda = 0$) for samples from Max Planck at different densities.

$\lambda = 0$. Observe in Figure 5 that the quality of VIPSS drops gracefully under decreasing sampling density. Despite the sparsity and non-uniformity of sampling in Figure 6, our method is able to faithfully reconstruct various shape features, such as the fingers of the Hand, protrusions of the Vertebra, ears of the Dog, etc.

Lastly, we test our method on noisy samples in Figure 7. Consistent with our observations earlier in 1D and 2D (Figure 2), larger λ values lead to smoother and less approximating surfaces, which are suited for higher noise levels.

6.2 Comparisons

Normal estimation methods. We compare the oriented vectors $\mathbf{g}$ resulted by our optimization (8) with those produced by existing normal estimation methods.

We first consider methods that separately estimate directions and orientations. Since errors in the un-oriented directions persist after orientation, we will focus on issues in the direction estimation step. We considered two prevalent methods for direction estimation, the plane-fitting method of [Hoppe et al. 1992] (referred to as PCA) and the Voronoi-based method of [Merigot et al. 2011] (referred to as VCM). Both methods are based on analysis of local point neighborhood, whose size is controlled by the number of nearest neighbors (k in PCA) or a sphere radius (r in VCM). We found that they both tend to fail on non-uniformly distributed samples, where oftentimes a suitable neighborhood size cannot be found. As shown in Figure 8, taking the Dog wireframe samples from Figure 6 as input, PCA requires a fairly large neighborhood ($k = 30$ in (a)) to get a reasonable direction estimate for most points, while still failing on some (see the red and blue boxes). Increasing the neighborhood size ($k = 50$ in (b)) improves the directions at some points (the red box), but makes others worse (the blue box), due to the interference with nearby features (the dog ear). Similarly, VCM at a small neighborhood size ($r = 0.1$ in (c)) produces many incorrect directions (red box). A larger neighborhood ($r = 0.25$ in (d)) improves some directions but results in over-smoothing at small shape features (blue box).

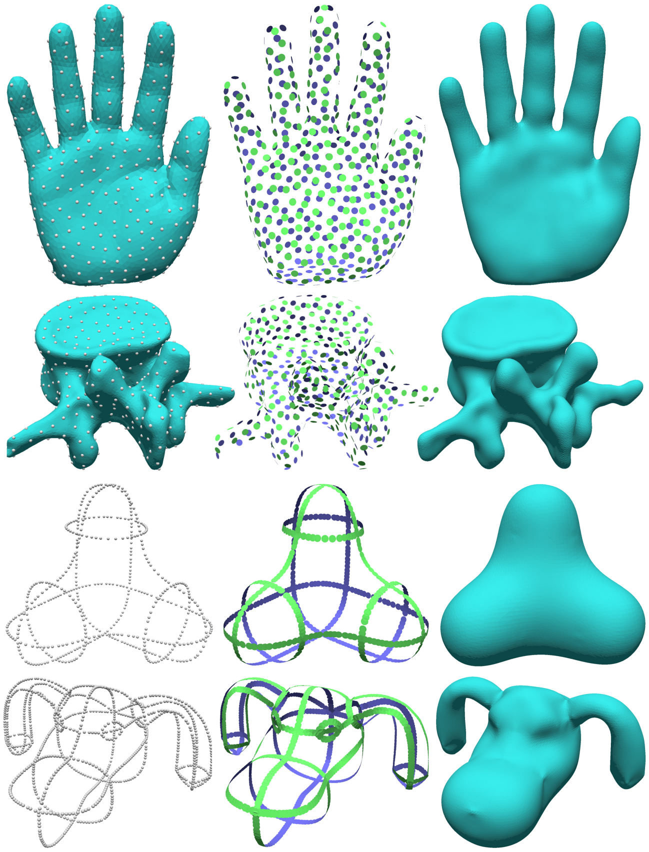

Fig. 6. Left: uniform samples from two surfaces (Hand and Vertebra, with 500 points each) and two wireframes (Trebol with 500 points and Dog with 1000 points). Middle: optimized vectors $\mathbf{g}$. Right: VIPSS at $\lambda = 0$.

Next we compare with the variational method of Wang et al. [2011], which estimates oriented normals in a single step by solving a quadratic optimization problem similar to ours (8). While our matrix H is derived from the global smoothness of the implicit function, theirs is based on two hand-crafted energy terms that measure correlation between nearby normals. We have observed that, while Wang's method produces more stable results than two-step methods for sparse and non-uniform inputs, careful tuning of its parameters (neighborhood size and energy weighting) is required for individual inputs. Even with our best effort in tuning, Wang's method can still fail for some inputs, such as the Walrus sketch (Figure 1 (e)) and the Bathtub (Figure 9, third row), particularly near sparsely sampled thin shape features (e.g., Walrus' flippers and Bathtub's curved wall). These errors, in turn, lead to poorly reconstructed surfaces. Observe that our method produces a plausible set of normals for both inputs, and in turn better reconstructions.

