

Decoupling Noise and Features via Weighted ℓ_1 -Analysis Compressed Sensing

RUIMIN WANG, ZHOUWANG YANG, LIGANG LIU, JIANSONG DENG, and FALAI CHEN
University of Science and Technology of China

Many geometry processing applications are sensitive to noise and sharp features. Although there are a number of works on detecting noise and sharp features in the literature, they are heuristic. On one hand, traditional denoising methods use filtering operators to remove noise, however, they may blur sharp features and shrink the object. On the other hand, noise makes detection of features, which relies on computation of differential properties, unreliable and unstable. Therefore, detecting noise and features on discrete surfaces still remains challenging.

In this article, we present an approach for decoupling noise and features on 3D shapes. Our approach consists of two phases. In the first phase, a base mesh is estimated from the input noisy data by a global Laplacian regularization denoising scheme. The estimated base mesh is guaranteed to asymptotically converge to the true underlying surface with probability one as the sample size goes to infinity. In the second phase, an ℓ_1 -analysis compressed sensing optimization is proposed to recover sharp features from the residual between base mesh and input mesh. This is based on our discovery that sharp features can be sparsely represented in some coherent dictionary which is constructed by the pseudo-inverse matrix of the Laplacian of the shape. The features are recovered from the residual in a progressive way. Theoretical analysis and experimental results show that our approach can reliably and robustly remove noise and extract sharp features on 3D shapes.

Categories and Subject Descriptors: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—*Physically based modeling*

General Terms: Algorithms, Theory

Additional Key Words and Phrases: Denoising, asymptotic optimality, sharp feature, ℓ_1 -analysis compressed sensing

The work is supported by the 973 Program 2011CB302400, the NSF of China (nos. 11031007, 11171322, 61222206), One Hundred Talent Project of the Chinese Academy of Sciences, the 111 Project (no. b07033), and the Program for New Century Excellent Talents in University (no. NCET-11-0881).

Authors' addresses: R. Wang, Z. Yang (corresponding author), L. Liu, J. Deng, and F. Chen, School of Mathematical Science, University of Science and Technology of China, Hefei, Anhui, China; email: yangzw@ustc.edu.cn.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2014 ACM 0730-0301/2014/03-ART18 \$15.00

DOI: <http://dx.doi.org/10.1145/2557449>

ACM Reference Format:

Ruimin Wang, Zhouwang Yang, Ligang Liu, Jiansong Deng, and Falai Chen. 2014. Decoupling noise and features via weighted ℓ_1 -analysis compressed sensing. *ACM Trans. Graph.* 33, 2. Article 18 (March 2014) 12 pages. DOI: <http://dx.doi.org/10.1145/2557449>

1. INTRODUCTION

With digital scanning devices becoming widespread, more and more acquired raw data of the sampled 3D models is available. Even with high-resolution scanners, the raw data contains inevitable noise from various sources. Although a large number of mesh denoising schemes already exist [Taubin 1995; Fleishman et al. 2003; Jones et al. 2003; Zheng et al. 2010], removing noise while preserving sharp features still remains a challenge.

The reasons are threefold. First, sharp features and noise are ambiguous since there are no precise mathematical definitions for them. Features are often hard to distinguish from large amounts of noise even for human beings (see Figures 10 and 6). Second, traditional approaches adopt local filtering operators which rely on various differential properties, such as normal (or tangent plane) and curvature, to average nearby points to “remove” the noise. On one hand, computation of the differential properties is a “chicken and egg” problem, since the definition of geometric differential assumes local smoothness, and its computation is not reliable and robust in the presence of noise. On the other hand, the filtering operators would blur sharp features and shrink the object. Third, most of the existing denoising methods require fine tuning of various parameters in order to produce the best results for different inputs, thus making it difficult for users.

In this work, we present an approach for decoupling noise and sharp features of 3D shapes. Suppose vertices of the input mesh are sampled from an underlying surface with additive independent and identically distributed (i.i.d.) random noise over the surface. Our idea is to estimate a base mesh to approximate the true underlying surface and then recover the features from the residual between base mesh and input mesh.

First, we propose a Discrete Laplacian Regularization Smoothing (DLRS) model to estimate the base mesh. The objective function consists of a data term which measures how far the smoothed points are from the original data, and a smoothness term which uses the discrete Laplacian of the points to measure the smoothness of the resulted points. Unlike previous approaches that heuristically select the smoothness parameter to balance the data term and the smoothness term, we present an automatic scheme, based on the Generalized Cross-Validation (GCV) scheme, to compute the optimal value of this parameter according to the input data.

If the true underlying surface is C^2 -smooth, we prove that the computed parameter is *asymptotically optimal*. That is, with this optimal parameter, the estimated base surface is guaranteed to

asymptotically converge to the true surface with probability one when the sample size goes infinity.

In general cases, an underlying surface is a piecewise C^2 -smooth surface with sharp features. As we mentioned earlier, it is difficult to identify these features in the presence of noise. We find that features are regarded as large-scale noise in the denoising phase and the large errors produced will be uniformly distributed to other regions in the base mesh. That is, the presence of sharp features may result in generating artifacts in regions of C^2 during the denoising phase.

We thus propose an approach for identifying and recovering features from the residual between the estimated base mesh and the input mesh. This is achieved by two observations. First, the residual inevitably mixes the information of features and the noise. We discover that the pseudo-inverse matrix of the Laplacian matrix of the mesh is a coherent dictionary for sparsely representing the feature signal on the shape. Second, we are highly inspired by the promising development of the technique of *compressed sensing* in recent years [Candès and Tao 2005; Donoho 2006; Eldar and Kutyniok 2012].

Therefore, we formulate the identification of sharp features as an ℓ_1 -analysis compressed sensing optimization problem. We propose an iterative procedure to recover the features progressively. In each iteration, we identify a part of features. After marking the identified features, we find more features in the next iteration. This is reasonable as the feature becomes sparser in the latter iterations as the identified features in previous iterations have already been marked and do not contribute to the feature detection.

We demonstrate the performance of our method on a number of diverse inputs, with either synthetic or real noise, and demonstrate its ability to denoise the surfaces and discover their features. We also compare with previous methods and the results show that our approach achieves much better results than the state-of-the-art methods.

Contributions. Our approach for decoupling noise and features is quite different from previous methods. To the best of our knowledge, this is the first time noise and features are analyzed and separated in such an elegant manner with guarantees by statistical theory. The contributions of our approach are summarized in the following.

—*Asymptotically optimal surface denoising.* Our denoising approach is fully automatic without tuning any parameter. The smoothness parameter is automatically computed. The denoised surface is guaranteed to asymptotically converge to the true underlying surface with probability one if the underlying surface is C^2 -smooth.

—*Faithful feature recovering by ℓ_1 -analysis.* We successfully apply the ℓ_1 -analysis compressed sensing technique to identify and recover sharp features from the residual between the estimated base surface and the input surface. This is based on our discovery that the pseudo-inverse matrix of the Laplacian matrix of the mesh is a coherent dictionary for sparsely representing the sharp feature signal on the shape.

2. RELATED WORK

2.1 Feature-Preserving Surface Denoising

A wide variety of surface denoising/smoothing algorithms have been proposed during the past two decades [Botsch et al. 2010]. A thorough review on this topic is out of the scope of this article.

The most common techniques are based on Laplacian operators. Taubin [1995] developed a fast and simple iterative local smoothing scheme based on the definition of the Laplacian operator on

meshes. This approach was extended to irregular meshes using a geometric flow analogy by Desbrun et al. [1999]. Other methods extended feature-preserving anisotropic diffusion in image processing to anisotropic geometric diffusion on surfaces [Clarenz et al. 2000; Bajaj and Xu 2003].

Fleishman et al. [2003] and Jones et al. [2003] extended the bilateral filter from image denoising [Tomasi and Manduchi 1998] to mesh denoising, which anisotropically averages the nearby vertices weighted by a monotonously decreasing function in terms of both spatial difference and vertex difference. Dugué et al. [2004] proposed a higher-order bilateral filter for mesh denoising. The bilateral filter was also applied to the facet normal field defined over the mesh by Zheng et al. [2010]. The normal field was first filtered and then the denoised surface was reconstructed from the filtered normal field. This scheme obtains better results than previous normal-based filtering methods [Yagou et al. 2002; Sun et al. 2007]. Su et al. [2009] adopted a mean filter to smooth the vertex Laplacians and then reconstructed the geometry from the filtered Laplacian. Fan et al. [2010] applied an anisotropic filter and second-order bilateral filter to smooth the normal field as well as the curvature tensor field, which can better preserve curvature details and alleviate volume shrinkage during denoising. Bian and Tong [2011] designed filter weights via the classification of vertices and presented a two-step denoising method that consists of normal vector filtering and vertex position updating.

