Computer-Aided Design 78 (2016) 179–187

Contents lists available at ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad

Surface approximation via sparse representation and parameterization optimization

Linlin Xu, Ruimin Wang, Zhouwang Yang, Jiansong Deng, Falai Chen, Ligang Liu [*]

School of Mathematical Sciences, University of Science and Technology of China, Hefei 230026, PR China

ARTICLE INFO

Keywords:
Surface approximation
Sparse representation
Parameterization optimization

ABSTRACT

Surface approximation with smooth functions suffers the problems of choosing the basis functions and representing non-smooth features. In this work, we introduce a sparse representation for surfaces with a set of redundant basis functions, which efficiently overcomes the overfitting artifacts. Moreover, we propose an approach of parameterization transformation, which makes the possibility to represent non-smooth features by the composition of a smooth function and a non-smooth domain optimization. We couple the sparse representation and the parameterization transformation in a global optimization to respect sharp features with smooth polynomial basis functions. Our approach is capable for approximating a wide range of surfaces with different level of sharp features. Experimental results have shown the feasibility and applicability of our proposed method in various applications.

1. Introduction

In geometric modeling, surface approximation is one of the most important and interesting research topics. Actually, it is the fundamental problem of data fitting. Fitting data with a linear combination of a family of basis functions has been widely studied in various applications such as data smoothing, data predicting, feature extraction, etc. [1,2]. The coefficients for basis functions are obtained by solving a least-squares solution [3,4]. The choice of basis functions will make a big difference on the fitting results.

To be flexible about basis functions, sparse representation is first proposed in image processing problems [5] like noise reduction, image compression, face recognition, and pattern classification. It represents input signal by automatically choosing proper basis from a redundant set of basis functions (upper row in Fig. 1). As many as possible functions can be put into basis set to satisfy more geometric features, what is more, sparse representation could effectively overcome the overfitting artifacts. However, it is nontrivial to represent non-smooth features using a set of smooth basis functions (middle row in Fig. 1).

Fortunately, we realized that in addition to the functions themselves, the definition domain (the parameterization of input data) could also affect the shape of function. Specifically, it is possible to obtain non-smooth shape by the composition of a smooth function and a non-smooth domain optimization. Furthermore, geometry itself contains no intrinsic parameterization, it is also quite unreasonable to fix the parameterization which can be obtained by existing algorithm. Then we consider the feasibility of representing various geometric features with parameterization optimization. The parameterization of input data has been widely studied in the literature [6]. Current methods try to compute parameterization to preserve geometric properties such as angle [7] and triangle rigidity [8]. But little effort has been put on this problem of finding a suitable parameterization optimization for good approximation result under a certain set of basis functions.

In this paper, by coupling *sparse representation* and *parameterization optimization*, we propose a novel surface approximation method for representing sharp features using *polynomial* basis functions. In a sense it is an implicit composite function (Section 3.2.1). Since sparse optimization and optimal parameterization will be solved simultaneously, we also present an augmented Lagrange method based formulation and solve it in an iterative way. Experimental results show that our new representation method has the ability to approximate sharp features even with a set of smooth functions (lower row in Fig. 1). That is to say, definition domains determine the representation ability of the linear space of predefined basis functions, our idea of optimizing parameterization just largens this space.

For complex geometric surface, we first decompose it into multiple small local patches where adjacent patches have overlaps. Aiming for the continuity of adjacent patches, we finally put

[*] Corresponding author.
E-mail address: lgliu@ustc.edu.cn (L. Liu).

http://dx.doi.org/10.1016/j.cad.2016.05.013

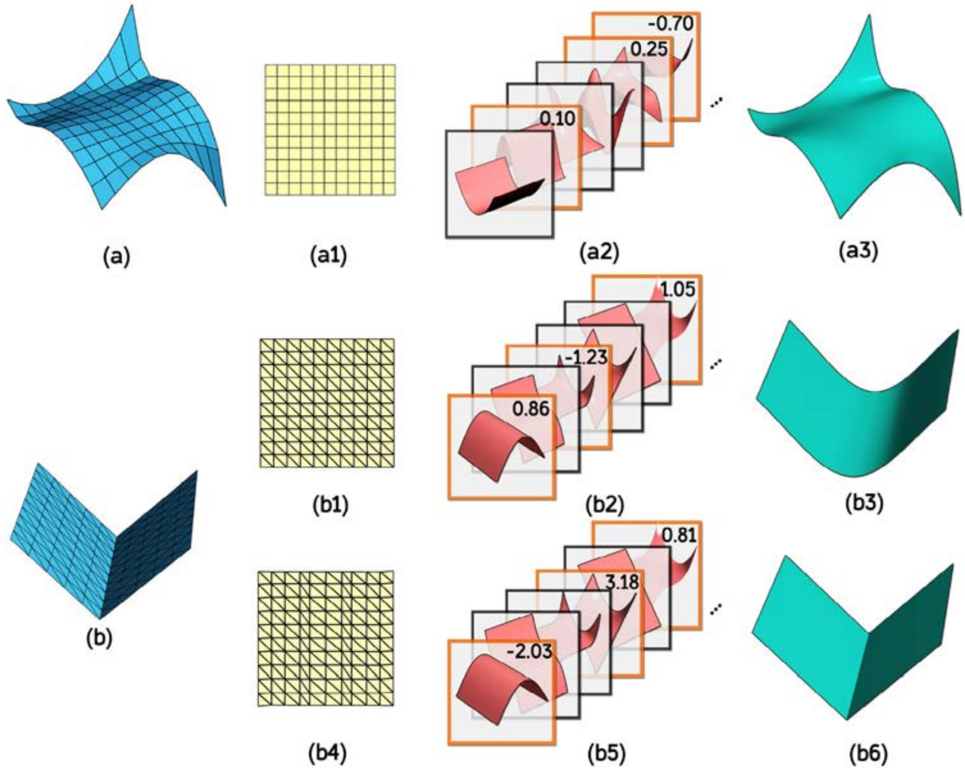


Fig. 1. Approximating surfaces with sparse representation and parameterization optimization. The upper row shows the result of approximating a smooth surface with sparse representation. Given a smooth surface patch (a) defined on a planar domain (a1), a subset of basis functions (highlighted in orange with the corresponding coefficients shown in the top-right) from a set of redundant basis functions (a2) are chosen to represent (a) as shown in (a3). The lower row shows the results produced by our approach. Given a surface patch with a crease feature (b) defined on the domain (b1), the naive approximation using the sparse representation (b2) cannot well represent the feature (b3). With the optimized parameterization transformation (b4), our approximation approach of using the sparse representation (b5) respects the sharp feature quite well (b6).

forward a consistent surface approximation method by globally processing small patches.

The contributions of our work are summarized as:

- A new composite representation method is proposed for surface approximation by coupling sparse representation and parameterization optimization.
- An augmented Lagrange method based numerical algorithm is presented to solve the composite problem efficiently.

