



Sparse RBF surface representations



Manyi Li^a, Falai Chen^{b,*}, Wenping Wang^c, Changhe Tu^{a,*}

^a Shandong University, China

^b University of Science and Technology of China, China

^c The University of Hong Kong, Hong Kong

ARTICLE INFO

Article history:

Received 21 May 2016

Accepted 1 August 2016

Available online 12 August 2016

Keywords:

Radial basis function

Sparse representation

Medial axis

ABSTRACT

In this paper, we propose a sparse surface representation for arbitrary surface models (point clouds, mesh models, continuous surface models, etc.). We approximate the input surface model with radial basis functions (RBF) whose centers are located on the medial axis of the input object surface. The sparsity of the RBF representation is achieved by solving an L_1 optimization problem. Experimental results demonstrate that our method needs much less number of parameters to represent the input surface model with good accuracy. The sparse representation is useful in various applications, such as saving memory space in storing the surface models, and saving time in transmission of the surface models on the internet.

© 2016 Elsevier B.V. All rights reserved.

1. Introduction

Surface representation is a fundamental issue in the community of Computer Aided Design and Computer Graphics. Popular representations include parametric representation, implicit representation and mesh representation. Among various surface representations, implicit representation is a flexible and effective tool to describe smooth surfaces with complex shape and topology. An implicit representation generally can be expressed as a linear combination of some basis functions, such as radial basis functions, B-spline functions, etc. There have been many approaches on reconstructing implicit surface representations from point clouds, such as radial basis functions (RBF) (Carr et al., 2001), multilevel partition of unity (MPU) (Ohtake et al., 2005b), Poisson reconstruction (Hoppe, 2008) and others, focusing on goals like improving accuracy, handling large and noisy data sets, etc. However, the resulting implicit function constructed by such methods generally consists of lots of terms—tens of thousands or even millions of terms, depending on the number of input points and the complexity of the surface model. Such representations add great burden on storage and on subsequent computations such as boolean operations. In this paper, we have a different goal—constructing sparse representation of the surface model, that is, representing the surface model with fewest possible terms while preserving the required accuracy of the model. With the sparse representation, less storage space is needed to store the surface model and less transmission time is needed to transfer the surface model on the internet, which may be useful in many online applications such as constructing 3D icons of objects in 3D repository on the websites.

There are two key issues in sparse representation of surface models. Firstly, how to balance the sparsity and accuracy of the surface model. Generally sparsity decreases the accuracy of the surface model. For a given accuracy, how many basis functions should be selected in the implicit representation is a big problem. Secondly, how to distribute the basis functions

* Corresponding author.

E-mail addresses: chenfl@ustc.edu.cn (F. Chen), chtu@sdu.edu.cn (C. Tu).

in appropriate locations. Previously, in implicit surface reconstruction, some techniques have been proposed to reduce the number of basis functions (Carr et al., 2001; Ohtake et al., 2005a, 2006; Samozino et al., 2006). The common method is to first select some possible centers, then solve a least-square approximation problem. However, the two major problems—how many basis functions to select and how to select the centers of the basis functions are not well solved in these methods. In this paper, we propose a sparse optimization method to solve the problem. We express the implicit representation using radial basis functions, i.e., the implicit function is a linear combination of radial basis functions. The centers of the basis functions are located on the medial axis of the input surface model. We propose an L_1 optimization algorithm to find the sparse centers. Compared with previous approaches, our method produces a more compact implicit representation for a surface model while preserving required accuracy.

We emphasize some subtle differences between our method and previous implicit surface reconstruction methods as they are closely related. As stated before, our goal is to compute a sparse representation for a surface model. Thus the input can be any surface representation—point cloud, mesh model, parametric model or even implicit model. From the input surface model, we find an approximation of the surface model with fewest possible terms with required accuracy. This is not exactly the same problem with implicit surface reconstruction from point clouds. So we will not consider problems in implicit surface reconstruction, e.g., point clouds with noise or incomplete data sets.

The outline of this paper is as follows. Section 2 discusses some related works. Section 3 reviews some preliminary knowledge about radial basis functions and sparse optimization. Section 4 presents the optimization model together with an algorithm to compute the sparse representation, including a strategy to sample the medial axis points. The experimental results and comparisons are demonstrated in section 5. We conclude the paper in section 6 with some future research problems.

2. Related work

In this section, we discuss some related works on sparse representation of surface models—implicit surface reconstruction, sparse optimization and medial axis transformation.

2.1. Implicit surface reconstruction

Given an input point cloud, implicit surface reconstruction is to find a smooth implicit function which is usually expressed as a linear combination of a set of basis functions to fit the point cloud. In the past several decades, there is a wealthy literature focusing on implicit surface reconstruction problem, and various approaches have been proposed, such as Blinn's blobs, signed distance functions, radial basis functions (RBF), multilevel partition of unity(MPU), moving least squares (MLS), level sets, Poisson, algebraic splines, etc. For a complete reference, the reader is referred to Berger et al. (2016), Wang et al. (2011), Pan et al. (2016).

One problem with these approaches is that a large body of data is needed to express the implicit function, especially for complex models. This will create a big burden for storage and transmission on the internet. To solve this problem, several techniques have been developed to reduce the amount of data. J.C. Carr put forward a method to construct an implicit surface with RBFs and presented a greedy algorithm to iteratively append centers with large residuals to reduce the number of basis functions (Carr et al., 2001). However, this reduction can't guarantee the most sparse solution. Y. Ohtake combined the adaptive partition of unity and least-square RBF fitting to approximate the point clouds (Ohtake et al., 2006). He first selected a set of surface points whose support volume can well cover the shape and overlap more than a threshold at each point, then used least-square method to solve the weights of RBF centers. Different from those methods with the centers of the basis functions on the surface or offset, M. Samozino (2006) proposed to place the RBFs' centers on the Voronoi vertices. This method first selects a user-specified number of centers by filtering and clustering from a subset of Voronoi vertices, then solves a least-square problem to get the reconstructed surface. However, it approximates the surface and center points equally, which causes larger approximation error on surface.

In all, although several techniques have been proposed to reduce the number of basis functions in implicit surface reconstruction, they all adopt the idea of selecting the centers of basis functions before fitting. The two key issues—balancing between sparsity and accuracy and finding the best locations of the centers of basis functions in the implicit representation are not well solved. In this paper, we will present a new approach based on sparse optimization to solve the problem.