A few global and noniterative mesh smoothing approaches have been developed during the last few years. Nealen et al. [2006] and Liu et al. [2007] proposed similar global smoothing schemes by setting the vertex Laplacians to zero and reconstructing the surface with geometric constraints. Hildebrandt and Polthier [2007] modeled mesh fairing as an optimization problem where a fairness measure is minimized subject to constraints that control the spatial deviation of the surface. He and Schaefer [2013] built a discrete gradient operator on arbitrary triangle meshes and extended the image ℓ_0 -smoothing method [Xu et al. 2011] to denoise triangulated meshes of 3D models.

It should be emphasized that most previous methods require the user to carefully tune the model parameters case by case and rarely have theoretical guarantees. In contrast, the results obtained from our approach are guaranteed to asymptotically converge to the true underlying surfaces by statistical theory.

2.2 Compressed Sensing (CS)

In recent years, Compressed Sensing (CS) has attracted considerable attention in areas of applied mathematics, computer science, and signal processing [Candès and Tao 2005; Donoho 2006; Candès and Wakin 2008; Candès et al. 2010; Eldar and Kutyniok 2012]. The central insight of CS is the sparsity, that is, signals are sparse with respect to some suitable basis or dictionary, such that signals can be recovered from very few measurements (undersampled data) by a convex optimization. Designing measurement/sensing matrices with favorable properties and constructing suitable dictionaries for sparse representations are the important research topics in CS. We will see in Section 5 that the pseudo-inverse matrix of Laplacian of a mesh is a coherent dictionary for representing the C^0 -signal on the mesh and its sharp features can be sparsely represented in this basis. Then we cast the feature recovery problem into an ℓ_1 -analysis CS optimization. Using our ℓ_1 -analysis approach on the residual between the denoised mesh and the original mesh, we can faithfully identify the locations of sharp features. Then the mesh can eventually be denoised while preserving sharp features by a modified Laplacian optimization.



Fig. 1. Main phases of our approach: (a) the input noisy mesh model; (b) Phase I: denoising the model using DLRS (Section 4); (c) Phase II: recovering sharp features progressively based on ℓ_1 -analysis (Section 5); (d) the final denoised result with recovered features.

The general idea of ℓ_1 -regularization for the purpose of sparse signal reconstruction has been used in the community of geometry processing [Avron et al. 2010; Habbecke and Kobbelt 2012] recently. Instead of applying the ℓ_1 -regularization in the optimization, our approach adopts the ℓ_1 -analysis CS framework to recover sparse features on 3D shapes.

3. OVERVIEW

Different from previous methods that simultaneously denoise the input data and preserve the features, our approach decouples noise and features in two phases. In the first phase, we obtain an estimated base mesh by denoising the input mesh. In the second phase, we recover sharp features from the residual between base mesh and input mesh. Figure 1 shows the pipeline of our approach, in which each iteration consists of: (I) one step of the discrete Laplacian regularization smoothing model followed by (II) one step of feature detection and Laplacian matrix modification.

Phase I: Mesh denoising. A global Laplacian regularization denoising scheme is developed to denoise the input mesh. The denoised mesh is considered as an estimated base mesh for an approximation to the true underlying surface (see Figure 1(b)). We use the Generalized Cross-Validation (GCV) method to automatically choose an optimal parameter which is used to balance the data term and the smoothness term in the objective function. We have proved that by using this parameter the resulting denoised mesh is asymptotically optimal, which means it approximates the true underlying surface with probability one as the sample size goes to infinity if the underlying surface is C^2 -smooth without any sharp features. See Section 4 for the detail.

Phase II: Feature recovering. However, for non- C^2 -smooth surfaces with sharp features, the residual between base mesh and input mesh inevitably mixes the information from features and noise. Then we perform the second phase to recover features from the residual. We discover that the pseudo-inverse matrix of the Laplacian of a mesh is a coherent dictionary for sparsely representing sharp feature signals on the shape. Also we are profoundly inspired by the emerging technique of compressed sensing in recent years. Therefore, we formulate the identification and recovery of sharp features as an ℓ_1 -analysis compressed sensing optimization on the residual. To handle shapes with many features, we employ an iterative process to recover the features (see Figure 1(c)). The detail is elaborated in Section 5.

4. MESH DENOISING WITH ASYMPTOTIC OPTIMALITY

Given a noisy mesh with sharp features, our first step is to seek a base mesh which approximates the original noise-free mesh as much

as possible. To this end, we propose a Discrete Laplacian Regularization Smoothing (DLRS) model for estimating the base surface. We also prove the theoretical properties of our DLRS model, such as convergence rate and asymptotic optimality.

4.1 Denoising with DLRS Model

Problem formulation. The problem of surface denoising can be stated as follows: Assume that we are given a mesh $P = \{\mathbf{p}_i\}_{i=1}^n$, where \mathbf{p}_i are sampled possibly with noise from a C^2 -smooth surface \tilde{S} , that is,

$$\mathbf{p}_i = \mathbf{s}_i + \varepsilon_i \mathbf{n}_i, \quad i = 1, \dots, n, \quad (1)$$

where \mathbf{n}_i is the unit normal of surface at $\mathbf{s}_i \in \tilde{S}$ and ε_i represents noise. The noise $\varepsilon_i, i = 1, \dots, n$, are assumed to be independent and identically distributed (i.i.d.) random variables with zero mean and finite variance σ^2 . The goal of denoising is to produce a smooth mesh surface \hat{S} to approximate the true underlying surface \tilde{S} as much as possible.

DLRS model. Our denoising model shares much similarity with other global Laplacian optimization approaches [Nealen et al. 2006; Liu et al. 2007]. To make the article self-contained and define the notations for the rest of the work, we derive our DLRS denoising model in this section. To find a C^2 -smooth surface \hat{S} which is a reasonable estimate of the true surface \tilde{S} , we formulate it as the following variational minimization problem

$$\hat{S} = \arg \min_S \frac{1}{n} \sum_{i=1}^n d^2(\mathbf{p}_i, S) + \lambda \int_S (2\mathcal{H})^2, \quad (2)$$

where $d(\mathbf{p}_i, S)$ is the geometric distance from \mathbf{p}_i to S , \mathcal{H} is the mean curvature of S , and λ is a smoothness parameter. The objective function consists of two terms: the data term which measures how well the surface approximates the points, and the smoothness (regularization) term which measures how smooth the surface. The smoothness parameter λ plays the role of balancing the two terms.

To simplify the computation, we replace the smoothness term by its discrete approximation

$$\begin{aligned} \mathcal{J}(S) &= \frac{1}{n} \sum_{i=1}^n 4\mathcal{H}^2(\mathbf{s}_i) \\ &= \frac{1}{n} \sum_{i=1}^n \|\Delta_S \mathbf{s}_i\|^2 \simeq \frac{1}{n} \sum_{i=1}^n \|L_i S\|^2, \end{aligned} \quad (3)$$

where $S = (\mathbf{s}_1, \dots, \mathbf{s}_n)^T \in \mathbb{R}^{n \times 3}$, Δ_S is the Laplace-Beltrami operator on the surface, and $L = (L_1^T, \dots, L_n^T)^T$ is the discrete Laplacian matrix. The second equality in Eq. (3) is derived by $\Delta_S \mathbf{s}_i = 2\mathcal{H} \mathbf{n}_i$.

Table I. The Performance Statistics of Our Algorithm

#Vertices (Model)	Eigen computation and comparison						Choosing λ / Linear system/ ℓ_1 -analysis	
	Spectrum extrapolation			Full eigenvalues				
	Time	$\text{tr}(M)$	λ_G	Time	$\text{tr}(M)$	λ_G		
2.5k (8-like in Figure 7)	4s	2.19×10^4	5.67	12s	2.25×10^4	5.56	0.2s/0.03s	1s
5k (upper model in Figure 12)	8s	8.76×10^4	10.41	76s	8.98×10^4	10.13	0.4s/0.07s	2s
7k (fandisk in Figure 1)	21s	3.22×10^5	3.78	150s	3.36×10^5	3.65	0.6s/0.10s	4s
9k (cube in Figure 6)	43s	4.20×10^5	102.23	560s	4.01×10^5	98.41	0.8s/0.13s	5s
10k (pipe in Figure 9)	48s	4.79×10^5	8.13	730s	5.01×10^5	7.84	0.8s/0.14s	5s
12k (torus in Figure 6)	65s	5.12×10^5	18.32	1059s	5.38×10^5	17.97	1.0s/0.15s	6s
28k (lower model in Figure 12)	202s	1.13×10^6	8.59	10428s	1.19×10^6	8.31	2.1s/0.27s	13s
50k (flower in Figure 10)	854s	1.67×10^6	18.40	–	–	–	3.4s/0.57s	19s

In the implementation, only the q smallest magnitude eigenvalues are computed to extrapolate the remaining ones using the acceleration technique of spectrum extrapolation. We set $q = 2\sqrt{n}$ and obtain a good estimate of λ_G while greatly reducing the computation time complexity from $O(n^3)$ to $O(n^2)$. For a mesh model with $50k$ vertices, there is no output of computing all the eigenvalues within a tolerable time range. The time costs of three remaining components of our algorithm are listed in the last two columns. “Choosing λ ” stands for obtaining the GCV choice according to (7); “Linear system” stands for solving the linear system (5); “ ℓ_1 analysis” stands for solving the weighted ℓ_1 -analysis optimization (14).