2. Background and motivation

Surface approximation and data fitting. In Computer Aided Design, B-splines are the de facto industrial standards for surface approximation schemes [9–11] though it is much difficult to fit models with complicated topology. Different from this explicit representation, implicit function is commonly defined as the sum of radial basis functions [12–14] or the signed distance [15]. Now that surface approximation is one specific field of data fitting, other applications like data mining [3], statistics [1] or biology [2] will also give much important reference data. Though experimental results have shown the validity of these representation methods, overfitting artifacts [16] cannot be avoided by directly minimizing squared-error or redundant basis functions [3,4].

Sparse representation. To overcome the overfitting artifacts in traditional fitting above, sparse representation is adopted to represent given data using a small number of basis functions [5] and has theoretical guarantee in many image applications such as noise reduction, super resolution, compression, etc. [17–19].

Being attracted by the advantages of sparse techniques, a good few researchers have extended these techniques to 3D geometry processing rapidly and successfully [20]. The work of [21] employs K-SVD algorithm on accurately compressing point cloud. [22] uses ℓ_1 optimization to compute normals of point cloud. [23] uses a ℓ_0-smoothing method based on discrete gradient operator inspired by [24]. Wang et al. [25] introduce a weighted ℓ_1-analysis compressed sensing which decouples noise and feature. These approaches all focus on geometric applications, here we adopt sparse representation to approximate geometry with a novel optimization framework.

Basis functions. There are many possible choices for basis functions. [26] uses polynomials for hole-filling problem. [13] adopts implicit representation with Radial Basis Functions (RBFs) to achieve surface reconstruction and mesh completion. [27] fits meshes using PHT-spline. These methods all find a suitable application for one specific type of basis functions. Besides the complexity of some basis functions, another weak point is the lack of universality, e.g., smooth functions cannot represent sharp features like crease, cusp, or corner. Our parameterization optimization (Section 3.2.1) successfully addresses these problems even with simple polynomials as basis functions.

Parameterization and geometry features processing. As mentioned in Section 1, parameterization of geometric data has been well studied in geometry processing [28,6]. [29,30] compute the parameterization in angle space, with the results minimizing angular distortion. [31,32] process the parameterization optimization guided by stretch. [33–35] try to preserve triangle angles. And [8] aims for the rigidity of triangles which inspires us with our as-rigid-as-possible (ARAP) term included in the regularization term (4) in Section 3.1.

To align geometry features, NURBS [36] can only represent features which are parameterized onto the grid of domain and it is hard to achieve. The methods of [37–39] are essentially driven by feature detection or feature-aware energy that will affect the results largely, as well as many other geometry processing problems like geometry inpainting [40,41], mesh smoothing [42,43] and reconstruction [44]. Without these additional detection techniques, our novel composite representation algorithm makes the parameterization automatically respect sharp features in the input data.

In summary, we develop a novel scheme for a good surface approximation. Specifically, we allow users to choose any number of simple polynomial basis functions and adopt a sparse fitting to automatically choose a small basis set to fit the given geometric surface.

3. Surface approximation of scalar data

Now that the new composite representation and its solver are our main contributions, to be clear and compact, we take the scalar data as input to introduce the methodology and do some experiments to show the validity of the algorithm in this section.

3.1. Formulation

Denote $\mathbf{b} = (b_1, \dots, b_m)^T$ as a set of scalar observations from some underlying Euclidean space $\mathbb{R}^k$. We use monomial functions $\mathcal{P} = \{p_j(\mathbf{u}) : \, degree(p_j) \le t\}_{j=1}^n$ defined on $\mathbb{R}^k$ as the basis functions, here we set $t = 14$. The fitting problem, also called regression, is actually to represent $\mathbf{b}$ as a linear combination of the basis $\mathcal{P}$, which aims to find the coefficients $\mathbf{c} = (c_1, \dots, c_n)^T$ by minimizing the error function

$$F(\mathbf{c}, U) = \sum_{i=1}^{m} \Big[b_i - \sum_{j=1}^{n} c_j p_j(\mathbf{u}_i) \Big]^2, \tag{1}$$

where $U = (\mathbf{u}_1, \dots, \mathbf{u}_m) \in \mathbb{R}^{k \times m}$ are the parameters of the data.

With the same optimization problem (1), we optimize a sparse vector $\mathbf{c}$ (with only a few non-zero values) and the parameter values U of the input data simultaneously as follows:

$$\begin{aligned} \min_{\mathbf{c}, U} \quad & F(\mathbf{c}, U) \\ \text{s.t.} \quad & \|\mathbf{c}\|_0 \le s, \end{aligned} \tag{2}$$

where s is the parameter for sparsity.

However, directly solving U in problem (2) means solving for every $\mathbf{u}_i$ independently. This trivial approach might lead to a bad parameterization as shown in Fig. 7. So we add a regularization term $Q(U)$ to (2) to prevent the parameterization (the triangulation of $\{\mathbf{u}_i\}$) from transforming badly like irregularity or flip, our final optimization problem becomes:

$$\begin{aligned} \min_{\mathbf{c}, U} \quad & F(\mathbf{c}, U) + Q(U) \\ \text{s.t.} \quad & \|\mathbf{c}\|_0 \le s. \end{aligned} \tag{3}$$

We will discuss about $Q(U)$ in Section 3.2.2.

3.2. Methodology

3.2.1. Parametric optimization

Parametric optimization. In fact, the parameterization is optimized discretely in our solving algorithm (Section 3.3) for problem (3). To illustrate easily and clearly, we temporarily formulate the parameterization optimization as a 2D to 2D mapping which needs further studies in the future. For the initialization of parameterization $\mathbf{u}^{(0)}$, it can be obtained by any existing approach, we will give a simple discussion later. Here we adopt the LSCM method [7].

Assume $\{\mathbf{u}_i\}_{i=1}^m$ is a set of vertices in domain M and $\{\widetilde{\mathbf{u}}_i\}_{i=1}^m$ is a set of vertices in domain N with $\Phi(\mathbf{u}_i) = \widetilde{\mathbf{u}}_i$. Let T_M be the triangulation of $\{\mathbf{u}_i\}_{i=1}^m$ and T_N the triangulation of $\{\widetilde{\mathbf{u}}_i\}_{i=1}^m$. Thus we define the mapping Φ as a piecewise linear transformation of two-dimensional triangulated structure as illustrated in Fig. 2, where $\Phi(\mathbf{u}) = w_1\Phi(\mathbf{u}_{i_1}) + w_2\Phi(\mathbf{u}_{i_2}) + w_3\Phi(\mathbf{u}_{i_3})$ for any $\mathbf{u} = w_1\mathbf{u}_{i_1} + w_2\mathbf{u}_{i_2} + w_3\mathbf{u}_{i_3}$ lying in the triangle $\triangle\mathbf{u}_{i_1}\mathbf{u}_{i_2}\mathbf{u}_{i_3}$ with its barycenter coordinates (w_1, w_2, w_3).