2.2. Sparse optimization

A signal can be expressed as a linear combination of some atom functions. A signal is called sparse if most of the combination coefficients are zero. Sparse optimization is to find a sparse solution in all the feasible solutions of a problem.

Sparse optimization has become a very popular technique in signal processing and computer vision (Elad, 2010; Wright et al., 2010). Comparatively, the use of sparse optimization in geometric modeling has just begun. Recently, this technique has been applied in geometric modeling and graphics problems, such as mesh denoising (He and Schaefer, 2013; Wang et al., 2014), shape matching (Bouaziz et al., 2013; Ovsjanikov et al., 2012), shape segmentation (Liu et al., 2015; Hu et al., 2012), point cloud consolidation (Avron et al., 2010; Mustafa et al., 2015), curve fitting (Kang et al., 2015), shape deformation (Deng et al., 2013), etc. For a survey on these researches, the reader is referred to Xu et al. (2015).

In this paper, we apply the sparse optimization technique to select a set of sparse radial basis functions such that the resulting implicit function approximates the input surface model with prescribed accuracy.

2.3. Medial axis transform

Medial axis transform (MAT) (Blum, 1967) is defined as the set of balls, the union of which is exactly the boundary of objects. Medial axis is the centers of the medial balls. Thus the medial axis is composed of all the points with more than two closest boundary points. Medial axis has been widely used in computer graphics and geometric processing problems (Siddiqi et al., 2008; Yoshizawa et al., 2007; Siddiqi and Pizer, 2008).

For a given geometric object, how to extract an accurate and stable medial axis transform has been a very hot topic. There is a lot of literature focusing on the problem. N. Amenta et al. (2001) constructed the approximated medial axis with poles, which is a subset of the Voronoi vertices of the sample points from the object. However, the medial axis based on Voronoi diagram is very sensitive to small perturbations on the input geometry. Furthermore, the extracted medial axis generally contains many tiny and messy branches which have to be removed to get a clear and stable medial axis. Several techniques have been explored to solve the problem, such as by separation angles (formed by the vectors connecting a medial axis point to the closest points on the boundary) (Chazal and Lieutier, 2005), by scale factors (Giesen et al., 2009; Miklos et al., 2010), or other criterions (Attali and Montanvert, 1997; Dey and Zhao, 2004; Foskey et al., 2003).

3. Preliminary knowledge

In this section, we present some preliminary knowledge about radial basis functions and sparse optimization.

3.1. Radial basis functions

An implicit function is generally expressed as a linear combination of some basis functions:

$$f(\mathbf{X}) := \sum_{i=1}^n \lambda_i \phi_i(\mathbf{X}) = 0, \quad (1)$$

where $\phi_i(\mathbf{X})$, $i = 1, 2, \dots, n$ are basis functions, and λ_i , $i = 1, 2, \dots, n$ are combination coefficients. \mathbf{X} is the variables in the geometric space, for example, in two dimensional space $\mathbf{X} = (X_1, X_2)$, and in three dimensional space $\mathbf{X} = (X_1, X_2, X_3)$.

In this paper, we adopt radial basis functions $\phi_i(\mathbf{X}) = \phi(\|\mathbf{X} - \mathbf{c}_i\|/r_i)$, where $\phi(\gamma)$ is a nonnegative function defined on $[0, +\infty)$ which generally has the following properties (Carr et al., 2001):

- (1) $\phi(0) = 1$;
- (2) $\lim_{\gamma \rightarrow +\infty} \phi(\gamma) = 0$;

\mathbf{c}_i is the center of the basis function $\phi_i(\mathbf{x})$, and $r_i > 0$ is the influence radius of the radial basis function which will be discussed in details in Section 4.

A typical choice of radial basis function is Gaussian function

$$\phi(\gamma) = e^{-\gamma^2}. \quad (2)$$

Besides the Gaussian function, there are other radial basis functions including compactly supported radial basis functions (CSRBF), for example, $\phi(\gamma) = (1 - \gamma)^4(1 + 4\gamma)$ for $0 \leq \gamma \leq 1$, and otherwise $\phi(\gamma) = 0$.

3.2. Sparse optimization

Let $\mathbf{u} = (u_1, u_2, \dots, u_n)$ be an n -dimensional vector. The L_p norm of \mathbf{u} is defined as $\|\mathbf{u}\|_p = (\sum_{i=1}^n |u_i|^p)^{1/p}$ ($0 < p < +\infty$). The L_0 pseudo norm of \mathbf{u} is defined as $\|\mathbf{u}\|_0 := \#\{u_i | u_i \neq 0\}$, i.e., the number of nonzero elements in \mathbf{u} . A vector \mathbf{u} is called sparse if $\|\mathbf{u}\|_0 \ll n$.

Consider solving a linear system of equations

$$\mathbf{A}\mathbf{u} = \mathbf{b}. \quad (3)$$

Assuming there are infinite number of solutions in the system, we hope to find the sparsest solution among all the feasible solutions. This amounts to solving the following optimization problem

$$\begin{aligned} \min \quad & \|\mathbf{u}\|_0 \\ \text{s.t.} \quad & \mathbf{A}\mathbf{u} = \mathbf{b}. \end{aligned} \quad (4)$$

The above optimization is a NP-hard problem. To efficiently solve the above problem, we relax the L_0 norm to L_1 norm:

$$\begin{aligned} \min \quad & \|\mathbf{u}\|_1 \\ \text{s.t.} \quad & \mathbf{A}\mathbf{u} = \mathbf{b}. \end{aligned} \quad (5)$$

This problem is convex and can be efficiently solved by techniques such as least absolute shrinkage and selection operator(LASSO). It is shown that the solutions of the two problems (Eq. 4) and (Eq. 5) are equivalent under certain conditions (Tropp, 2006).

The problem (Eq. 5) can be rewritten as a unconstrained optimization problem

$$\min \quad \mu\|\mathbf{u}\|_1 + \|\mathbf{u} - \mathbf{b}\|_2^2 \quad (6)$$

where $\mu > 0$ is a parameter which balances the two targets: sparsity and accuracy of solutions.

4. Model and algorithm

In this section, we first give a brief overview about the problem and the overall approach, then present the strategy to sample points on the medial axis of the input object. Finally, we put forward the optimization model and algorithm to compute the sparse representation of the input surface.

4.1. Overview

Given an input model (point cloud, polygonal mesh, parametric equation or even implicit equation), our goal is to find an RBF implicit function $f(\mathbf{X})$ as defined in (Eq. 1) to approximate the input model such that the expression of $f(\mathbf{X})$ has fewest possible terms with a prescribed accuracy. In this paper, we use Gaussian basis functions as described in Section 3.1, even though other representations of radial basis functions such as CSRBF should also apply.