Thus we arrive at the following denoising model:

$$\hat{S} = \arg \min_S \sum_{i=1}^n \|\mathbf{p}_i - \mathbf{s}_i\|^2 + \lambda \sum_{i=1}^n \|L_i S\|^2. \quad (4)$$

We call it the Discrete Laplacian Regularization Smoothing (DLRS) model.

Denote $P = (\mathbf{p}_1, \dots, \mathbf{p}_n)^T \in \mathbb{R}^{n \times 3}$. The DLRS model eventually leads to linear systems

$$(I_n + \lambda M)\hat{S} = P, \quad (5)$$

where $M = L^T L = \sum_{i=1}^n L_i^T L_i$. Thus, given a specific value of the smoothness parameter λ , we have the solution

$$\hat{S}_n(\lambda) = (I_n + \lambda M)^{-1} P$$

as the estimated base mesh of the true underlying surface, where I_n is the $n \times n$ identity matrix.

Choosing the optimal value of λ . As we mentioned before, our DLRS model is similar to global Laplacian optimization approaches [Nealen et al. 2006; Liu et al. 2007]. However, previous works select the smoothness parameter λ in a heuristic manner or they allow users to adjust λ to control the smoothness of the denoised results.

We observe that there is no specific value of λ which works for all input data. That is, the parameter λ in Eq. (4) should be chosen by different values for different input data to gain the best denoised results. Inspired by the research on smoothing splines in statistics, we adopt the *Generalized Cross-Validation* (GCV) [Wahba 1990, Chapter 4] to determine the smoothness parameter λ in our DLRS model.

Specifically, the merit function of GCV is defined as

$$\text{GCV}_n(\lambda) = \frac{\frac{1}{n} \|P - \hat{S}_n(\lambda)\|_F^2}{\left(1 - \frac{1}{n} \text{tr}[A_n(\lambda)]\right)^2}, \quad (6)$$

where $A_n(\lambda) = (I_n + \lambda M)^{-1}$. And the optimal value of λ can be computed by minimizing the previous GCV function, that is,

$$\hat{\lambda}_G = \arg \min_{\lambda > 0} \text{GCV}_n(\lambda), \quad (7)$$

which can be easily solved by a line search optimization.

Computation acceleration. The straightforward calculation of $\text{tr}[A_n(\lambda)]$ in Eq. (6) is expensive due to the costly computation of the inverse of $I_n + \lambda M$ for various values of λ . Here we compute

the eigenvalues $\mu_1 \leq \mu_2 \leq \dots \leq \mu_n$ of the matrix M in ascending order. Then $\text{tr}[A_n(\lambda)]$ can be simply calculated as

$$\text{tr}[A_n(\lambda)] = \sum_{i=1}^n \frac{1}{1 + \lambda \mu_i}. \quad (8)$$

Note that M is a semi-definite sparse matrix. The number of nonzeros of M is strictly less than $(k+1)^2 n$ and empirically $3kn$ when k nearest neighbors are used to build the Laplacian matrix L . Our algorithm benefits greatly from the sparsity of M and the asymptotic behavior of its eigenvalues. We develop a technique, namely spectrum extrapolation, to accelerate the computation in our implementation as follows.

As the eigenvalues of M in ascending order asymptotically satisfy $\mu_j = O(j^2)$ (according to Lemma 1.7 in the supplementary material), we can use partial eigenvalues to extrapolate a function that forecasts the remaining ones. Let $\mu(j) = c_0 + c_1 j + c_2 j^2$ be the quadratic function for depicting the asymptotic behavior of eigenvalues wherein c_0, c_1, c_2 are unknown coefficients to be determined. Specifically, we only compute the q smallest (or largest) magnitude eigenvalues $\{\mu_{j_k}\}_{k=1}^q$ using the ARPACK library [Lehoucq et al. 1998] within $O(nq^2)$ complexity. We obtain the coefficients by least-squares fitting the truncated eigenvalues and predict the remaining eigenvalues $\{\mu_{j_k}\}_{k=q+1}^n$ by the resulting $\mu(j)$. Then $\text{tr}[A_n(\lambda)]$ can be calculated according to Eq. (8). We set $q = 2\sqrt{n}$ in our implementation. The error between estimated and true $\text{tr}[A_n(\lambda)]$ is around 5% from our experiments, as shown in Table I. Note that the ARPACK package is based upon a variant of the Lanczos iteration called Implicitly Restarted Lanczos Method (IRLM) [Sorensen 1997] which is stable in computing either the smallest or largest magnitude eigenvalues.

Choice of Laplacian. The discrete Laplacian matrix L of a mesh could be defined in various ways. The geometric Laplacian approximates the exact Laplacian better than the graph Laplacian. But for noisy data, the graph Laplacian that contains only information of topological connections is more reliable than the geometric Laplacian. We employ the graph Laplacian since it is robust to different levels of noise and we have shown its good performance in our experiments. Undoubtedly, other modifications of geometric Laplacian can also be optional in our algorithm.

Pseudocode. The pseudocode of computing the estimated base surface can be seen in Algorithm 1.

ALGORITHM 1: Computing the estimated base mesh \hat{S} by our DLRS model.

Input: the set of points P and its Laplacian matrix L

Output: $\hat{S} = \text{DLRS}(P, L)$

- 1: Calculate $M = L^T L$ and its eigenvalues μ_1, \dots, μ_n .
 - 2: Compute the optimal smoothness parameter by minimizing the GCV function in (6), i.e., $\hat{\lambda}_G = \arg \min_{\lambda > 0} \text{GCV}_n(\lambda)$.
 - 3: Obtain \hat{S} by solving linear systems $(I_n + \hat{\lambda}_G M)\hat{S} = P$.
-

4.2 Asymptotic Properties

We have established the theoretical properties (convergence rate and asymptotic optimality) of our DLRS model under some regularity conditions. All the technical proofs can be found in the supplementary material. Specifically, we have the following two main theorems.

Denote the error between the estimated base mesh surface and the true underlying surface as

$$r_n(\lambda) = \frac{1}{n} \|\hat{S}_n(\lambda) - \bar{S}\|_F^2.$$

THEOREM 1. Assume that P is the equidistributed sample of a C^2 -smooth surface. As $n \rightarrow \infty$ and $\lambda \sim n^{-2/3}$ is chosen, we have with probability one

$$\mathbf{E}[r_n(\lambda)] = O(n^{-\frac{2}{3}}).$$

THEOREM 2. If the smoothness parameter $\hat{\lambda}_G$ is the GCV choice according to Eq. (7), then the estimated base mesh surface $\hat{S}_n(\hat{\lambda}_G)$ from our DLRS model is asymptotically optimal, that is,

$$\frac{r_n(\hat{\lambda}_G)}{\inf_{\lambda \in \mathbb{R}_+} r_n(\lambda)} \rightarrow_p 1, \quad (9)$$

where \rightarrow_p means the convergence in probability.

Theorem 1 and Theorem 2 give the theoretical guarantees by statistical theory for our denoising scheme. It is seen that the estimated base mesh \hat{S} asymptotically converges to the ground-truth surface with probability one as the sample size goes to infinity. In particular, Theorem 1 ensures the existence of smoothness parameter λ to achieve the convergence rate of $n^{-\frac{2}{3}}$, and Theorem 2 provides a numerical means for choosing an asymptotically optimal parameter.

Remark. The DLRS model is similar to an implicit integration step of the biharmonic flow, which corresponds to a geometric evolution equation. Usually, it is hard to decide the step size and terminal condition for the evolution equations. We have proved that the DLRS model guarantees the asymptotically optimal convergence by using statistical theory. To the best of our knowledge, the property of asymptotic optimality cannot be obtained using the biharmonic flow formulation for smoothing noisy data.

5. FEATURE RECOVERING VIA ℓ_1 -ANALYSIS COMPRESSED SENSING

If the true underlying surface is C^2 -smooth, the denoised mesh \hat{S} obtained by our model DLRS with the parameter $\hat{\lambda}_G$ is guaranteed to be an asymptotically optimal approximation (Theorem 2). However, the true surface might not be a C^2 -smooth surface as a whole but a piecewise C^2 surface with C^0 sharp features.