Implicit composite function. By the aid of Φ, we directly obtain a composite function (Fig. 2). More precisely, a function f defined on domain $M \subset \mathbb{R}^2$ consists of some linear combined function $h : \mathbb{R}^2 \to \mathbb{R}$ and a parameterization mapping $\Phi : \mathbb{R}^2 \to \mathbb{R}^2$, that is, $f(\mathbf{u}) = h \circ \Phi(\mathbf{u})$. For our fitting problem, h is a linear combination of monomial functions with $h(\mathbf{u}) = \sum_{j=1}^n c_j p_j(\mathbf{u})$. Then the composite function will be written as follows:

$$f(\mathbf{u}) = h \circ \Phi(\mathbf{u}) = \sum_{j=1}^{n} c_j p_j(\Phi(\mathbf{u})).$$

This composite function is no longer the simple linear combination of some basis. The existence of the implicit Φ successfully overcomes the limitations in trivial linear combination as mentioned in Section 1 (middle row in Fig. 1), thus the flexibility in basis selection is obvious.

3.2.2. Regularization of parameterization

Our regularization term $Q(U)$ for parameterization consists of a data term $Q_1(U)$ and an ARAP term [8] $Q_2(U)$ which aims at shape preserving for mesh parameterization:

$$Q(U) = \beta_1 Q_1(U) + \beta_2 Q_2(U), \tag{4}$$

here,

$$Q_1(U) = \|U - U^{(0)}\|^2,$$

$$Q_2(U) = \sum_t \Delta_t((\sigma_t^1 - 1)^2 + (\sigma_t^2 - 1)^2),$$

and β_1, β_2 are scalar parameters, Δ_t is the area of the tth triangle, σ_t^1 and σ_t^2 are the signed singular values of the Jacobian of the tth triangle's transformation. For more details, please refer to Section 4.4 in [8].

3.3. Solver

It is nontrivial to solve the non-linear optimization (3) when $\mathbf{c}$ and U are coupled together.

Auxiliary variables. To this end, we introduce auxiliary variables $\mathbf{f} = (f_1, \dots, f_m)^T$ for converting problem (3) to a new optimization:

$$\begin{aligned} \min_{\mathbf{c}, U, \mathbf{f}} \quad & \|\mathbf{b} - \mathbf{f}\|^2 + Q(U) \\ \text{s.t.} \quad & \|\mathbf{c}\|_0 \le s, \\ & f_i = \sum_{j=1}^{n} c_j p_j(\mathbf{u}_i), \quad i = 1, \dots, m. \end{aligned} \tag{5}$$

Augmented Lagrange method. The augmented Lagrange method (ALM) [45] is adopted to solve (5) as follows:

$$\begin{aligned} \min_{\mathbf{c}, U, \mathbf{f}} \quad & \|\mathbf{b} - \mathbf{f}\|^2 + Q(U) + \frac{\rho}{2} \sum_{i=1}^{m} \Big[f_i - \sum_{j=1}^{n} c_j p_j(\mathbf{u}_i) + \frac{\lambda_i}{\rho} \Big]^2 \\ \text{s.t.} \quad & \|\mathbf{c}\|_0 \le s, \end{aligned}$$

where $\Lambda = (\lambda_1, \dots, \lambda_m)$ are the Lagrange multipliers, ρ is the penalty factor. We apply the alternating direction method of multipliers (ADMM) to alternately update variables $\mathbf{c}$, U, $\mathbf{f}$ and Λ.

Iteration process. The parameters $\beta_1, \beta_2, s, \rho$ and τ are fixed. With the initialization of triangulation $U^{(0)}$, $\Lambda^{(0)} = \mathbf{0}$ and $\mathbf{f}^{(0)} = \mathbf{0}$, we update each variable iteratively via:

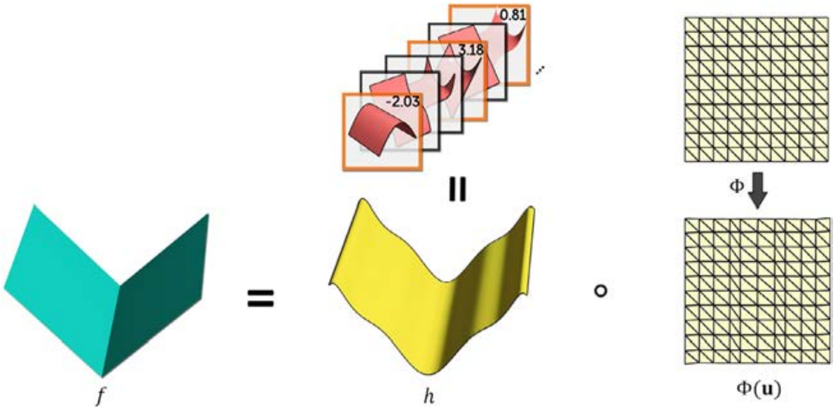


Fig. 2. A surface patch with a crease feature f is represented as a composition of a smooth function h and a parameterization $\Phi(u)$, where f is a sparse representation from a set of redundant basis functions and $\Phi(u)$ is a planar transformation from a predefined triangulation domain.

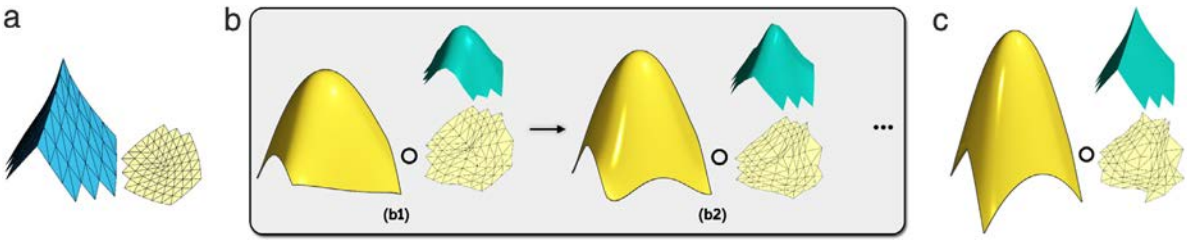


Fig. 3. Overview of our approach. (a) The input surface patch (in blue) with an initial parameterization obtained by the LSCM (in buff); (b1) and (b2) show the two results of intermediate iterations during the optimization. In each iteration (b1) and (b2), a surface (in cyan) is obtained by the composition of the surface (in yellow) represented by the sparse representation and the optimized triangular parameterization domain (in buff). (c) Shows the final result.