We also compared with the deep-learning-based normal estimation method, PCPNet [Guerrero et al. 2018] on the same Bathtub example (Figure 9, bottom row). This method was unable to give any

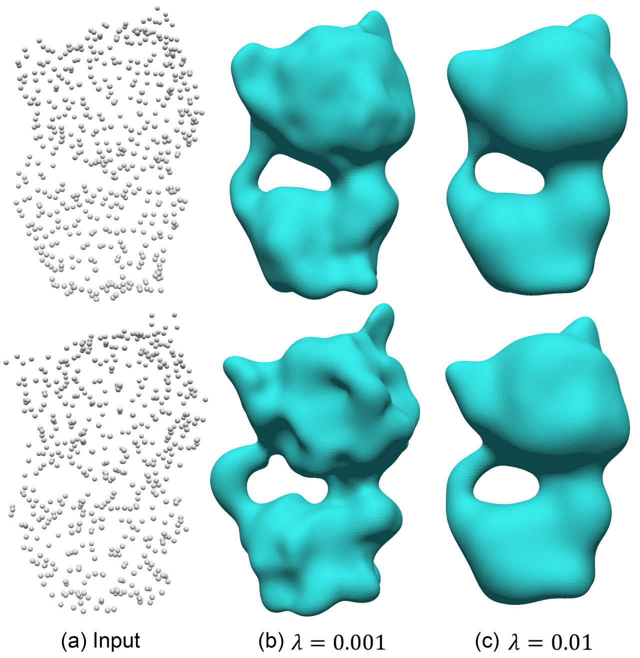

Fig. 7. (a): Two sampling of the Kitten (500 points each) at low (top, 1%) and high (bottom, 5%) noise rate. (b,c): VIPSS with $\lambda = 0.001$ and 0.01.

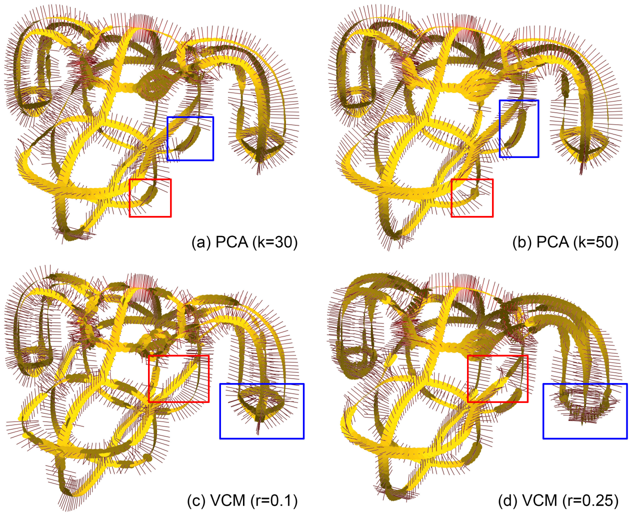

Fig. 8. Comparing direction estimation on samples from a 3D wireframe using PCA [Hoppe et al. 1992] and VCM [Merigot et al. 2011] with different parameters. Each un-oriented direction is shown by a yellow tangent disk and a line segment.