The presence of shape features may result in generating artifacts in C^2 regions during the denoising phase (see the small bumps

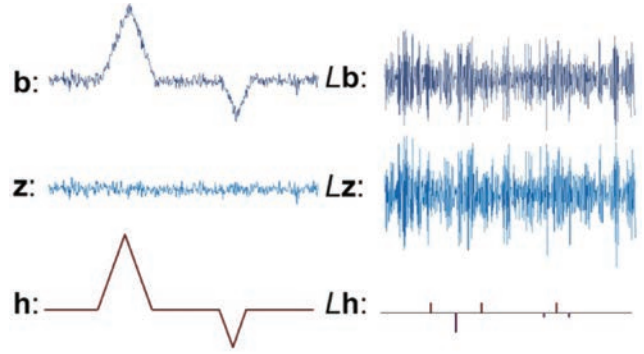


Fig. 2. 1D illustration of signals and their Laplacians. Left: the residual \mathbf{b} (upper) is a mixture of noise \mathbf{z} (middle) and features \mathbf{h} (lower). Right: the corresponding Laplacians of signals on the left.

in the denoised meshes shown in Figures 1(b) and 9(c)). This is because features are regarded as large magnitude noise and the errors produced will be slightly diffused to other regions in \hat{S} by the global optimization (4). But compared to the magnitude of these small bumps, the errors located at sharp features are dominant. In this section we propose an ℓ_1 -analysis compressed sensing-based approach for recovering the features from the residual between the base mesh \hat{S} and the given noisy mesh P .

5.1 ℓ_1 -Analysis Compressed Sensing on Residual

Residual. By the denoising phase we now have the estimated base mesh $\hat{S} = \{\hat{\mathbf{s}}_i\}_{i=1}^n$ from the input noisy mesh $P = \{\mathbf{p}_i\}_{i=1}^n$. The residual between \hat{S} and P is defined as

$$b_i = (\mathbf{p}_i - \hat{\mathbf{s}}_i)^T \hat{\mathbf{n}}_i, \quad i = 1, \dots, n, \quad (10)$$

where $\hat{\mathbf{n}}_i$ is the unit normal vector of surface \hat{S} at $\hat{\mathbf{s}}_i$. Denote $\mathbf{b} = (b_1, \dots, b_n)^T$ as the residual vector.

As an inference of asymptotic properties presented in Section 4, the residual signal \mathbf{b} is essentially i.i.d. noise if the true underlying surface of input mesh P is at least C^2 -continuous as a whole. When the underlying surface contains sharp features, the residual \mathbf{b} inevitably mixes the information from features and the noise, as illustrated in Figure 2 (left). The residual can be decomposed as

$$\mathbf{b} = \mathbf{h} + \mathbf{z}, \quad (11)$$

where $\mathbf{h} = (h_1, \dots, h_n)^T$ is the unknown signal of shape features and $\mathbf{z} = (z_1, \dots, z_n)^T$ is the measurement error.

ℓ_1 -Analysis compressed sensing. The compressed sensing theory asserts that if the unknown signal is reasonably sparse, it is possible to recover it under suitable conditions on a sensing matrix by an ℓ_1 -norm convex programming [Candès and Tao 2005; Donoho 2006]. The techniques hold for signals which are sparse in the standard coordinate basis or sparse with respect to some orthogonal basis. However, there are numerous applications in which a signal of interest is usually not sparse in an orthogonal basis but in a *coherent dictionary* (see detail in Candès et al. [2010]).

Coherent dictionary for shape features. Considering the feature signal \mathbf{h} in our case, we discover that shape features can be represented as sparse in some coherent dictionary which is constructed by the pseudo-inverse matrix L^+ of the Laplacian matrix L of the shape. The key observation is that Lh (Laplacian of feature signal) has quickly decaying coefficients and is indeed sparse, as illustrated in Figure 2 (right). Furthermore, the matrix L^+ has n (number of

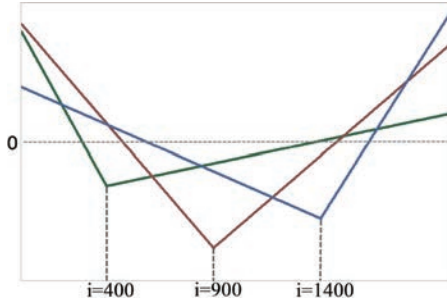


Fig. 3. Three basis functions corresponding to the 400-th, 900-th, and 1400-th columns of L^+ respectively. We use the 1D case as an illustration.

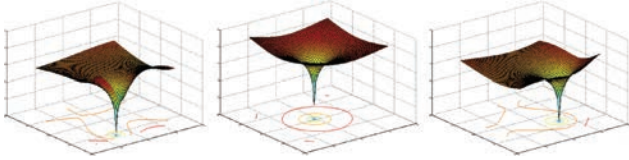


Fig. 4. The discrete Greens functions corresponding to three columns of L^+ in the 2D case. This evidence suggests that L^+ is a coherent dictionary for representing the C^0 feature signals on surfaces.

vertices) columns and each column (say, the i -th column) of L^+ can be regarded as a discrete Greens function which is C^0 at v_i and smooth over all other vertices. Figure 3 and Figure 4 illustrate the basis functions corresponding to columns of L^+ in the 1D case and the 2D case, respectively. This evidence suggests that the columns of L^+ form a basis (a coherent dictionary) for representing the C^0 feature signals on the shape.

ℓ_1 -Analysis on residual. As we have found a coherent dictionary $D = L^+$ for representing \mathbf{h} sparsely, that is, $L\mathbf{h}$ is sparse, we can thus formulate the problem of recovering feature signals \mathbf{h} from the residual \mathbf{b} as an ℓ_1 -analysis compressed sensing optimization

In our setting, the sensing matrix is the identity matrix. Thus we have the following ℓ_1 -analysis compressed sensing optimization

$$\min_{\mathbf{h}} \|L\mathbf{h}\|_1 \quad \text{s.t.} \quad \|\mathbf{h} - \mathbf{b}\|_2^2 \leq \epsilon^2, \quad (12)$$

where ϵ^2 is a likely upper bound on the noise power $\|\mathbf{z}\|_2^2$. The roles of the penalty and the constraint in Eq. (12) might also be reversed if we choose to constrain the sparsity and obtain the best fit for that sparsity. An equivalent optimization is given as

$$\min_{\mathbf{h}} \|\mathbf{h} - \mathbf{b}\|_2^2 \quad \text{s.t.} \quad \|L\mathbf{h}\|_1 \leq \tau, \quad (13)$$

where τ is a tuning parameter controlling the sparsity. We prefer to solve Eq. (13) instead of (12), since the sparsity parameter τ holds more intuitive geometric meaning and can be actually controlled (see Figure 8). Denote $\hat{\mathbf{h}}$ as the solution to the optimization (13). We assert that $L\hat{\mathbf{h}}$ provides accurate and reliable locations of sharp features with τ smaller than exact $\|L\mathbf{h}\|_1$. Otherwise, if $L\hat{\mathbf{h}}$ incorrectly locates the features, the optimal solution $\hat{\mathbf{h}}$ might bias the original feature signal in large error that implies a contradiction.

Weighted ℓ_1 -analysis. Consider the weighted ℓ_1 -analysis on the residual

$$\begin{aligned} \min_{\mathbf{h}} \quad & \|\mathbf{h} - \mathbf{b}\|_2^2 \\ \text{s.t.} \quad & \|W(L\mathbf{h})\|_1 = \sum_{i=1}^n w_i |L_i \mathbf{h}| \leq \tau, \end{aligned} \quad (14)$$

where $W = \text{diag}(w_1, \dots, w_n)$ and w_1, \dots, w_n are positive weights. The weighted ℓ_1 -analysis optimization (14) can be regarded as a relaxation of an ℓ_0 -minimization problem. It is desired that the weights could be to counteract the influence of the signal magnitude on the ℓ_1 -penalty function. Ideally, the weights are inversely proportional to the true signal magnitude, that is,

$$w_i = \begin{cases} \frac{1}{|L_i \mathbf{h}|} & L_i \mathbf{h} \neq 0, \\ \infty & L_i \mathbf{h} = 0. \end{cases} \quad (15)$$

The large entries in W force the Laplacian of feature $L\mathbf{h}$ to concentrate on the indices where w_i is small. These constructed weights precisely correspond to the indices where $L\mathbf{h}$ is nonzero. It is of course impossible to construct the precise weights (15) without knowing the feature signal \mathbf{h} itself, but this suggests more generally that large weights could be used to discourage nonzero entries in the recovered $L\mathbf{h}$, while small weights could be used to encourage nonzero entries.

An iterative algorithm of reweighted ℓ_1 -minimization is proposed by Candès et al. [2008] to enhance the sparsity in signal recovery. There exists such a possibility of constructing a favorable set of weights based on an approximation of $L\mathbf{h}$ or on other information about the vector magnitudes. Based on the geometric information of the estimated base surface \hat{S} , we design the weights as

$$w_i = \frac{1}{\rho + \|L_i \hat{S}\|}, \quad i = 1, \dots, n,$$

where ρ is a small number ($\rho = 10^{-7}$ by default) that provides numerical stability and should be set slightly smaller than the expected nonzero magnitudes of $L\mathbf{h}$. With these well-designed weights, we then perform a weighted ℓ_1 -analysis (14) on the residual for recovering the feature signal. We employ the interior-point method [Boyd and Vandenberghe 2004] to solve this convex optimization problem (14) in our implementation.

5.2 Iterative Feature Recovering

If we set large value of the sparsity parameter τ , the optimization (14) will recover features but may introduce some nonfeature points in the result. Thus we prefer to choose a small value of τ . For some models with large portion of features, the solution to Eq. (14) returns only the most prominent (sharpest) features.