(1) **c**-subproblem/sparsity optimization:

$$\min_{\mathbf{c}} \quad \frac{\rho}{2}\sum_{i=1}^{m}\Big[f_i^{(k-1)} - \sum_{j=1}^{n} c_j p_j(\mathbf{u}_i^{(k-1)}) + \frac{\lambda_i^{(k-1)}}{\rho}\Big]^2$$
$$\text{s.t.} \quad \|\mathbf{c}\|_0 \le s.$$

(2) U-subproblem/parameterization optimization:

$$\min_{U} \frac{\rho}{2}\sum_{i=1}^{m}\Big[f_i^{(k-1)} - \sum_{j=1}^{n} c_j^{(k)} p_j(\mathbf{u}_i) + \frac{\lambda_i^{(k-1)}}{\rho}\Big]^2 + Q(U).$$

(3) **f**-subproblem/data fitting:

$$\min_{\mathbf{f}} \|\mathbf{b}-\mathbf{f}\|^2 + \frac{\rho}{2}\sum_{i=1}^{m}\Big[f_i - \sum_{j=1}^{n} c_j^{(k)} p_j(\mathbf{u}_i^{(k)}) + \frac{\lambda_i^{(k-1)}}{\rho}\Big]^2.$$

This subproblem has a closed-form solution $f_i^{(k)} = \frac{2b_i+\rho a_i}{2+\rho}$ with $a_i = \sum_{j=1}^{n} c_j^{(k)} p_j(\mathbf{u}_i^{(k)}) - \frac{\lambda_i^{(k-1)}}{\rho}$ for $i = 1, \dots, m$.

(4) Λ-update:

$$\lambda_i^{(k)} = \lambda_i^{(k-1)} + \tau\rho\Big[f_i^{(k)} - \sum_{j=1}^{n} c_j^{(k)} p_j(\mathbf{u}_i^{(k)})\Big], \quad i = 1, \dots, m.$$

Implementation. There are in all three subproblems and one update in the iteration steps:

- **c**-subproblem is the traditional sparse representation problem which can be solved by the classical orthogonal matching pursuit(OMP) algorithm [46];
- U-subproblem is solved using a function named *find_min_using_approximate_derivatives* in dlib library using Broyden–Fletcher–Goldfarb–Shanno (BFGS) search strategy [47], it numerically approximates the gradient instead of taking a gradient function;
- **f** subproblem and Λ update are simple scalar operations.
- The algorithm terminates if the error between two steps is smaller than a threshold or the number of iteration steps is bigger than a threshold.

3.4. Performance

For the parameters in above algorithm, we experimentally set them as $\beta_1 = 0.2, \beta_2 = 0.1, s = 15, \rho = 0.1, \tau \le 1.0$.

Fig. 3 shows the overview of our algorithm applied on a scalar data with sharp feature (a). Two intermediate iterations are illustrated (b1) and (b2). As the solver iterates, the representation gradually approximates input data. When the algorithm reaches the terminal condition, we get the final representation result (c).

Fig. 4 shows two more examples with the comparison between our approach and traditional sparse representation (1). Input (a) with initial parameterization, both the representation results of traditional sparse representation (b) and our method (c) are computed. The colormap of fitting error makes the approximation performance visible and clearly tells that no matter the input signal contains feature or not, our composite representation performs much better. Especially, it respects sharp feature well even under smooth polynomial functions.

Fig. 5 illustrates how energy changes for some examples. Generally, our algorithm terminates at about 15 steps. The time cost is also shown in the figure.

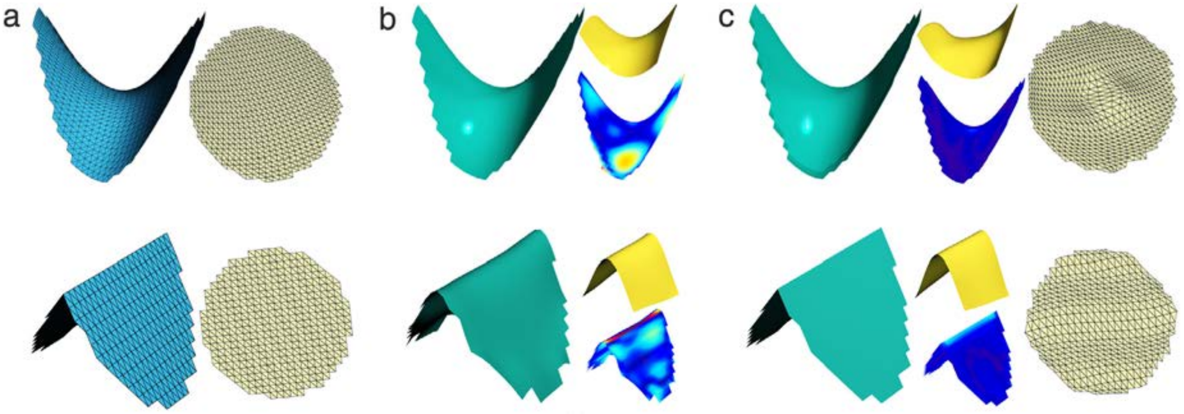
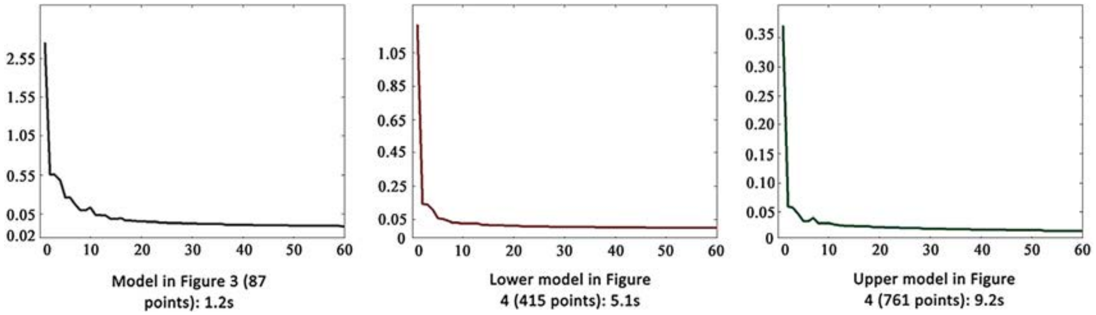


Fig. 4. Comparison between our approach and traditional sparse representation. (a) The two input surface patches (in blue) and initial parameterizations (in buff). (b) and (c) are the results of traditional sparse representation and our approach respectively. With fixed parameterization, sparse representation (yellow models in (b)) directly makes the final results (in cyan). From the colormaps of approximation error in the bottom-right, smooth surface (upper row) can be approximated well, but the crease feature (lower row) is difficult to represent. Using our new parameterization optimization (in buff), in addition to the better approximation result of smooth patch, the feature is also successfully respected.



Model in Figure 3 (87 points): 1.2s

Lower model in Figure 4 (415 points): 5.1s

Upper model in Figure 4 (761 points): 9.2s

Fig. 5. Statistics of our algorithm on changes of fitting error and time cost. The models come from Figs. 3 and 4.