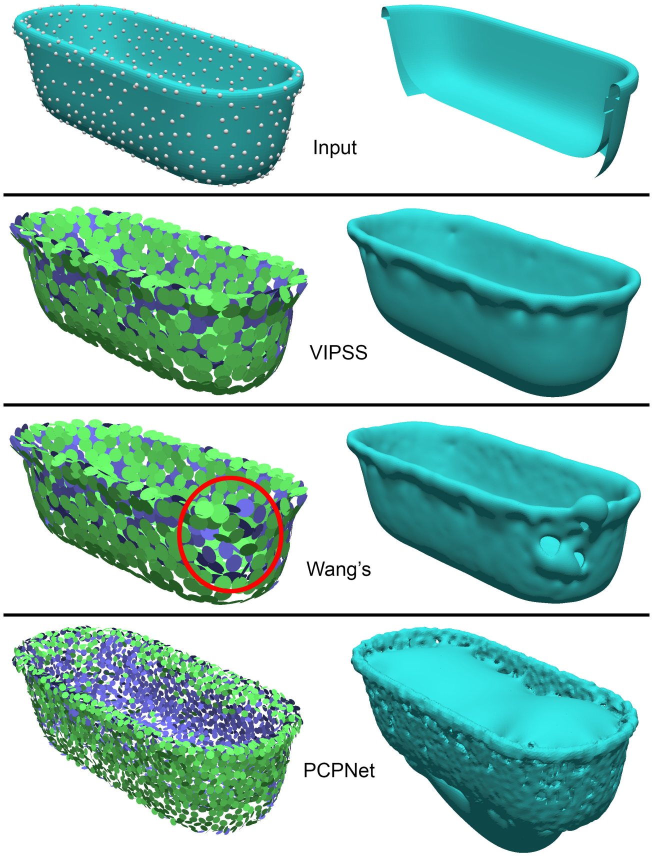

Fig. 9. Comparing normal estimation from a 800-point sample from the Bathtub (cross-section shown in top-right) using VIPSS ($\lambda = 0$), variational method of [Wang et al. 2011], and PCPNet [Guerrero et al. 2018] (on a 3000-point sampling). The surfaces for these methods are generated using Hermite RBF interpolation (i.e., zero-level set of $f_{\mathbf{s},\mathbf{g}}$ where $\mathbf{s} = 0$ and $\mathbf{g}$ are the estimated normals).

reasonable normals for the original 800-point sample, so we showed their result on a denser sampling (3000 points). Even at this density, PCPNet produces incorrect normal orientations for a significant portion of the points, which leads to a poor reconstruction.

Surface reconstruction methods. As reviewed earlier, most implicit reconstruction methods require oriented points as input. Hence improved normal estimation (e.g., our optimized vectors $\mathbf{g}$) would benefit these existing methods.

In our setting, using Duchon's interpolant (or so-called Hermite RBF) has several advantages over other implicit reconstruction methods. First, the interpolant $f_{\mathbf{s},\mathbf{g}}$ can fully utilize the optimized Hermite data $\mathbf{s}, \mathbf{g}$, not just the vectors $\mathbf{g}$, in the case of approximation with a non-zero λ (and hence non-zero $\mathbf{s}$). Second, by Proposition 3.2, Duchon's interpolant using optimized Hermite data is theoretically optimal in terms of the objective (1). Third, and in practice, we observed that Duchon's interpolant outperforms existing methods for interpolating sparse, oriented points. In Figure 10, we compared Duchon's interpolant with Screened Poisson reconstruction [Kazhdan and Hoppe 2013] and Albegraic Point Set Surface (APSS) [Guennebaud and Gross 2007] (an IMLS-type method) on the same input, which is a 1000-point sampling of the Stanford Bunny with normals computed by our method. Each of these methods has a parameter that trades off smoothness with closeness of approximation (fitting weight α in Screened Poisson, larger for a closer fit, and filter scale h in APSS, smaller for a closer fit). Observe that closer approximation using these methods leads to surface artifacts (dimples in (d) and

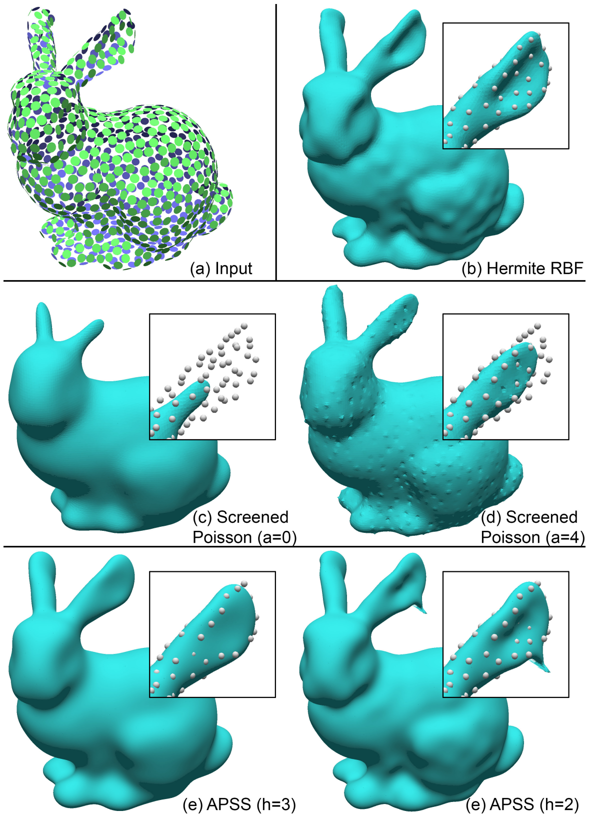

Fig. 10. Comparing Duchon's interpolant (Hermite RBF) (b) with Screened Poisson [Kazhdan and Hoppe 2013] (c,d) and APSS [Guennebaud and Gross 2007] (e,f) at different parameters on the same oriented input with 1000 points (a). Orientations in (a) are computed by our method with $\lambda = 0$.

tearing in (e)). On the other hand, Duchon's interpolant results in exact and smooth interpolation (b).