Unlike traditional approaches which put the centers of the radial basis functions on the input surface and/or offset points, we propose to put the centers $\{\mathbf{c}_i\}_{i=1}^n$ of the RBFs on the medial axis of the input object. This idea is motivated by the extreme case—a sphere can be expressed by just one RBF with its center in the center of the sphere, which is the sparsest representation of the sphere. The parameter r_i is taken as the radius of the center point \mathbf{c}_i , i.e., r_i is the distance from the center \mathbf{c}_i to the boundary of the input model.

To begin with, we sample the input model to get a set of points $\{\mathbf{P}_i\}_{i=1}^N$ on the given surface and extract the medial axis M of the input model. Following the work in Amenta et al. (2001), we compute poles from the Voronoi diagram of the input points. It has been proved that when the input points are dense enough, the extracted poles should be a good approximation of the medial axis.

In most cases, the number of medial axis points computed in the first step is very large, making it hard to compute in the later optimization process. As a result, a sampling process is necessary to select a sufficient subset of medial axis points. We propose a sampling algorithm to select the sampled points based on the surface details, i.e. the radii of medial axis points. After this step, we obtain a set of points $C := \{\mathbf{c}_i\}_{i=1}^n$ on the medial axis which are used as the initial centers of the RBFs.

After the sampling process, we further reduce the number of RBFs by proposing an optimization model which fits the given point set $\{\mathbf{P}_i\}_{i=1}^N$ and at the same time produces a sparse solution. The optimization is a trade-off between the accuracy and sparsity, which can be controlled by some parameters. Our optimization algorithm includes two stages. The first stage—sparse optimization is to obtain a sparse set of basis functions, while the second stage—accuracy optimization is to improve approximation accuracy. The sparse optimization problem is solved by alternating direction method of multipliers (ADMM).

Fig. 1 illustrates a specific example to demonstrate the process of our algorithm and the results of each step.

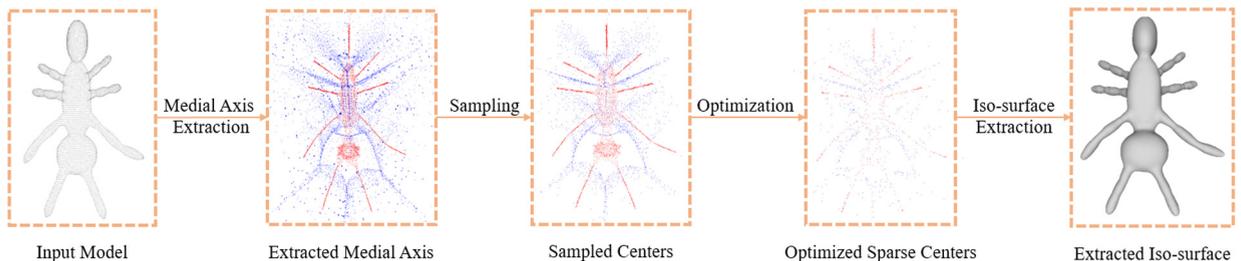


Fig. 1. The process of our algorithm and results in each step. The inner medial axis points are drawn in red, the outer ones are blue. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4.2. Sampling medial axis

The goal of the sampling step is to reduce the scale of optimization problem, making the problem solvable in memory and time. Via sampling, we find a subset of medial axis points as sampled centers, with which the optimization can approximate the surface well. Thus, the sampled points should be reasonably distributed on the medial axis with certain density. There is a balance between the number of sampled points and the approximation error. While few sampling points will increase the approximation error, too many sampled points may contain redundancy and make the scale of the problem too large. Secondly, local features need more samples. One reason is that local features call for more basis functions to be approximated. The other reason is that the corresponding basis functions at local features often have smaller local supports.

Considering these requirements, we develop a sampling algorithm to decide the number of sampled points automatically. The sampled points are selected based on the surface details, i.e. the radii of medial axis points.

In the sampling, all the surface points should be covered by the sampled centers with slightly expanded radii, i.e. all surface points should satisfy $\min_{\mathbf{c}_i \in C_{inner}} \frac{d(\mathbf{p}, \mathbf{c}_i)}{r_i} < s$ and $\min_{\mathbf{c}_i \in C_{outer}} \frac{d(\mathbf{p}, \mathbf{c}_i)}{r_i} < s$, where C_{inner} and C_{outer} are the set of inner and outer samples, $d(\mathbf{p}, \mathbf{c}_i)$ is the distance between surface point \mathbf{p} and sampled center \mathbf{c}_i , $s > 1$ is a parameter to control the sampling density. The selection is to iteratively append medial axis points as sampled centers, until the requirements are satisfied. Compared with the random method and the clustering method in [Samožino et al. \(2006\)](#), our algorithm is more robust and doesn't need the users to decide the number of sampled points for different models.

The sampling parameter s controls the sampling density which influences the sparsity of the implicit representation and the approximation error. Generally, as s increases, the sampling density decreases, whereas the sparsity of the implicit and the approximation error increases. Further details will be discussed in section 5.

After the sampling, some outer distant points on the medial axis are filtered out. Because the outer distant points have little influence on the approximation of surface, and are less necessary to be approximated.

4.3. Optimization

After the sampling step, we hope to find a sparse weight vector $\lambda = (\lambda_1, \dots, \lambda_n)$ of $f(\mathbf{X})$ as defined in (Eq. 1) such that $f(\mathbf{X}) = 0$ is a good fit to the point set $\{\mathbf{P}_i\}_{i=1}^N$. Our optimization process contains two stages—sparse optimization and accuracy approximation. We first apply the L_1 minimization to perform the sparse optimization, and drop the terms whose weights are close to zero to find the sparse centers of the radial basis functions. Then we perform a second approximation without sparsity term to obtain a more accurate solution.

The inputs of the optimization algorithm include the surface points $\{\mathbf{P}_i\}_{i=1}^N$, the sampled medial axis points $\{\mathbf{c}_i\}_{i=1}^n$ which are served as the centers of the RBFs, and the offset points $S = \{s_{i_{inner}}, s_{i_{outer}}\}_{i=1}^m$. The offset points are computed along two opposite oriented normals, which can be approximated by the directions from each surface point to its corresponding medial axis points. We can sample one pair of inner and outer offset points for a few surface points. The output is a sparse representation of the input surface model in the form (Eq. 1) but with most of the terms being removed.