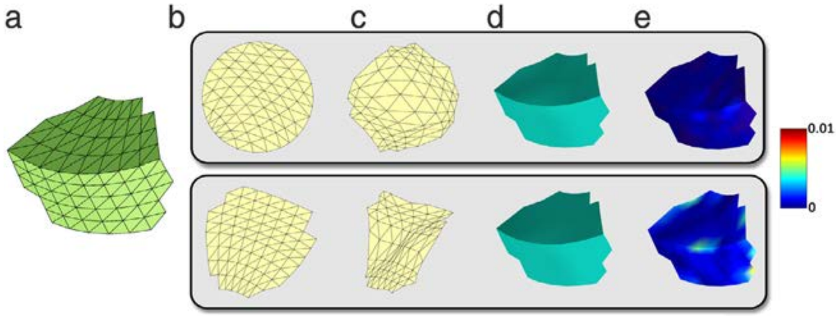


Fig. 6. The approximation results with different initial domains (in buff). (a) The input surface patch. (b) Shows the initial parameterizations obtained via two different approaches. The upper row is computed using discrete conformal map with circulator border, the lower row comes from least square conformal map. Starting from different initializations, our algorithm can always find the optimal parameterizations (c) to well represent the input surface (d). It is clear from the colormaps of approximation errors (d).

3.5. Discussions

Parameterization initialization. Fig. 6 shows the results of our algorithm with different approaches for initial parameterization. The upper row uses discrete conformal map [48], the lower uses least square conformal map. Although the initialization (b) and optimal parameterizations (c) are different, both final representations approximate input signal (a) well, this is clear from the error colormaps (e). It means that our algorithm can always find the optimal parameterization according to the initial domain and simultaneously find a good representation for input signal under this transformation.

Influence of regularization. There are two regularization terms in our formulation (4). Fig. 7 illustrates the difference without and with triangular regularization term (4). As we can see, this term helps our algorithm obtain geometry-reasonable transformation (c). The result without this term (b) has singularities in the parametric optimization.

4. Consistent surface approximation for meshes

4.1. Approximation of local geometry

For an input geometric data, we separate it into three signals of x, y, z coordinates (scalar data in Section 3) and process each coordinate respectively starting from the initial parameterization. Fig. 8 gives a intuitive illustration with a simple geometry patch.

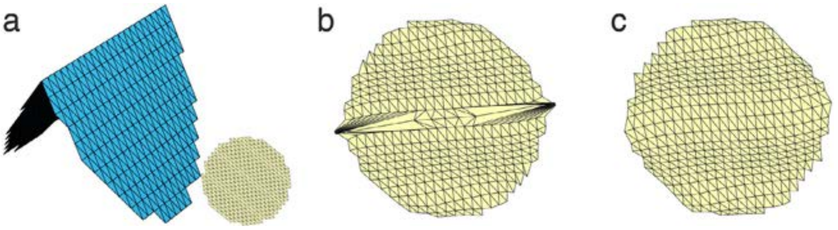


Fig. 7. The influence of triangular regularization term. (a) The input surface patch (blue) with initial parameterization (buff). With regularization term for parameterization, the transformed result (c) is more reasonable than that without any constrain term.

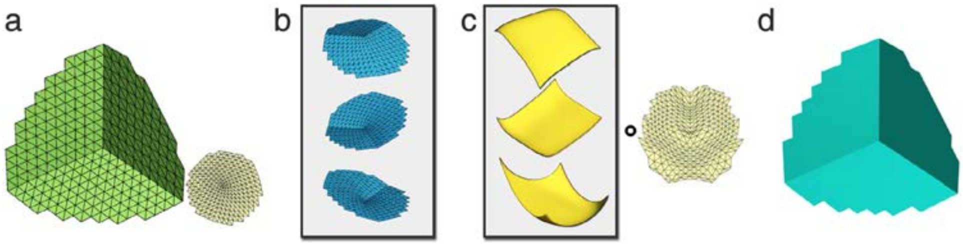


Fig. 8. Approximation of a geometry patch. (a) Shows an geometric patch (green) with the initial parameterization (buff). Signals along three coordinates are extracted (b) and processed respectively (c). For each coordinate, it is a composition of sparse representation (yellow) and the common transformed parameterization (buff). (d) Is the final approximation result.

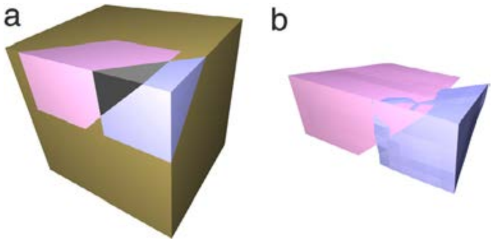


Fig. 9. Disadvantage resulting from processing each geometric patch individually. Gaps (b) will be generated in the overlapping region (gray) in (a) by individually processing adjacent patches (in red and purple).

4.2. Consistent surface approximation

For complex geometric surface, we first decompose it into multiple small local patches where adjacent patches have overlaps. Here, we must pay attention to a hidden trouble that individually approximating the geometric data on each patch using Eq. (3) may result in gaps. Fig. 9 illustrates this problem with two adjacent patches. The patches in pink and purple in (a) have an overlap region in gray, (b) is the result with gaps by individual processing. To address it, we develop a global approximation method by blending all local patches in a consistent manner as follows.

Approximation with global consistency Since the adjacent patches extracted above have overlaps, a geometric point $\mathbf{p}$ may appear in several different patch $M_\ell^{\mathbf{p}}$, $\ell = 1, \ldots, n_{\mathbf{p}}$. Assume the parameters of $\mathbf{p}$ on those domains are $\{\mathbf{u}_\ell^{\mathbf{p}}\}$ and sparse coefficients on these

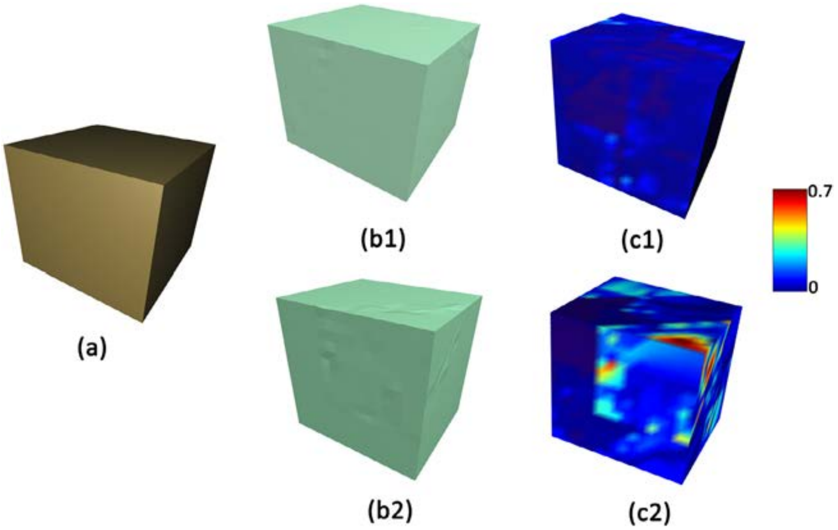


Fig. 10. The influence of patch size. (a) Input geometry. (b1) shows the approximation result using 7-ring neighborhood of surface points as local patches, while (b2) uses 10-ring. (c1) and (c2) are the colormaps of (b1) and (b2).

