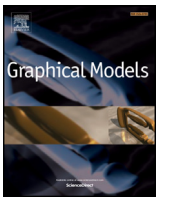
Graphical Models 000 (2015) 1–21

Contents lists available at ScienceDirect

Graphical Models

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



Survey on sparsity in geometric modeling and processing

Linlin Xu, Ruimin Wang, Juyong Zhang, Zhouwang Yang, Jiansong Deng, Falai Chen, Ligang Liu*

University of Science and Technology of China, Hefei, China

ARTICLE INFO

Article history:
Received 14 November 2014
Revised 18 June 2015
Accepted 18 June 2015
Available online xxx

Keywords:
Geometric processing
Sparse regularization
Dictionary learning
Low rank

ABSTRACT

Techniques from sparse representation have been successfully applied in many areas like digital image processing, computer vision and pattern recognition in the past ten years. However, sparsity based methods in geometric processing is far from popular than its applications in these areas. The main reason is that geometric signal is a two-dimensional manifold and its discrete representations are always irregular, which is different from signals like audio and image. Therefore, existing techniques cannot be directly extended to handle geometric models. Fortunately, sparse models are beginning to see significant success in many classical geometric processing problems like mesh denoising, point cloud compression, etc. This review paper highlights a few representative examples of how the interaction between sparsity based methods and geometric processing can enrich both fields, and raises a number of open questions for future study.

1. Introduction

Sparsity based regularization and sparse signal representation [1,2] have proven to be an extremely powerful tool for processing signals like audio, image and video. Sparse techniques have become state-of-the-art tools in many fields like machine learning [3,4], signal processing [5,6], neuroscience [7,8] and statistics [9–11]. This success is mainly due to the fact that important classes of signals such as audio and images have naturally sparse representations with respect to fixed bases (i.e., Fourier, Wavelet), or concatenations of such bases. Moreover, efficient and provably effective algorithms based on convex optimization or greedy pursuit are available for computing such representations with high fidelity [12].

While these successes in classical signal processing applications are inspiring, in geometric processing we are dealing with two-dimensional manifold signals with irregular domain, which is totally different from audio, image and video. One might justifiably wonder whether sparsity based regularization and sparse representation can be useful at all for geometric processing tasks. The answer has been largely positive: in the past few years, variations and extensions of ℓ_1 minimization have been applied to many geometric processing tasks, including mesh denoising [13–15], surface reconstruction [16], point cloud consolidation [17–21], mesh segmentation [22–25], and point cloud registration [26]. In almost all of these applications, using sparsity as a prior leads to state-of-the-art results.

Before going any further, we would like to briefly analyze the difference between using sparse techniques in geometry and in traditional fields. Sparse signal techniques have been successfully applied on many aspects as acquiring, representing and compressing high-dimensional signals. This is because that signals like audio and images can be sparsely represented by fixed basis like Fourier, Wavelet and Discrete Cosine Transform (DCT). Another important property of these signals is that they have a natural domain on which functions can be defined. For instance the domain of an audio is time or frequency and the domain of an image is a regular planar grid.

* Corresponding author. fax: 86 551 3600 985.
E-mail address: lgliu@ustc.edu.cn (L. Liu).

http://dx.doi.org/10.1016/j.gmod.2015.06.012
1524-0703/

Geometric signals usually consist of geometric positions and sometimes connection relationships. The connection relationships are represented by a 3D graph on which we can take geometric positions as sampling for a certain three dimensional functions. We can take these relationships as the domain of geometric signals. However these domains usually cannot embed onto a planar region and they are irregular compared to previous domains. Sometimes we cannot handle geometry directly and we have to transform the geometry into feature space which might be some Euclidean space. With these irregular domains another important issue is that how to define the basis. Besides, famous sparse related regularization terms for image processing like TV model assume that image is piece-wise constant. Geometric signals on the other hand are at least continuous. Thus most regularization terms cannot be directly employed on geometric problems.

Above all, applying sparse techniques on geometric signals generally faces the problems of handling irregular domain, defining basis functions and the geometric specified regularization terms. Fortunately, many creative researchers have found a lot of effective methods tackling these problems and successfully used sparse techniques to solve geometric problems as mentioned above. And the experiment results sufficiently show the advantages of sparse techniques, such as robustness to noise, local controllability and feature preserving.

In the rest of this paper, we would like to first introduce traditional sparse models (Section 2) used in previous fields like machine learning, computer vision, etc. Then according to the different sparse models, we classify all the papers into three parts (Sections 3–5) where we illustrate how these techniques are successfully applied on geometric processing problems. By giving a survey about the usage and the effectiveness of sparse techniques, we would like to achieve the goal of inspiring the researchers who are interested to discover more applications. In the end, Section 6 gives a summary and possible future works.

2. Preliminaries

Before illustrating how sparsity is applied on geometry processing problems, we would like to introduce some notations and general sparsity models.