To recover the other features, our idea is to modify the rows (of Laplacian matrix L) corresponding to the identified features and perform the weighted ℓ_1 -analysis optimization (14) in an iterative manner. In each iteration, the most salient features in the current residual are identified. After marking the identified features, we find more features in the next iteration. This is reasonable as features become sparser in following iterations as the identified features in previous iterations have already been identified and do not contribute to the residual \mathbf{b} in feature recovering model (14).

Feature classification. Generally there are two types of sharp features, namely corners and creases, as shown in Figure 5, on 3D shapes. A corner point is the one at which the tangent of any passing curve on the surface is discontinuous. A crease curve introduces the discontinuities of first derivatives across it, but preserves C^2 -continuity along it. Corners can also be the intersections of several creases.

After identifying locations of the features by the weighted ℓ_1 -analysis (14) on the residual, we adopt a simple scheme to classify their types. If a feature point has a few feature points in its neighbor, we compute a dominant direction by PCA and classify it as a crease along this direction. If a feature is isolated from others or the PCA direction is degenerated, it is identified as a corner (see Figure 5).

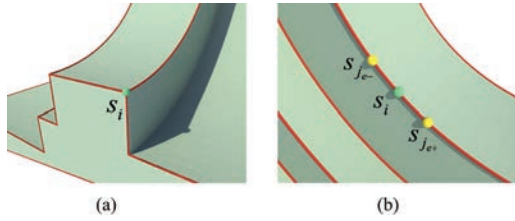


Fig. 5. Two types of feature points: (a) corner; (b) crease.

Feature-aware modification of Laplacian matrix. If the vertex \mathbf{s}_i is identified as a corner, we remove the Laplacian penalty $\mathcal{L}(\mathbf{s}_i) = L_i S$ by setting the i -th row of the Laplacian matrix $L_i = 0$. If the vertex \mathbf{s}_i is identified as a point on a crease \mathcal{E} , we only remove the straddling smoothness penalties and yield a term of the form

$$\mathcal{L}(\mathbf{s}_i) = L_i S = \sum_{j \in \mathcal{N}(i) \cap \mathcal{E}} (\mathbf{s}_j - \mathbf{s}_i) = \mathbf{s}_{j_{e-}} + \mathbf{s}_{j_{e+}} - 2\mathbf{s}_i,$$

where $\mathbf{s}_{j_{e-}}$ and $\mathbf{s}_{j_{e+}}$ are the adjacent neighbors of \mathbf{s}_i along the crease \mathcal{E} , as illustrated in Figure 5(b).

Termination criterion of iterations. We adopt a statistical method of nonparametric test for checking whether the residual \mathbf{b} contains more features. Specifically, residuals were randomly divided into two sets \mathbf{b}_1 and \mathbf{b}_2 . We use a two-sample Kolmogorov-Smirnov test [Massey 1951] to compare the distributions of the values in the two sets \mathbf{b}_1 and \mathbf{b}_2 . We state that:

- the null hypothesis H_0 that \mathbf{b}_1 and \mathbf{b}_2 are from the same continuous distribution;
- the alternative hypothesis H_1 that they are from different continuous distributions.

This hypothesis does not specify what the common distribution is (e.g., normal or not normal). In statistics, a result is called statistically significant if it is unlikely to have occurred by chance alone, according to a predetermined threshold probability, the significance level. The result is 1 if the test rejects the null hypothesis at the α significance level; 0 otherwise. We use the significance level $\alpha = 0.05$ in our implementation.

Pseudocode. The pseudocode of the iterative feature recovering via ℓ_1 -analysis can be found in Algorithm 2.

ALGORITHM 2: Iterated procedure for feature recovering

Input: the points P , its Laplacian L , and the sparsity parameter τ

Output: denoised mesh with features

- 1: Call Algorithm 1, $\hat{S} = \mathbf{DLRS}(P, L)$.
 - 2: Calculate the residual \mathbf{b} according to (10).
 - 3: If the result of the two-sample Kolmogorov-Smirnov test on residual \mathbf{b} is 1, go to Step 4; Otherwise, exit and output the current \hat{S} .
 - 4: Recover the features $\hat{\mathbf{h}}$ by the weighted ℓ_1 -analysis (14), and get the reliable locations of $\hat{\mathbf{h}}$ indicated by $L\hat{\mathbf{h}}$.
 - 5: Classify the features $\hat{\mathbf{h}}$ and modify accordingly the Laplacian matrix L based on their feature types.
 - 6: Go to Step 1.
-

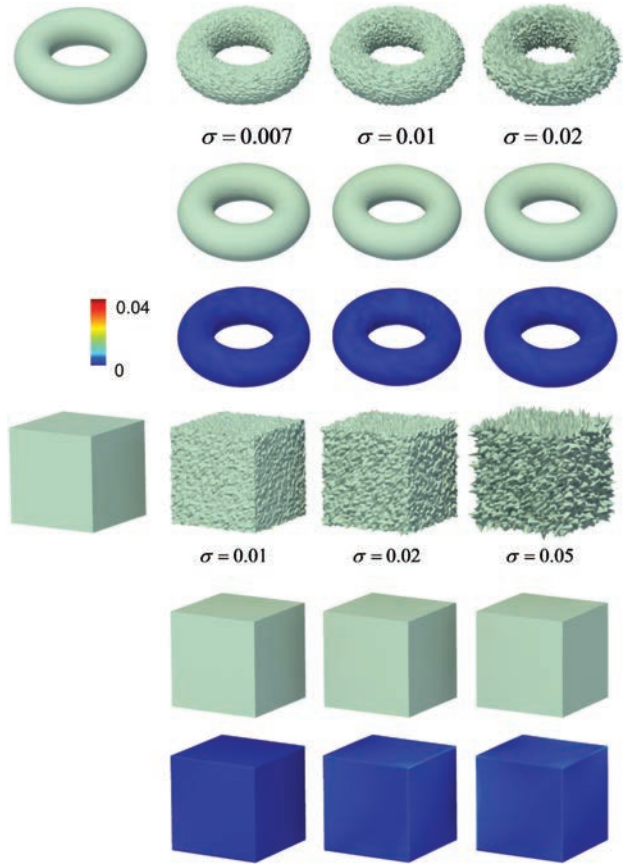


Fig. 6. Our approach is robust to different levels of noise. Upper example: a C^2 -smooth torus shape; lower example: a cube model with sharp features. For each example, the first row shows the ground-truth model followed by the noisy models with different levels of noise (the standard deviation is shown below each model), the second row shows the denoised models, and the third row visualizes the differences between the denoised models and the ground-truth model.

6. EXPERIMENTAL RESULTS

We have implemented our approach and tested it on a large variety of models with different types of features. All the examples presented in this article were made on a dual-core 3 GHz machine with 8G memory (see more results in the accompanying video accessible from the ACM Digital Library).

Like previous works, we mostly use ground-truth models with synthetic noise for evaluating our approach and comparing with other methods. The synthetic noise is generated by an i.i.d. random variable generator with zero mean and a standard deviation σ which is proportional to the diagonal of the bounding box of the model.

Figure 7 shows a denoising example on a model which is discretized from a C^2 -smooth 8-like surface. We added severe synthetic noise ($\sigma = 0.02$) on the model as shown in Figure 7(b). The denoised results with different values of the smoothness parameter λ are shown in Figure 7(c)–(g). It is seen that small λ cannot filter out the noise (see (c) and (d)) while large λ may shrink the object (see (f) and (g)). Thanks to the GCV method, our algorithm can

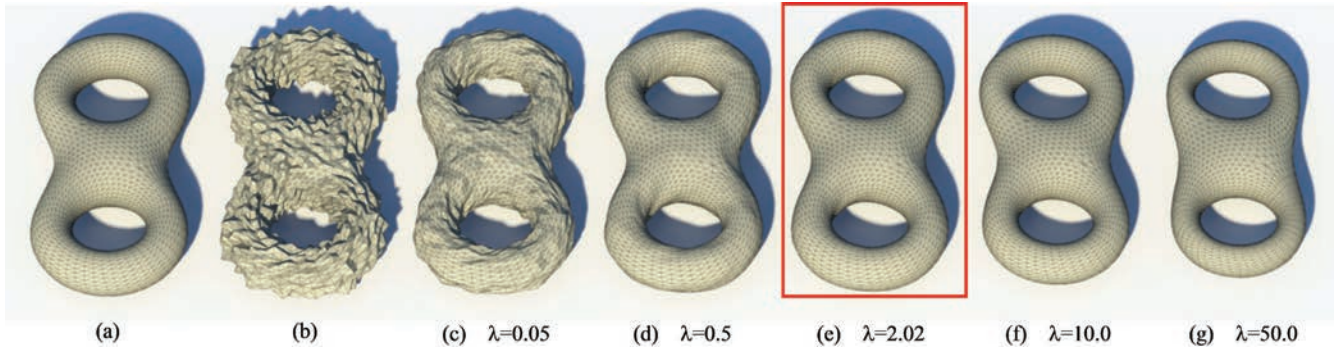


Fig. 7. Our approach computes the optimal smoothness parameter to denoise the mesh. (a) the ground-truth 8-like model which is a C^2 -smooth surface; (b) the model artificially corrupted by severe synthetic noise; (c)–(g) denoised results by the global smoothing approach with various parameters λ in (4). Our approach obtains the best smoothing result (e) via the automatically chosen optimal parameter $\lambda = 2.02$.