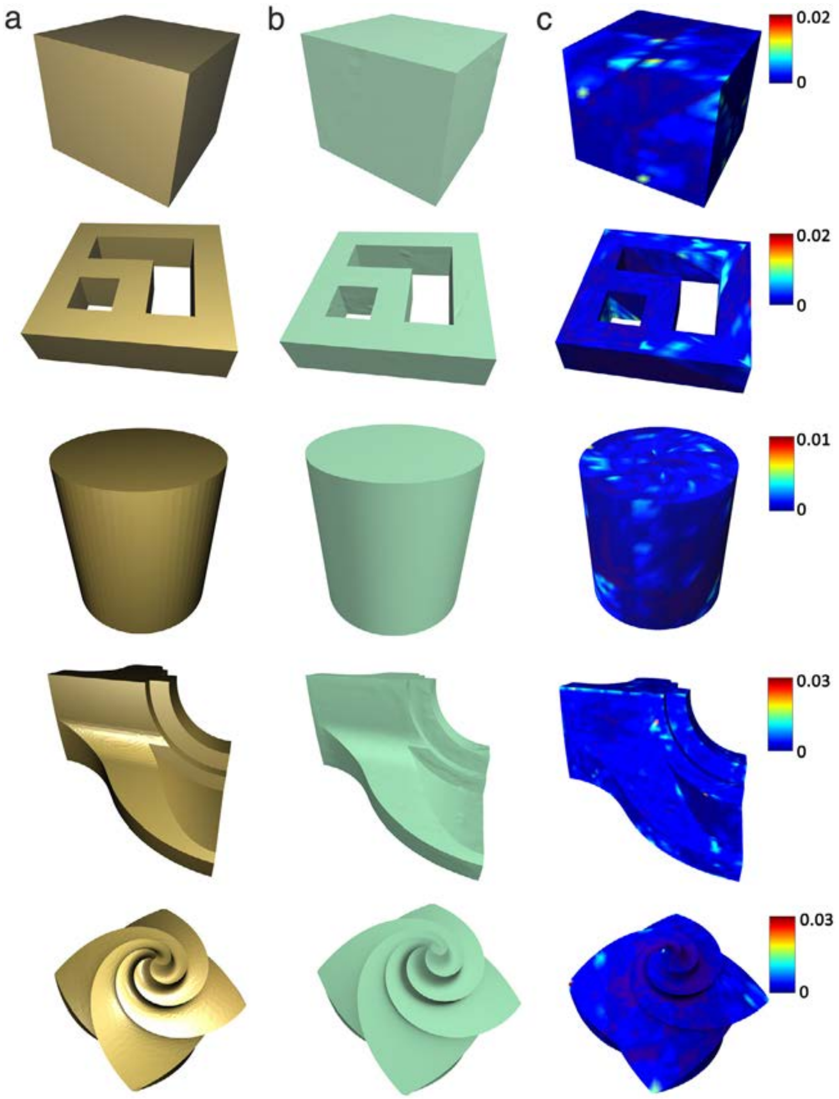


Fig. 11. Approximation results for several models with crease features. (a) The input geometries. (b) Our approximation results. (c) Colormaps of approximation errors.

domains are $\{\mathbf{c}_\ell\}$, the local approximation is then represented as $\sum_{j=1}^{n} c_j p_j(\mathbf{u}_\ell^{\mathbf{p}})$.

Factually, global consistency means that the final function value at each point is decided by several local patch representations, we compute it by directly averaging the approximations on each $M_\ell^{\mathbf{p}}$ with $\frac{1}{n_{\mathbf{p}}}\sum_{\ell=1}^{n_{\mathbf{p}}}\sum_{j=1}^{n} c_j p_j(\mathbf{u}_\ell^{\mathbf{p}})$. Then We have:

$$\begin{aligned}\min_{\mathbf{C},\mathbf{U}} \quad & \sum_{\mathbf{p}}\Big[b(\mathbf{p}) - \frac{1}{n_{\mathbf{p}}}\sum_{\ell=1}^{n_{\mathbf{p}}}\sum_{j=1}^{n} c_j p_j(\mathbf{u}_\ell^{\mathbf{p}})\Big]^2 + \sum_{k=1}^{K} Q(U_k). \\ \text{s.t.} \quad & \|\mathbf{c}_k\|_0 \leq s,\ k = 1, \ldots, K.\end{aligned} \tag{6}$$

Here, $b(\mathbf{p})$ is the geometric data of point $\mathbf{p}$ on the input surface, K is the number of local patches, $\mathbf{C} = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_K\}$ and $\mathbf{U} = \{U_1, U_2, \ldots, U_K\}$ are respectively the local coefficients and parameterizations on each patch.

Solver. Obviously, problem (6) is a large-scale sparse optimization. We first evaluate every $\mathbf{c}_k$ and U_k on each extracted patch, then iteratively update each pair of them in a Gauss–Seidel way by taking $\{\mathbf{c}_i\}_{i\neq k}$ and $\{U_i\}_{i\neq k}$ as known on the other patches. This global consistent optimization achieves continuity between patches.

5. Results

We experiment on various unitized examples to illustrate the performance of our globally consistent approximation. All the examples presented in this article are made on a dual-core 3.8 GHz machine with 8G memory.

Using the same values of parameters introduced in Section 3.4, the final result can be obtained by 1 or 2 consistent blending iterations in general, and we select the 7-ring neighborhood of surface vertices as local patches in all the experiments. Suppose V is the number of the input surface, we generally obtain about 2% patches of V for each geometry surface. Obviously, the number of patches will decrease greatly when the size of each local patch increases, but there is no guarantee about the fitting result. Fig. 10 illustrates this problem using Cube model under different patch sizes. Table 1 lists all the time cost.

Fig. 11 shows several approximation results with sharp features, Fig. 12 gives more results with much details. According to the colormaps (c), the approximation error is about 0.03. This sufficiently tells that our consistent sparse representation performs quite well on surface approximation. In addition to the good approximation of smooth regions, it successfully preserves different levels of geometry features. From the computation time in Table 1, we will observe that the models with details in Fig. 12

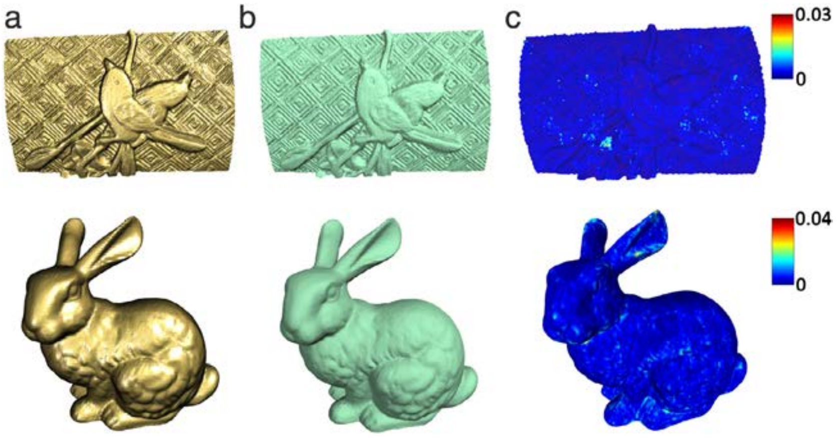


Fig. 12. Approximation results for several models with much details. (a) The input geometries. (b) Our approximation results. (c) The colormaps of approximation errors.