4.3.1. Sparse optimization

In this stage, we compute a sparse solution of the implicit representation (Eq. 1) which fits the given surface points and offset points, and at the same time loosely fits the signed distance function of the input object. The objective function contains four terms: sparsity term $\|\lambda\|_1$, surface point term $E(P)$, medial axis point term $E(C)$, and offset point term $E(S)$. The objective function of sparse optimization can be expressed as

$$\min \rho_w \|\lambda\|_1 + \rho_s E(P) + \rho_m E(C) + \rho_o E(S). \quad (7)$$

The sparsity term is the L_1 -norm of the variables $\lambda = (\lambda_1, \dots, \lambda_n)$, which are the weights of the radial basis functions in $f(\mathbf{X})$. The other terms are to approximate the signed distance functions of the input object at different points, as discussed below.

The surface point term is to reconstruct the shape of the surface, and the function values at surface points should be all zero. This approximation term can be written as

$$E(P) = \sum_{i=1}^N (f(\mathbf{P}_i))^2. \quad (8)$$

The medial axis point term is to avoid collapse of the implicit function inside the shape. The corresponding term is

$$E(C) = \sum_{j=1}^M (f(\mathbf{c}_j) - r_j * \text{label}(\mathbf{c}_j))^2, \quad (9)$$

here $\text{label}(\mathbf{c}_j) = 1$ if \mathbf{c}_j is an inner medial axis point, and $\text{label}(\mathbf{c}_j) = -1$ if \mathbf{c}_j is an outer medial axis point.

The offset point term is an additional constraint in order to avoid occurrence of extra connected iso-surfaces of the implicit function $f(\mathbf{X})$. The offset points contain inner and outer offset points. The inner offset points have positive signed

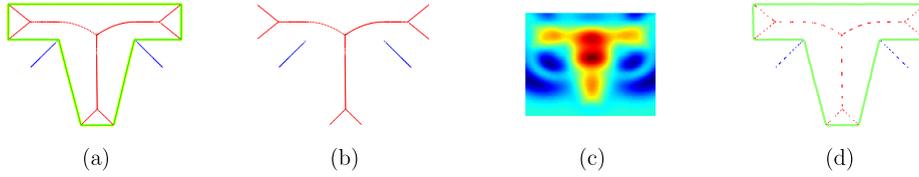


Fig. 2. Illustration of the optimization process. From left to right: (a) shows the constraint points: surface points (green), offset points (yellow) and sampled centers (inner centers are red and outer centers are blue). (b) shows the centers of basis functions: inner sampled centers (red) and outer sampled centers (blue). (c) is the approximated distance field, where color represents the implicit function values. (d) is the extracted contour (green) and the sparse centers (inner centers are red and outer centers are blue) after optimization. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

distances and the outer offset points have negative ones. The distances are their offset values. The offset approximation term can be written as

$$E(S) := \sum_{k=1}^m [(f(s_{k_{inner}}) - d_k)^2 + (f(s_{k_{outer}}) + d_k)^2], \quad (10)$$

where d_k is the offset value for each pair of points.

According to the importance of each term, the parameters are chosen with different values. Generally, ρ_s should be given a large value since the corresponding term measures the accuracy of the approximation. The parameters ρ_m and ρ_o can be relatively small since these terms are used just to control the overall shape of the implicit function $f(\mathbf{X})$, and exact approximation is not necessary. The parameter ρ_w is used to control the sparsity of the solution. Larger value of ρ_w would induce a sparser solution.

After the variable λ is optimized, we filter out those points in $\{\mathbf{c}_i\}_{i=1}^n$ whose absolute values of weights λ_i are smaller than a specified constant δ . The retained points are the sparse centers of the radial basis functions in the implicit representation $f(\mathbf{X})$. Fig. 2 gives an example to illustrate the optimization process in 2D case.

4.3.2. Optimization algorithm

With abuse of notations, the objective function of the optimization problem (Eq. 7) can be written as a quadratic function

$$E(\mathbf{x}) := \omega|\mathbf{x}|_1 + \mathbf{x}^T \mathbf{A} \mathbf{x} + \mathbf{b}^T \mathbf{x}, \quad (11)$$

where the variable $\mathbf{x} = \lambda$ is the weight vector of the basis functions in the implicit function $f(\mathbf{X})$. A is a positive definite matrix of size $n \times n$, and \mathbf{b} is a n -dimensional column vector.

We apply ADMM algorithm to solve the optimization problem. Rewrite (Eq. 11) as

$$\begin{aligned} \min \quad & \omega|\mathbf{y}|_1 + \mathbf{x}^T \mathbf{A} \mathbf{x} + \mathbf{b}^T \mathbf{x} \\ \text{s.t.} \quad & \mathbf{y} - \mathbf{x} = \mathbf{0}. \end{aligned} \quad (12)$$

The augmented Lagrangian of the above problem is:

$$L_\rho(\mathbf{x}, \mathbf{y}) = \omega|\mathbf{y}|_1 + \mathbf{x}^T \mathbf{A} \mathbf{x} + \mathbf{b}^T \mathbf{x} + \mathbf{z}^T (\mathbf{y} - \mathbf{x}) + \frac{\rho}{2} \|\mathbf{y} - \mathbf{x}\|_2^2. \quad (13)$$

In the ADMM solver, we iteratively update the variables \mathbf{x} , \mathbf{y} , \mathbf{z} as described below:

$$\mathbf{x}^{k+1} = \min (\mathbf{x}^k)^T \mathbf{A} \mathbf{x}^k + \mathbf{b}^T \mathbf{x}^k - \mathbf{z}^T \mathbf{x}^k + \frac{\rho}{2} (\mathbf{x}^k)^T \mathbf{x}^k - \rho (\mathbf{x}^k)^T \mathbf{y}^k, \quad (14)$$

$$\mathbf{y}^{k+1} = \min \omega|\mathbf{y}^k|_1 + (\mathbf{z}^k)^T \mathbf{y}^k + \frac{\rho}{2} (\mathbf{y}^k)^T \mathbf{y}^k - \rho (\mathbf{x}^{k+1})^T \mathbf{y}^k, \quad (15)$$

$$\mathbf{z}^{k+1} = \mathbf{z}^k + \rho (\mathbf{y}^{k+1} - \mathbf{x}^{k+1}). \quad (16)$$

For most examples, the algorithm converges in about 200 iterations.