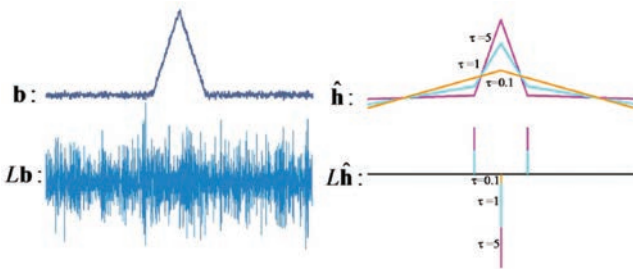


Fig. 8. 1D illustration of results using different sparsity parameters τ in the ℓ_1 -analysis optimization. Left: the residual signal \mathbf{b} (upper) and its Laplacian (lower); Right: the recovered feature signals $\hat{\mathbf{h}}$ corresponding to different values 0.1 (orange), 1 (blue), and 5 (magenta) of τ (upper) and their Laplacians (lower). It is seen that only the prominent feature is identified using small $\tau = 0.1$ and more features are identified using larger $\tau = 1$ and $\tau = 5$. The ℓ_1 -analysis optimization always provides accurate and reliable locations (nonzeros of the Laplacians) of features.

automatically choose an optimal value of $\lambda = 2.02$ and obtain the best denoised result as shown in Figure 7(e).

The torus model in Figure 6 (upper) is also a C^2 -smooth surface. We added different levels of noise on the model and our algorithm obtains good denoised results as shown in Figure 6 (also see the color maps which encode the difference between the denoised models and the ground-truth model).

For C^2 -smooth surfaces, we only need to perform the first phase to obtain the final denoised results. For nonsmooth surfaces with sharp features, we need to perform the second phase to recover the features. Figure 9(a) shows a surface with sharp features. We added some synthetic noise on it shown in Figure 9(b) and applied our smoothing algorithm on (b). The denoised result filters out the noise but also blurs the sharp features as shown in Figure 9(c). The sharp features are progressively recovered using our approach, as shown in Figure 9(d) and (e). All sharp features are correctly recovered in the final result (f).

We also tested our algorithm on the cube model with corners and creases, as shown in Figure 6. It is seen from the results that our approach can faithfully recover features corrupted by different levels of noise, even by heavily added noise. Figure 10 shows a more complex model with many sharp features. Sharp features are correctly recovered and the resulting mesh has slight difference with the ground truth from the color map.

Comparisons. We compare our approach with some state-of-the-art methods, including the anisotropic geometric diffusion [Clarenz et al. 2000], vertex-based bilateral filtering [Fleishman et al. 2003], noniterative smoothing [Jones et al. 2003], anisotropic filtering [Hildebrandt and Polthier 2004], Laplacian mesh optimization [Nealen et al. 2006], and normal-based bilateral filtering [Zheng et al. 2010]. The comparisons are given in Figure 11, with close-up views showing the differences. For all those methods to be compared, we carefully chose the parameters to enable them to produce visually the best denoised results as shown in the figure. In the methods of Clarenz et al. [2000] and Hildebrandt and Polthier [2004], the selection of step size has serious impact on the denoised results and an inappropriate step length will often lead to the self-intersection of resulting mesh. The anisotropic filtering method [Hildebrandt and Polthier 2004] also provides a scheme to preserve the features while denoising, which performs much better if taking the precise location of features as an input. The methods of Clarenz et al. [2000], Fleishman et al. [2003], and Hildebrandt and Polthier [2004] always blur sharp features (see (a1), (a2), (b1), (b2), (d1), (d2) in Figure 11). The method in Nealen et al. [2006] uses a weighted goodness-of-fit measure and the geometric Laplacian to fair the input mesh, while sharp features in the resulting mesh have been largely oversmoothed (see (e1), (e2) in Figure 11). The methods of Jones et al. [2003] and Zheng et al. [2010] preserve the sharp features for input meshes with light noise. However, for heavy, noisy meshes they may introduce bumps in smooth regions and visual artifacts in the results (see (c1), (c2), (f1), (f2) in Figure 11).

Real data. Our algorithm can also be applied for denoising real scanning data which is generally point cloud without topological connections. The idea is to build a graph and then denoise the data using its graph Laplacian. For each point in the cloud, we connect this point and its neighbors within a certain distance and obtain a graph for all the points. The Laplacian matrix can be defined on the graph, thus our algorithm can be applied. Figure 12 shows two examples of denoised results on raw scanning data of real objects, which demonstrates that our algorithm removes the noise in the raw data while preserving the features well.

Parameter. Previous methods need to fine tune various parameters to produce the best results for different inputs. There is only one parameter in our approach, namely the sparsity parameter τ used in the weighted ℓ_1 -analysis (14). We prefer to set a small value of τ that is necessarily smaller than exact $\|W(L\mathbf{h})\|_1$ and perform the iterative process to recover features from the residual sequentially. Using

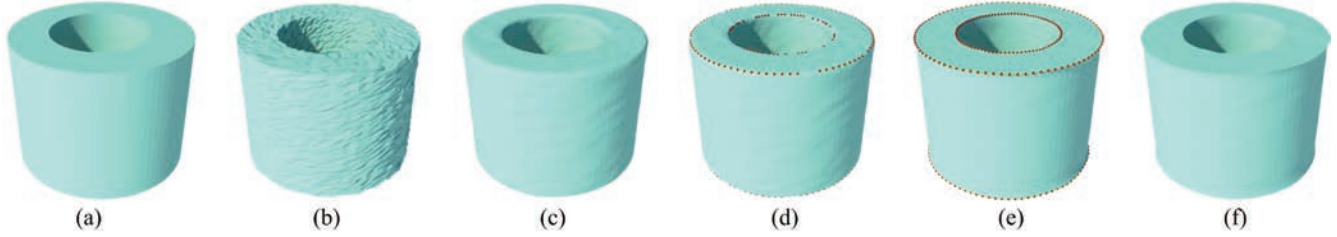


Fig. 9. Our algorithm progressively recovers sharp features. (a) the ground-truth model; (b) the noisy model; (c) the denoised model using our DLRS algorithm; (d)–(e) progressively recovered features shown in red dots with two iterations; (f) the final result.

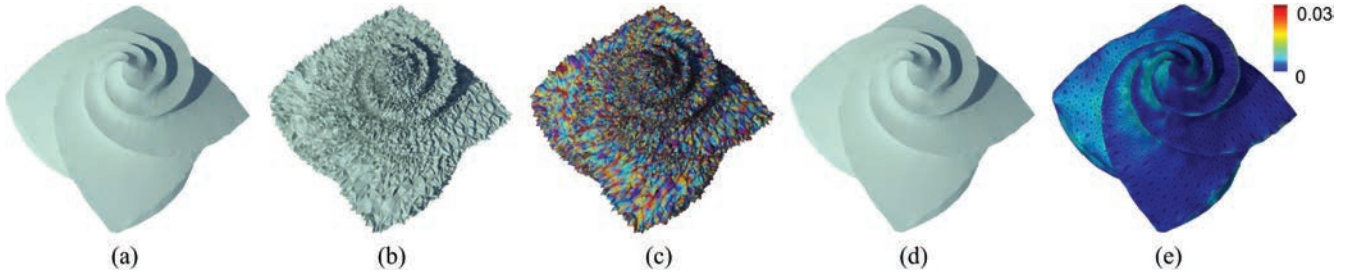


Fig. 10. Our approach is able to faithfully recover sharp features corrupted by noise. (a) The octa-flower model as a ground truth; (b) the model artificially corrupted by independent and identically distributed (i.i.d.) noise (with zero mean and standard deviation $\sigma = 0.01$ of the diagonal of the bounding box of the model); (c) the error map of (b); (d) the denoising result by our approach; (e) the error map of (d). The colored models (c,e) visualize the errors between the processed models and the ground-truth model (a). Note that sharp features such as creases and corners are well retained in our result (d).

an iterative procedure to recover the features on shapes tends to allow for successively better estimation of the nonzero coefficients of Laplacian of features. Even though the iteration may find inaccurate feature estimates with an inappropriate sparsity parameter, the largest coefficients of $L\mathbf{h}$ are most likely to be identified as nonzero. As shown in Figure 8, the solutions $\hat{\mathbf{h}}$ from the weighted ℓ_1 -analysis optimization (14) with different $\tau = 0.1, 1, 5$ provide accurate and reliable location of features. Once these locations are identified, their influence is eliminated by modifying the corresponding rows in the Laplacian matrix. Then it allows more sensitivity for identifying the remaining features whose Laplacian coefficients are small but nonzero.