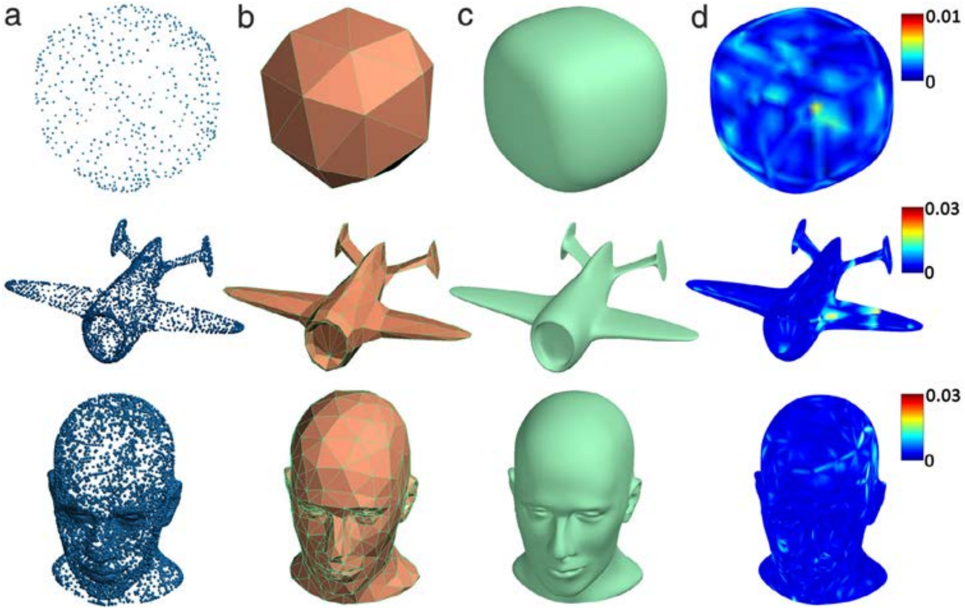


Fig. 13. Surface reconstruction result. (a) The input point clouds. (b) The underlying triangular structures. (c) The reconstruction results. (d) Colormaps of fitting errors.

Table 1
Time cost for the examples in Figs. 11–13.

Models	Fig. 11					Fig. 12		Fig. 13		
	Cube	Cubetorus	Cylinder	Fandisk	Flower	Bird	Bunny	Ball	Airplane	Head
Vertex	1538	2686	2732	6475	7919	55 955	35 947	1410/24	9730/154	26 909/428
Patch	26	40	39	88	97	882	551	6	41	113
Time (m)	1.3	2.33	1.63	4.5	5.01	27.8	19.1	0.3	1.71	4.62

take much less time than those with sharp features in Fig. 11. This is due to the higher smoothness of the local patch with details.

Surface reconstruction. A natural application is surface reconstruction which is also the approximation problem. Fig. 13 shows the process of reconstruction from point cloud (a) using our method. We regard point cloud as signals defined on a underlying triangular structure (b) which can obtained by simplifying (a). By projecting the points onto this structure, we get the local barycenter coordinates on corresponding triangles. Then we find a cover of (b) as above and solve the representations taking all points from (a) as observations. At last the representation for (a) is obtained (c). (d) shows the fitting error between (a) and (c). As we can see, the error is much lower than 0.01 meaning that our algorithm can well represent the input point cloud. The features are also preserved (second model of plane). Here, we select 1-ring neighborhood of the underlying triangular structure as a local patch. The number of patches is showed in Table 1 where the sizes of point cloud (a) and the triangular structure (b) are both listed in the Vertex term.

6. Conclusion and future work

In this paper, we propose a novel surface approximation method by coupling sparse representation and parameterization optimization in a global optimization. The introduction of parameterization optimization makes it possible to represent non-smooth features by the composition of a smooth function and a non-smooth domain optimization, and to some extent this novel idea addresses the excessive dependency on parameterization in fitting problem. Our new approach is capable for approximating a wide range of surfaces with different level of sharp features. Experimental results show its powerfulness in different applications and potential in geometric processing.

Insight. The parameterization optimization aims to find a more proper domain for predefined basis functions, it is in fact learned from the input geometry. In the point view of deep learning [49], our method is a multi-layer representation with two layers. This deeper representation definitely has stronger representation ability than just one layer. So this multi-layer property illustrates well why we can obtain promising results in surface approximation.

Limitation. As we introduce a more complex formulation for sparse representation, more time is needed to solve the problem as shown in Table 1. For a simple example, our algorithm has to cost 9.2 s to reach the ideal result for the upper model in Fig. 4, traditional sparse representation costs only 0.5 s. Application on higher dimension will definitely cost too much time solving it.

Future work. As discussed above, we regard our work as a pioneer for representing geometry with multi layers (two in our approach), one for parameterization transformation and one for linear combination. Since we have not found a suitable formulation for the parameterization transformation (Section 3.2.1), how to set the parameterization layer is still unknown. And we believe there are still a lot of interesting and challenging problems in geometry processing that can be formulated with multi-layer representation. We would like to dig more on this novel composite idea to solve more problems.

Acknowledgments

The authors thank the reviewers for providing useful comments and suggestion. This work is supported by the National Natural Science Foundation of China (11371341, 11526212, 11571338, 61222206), One Hundred Talent Project of the Chinese Academy of Sciences, and the Fundamental Research Funds for the Central Universities (WK0010000051).

References

[1] Neter J, Kutner MH, Nachtsheim CJ, Wasserman W. Applied linear statistical models. Chicago: Irwin; 1996.

[2] Motulsky H, Christopoulos A. Fitting models to biological data using linear and nonlinear regression: A practical guide to curve fitting. Oxford University Press; 2004.

[3] Pyle D. Data preparation for data mining, Vol. 1. Morgan Kaufmann; 1999.

[4] Guest PG. Numerical methods of curve fitting. Cambridge University Press; 2012.

[5] Elad M. Sparse and redundant representations: From theory to applications in signal and image processing. 1st ed. Springer Publishing Company, Incorporated; 2010. 9781441970107.

[6] Sheffer A, Praun E, Rose K. Mesh parameterization methods and their applications. Found Trends Comput Graph Vis 2006;(2):105–71.

[7] Lévy B, Petitjean S, Ray N, Maillot J. Least squares conformal maps for automatic texture atlas generation. In: ACM transactions on graphics (TOG), Vol. 21. ACM; 2002. p. 362–71.

[8] Liu L, Zhang L, Xu Y, Gotsman C, Gortler SJ. A local/global approach to mesh parameterization. Comput Graph Forum 2008;27(5):1495–504. (Proceedings of Eurographics Symposium on Geometry Processing).