4.3.3. Accuracy optimization

After sparse optimization, we get a set of sparse radial basis functions. Using this set of basis functions, we compute the corresponding weights using a similar optimization as (Eq. 7)—only with the absence of sparsity term (i.e., $\rho_w = 0$). We continue to use the same ADMM algorithm to iterate. This accuracy optimization is a supplement to improve the accuracy with generated sparse basis functions. Generally, the accuracy optimization takes at most a few seconds, and it can reduce by more than a half of the approximation error with little cost.

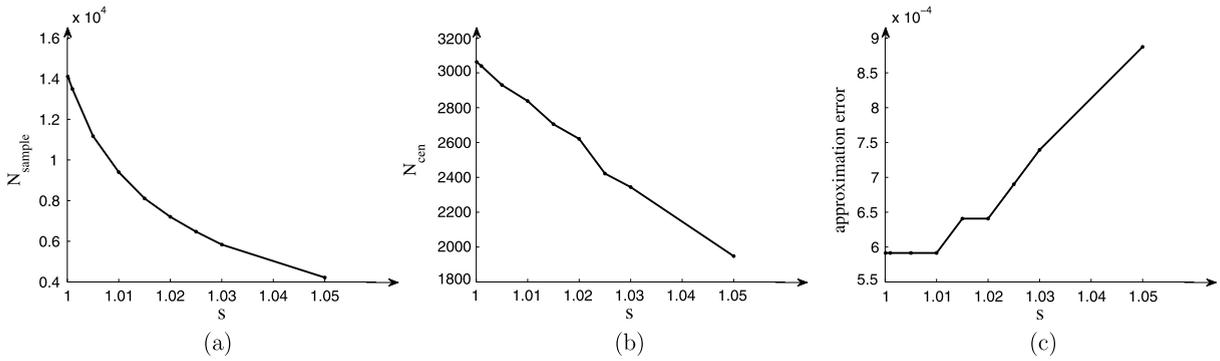


Fig. 3. The effect of the parameter s on sampling density, sparsity of the implicit representation and approximation error for the Bimba model. From left to right: (a) the number of sampling points with respect to the parameter s ; (b) the number of sparse centers over the parameter s ; (c) the approximation error with respect to the parameter s .

5. Results and discussion

In this section, we present some experimental examples to illustrate sparsity of our representation. Comparisons are made with related approaches such as the Voronoi diagram based RBFs (Samozino et al., 2006), and the adaptive partition of unity combined with sparse RBFs (Ohtake et al., 2006). An application of our sparse representation in 3D icon construction is also presented.

5.1. Implementation details

We use following formula to measure the approximation error between the input model and the implicit surface $f(\mathbf{X}) = 0$ (Taubin, 1994):

$$e_{avg} = \frac{1}{L} \sqrt{\frac{1}{N} \sum_{i=1}^N \frac{(f(p_i))^2}{\|\nabla f(p_i)\|^2}}, \quad (17)$$

where L is the diagonal length of the object size. In this paper, we scale the input model into the bounding box $[-1, 1]^3$. The computations are performed using Matlab, running on a 3.50 GHz computer with 16 GB RAM.

In our algorithm, there are some parameters to control the results. For sampling process, the only parameter s is for the sampling density which also influences the sparsity of our implicit representation and approximation error. Generally, as s increases, the sampling density decreases, whereas the sparsity of the implicit representation and the approximation error increases. However, when the sampling density reaches a certain level, improvement on the approximation error is not obvious. Fig. 3 demonstrates the influence of parameter s on the sampling density, sparsity of the implicit representation, and the approximation error for the Bimba model. From our experiments, $s = 1.005$ is a reasonable choice to get sufficient sampling points to reach good approximation and sparsity.

For optimization process, there are four parameters in the objective function (Eq. 7) to control the sparsity and quality of the approximating result. The parameter ρ_w controls the sparsity, while the others ρ_s , ρ_m , ρ_o are relevant to the quality of the approximation. Generally, ρ_s should be chosen large in order to get a good approximation to the input model, while ρ_m and ρ_o should be relatively small. In our experiments, we set these parameters as $\rho_w = 0.1$, $\rho_s = 1000$, $\rho_m = 10$, $\rho_o = 300$.

The threshold δ controls the removal of redundant centers. This threshold should filter out most redundant centers while keep the quality of the implicit representation. Fig. 4 presents the average sparsity and accuracy of a set of models with different δ . Generally, $\delta = 0.01$ is a reasonable choice to preserve the accuracy.

5.2. Results

Fig. 5 illustrates some examples to demonstrate the fitting results of sparse approximation algorithm. The corresponding color maps of the approximation errors are also shown, where blue means smaller values and red indicates larger values. The color maps in this paper are all drawn in the same metric. Column 3 in Table 1 shows the statistics of the fitting results, including the number of sparse centers of the radial basis functions and approximation error for each model. From the statistics, we can see that the number of sparse centers N_{cen} is significantly smaller than the number of input points N with an average error of order 10^{-4} . For models (a)–(d), N_{cen} is about one fifth of N . The sparsity gets more obvious when the models are smooth and fair, i.e., the curvatures of the models do not oscillate too much. For the models (e) and (f), the ratio of N_{cen} over N is even smaller, it reaches $1/15$ and $1/20$ respectively. Note that the storage of the implicit representation is measured by the number of parameters N_{para} in the representation which is the number of sparse centers

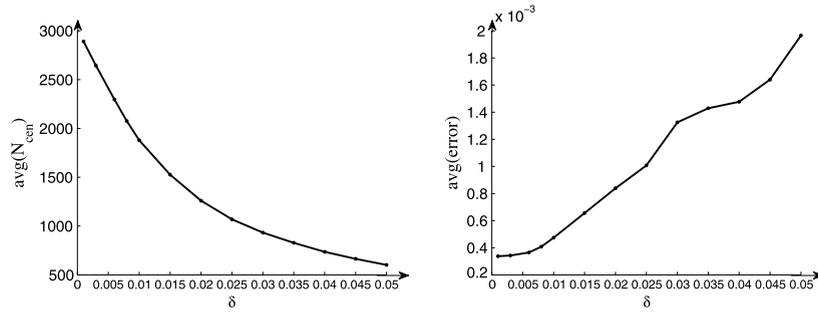


Fig. 4. The sparsity and accuracy with different δ in optimization. N_{cen} is the number of remained centers after sparse optimization, $error$ is the approximation error computed with Eq. 17. The figures show the average N_{cen} and $error$ of a set of models. Thus, $\delta = 0.01$ is a reasonable choice to balance between sparsity and accuracy.