We next compare with a few methods that do not require oriented points in Figure 11. The variational method of [Alliez et al. 2007] (b) has trouble handling sparse samples on thin shapes (e.g., the Bathtub), even after we used a very high data-fitting weight in their formulation. Also, since the variational problem is solved on a tetrahedralization of the domain, their method produces less smooth surfaces than our method (e.g., the Dog). On the other hand, combinatorial methods, such as the ball-pivoting method [Bernardini et al. 1999], the tight cocone [Dey and Goswami 2003], and the power crust [Amenta et al. 2001], are designed for dense samples and generally unsuited for sparse and non-uniform inputs like these.

6.3 Performance

Table 1 reports the running time of the four steps of our method (see Section 5) on the Max Plank data set in Figure 5. Observe that these statistics agree with the complexity analysis in Section 5.2. The running time is dominated by the computation of H (inverting A) and the initialization of optimization (computing eigenvectors of H), both exhibiting cubic growth with the input size.

Table 1. Running time (in seconds) of each step for Figure 5 on a MacBook Pro with 2.5GHz Intel Core i7 CPU and 16 GB memory (implementation in C++). Timing for Optimization is written as initialization time (using the offset method of Section 5.1) + NLopt time. Surfacing uses a 100^3 grid.

# Points	Compute H	Optimize g	Build $f_{s,g}$	Polygonize
500	0.3	2.5 + 0.2	0.0	6.6
1000	1.8	20.0 + 1.1	0.1	13.0
2000	14.6	165.6 + 10.8	0.3	26.0
4000	98.5	1428.2 + 45.4	0.7	51.7

6.4 Application: sketch surfacing

AR/VR sketching tools, such as Google's Tilt Brush, allow a user to create 3D curvilinear designs in a fully immersive manner. While existing tools only display the sketches, creating a surface directly from the sketches could give the user a more intuitive grasp of their design. However, freehand 3D sketches are highly unstructured (e.g., incomplete strokes, over-sketching, missing junctions, etc.), whereas existing methods for surfacing 3D wireframes generally require a clean, connected graph [Bessmeltsev et al. 2012; Pan et al. 2015; Stanko et al. 2016].

Due to its resilience to sparse and non-uniform sampling, our method is well-suited to perform this challenging surfacing task. Since we were not able to find a public data set of curves produced by these sketching systems, we simulate the sketches by freehand curves drawn on top of existing surfaces. Two examples are shown in Figure 12 on a sketch of a telephone and a hand (we use 1000 sample points for each). Our method ($\lambda = 0$) is able to surface both examples despite the many incomplete curves. Of particular note is that our method is able to resolve the small gap between two nearby fingers of the Hand (see inserts) despite the sparse sampling there.

An even more challenging example is shown in Figure 1, where we further introduced random perturbation to each curve segment (by maximally 5% of the longest dimension of the input) to simulate inaccuracies in free-hand 3D sketches (e.g., over-sketching and missing junctions). Our method is able to create a smooth approximation of the Walrus at $\lambda = 0.003$.

7 CONCLUSION AND LIMITATIONS

We describe a novel implicit surface definition (VIPSS) from unoriented point sets that involves a single parameter (zero for exact interpolation), applies to any dimensions, and does not require discretization. Reconstruction using the definition can be easily implemented using standard linear algebra and optimization packages, and the results appear more robust under sparse and non-uniform sampling than existing methods.

Limitations. The main limitation of our method is its computational complexity (cubic on the number of points). There are several promising directions that wish to pursue for improving its scalability. Since the matrix M is the Gram matrix of inner products in a semi-Hilbert space, low-rank approximation methods for such matrices [Drineas and Mahoney 2005; Smola and Schökopf 2000] can potentially reduce the complexity of various matrix operations in this work. For large and dense point sets, we may consider an incremental approach akin to [Carr et al. 2001] that starts with fitting a small subset of points and then incrementally adds more points where the approximation errors are large.