[9] Hoschek J, Lasser D, Schumaker LL. Fundamentals of computer aided geometric design. AK Peters, Ltd.; 1993.

[10] Piegl LA, Tiller W. The NURBS book. Springer; 1995.

[11] Deng C, Lin H. Progressive and iterative approximation for least squares b-spline curve and surface fitting. Comput-Aided Des 2014;47:32–44.

[12] Muraki S. Volumetric shape description of range data using "blobby model". In: ACM SIGGRAPH computer graphics, Vol. 25. ACM; 1991. p. 227–35.

[13] Carr JC, Beatson RK, Cherrie JB, Mitchell TJ, Fright WR, McCallum BC, et al. Reconstruction and representation of 3D objects with radial basis functions. In: Proceedings of the 28th annual conference on computer graphics and interactive techniques. ACM; 2001. p. 67–76.

[14] Turk G, O'brien JF. Modelling with implicit surfaces that interpolate. ACM Trans Graph 2002;21(4):855–73.

[15] Hoppe H, DeRose T, Duchamp T, McDonald J, Stuetzle W. Surface reconstruction from unorganized points, Vol. 26. ACM; 1992.

[16] Leinweber DJ. Stupid data miner tricks. J Invest 2007;(1):15–22.

[17] Donoho DL. De-noising by soft-thresholding. IEEE Trans Inf Theory 1995; 41(3):613–27.

[18] Yang J, Wright J, Huang TS, Ma Y. Image super-resolution via sparse representation. IEEE Trans Image Process 2010;19(11):2861–73.

[19] Wright J, Yang AY, Ganesh A, Sastry SS, Ma Y. Robust face recognition via sparse representation. IEEE Trans Pattern Anal Mach Intell 2009;31(2):210–27.

[20] Xu L, Wang R, Zhang J, Yang Z, Deng J, Chen F, et al. Survey on sparsity in geometric modeling and processing. Graph Models 2015;82:160–80.

[21] Digne J, Chaine R, Valette S. Self-similarity for accurate compression of point sampled surfaces. In: EUROGRAPHICS 2014. Eurographics Association; 2014. p. 155–64.

[22] Avron H, Sharf A, Greif C, Cohen-Or D. ℓ_1-sparse reconstruction of sharp point set surfaces. ACM Trans Graph 2010;29(5):135.

[23] He L, Schaefer S. Mesh denoising via ℓ_0 minimization. ACM Trans Graph 2013; 32(4):64.

[24] Xu L, Lu C, Xu Y, Jia J. Image smoothing via ℓ_0 gradient minimization. ACM Trans Graph 2011;30(6):174.

[25] Wang R, Yang Z, Liu L, Deng J, Chen F. Decoupling noise and features via weighted ℓ_1-analysis compressed sensing. ACM Trans Graph 2014;33(2):18.

[26] Wang J, Oliveira MM. A hole-filling strategy for reconstruction of smooth surfaces in range images. In: XVI Brazilian symposium on computer graphics and image processing, 2003. SIBGRAPI 2003. IEEE; 2003. p. 11–8.

[27] Li X, Deng J, Chen F. Surface modeling with polynomial splines over hierarchical t-meshes. Vis Comput 2007;23(12):1027–33.

[28] Floater MS, Hormann K. Surface parameterization: a tutorial and survey. In: Advances in multiresolution for geometric modelling. Springer; 2005. p. 157–86.

[29] Sheffer A, Lévy B, Mogilnitsky M, Bogomyakov A. Abf++: fast and robust angle based flattening. ACM Trans Graph 2005;24(2):311–30.

[30] Zayer R, Lévy B, Seidel HP. et al. Linear angle based parameterization. In: Fifth eurographics symposium on geometry processing-SGP 2007. 2007. p. 135–141.

[31] Sander PV, Snyder J, Gortler SJ, Hoppe H. Texture mapping progressive meshes. In: Proceedings of the 28th annual conference on computer graphics and interactive techniques. ACM; 2001. p. 409–16.

[32] Maillot J, Yahia H, Verroust A. Interactive texture mapping. In: Proceedings of the 20th annual conference on computer graphics and interactive techniques. ACM; 1993. p. 27–34.

[33] Jin M, Wang Y, Yau ST, Gu X. Optimal global conformal surface parameterization. In: Visualization, 2004. IEEE. IEEE; 2004. p. 267–74.

[34] Ben-Chen M, Gotsman C, Bunin G. Conformal flattening by curvature prescription and metric scaling. In: Computer graphics forum, Vol. 27. Wiley Online Library; 2008. p. 449–58.

[35] Springborn B, Schröder P, Pinkall U. Conformal equivalence of triangle meshes. ACM Trans Graph 2008;27(3):77.

[36] Rogers DF. An introduction to NURBS, with historical perspective. Morgan Kaufmann; 2001.

[37] Ray N, Li WC, Lévy B, Sheffer A, Alliez P. Periodic global parameterization. ACM Trans Graph 2006;25(4):1460–85.

[38] Kälberer F, Nieser M, Polthier K. Quadcover-surface parameterization using branched coverings. In: Computer graphics forum, Vol. 26. Wiley Online Library; 2007. p. 375–84.

[39] Bommes D, Zimmer H, Kobbelt L. Mixed-integer quadrangulation. In: ACM transactions on graphics (TOG), Vol. 28. ACM; 2009. p. 77.

[40] Wang J, Oliveira MM. Filling holes on locally smooth surfaces reconstructed from point clouds. Image Vis Comput 2007;25(1):103–13.

[41] Liu Z, Pan M, Yang Z, Deng J. Recovery of sharp features in mesh models. Commun Math Stat 2015;3(2):263–83.

[42] Chen CY, Cheng KY. A sharpness-dependent filter for recovering sharp features in repaired 3D mesh models. IEEE Trans Vis Comput Graphics 2008;14(1): 200–12.

[43] Fleishman S, Drori I, Cohen-Or D. Bilateral mesh denoising. In: ACM transactions on graphics (TOG). ACM; 2003. p. 950–3.

[44] Fleishman S, Cohen-Or D, Silva CT. Robust moving least-squares fitting with sharp features. In: ACM transactions on graphics (TOG). ACM; 2005. p. 544–52.

[45] Jorge N, Stephen JW. Numerical optimization. USA: Springerverlang; 1999.

[46] Pati YC, Rezaiifar R, Krishnaprasad P. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In: 1993 conference record of the twenty-seventh asilomar conference on signals, systems and computers, 1993. IEEE; 1993. p. 40–4.

[47] Shanno DF. Conditioning of quasi-Newton methods for function minimization. Math Comp 1970;24(111):647–56.

[48] Eck a, DeRose T, Duchamp T, Hoppe H, Lounsbery M, Stuetzle W. Multiresolution analysis of arbitrary meshes. In: Proc. SIGGRAPH 95. ACM; 1995. p. 173–82.

[49] Hinton GE, Osindero S, Teh YW. A fast learning algorithm for deep belief nets. Neural Comput 2006;(7):1527–54.