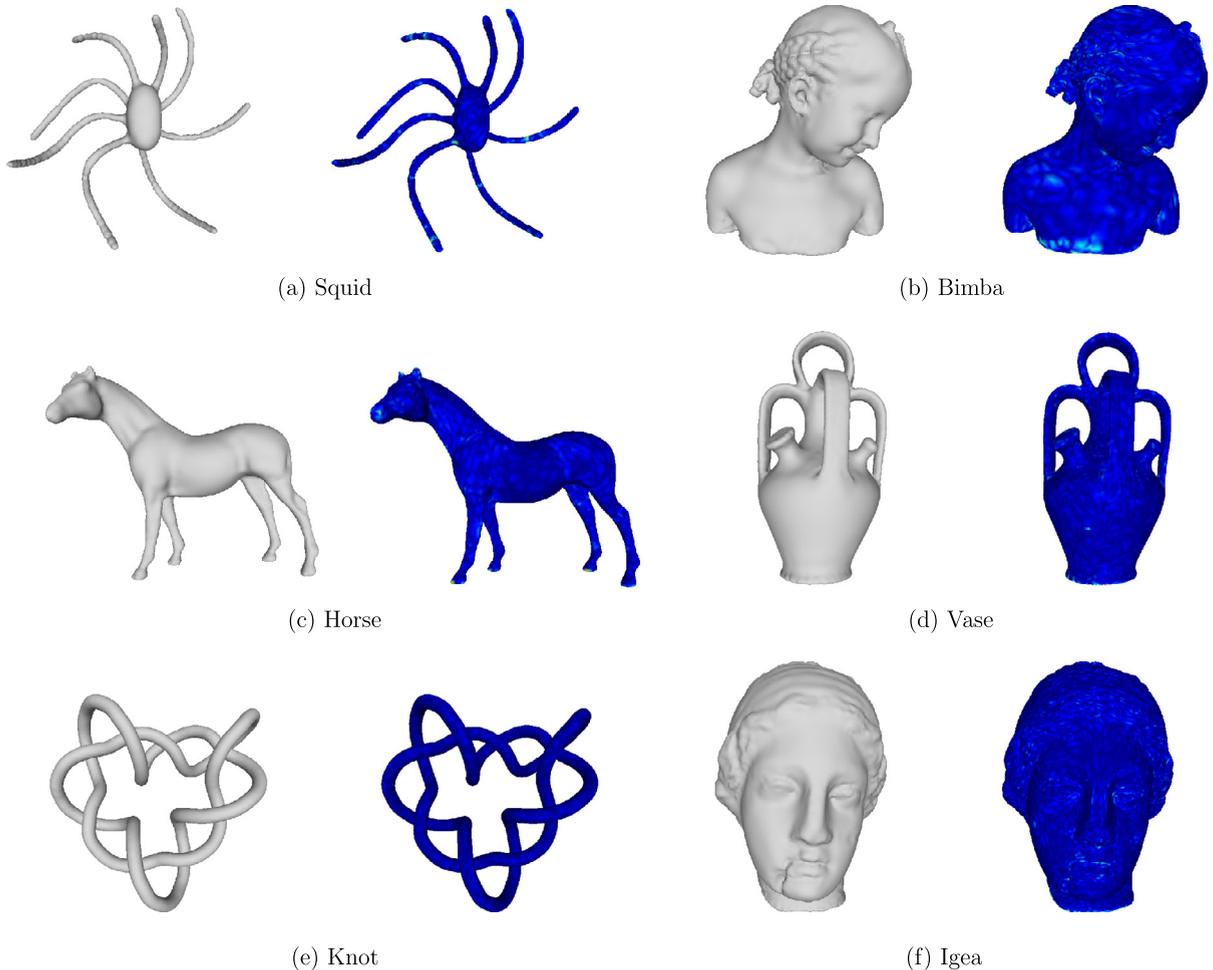


Fig. 5. Fitting results of our optimization algorithm. (For interpretation of the colors in this figure, the reader is referred to the web version of this article.)

N_{cen} multiplied by five (the center \mathbf{c}_i , radius r_i and the coefficient λ_i), while the storage of the input model is the number of points N multiplied by three.

5.3. Comparisons

Next we compare our sparse optimization method with related works, e.g. the Voronoi diagram based RBF method (Samozino et al., 2006) and the partition of unity based RBF method (Ohtake et al., 2006). Table 1 lists the statistic results for the three methods. For a comparison with the Voronoi diagram based RBF method, both methods locate the centers of

Table 1

Statistics for our method, the Voronoi centered RBFs (Samozino et al., 2006) and the adaptive partition of unity combined with sparse RBFs (Ohtake et al., 2006). The sparsity of each method is measured by the number of parameters N_{para} with the given number of input points.

Model	Input points	Our method	Voronoi-based RBF method (Samozino et al., 2006)	Sparse PU+RBF method (Ohtake et al., 2006)
Squid	6325	$N_{cen} = 1216$ $N_{para} = 6080$ $e_{avg} = 4.63e - 4$	$N_{cen} = 5589$ $N_{para} = 27945$ $e_{avg} = 4.29e - 4$	$N_{cen} = 3774$ $N_{para} = 52836$ $e_{avg} = 4.31e - 4$
Bimba	15516	$N_{cen} = 3156$ $N_{para} = 15780$ $e_{avg} = 6.24e - 4$	$N_{cen} = 9458$ $N_{para} = 47290$ $e_{avg} = 7.89e - 4$	$N_{cen} = 3874$ $N_{para} = 54236$ $e_{avg} = 5.83e - 4$
Horse	8078	$N_{cen} = 1619$ $N_{para} = 8095$ $e_{avg} = 4.17e - 4$	$N_{cen} = 4950$ $N_{para} = 24750$ $e_{avg} = 7.52e - 4$	$N_{cen} = 3732$ $N_{para} = 52248$ $e_{avg} = 4.46e - 4$
Vase	14859	$N_{cen} = 2664$ $N_{para} = 13320$ $e_{avg} = 3.82e - 4$	$N_{cen} = 8937$ $N_{para} = 44685$ $e_{avg} = 5.77e - 4$	$N_{cen} = 9661$ $N_{para} = 135254$ $e_{avg} = 4.15e - 4$
Knot	40848	$N_{cen} = 2778$ $N_{para} = 13890$ $e_{avg} = 1.18e - 4$	$N_{cen} = 4809$ $N_{para} = 24045$ $e_{avg} = 1.03e - 4$	$N_{cen} = 10526$ $N_{para} = 147364$ $e_{avg} = 2.21e - 4$
Igea	116737	$N_{cen} = 6001$ $N_{para} = 30005$ $e_{avg} = 2.68e - 4$	$N_{cen} = 14765$ $N_{para} = 73825$ $e_{avg} = 6.53e - 4$	$N_{cen} = 9346$ $N_{para} = 130844$ $e_{avg} = 3.34e - 4$