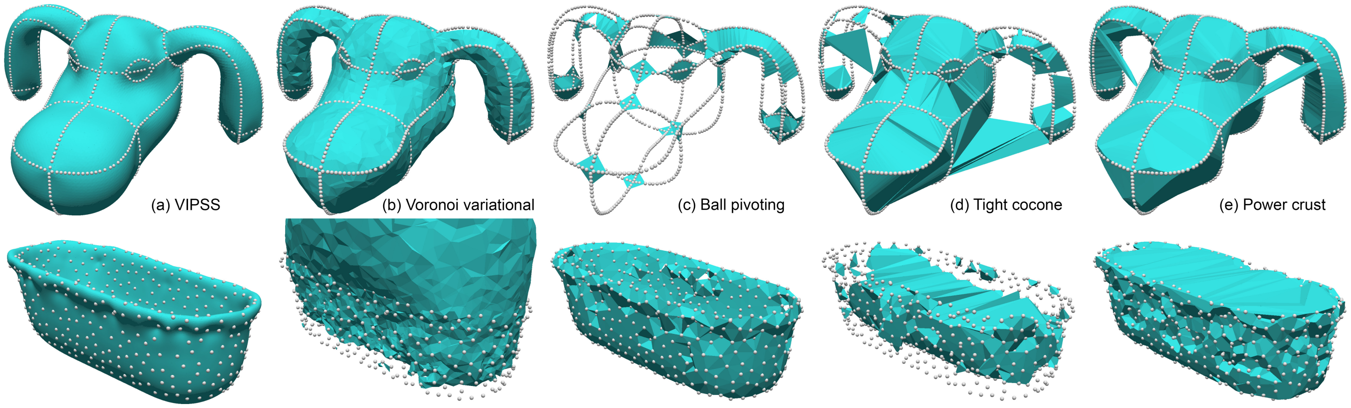

Fig. 11. Comparing VIPSS ($\lambda = 0$) with methods that do not require oriented inputs: the Voronoi-based variational method of [Alliez et al. 2007], the ball-pivoting method [Bernardini et al. 1999], the tight cocone [Dey and Goswami 2003], and the power crust [Amenta et al. 2001].

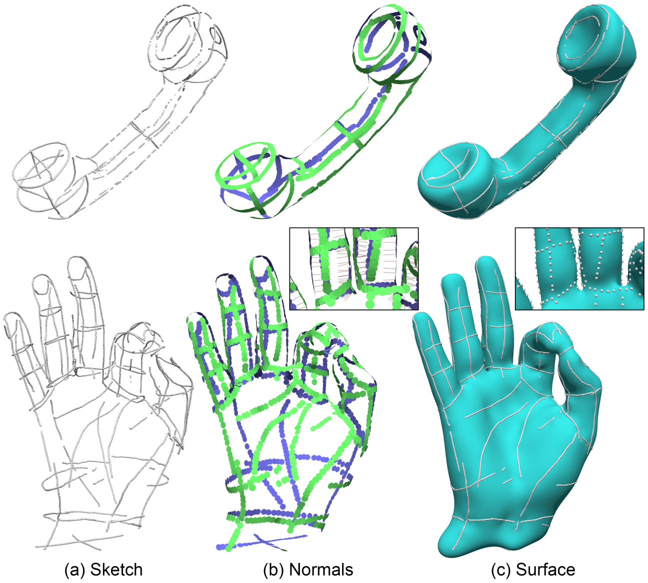

Fig. 12. VIPSS ($\lambda = 0$) vectors (b) and surface (c) for samples from an unstructured sketch (a) of a Phone (top) and Hand (bottom). The inserts take a closer look between the middle and ring fingers of the Hand (the line segments in the insert of (b) indicate $-g$).

Our current approach for initializing the optimization (by trying different offsets of λ) is neither efficient nor guaranteed to work for all inputs (particular those exhibiting a varying level of noise and complexity). A possible remedy to the latter is to explore spatially varying offsets. Alternatively, one may "warm-start" the optimization using results obtained from a subset of the points in an incremental framework, thereby avoiding the need for computing eigenvectors and trying different offsets.

The robustness of VIPSS to sparse and non-uniform sampling raises the theoretical question of what is the sampling condition under which VIPSS has guaranteed reconstruction quality (geometrically and topologically). We would also like to explore how the current formulation can be extended to reconstruct shapes with sharp (C^0) features, such as man-made shapes.

ACKNOWLEDGMENTS

This work is supported by NSF grants RI-1618685 and DBI-1759836, NIH grant CA233303-1, and a gift from Adobe. We would like to thank MeshLab, CGAL, researchers who distribute code, and Tong Zhao who helped with comparisons with [Alliez et al. 2007].

REFERENCES

Marc Alexa, Johannes Behr, Daniel Cohen-Or, Shachar Fleishman, David Levin, and Claudio T Silva. 2003. Computing and rendering point set surfaces. *IEEE Transactions on visualization and computer graphics* 9, 1 (2003), 3–15.

P. Alliez, D. Cohen-Steiner, Y. Tong, and M. Desbrun. 2007. Voronoi-based Variational Reconstruction of Unoriented Point Sets. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing (SGP '07)*. 39–48.

Nina Amenta and Marshall Bern. 1999. Surface reconstruction by Voronoi filtering. *Discrete & Computational Geometry* 22, 4 (1999), 481–504.

Nina Amenta, Sunghee Choi, and Ravi Krishna Kolluri. 2001. The power crust. In *Proceedings of the sixth ACM symposium on Solid modeling and applications*. ACM, 249–266.

Nina Amenta and Yong Joo Kil. 2004. Defining point-set surfaces. *ACM Transactions on Graphics (TOG)* 23, 3 (2004), 264–270.