The sparsity of features on surface means $O(\sqrt{n})$, where n is the number of vertices, because this is the amount of points on smooth feature lines. In the implementation, we set a default value of $\tau = 0.1 * \sqrt{n}$ which works quite well for most models tested in our experiments. We also allow the user to adjust τ . As τ constrains the sparsity of the features detected from the residual, it is very intuitive for the user to adjust its value. If the input mesh is supposed to have lots of features, the user can set a slightly larger value of τ to recover features faster. However, large τ likely returns fake features in the results.

Timing. Table I shows exhaustively the time cost of our algorithm. The most time-consuming part in our algorithm is the computation of the eigenvalues of the matrix M in the denoising phase. We accelerate the algorithm by developing a technique of spectrum extrapolation which only requires to compute the q smallest magnitude eigenvalues as mentioned in Section 4. The spectrum extrapolation has a significant acceleration compared to full eigenvalues computation (see Table I). The Reverse Cuthill-McKee (RCM) ordering [Cuthill and McKee 1969] is employed as a preprocessing for the full eigenvalues computation based on the QR factorization. For

the spectrum extrapolation, we use the well-known ARPACK library which is based upon the Implicitly Restarted Arnoldi Method (IRAM) to compute the truncated eigenvalues. IRAM reduces to the Implicitly Restarted Lanczos Method (IRLM) when the input matrix is symmetric. These variants can be viewed as synthesis of the Arnoldi/Lanczos process and the Implicitly Shifted QR technique, which ensures they are numerically stable in computing either the smallest or largest magnitude eigenvalues.

We have tested our algorithm on a variety of models in our article with mesh size n from $2k$ to $50k$. The statistics of performance in Table I show that we can obtain a good estimate of λ_G while reducing the computation complexity from $O(n^3)$ to $O(n^2)$ when we set $q = 2\sqrt{n}$ in the implementation. In the last two columns of Table I, we also give the time costs of other components including the GCV choice in Eq. (7), the linear system in (5), and the weighted ℓ_1 -analysis optimization in (14). It is obvious that the computation of eigenvalues is the dominant cost in our algorithm.

Limitations. Our approach relies on the assumption that the noise is i.i.d. random variable, which is correct for most of the noise and assumed in most of the previous papers. However, if the noise is not i.i.d., our DLRS model in the denoising phase may not guarantee to obtain the optimal base mesh, which might affect the result of our ℓ_1 -analysis feature recovering scheme. Fortunately, our experiments show that our algorithm can successfully decouple the noise and features on real scanning data (see Figure 12), which means that our approach is practically useful.

Our approach can be applied to either meshes or point clouds. The advantage of denoising a mesh rather than a point cloud is that the connectivity information implicitly defines the surface topology and serves as a means for fast access to neighboring samples. Thus if we have the correct topology of the underlying surface, our scheme is very robust to noise, even for very heavy noise as shown in

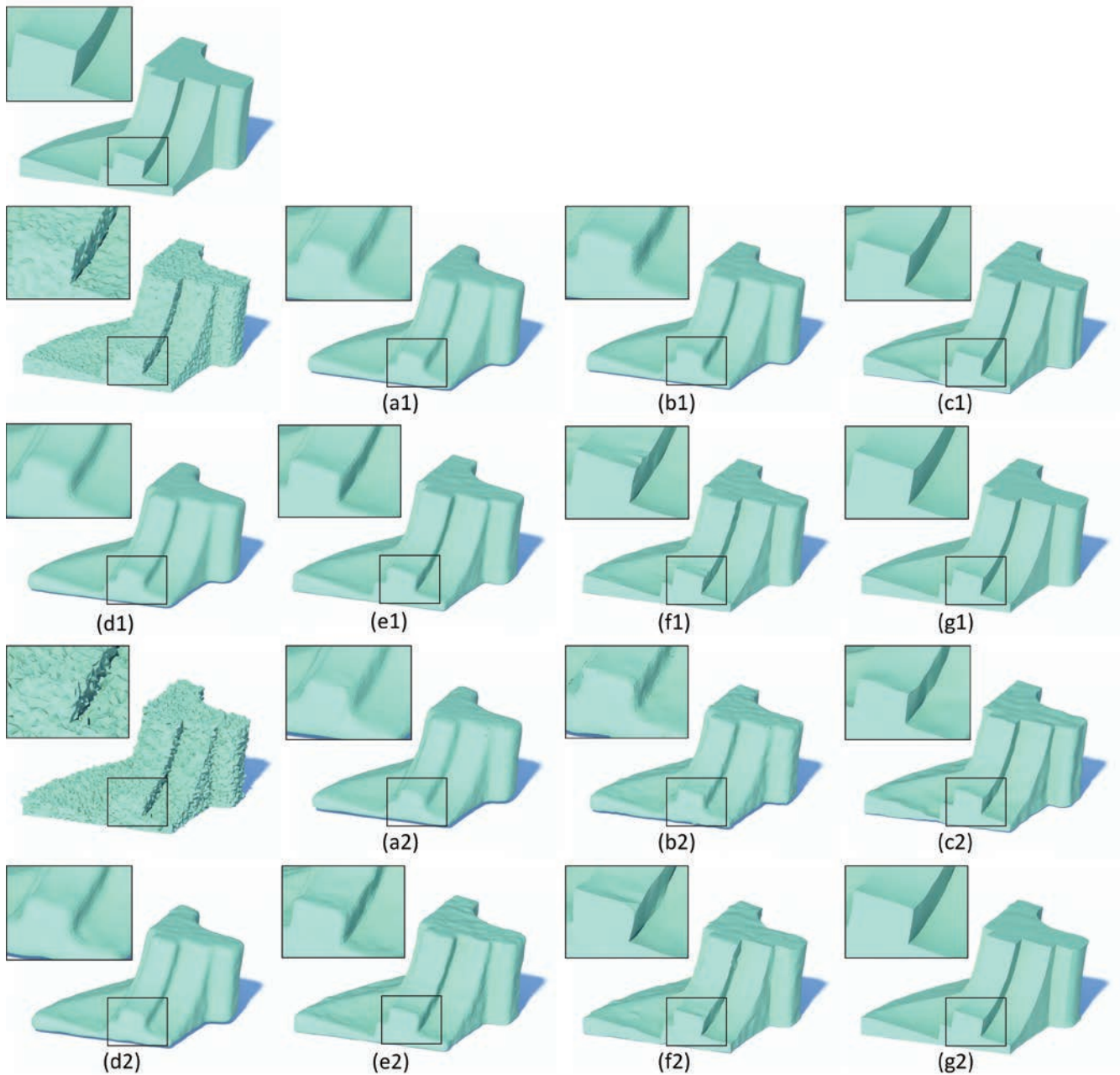


Fig. 11. Comparisons with previous methods. The first column from top to bottom: the ground-truth Fandisk and the models corrupted by different levels of artificial noise ($\sigma = 0.005$ in the upper row and $\sigma = 0.01$ in the lower row). (a1)–(a2) denoised results by the anisotropic geometric diffusion [Clarenz et al. 2000]; (b1)–(b2) by the vertex-based bilateral filtering [Fleishman et al. 2003]; (c1)–(c2) by the noniterative smoothing [Jones et al. 2003]; (d1)–(d2) by the anisotropic filtering [Hildebrandt and Polthier 2004]; (e1)–(e2) by the Laplacian mesh optimization [Nealen et al. 2006]; (f1)–(f2) by the normal-based bilateral filtering [Zheng et al. 2010]; (g1)–(g2) by our approach. The small regions with the frames are magnified to clearly show the differences. All meshes are flat-shaded to show faceting.

Figure 6. For point clouds, it is not easy to get the correct topology information. In our implementation, we adopt the nearest neighbors to build the graph of the points. Thus, if the graph incorrectly reflects the topology, the result might not be reliable. This is a fundamental limitation of any point-based processes.

7. CONCLUSION AND FUTURE WORK

This article presents a two-phase approach for decoupling features and noise on discrete surfaces. The first phase generates a base mesh which is obtained by denoising the input data using a global

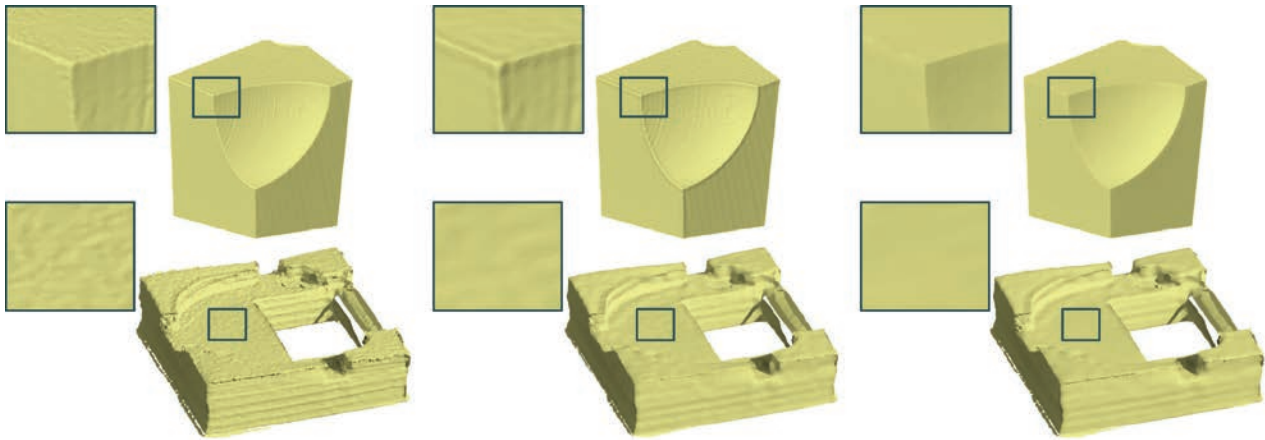


Fig. 12. Applying our approach to scanning data of real objects. From left to right: the input scanning raw data; the denoised results using the method of Jones et al. [2003]; the denoised results using our approach. The close-up views show the details in framed regions.