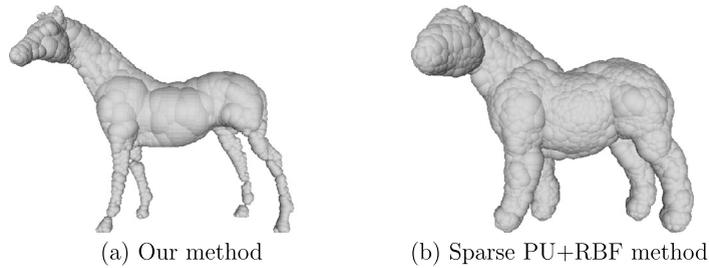


Fig. 6. Illustration of the center locations of the basis functions with our method and sparse PU+RBF method. The radii of the spheres are the radii of sparse centers, or half of the support of the basis functions with sparse PU+RBF method. This example demonstrates that placing centers on the medial axis results in more sparse representation.

basis functions on the medial axis. The results indicate that the number of sparse centers by our method is only about one fourth of that by the Voronoi diagram based RBF method with about the same approximation errors, except for the Knot model. The comparison with the partition of unity based RBF method demonstrates the effect of center locations on sparsity. The partition of unity based RBF method locates the centers on the object surface. Each basis function is composed of a quadratic polynomial in a local coordinate system and a normalized radial basis function, thus needs 14 floating numbers and calls for more mathematical operations. From Table 1, we can see that, with about the same accuracy the number of parameters by our method is 1/3 to 1/10 of that of the partition of unity based sparse RBF method, depending on the complexity of models. Fig. 6 illustrates the unions of the spheres corresponding to the basis functions by our method and by the partition of unity based RBF method respectively for the Horse model. This example demonstrates that placing centers on the medial axis results in more sparse representation.

Because the Voronoi diagram based RBF method and our method both adopt global optimization, the running time for the two methods is much longer than that of sparse partition of unity based RBF method. For example, for the Vase model which has about 15,000 input surface points, our method takes about 5 minutes in which about 3 minutes are expended in forming the matrix A in the optimization problem (Eq. 11). The computational time for the Voronoi-based RBF method is about 7 minutes whereas it is only 15 seconds for the sparse partition of unity based RBF method.

5.4. Applications in 3D icon construction

Nowadays it's a common requirement to preview the models or scenes as 3D icons in some virtual applications. The preview requires to display the models fast, where the transmission may be a limitation. The problem is how to keep more details using as little data as possible. Thus, the sparse representation is useful for such application.

To illustrate the advantage of sparse representation, we compare the file storage of our representation and some popular 3D file formats. Although many file formats compress data, the data amount is still large when the models consist of many

Table 2

The comparison of file storage (KB) between our representation and popular 3D file formats. The last column is the file storage of simplified meshes with the same accuracy with our representation.

Models	Ours	LWO	PLY	OBJ	OFF	WRL	STL	Mesh simplification (Garland and Heckbert, 1997)
Squid	72	173	404	411	422	590	618	162
Bimba	184	425	1026	1048	1070	1480	1516	357
Horse	95	221	516	527	539	753	789	122
Vase	155	407	987	1003	1030	1422	1452	226
Knot	161	1118	2737	2858	2856	3934	3990	3025
Igea	350	3717	8257	8365	8599	11677	11400	1379

complex primitives. We also compare the file storage with a mesh simplification method (Garland and Heckbert, 1997) as listed in the last column of Table 2. From the table, it can be seen that our representation uses much less storage space than other file formats to represent the same models with about the same accuracy, especially for complex models.

5.5. Limitations

There are some limitations for our sparse implicit representations. Firstly, our representation is not efficient for planar models and models with sharp features (such as a cube) due to the nature of Gaussian radial basis functions. One possibility for improvement is to introduce other types of basis functions in the sparse representations. Secondly, for complex models, enough sampling points have to be sampled in the detailed places, which leads to larger optimization problem and takes longer time to solve. Finally, the running time is also a limitation of our sparse representation. Currently the most expensive step in our algorithm is to form the matrix A in the optimization problem (Eq. 11). This should be able to greatly speed up by GPU implementation. Furthermore, the size of the optimization problem depends on the number of sampling points. Thus the suitable choice of the sampling density is also helpful for reducing the computation time.

6. Conclusion

In this paper, a sparse surface representation is proposed for arbitrary surface models. The input surface model is approximated with radial basis functions whose centers are located on the medial axis of the input object surface. The sparsity of the RBF representation is computed by solving an L_1 optimization problem. Comparisons and experimental results indicate that our method needs much less number of parameters to represent the input surface models. We demonstrate an application of our sparse representation in 3D icon construction which shows that our representation is much more compact than other file formats.

As for future work, there are some possible directions to improve our sparse representation. Firstly, radial basis functions in other forms can be investigated to improve the performance of approximation; And other types of basis functions may increase the ability to represent planar models and models with sharp features. Secondly, it will reduce lots of running time if the sparse optimization is combined with some local approximation technique. Furthermore, accelerating the algorithm of our method by GPU implementation is worthy of further investigation.

Acknowledgements

The authors would like to thank AIM@Shape Repository for the models, and Yusheng Li for the help on the optimization code. The work is supported by the NSFC Key Project (No. 61332015), NSFC project (No. 61272242), and also supported in part by the 973 Program of China under Grant 2015CB352502. And Falai Chen is supported by NSFC (No. 11571338).