Gavin Barill, Neil G Dickson, Ryan Schmidt, David IW Levin, and Alec Jacobson. 2018. Fast winding numbers for soups and clouds. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 43.

Matthew Berger, Andrea Tagliasacchi, Lee M Seversky, Pierre Alliez, Gael Guennebaud, Joshua A Levine, Andrei Sharf, and Claudio T Silva. 2017. A survey of surface reconstruction from point clouds. *Computer Graphics Forum* 36, 1 (2017), 301–329.

Fausto Bernardini, Joshua Mittleman, Holly Rushmeier, Cláudio Silva, and Gabriel Taubin. 1999. The ball-pivoting algorithm for surface reconstruction. *IEEE transactions on visualization and computer graphics* 5, 4 (1999), 349–359.

Mikhail Bessmeltsev, Caoyu Wang, Alla Sheffer, and Karan Singh. 2012. Design-driven quadrangulation of closed 3D curves. *ACM Transactions on Graphics (TOG)* 31, 6 (2012), 178.

Jules Bloomenthal. 1994. Graphics Gems IV. Chapter An Implicit Surface Polygonizer, 324–349.

Jean-Daniel Boissonnat and Frédéric Cazals. 2002. Smooth surface reconstruction via natural neighbour interpolation of distance functions. *Computational Geometry* 22, 1-3 (2002), 185–203.

Jean-Daniel Boissonnat and Steve Oudot. 2005. Provably Good Sampling and Meshing of Surfaces. *Graph. Models* 67, 5 (Sept. 2005), 405–451.

Alexandre Boulch and Renaud Marlet. 2012. Fast and Robust Normal Estimation for Point Clouds with Sharp Features. *Computer Graphics Forum* (2012).

Alexandre Boulch and Renaud Marlet. 2016. Deep Learning for Robust Normal Estimation in Unstructured Point Clouds. *Computer Graphics Forum* (2016).

E Brazil, Ives Macedo, M Costa Sousa, Luiz Henrique de Figueiredo, and Luiz Velho. 2010. Sketching variational hermite-rbf implicits. In *Proceedings of the Seventh Sketch-Based Interfaces and Modeling Symposium*. Eurographics Association, 1–8.

Martin D Buhmann. 2003. *Radial basis functions: theory and implementations*. Vol. 12. Cambridge university press.

Jonathan C Carr, Richard K Beatson, Jon B Cherrie, Tim J Mitchell, W Richard Fright, Bruce C McCallum, and Tim R Evans. 2001. Reconstruction and representation of

3D objects with radial basis functions. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. ACM, 67–76.

F. Cazals and M. Pouget. 2003. Estimating Differential Quantities Using Polynomial Fitting of Osculating Jets. In *Proceedings of the 2003 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing (SGP '03)*. 177–187.

Massimiliano Corsini, Paolo Cignoni, and Roberto Scopigno. 2012. Efficient and flexible sampling with blue noise properties of triangular meshes. *IEEE Transactions on Visualization and Computer Graphics* 18, 6 (2012), 914–924.

Tamal K Dey. 2006. *Curve and surface reconstruction: algorithms with mathematical analysis*. Vol. 23. Cambridge University Press.

Tamal K Dey and Samrat Goswami. 2003. Tight cocone: a water-tight surface reconstructor. *Journal of Computing and Information Science in Engineering* 3, 4 (2003), 302–307.

Tamal K. Dey and Samrat Goswami. 2004. Provable surface reconstruction from noisy samples. In *Proceedings of the 20th ACM Symposium on Computational Geometry, Brooklyn, New York, USA, June 8-11, 2004*. 330–339.

Tamal K Dey and Jian Sun. 2005. An Adaptive MLS Surface for Reconstruction with Guarantees.. In *Symposium on Geometry processing*. 43–52.

Huong Quynh Dinh, Greg Turk, and Greg Slabaugh. 2002. Reconstructing surfaces by volumetric regularization using radial basis functions. *IEEE transactions on pattern analysis and machine intelligence* 24, 10 (2002), 1358–1371.

Petros Drineas and Michael W. Mahoney. 2005. On the Nyström Method for Approximating a Gram Matrix for Improved Kernel-Based Learning. *J. Mach. Learn. Res.* 6 (Dec. 2005), 2153–2175.

Jean Duchon. 1977. Splines minimizing rotation-invariant semi-norms in Sobolev spaces. In *Constructive theory of functions of several variables*. Springer, 85–100.

Simon Giraudot, David Cohen-Steiner, and Pierre Alliez. 2013. Noise-adaptive shape reconstruction from raw point sets. In *Proceedings of the Eleventh Eurographics/ACMSIGGRAPH Symposium on Geometry Processing*. Eurographics Association, 229–238.

Gaël Guennebaud and Markus Gross. 2007. Algebraic point set surfaces. *ACM Transactions on Graphics (TOG)* 26, 3 (2007), 23.