Laplacian regularization smoothing optimization. The smoothness parameter is automatically chosen by adopting the generalized cross-validation scheme and is proved asymptotically optimal. The second phase extracts sharp features from the residual between base mesh and input mesh based on an ℓ_1 -analysis compressed sensing optimization. This is achieved by our insight that the pseudo-inverse of Laplacian matrix of a mesh is a coherent dictionary for representing the C^0 -signal on the mesh and its sharp features can be sparsely represented in this basis. We have tested our approach on a large number of mesh surfaces with various feature types. Experimental results have shown that our method can faithfully decouple noise and sharp features.

It is not surprising to see that features can be detected by the ℓ_1 -analysis Compressed Sensing (CS) techniques since Lh (the Laplacian of the feature signal) is indeed a sparse signal. By removing the unnecessary smoothness penalty at locations of features, the DLRS scheme with the optimal parameter $\hat{\lambda}_G$ determined by GCV performs equivalently as denoising each C^2 patch individually and thus eventually obtains the optimal approximation to the underlying surface with theoretical guarantee.

Future work. First, we would like to apply the emerging framework of ℓ_1 -analysis compressed sensing to other problems in geometry processing. Second, the framework of ℓ_1 -analysis has potential in detecting higher-order features. The key is to design some coherent dictionaries which can sparsely represent the features. It is worthwhile to investigate it. Third, we observe that the information of a shape can be separated into two orthogonal components: a normal component which encodes the geometric information, and a tangential component which encodes the parameter information. Our current scheme modifies the geometry of the mesh by moving vertices along their normal directions. It is highly worthwhile to analyze and remove the noise of the input data in the parameterization domain. We believe that this is feasible but not straightforward.

ELECTRONIC APPENDIX

The electronic appendix to this article is available in the ACM Digital Library.

ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for their constructive comments.

REFERENCES

- H. Avron, A. Sharf, C. Greif, and D. Cohen-Or. 2010. ℓ_1 -sparse reconstruction of sharp point set surfaces. *ACM Trans. Graph.* 29, 5, 135–154.
- C. Bajaj and G. Xu. 2003. Anisotropic diffusion of surfaces and functions on surfaces. *ACM Trans. Graph.* 22, 1, 4–32.
- Z. Bian and R. Tong. 2011. Feature-preserving mesh denoising based on vertices classification. *Comput.-Aided Geom. Des.* 28, 1, 50–64.
- M. Botsch, L. Kobbelt, M. Pauly, P. Alliez, and B. Levy. 2010. *Polygon Mesh Processing*. AK Peters/CRC Press.
- S. Boyd and L. Vandenberghe. 2004. *Convex Optimization*. Cambridge University Press.
- E. Candes, Y. Eldar, D. Needell, and P. Randall. 2010. Compressed sensing with coherent and redundant dictionaries. *Appl. Comput. Harmonic Anal.* 31, 1, 59–73.
- E. Candes and T. Tao. 2005. Decoding by linear programming. *IEEE Trans. Inf. Theory* 51, 12, 4203–4215.
- E. Candes and M. Wakin. 2008. An introduction to compressive sampling. *IEEE Signal Process. Mag.* 25, 2, 21–30.
- E. Candes, M. Wakin, and S. Boyd. 2008. Enhancing sparsity by reweighted ℓ_1 minimization. *J. Fourier Anal. Appl.* 14, 5–6, 877–905.
- U. Clarenz, U. Diewald, and M. Rumpf. 2000. Anisotropic geometric diffusion in surface processing. In *Proceedings of the Conference on Visualization (VIS'00)*. 397–405.
- E. Cuthill and J. McKee. 1969. Reducing the bandwidth of sparse symmetric matrices. In *Proceedings of the 24th ACM National Conference*. 157–172.
- M. Desbrun, M. Meyer, P. Schroder, and A. Barr. 1999. Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'99)*. 317–324.
- D. Donoho. 2006. Compressed sensing. *IEEE Trans. Inf. Theory* 25, 4, 1289–1306.
- F. Duguet, F. Durand, and G. Drettakis. 2004. Robust higher-order filtering of points. Tech. rep. RR-5165, INRIA. <http://www-sop.inria.fr/reves/Basilic/2004/Dug04a/RR-5165.pdf>.
- Y. Eldar and G. Kutyniok. 2012. *Compressed Sensing: Theory and Applications*. Cambridge University Press.
- H. Fan, Y. Yu, and Q. Peng. 2010. Robust feature-preserving mesh denoising based on consistent sub-neighborhoods. *IEEE Trans. Vis. Comput. Graph.* 16, 2, 312–324.

- S. Fleishman, I. Drori, and D. Cohen-Or. 2003. Bilateral mesh denoising. *ACM Trans. Graph.* 22, 3, 950–953.
- M. Habbecke and L. Kobbelt. 2012. Linear analysis of nonlinear constraints for interactive geometric modeling. *Comput. Graph. Forum* 31, 2, 1–10.
- L. He and S. Schaefer. 2013. Mesh denoising via l0 minimization. *ACM Trans. Graph.* 32, 4, 64:1–64:8.
- K. Hildebrandt and K. Polthier. 2004. Anisotropic filtering of nonlinear surface features. *Comput. Graph. Forum* 23, 3, 391–400.
- K. Hildebrandt and K. Polthier. 2007. Constraint-based fairing of surface meshes. In *Proceedings of the 5th Eurographics Symposium on Geometry Processing (SGP'07)*. 203–212.
- T. Jones, F. Durand, and M. Desbrun. 2003. Non-iterative, feature-preserving mesh smoothing. *ACM Trans. Graph.* 22, 3, 943–949.
- R. B. Lehoucq, D. C. Sorensen, and C. Yang. 1998. *ARPACK Users' Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*, Vol. 6, SIAM.
- L. Liu, C. Tai, Z. Ji, and G. Wang. 2007. Non-iterative approach for global mesh optimization. *Comput.-Aided Des.* 39, 9, 772–782.
- F. J. Massey. 1951. The kolmogorov-smirnov test for goodness of fit. *J. Amer. Statist. Assoc.* 46, 253, 68–78.
- A. Nealen, T. Igarashi, O. Sorkine, and M. Alexa. 2006. Laplacian mesh optimization. In *Proceedings of the 4th International Conference on Computer Graphics and Interactive Techniques in Australasia and Southeast Asia (GRAPHITE'06)*. 381–389.
- D. C. Sorensen. 1997. *Implicitly Restarted Arnoldi/Lanczos Methods for Large Scale Eigenvalue Calculations*. Springer.
- Z. Su, H. Wang, and J. Cao. 2009. Mesh denoising based on differential coordinates. In *Proceedings of the IEEE International Conference on Shape Modeling and Applications (SMT'09)*. 1–6.
- X. Sun, P. Rosin, R. Martin, and F. Langbein. 2007. Fast and effective feature-preserving mesh denoising. *IEEE Trans. Vis. Comput. Graph.* 13, 5, 925–938.
- G. Taubin. 1995. A signal processing approach to fair surface design. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'95)*. 351–358.
- C. Tomasi and R. Manduchi. 1998. Bilateral filtering for gray and color images. In *Proceedings of the 6th International Conference on Computer Vision (ICCV'98)*. 839–846.
- G. Wahba. 1990. *Spline Models for Observational Data*. SIAM, Philadelphia, PA.
- L. Xu, C. Lu, Y. Xu, and J. Jia. 2011. Image smoothing via L0 gradient minimization. *ACM Trans. Graph.* 30, 6, 174:1–174:12.
- H. Yagou, Y. Ohtake, and A. Belyaev. 2002. Mesh smoothing via mean and median filtering applied to face normals. In *Proceedings of the Geometric Modeling and Processing – Theory and Applications (GMP'02)*. 124–131.
- Y. Zheng, H. Fu, O. K.-C. Au, and C.-L. Tai. 2010. Bilateral normal filtering for mesh denoising. *IEEE Trans. Vis. Comput. Graph.* 17, 10, 1521–1530.

Received December 2012; revised December 2013; accepted December 2013