References

- Amenta, N., Choi, S., Kolluri, R.K., 2001. The power crust. In: Proceedings of the Sixth ACM Symposium on Solid Modeling and Applications. ACM, pp. 249–266.
- Attali, D., Montanvert, A., 1997. Computing and simplifying 2d and 3d continuous skeletons. *Comput. Vis. Image Underst.* 67 (3), 261–273.
- Avron, H., Sharf, A., Greif, C., Cohen-Or, D., 2010. l_1 -sparse reconstruction of sharp point set surfaces. *ACM Trans. Graph.* 29 (5), 135.
- Berger, M., Tagliasacchi, A., Seversky, L.M., Alliez, P., Guennebaud, G., Levine, J.A., Sharf, A., Silva, C., 2016. State of the art in surface reconstruction from point clouds. *Comput. Graph. Forum.*
- Blum, H., 1967. A transformation for extracting new descriptors of shape. In: Wathen-Dunn, W. (Ed.), *Proc. Models for the Perception of Speech and Visual Form*. MIT Press, Cambridge, MA, pp. 362–380.
- Bouaziz, S., Tagliasacchi, A., Pauly, M., 2013. Sparse iterative closest point. *Comput. Graph. Forum*, vol. 32. Wiley Online Library, pp. 113–123.
- Carr, J.C., Beatson, R.K., Cherrie, J.B., Mitchell, T.J., Fright, W.R., McCallum, B.C., Evans, T.R., 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, pp. 67–76.
- Chazal, F., Lieutier, A., 2005. The “ λ -medial axis”. *Graph. Models* 67 (4), 304–331.
- Deng, B., Bouaziz, S., Deuss, M., Zhang, J., Schwartzburg, Y., Pauly, M., 2013. Exploring Local Modifications for Constrained Meshes. *Comput. Graph. Forum*, vol. 32. Wiley Online Library, pp. 11–20.
- Dey, T.K., Zhao, W., 2004. Approximate medial axis as a Voronoi subcomplex. *Comput. Aided Des.* 36 (2), 195–202.

- Elad, M., 2010. *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. Incorporated, 1st edition. Springer Publishing Company.
- Foskey, M., Lin, M.C., Manocha, D., 2003. Efficient computation of a simplified medial axis. *J. Comput. Inf. Sci. Eng.* 3 (4), 274–284.
- Garland, M., Heckbert, P.S., 1997. Surface simplification using quadric error metrics. In: *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*. ACM Press/Addison-Wesley Publishing Co., pp. 209–216.
- Giesen, J., Miklos, B., Pauly, M., Wormser, C., 2009. The scale axis transform. In: *Proceedings of the Twenty-Fifth Annual Symposium on Computational Geometry*. ACM, pp. 106–115.
- He, L., Schaefer, S., 2013. Mesh denoising via l_0 minimization. *ACM Trans. Graph.* 32 (4), 64.
- Hoppe, H., 2008. Poisson surface reconstruction and its applications. In: *Proceedings of the 2008 ACM Symposium on Solid and Physical Modeling*. ACM, p. 10.
- Hu, R., Fan, L., Liu, L., 2012. Co-Segmentation of 3d Shapes via Subspace Clustering. *Comput. Graph. Forum*, vol. 31. Wiley Online Library, pp. 1703–1713.
- Kang, H., Chen, F., Li, Y., Deng, J., Yang, Z., 2015. Knot calculation for spline fitting via sparse optimization. *Comput. Aided Des.* 58, 179–188.
- Liu, X., Zhang, J., Liu, R., Li, B., Wang, J., Cao, J., 2015. Low-rank 3d mesh segmentation and labeling with structure guiding. *Comput. Graph.* 46, 99–109.
- Miklos, B., Giesen, J., Pauly, M., 2010. Discrete Scale Axis Representations for 3d Geometry. *ACM Trans. Graph.*, vol. 29. ACM, p. 101.
- Mustafa, G., Li, H., Zhang, J., Deng, J., 2015. l_1 -regression based subdivision schemes for noisy data. *Comput. Aided Des.* 58, 189–199.
- Ohtake, Y., Belyaev, A., Alexa, M., 2005a. Sparse low-degree implicit surfaces with applications to high quality rendering, feature extraction, and smoothing. In: *Proc. Symp. Geometry Processing*. Citeseer, pp. 149–158.
- Ohtake, Y., Belyaev, A., Alexa, M., Turk, G., Seidel, H.-P., 2005b. Multi-level partition of unity implicits. In: *ACM SIGGRAPH 2005 Courses*. ACM, p. 173.
- Ohtake, Y., Belyaev, A., Seidel, H.-P., 2006. Sparse surface reconstruction with adaptive partition of unity and radial basis functions. *Graph. Models* 68 (1), 15–24.
- Ovsjanikov, M., Ben-Chen, M., Solomon, J., Butscher, A., Guibas, L., 2012. Functional maps: a flexible representation of maps between shapes. *ACM Trans. Graph.* 31 (4), 30.
- Pan, M., Tong, W., Chen, F., 2016. Compact implicit surface reconstruction via low-rank tensor approximation. *Comput. Aided Des.*
- Samozino, M., Alexa, M., Alliez, P., Yvinec, M., 2006. Reconstruction with Voronoi centered radial basis functions. In: *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*. SGP '06, Switzerland. Eurographics Association, Aire-la-Ville, Switzerland, pp. 51–60.
- Siddiqi, K., Pizer, S., 2008. *Medial Representations: Mathematics, Algorithms and Applications*, vol. 37. Springer Science & Business Media.
- Siddiqi, K., Zhang, J., Macrini, D., Shokoufandeh, A., Bouix, S., Dickinson, S., 2008. Retrieving articulated 3-d models using medial surfaces. *Mach. Vis. Appl.* 19 (4), 261–275.
- Taubin, G., 1994. Distance approximations for rasterizing implicit curves. *ACM Trans. Graph.* 13 (1), 3–42.
- Tropp, J.A., 2006. Just relax: convex programming methods for identifying sparse signals in noise. *IEEE Trans. Inf. Theory* 52 (3), 1030–1051.
- Wang, J., Yang, Z., Jin, L., Deng, J., Chen, F., 2011. Parallel and adaptive surface reconstruction based on implicit PHT-splines. *Comput. Aided Geom. Des.* 28 (8), 463–474.
- Wang, R., Yang, Z., Liu, L., Deng, J., Chen, F., 2014. Decoupling noise and features via weighted l_1 -analysis compressed sensing. *ACM Trans. Graph.* 33 (2), 18.
- Wright, J., Ma, Y., Mairal, J., Sapiro, G., Huang, T.S., Yan, S., 2010. Sparse representation for computer vision and pattern recognition. *Proc. IEEE* 98 (6), 1031–1044.
- Xu, L., Wang, R., Zhang, J., Yang, Z., Deng, J., Chen, F., Liu, L., 2015. Survey on sparsity in geometric modeling and processing. *Graph. Models* 82, 160–180.
- Yoshizawa, S., Belyaev, A., Seidel, H.-P., 2007. Skeleton-based variational mesh deformations. *Comput. Graph. Forum*, vol. 26. Wiley Online Library, pp. 255–264.