Paul Guerrero, Yanir Kleiman, Maks Ovsjanikov, and Niloy J. Mitra. 2018. PCPNet: Learning Local Shape Properties from Raw Point Clouds. *Computer Graphics Forum* 37, 2 (2018), 75–85.

Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. 1992. *Surface reconstruction from unorganized points*. Vol. 26. ACM.

Alexander Hornung and Leif Kobbelt. 2006. Robust reconstruction of watertight 3 d models from non-uniformly sampled point clouds without normal information. In *Symposium on geometry processing*. Citeseer, 41–50.

Hui Huang, Dan Li, Hao Zhang, Uri Ascher, and Daniel Cohen-Or. 2009. Consolidation of Unorganized Point Clouds for Surface Reconstruction. *ACM Trans. Graph.* 28, 5, Article 176 (Dec. 2009), 176:1–176:7 pages.

Zhiyang Huang and Tao Ju. 2016. Extrinsically Smooth Direction Fields. *Comput. Graph.* 58, C (Aug. 2016), 109–117.

Takashi Ijiri, Shin Yoshizawa, Yu Sato, Masaaki Ito, and Hideo Yokota. 2013. Bilateral Hermite Radial Basis Functions for Contour-based Volume Segmentation. *Computer Graphics Forum* 32, 2 (2013), 123–132. Proc. of EUROGRAPHICS'13.

Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. 2006. Poisson Surface Reconstruction. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing (SGP '06)*. 61–70.

Michael Kazhdan and Hugues Hoppe. 2013. Screened poisson surface reconstruction. *ACM Transactions on Graphics (ToG)* 32, 3 (2013), 29.

Ravikrishna Kolluri. 2008. Provably good moving least squares. *ACM Transactions on Algorithms (TALG)* 4, 2 (2008), 18.

Sören König and Stefan Gumhold. 2009. Consistent Propagation of Normal Orientations in Point Clouds. In *VMV*.

Bao Li, Ruwen Schnabel, Reinhard Klein, Zhiquan Cheng, Gang Dang, and Jin Shiyao. 2010. Robust normal estimation for point clouds with sharp features. *Computers & Graphics* 34, 2 (2010), 94–106.

Shengjun Liu, Charlie C.L. Wang, Guido Brunnett, and Jun Wang. 2016. A Closed-form Formulation of HRBF-based Surface Reconstruction by Approximate Solution. *Comput. Aided Des.* 78, C (Sept. 2016), 147–157.

Xiuping Liu, Jie Zhang, Junjie Cao, Bo Li, and Ligang Liu. 2015. Quality Point Cloud Normal Estimation by Guided Least Squares Representation. *Comput. Graph.* 51, C (Oct. 2015), 106–116.

DanFeng Lu, HongKai Zhao, Ming Jiang, ShuLin Zhou, and Tie Zhou. 2005. A surface reconstruction method for highly noisy point clouds. In *International Workshop on Variational, Geometric, and Level Set Methods in Computer Vision*. Springer, 283–294.

Wenjia Lu, Zuoqiang Shi, Jian Sun, and Bin Wang. 2018. Surface Reconstruction Based on the Modified Gauss Formula. *ACM Trans. Graph.* 38, 1, Article 2 (Dec. 2018), 18 pages.

Josiah Manson, Guergana Petrova, and Scott Schaefer. 2008. Streaming surface reconstruction using wavelets. *Computer Graphics Forum* 27, 5 (2008), 1411–1420.

Quentin Merigot, Maks Ovsjanikov, and Leonidas J. Guibas. 2011. Voronoi-Based Curvature and Feature Estimation from Point Clouds. *IEEE Transactions on Visualization and Computer Graphics* 17, 6 (June 2011), 743–756.

N. J. Mitra, A. Nguyen, and L. Guibas. 2004. Estimating Surface Normals in Noisy Point Cloud Data. In *special issue of International Journal of Computational Geometry and Applications*, Vol. 14. 261–276.

B. S. Morse, T. S. Yoo, P. Rheingans, D. T. Chen, and K. R. Subramanian. 2001. Interpolating implicit surfaces from scattered surface data using compactly supported radial basis functions. In *Proceedings International Conference on Shape Modeling and Applications*. 89–98.

Patrick Mullen, Fernando De Goes, Mathieu Desbrun, David Cohen-Steiner, and Pierre Alliez. 2010. Signing the unsigned: Robust surface reconstruction from raw pointsets. *Computer Graphics Forum* 29, 5 (2010), 1733–1741.

Yutaka Ohtake, Alexander Belyaev, Marc Alexa, Greg Turk, and Hans-Peter Seidel. 2003a. Multi-level partition of unity implicits. In *ACM Transactions on Graphics (TOG)*, Vol. 22. ACM, 463–470.

Yutaka Ohtake, Alexander Belyaev, and Hans-Peter Seidel. 2003b. A multi-scale approach to 3D scattered data interpolation with compactly supported basis functions. In *Shape Modeling International, 2003*. IEEE, 153–161.

A Cengiz Öztireli, Gael Guennebaud, and Markus Gross. 2009. Feature preserving point set surfaces based on non-linear kernel regression. *Computer Graphics Forum* 28, 2 (2009), 493–501.

Hao Pan, Yang Liu, Alla Sheffer, Nicholas Vining, Changjian Li, and Wenping Wang. 2015. Flow Aligned Surfacing of Curve Networks. *ACM Trans. Graph. (SIGGRAPH)* 34, 4 (2015).

Rongjiang Pan and Vaclav Skala. 2012. Surface Reconstruction with higher-order smoothness. *The Visual Computer* 28, 2 (2012), 155–162.

Mark Pauly, Richard Keiser, Leif P. Kobbelt, and Markus Gross. 2003. Shape Modeling with Point-sampled Geometry. *ACM Trans. Graph.* 22, 3 (July 2003), 641–650.

Roi Poranne, Craig Gotsman, and Daniel Keren. 2010. 3D surface reconstruction using a generalized distance function. In *Computer Graphics Forum*, Vol. 29. Wiley Online Library, 2479–2491.

Vaughan Pratt. 1987. Direct Least-squares Fitting of Algebraic Surfaces. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '87)*. 145–152.

M. Samozino, M. Alexa, P. Alliez, and M. Yvinec. 2006. Reconstruction with Voronoi Centered Radial Basis Functions. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*. 51–60.

Nico Schertler, Bogdan Savchynskyy, and Stefan Gumhold. 2017. Towards Globally Optimal Normal Orientations for Large Point Clouds. *Comput. Graph. Forum* 36, 1 (Jan. 2017), 197–208.

Bernhard Schölkopf, Joachim Giesen, and Simon Spalinger. 2004. Kernel Methods for Implicit Surface Modeling. In *Proceedings of the 17th International Conference on Neural Information Processing Systems (NIPS'04)*. 1193–1200.

Chen Shen, James F. O'Brien, and Jonathan R. Shewchuk. 2004. Interpolating and Approximating Implicit Surfaces from Polygon Soup. *ACM Trans. Graph.* 23, 3 (Aug. 2004), 896–904.

Alex J. Smola and Bernhard Schökopf. 2000. Sparse Greedy Matrix Approximation for Machine Learning. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML '00)*. 911–918.

Tibor Stanko, Stefanie Hahmann, Georges-Pierre Bonneau, and Nathalie Saguin-Sprynski. 2016. Surfacing curve networks with normal control. *Computers & Graphics* 60 (2016), 1–8.

Gabriel Taubin. 2012. Smooth signed distance surface reconstruction and applications. In *Iberoamerican Congress on Pattern Recognition*. Springer, 38–45.

Greg Turk and James F. O'Brien. 2002. Modelling with Implicit Surfaces That Interpolate. *ACM Trans. Graph.* 21, 4 (Oct. 2002), 855–873.

Amir Vaxman, Marcel Campen, Olga Diamanti, Daniele Panozzo, David Bommes, Klaus Hildebrandt, and Mirela Ben-Chen. 2016. Directional field synthesis, design, and processing. In *Computer Graphics Forum*, Vol. 35. Wiley Online Library, 545–572.

C. Walder, O. Chapelle, and B. Schölkopf. 2005. Implicit Surface Modelling as an Eigenvalue Problem. In *Proceedings of the 22nd International Conference on Machine Learning*. ACM, 937–944.

Christian Walder, Olivier Chapelle, and Bernhard Schölkopf. 2007. Implicit surfaces with globally regularised and compactly supported basis functions. In *Advances in Neural Information Processing Systems*. 273–280.

Jun Wang, Zhouwang Yang, and Falai Chen. 2011. A variational model for normal computation of point clouds. *The Visual Computer* 28 (2011), 163–174.

Holger Wendland. 2004. *Scattered data approximation*. Vol. 17. Cambridge university press.

Hui Xie, Jianning Wang, Jing Hua, Hong Qin, and Arie E. Kaufman. 2003. Piecewise C1 Continuous Surface Reconstruction of Noisy Point Cloud via Local Implicit Quadric Regression. In *14th IEEE Visualization 2003 Conference, VIS 2003, Seattle, WA, USA, October 19-24, 2003*. 91–98.

Hong-Kai Zhao, Stanley Osher, and Ronald Fedkiw. 2001. Fast surface reconstruction using the level set method. In *Proceedings IEEE Workshop on Variational and Level Set Methods in Computer Vision*. IEEE, 194–201.