2.1. Notation

To make this survey self-contained, here we introduce some basic notations. Let $\mathbf{x} = (x_1, x_2, \ldots, x_k)^T$ be any vector in Euclidean space $\mathbb{R}^k$, $\|\mathbf{x}\|_p$ denotes the ℓ_p norm of $\mathbf{x}$ with $\|\mathbf{x}\|_p = (\sum_{i=1}^k |x_i|^p)^{1/p}$. And the ℓ_0 pseudo-norm of $\mathbf{x}$ is defined as $\|\mathbf{x}\|_0 = \#\{i|x_i \neq 0\} = \sum_{i=1}^k |x_i|^0$. $\mathbf{M} = (m_{ij})$ represents a matrix in $\mathbb{R}^{m\times n}$. Its frobenius norm is defined as $\|\mathbf{M}\|_F = (\sum_{i=1}^m \sum_{i=1}^n m_{ij}^2)^{1/2}$, and its nuclear norm is defined as $\|\mathbf{M}\|_* = \sum_i \sigma_i(\mathbf{M})$ where $\sigma_i(\mathbf{M})$ is the ith singular value of $\mathbf{M}$. Nuclear norm is the convex envelope of rank($\mathbf{M}$), which makes that $\|\mathbf{M}\|_*$ can be considered the relaxation of the rank of $\mathbf{M}$.

2.2. Sparse techniques

Generally, there are some basic assumptions in sparse techniques. For example, a signal can be represented by a sparse linear combination of dictionary elements, some special signals can be approximated by a low rank matrix. In the following, we will discuss these essential issues and general models raised in this field.

2.2.1. Sparsity in vector

A vector signal $\mathbf{u} = (u_1, u_2, \ldots, u_n)^T \in \mathbb{R}^n$ can be approximated by a linear combination of dictionary elements $\{\mathbf{d}_i \in \mathbb{R}^n\}_{i=1}^m$, which can be formulated as

$$\mathbf{u} \approx \sum_{i=1}^{m} x_i \mathbf{d}_i, \tag{1}$$

where $\mathbf{x} = (x_1, x_2, \ldots, x_m)^T$ is the coefficient vector. If $\mathbf{x}$ is sparse, it means that signal $\mathbf{u}$ can be represented by a linear combination of few dictionary elements. However there are also different explanations about sparsity that the vector becomes sparse under a certain transformation. For instance, the gradient of natural image is always sparse, and total variation model catches this observation well. In the following, we will give three general models which are quite popular in signal processing.

Sparse coding. Sparse coding method is widely used in computer vision tasks like face recognition, image classification. It assumes that the input signal can be sparsely represented by a set of dictionary elements. The target of sparse coding is to pursuit the sparse coefficient vector $\mathbf{x}$. The formulation is

$$\min_{\mathbf{x}} \frac{\lambda}{2} \|\mathbf{u} - \mathbf{D}\mathbf{x}\|_2^2 + \|\mathbf{x}\|_p, \tag{2}$$

where $0 \leq p \leq 1$. If $p = 0$, $\|\mathbf{x}\|_p$ is equivalent to the number of non-zero elements. However $\|\mathbf{x}\|_0$ is a nonconvex norm such that it is quite hard to obtain the optimal result and most methods use greedy strategy to get an approximation result [27,28]. On the other hand, Eq. (2) would be a convex problem if we set $p = 1$, and it is the well-known least absolute shrinkage and selection operator (LASSO) [29]. The relationship of the ℓ_1 relaxation and its origin sparse ℓ_0 model is an open problem and [30] proves that under certain conditions the results are equivalent. Recently researchers develop algorithms solving (2) when $0 < p < 1$ which also approximates the sparse solution. As shown in Fig. 1, the iso-level curve of $\|\mathbf{x}\|_p = 1$ concentrates toward axes with p decreasing, and thus model (2) returns more sparse result with smaller p value.

As shown in Fig. 2, the input signal is a smooth curve with random noise, and we reconstruct the curve with sparse coding formulation Eq. (2) with DCT as the dictionary. As shown in the second row, we can use few dictionary elements to approximate the signal with the help of ℓ_1 norm on the coefficient vector $\mathbf{x}$. Sparse coding has been applied on many kinds of problems [2] as face recognition [31], image super-resolution [32], image classification [33].

Dictionary learning. As discussed above, the problem of sparse coding focuses on the searching of sparse coefficient vector $\mathbf{x}$. And popular basis functions or vectors are

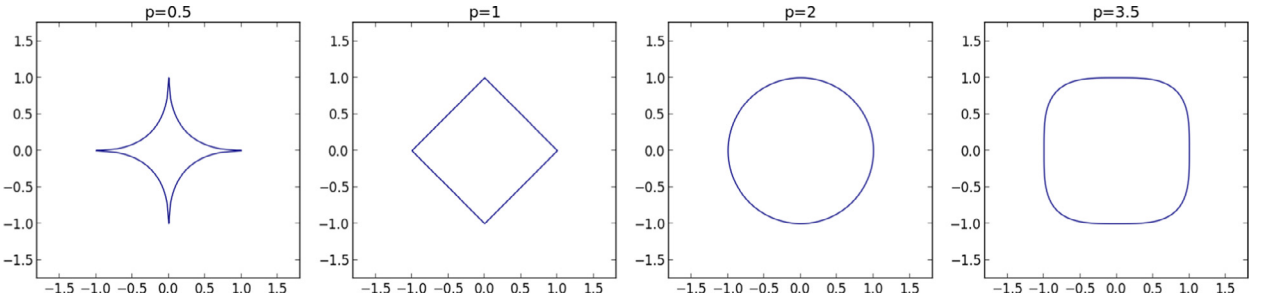


Fig. 1. Four unit ℓ_p-balls with different values of p in $\mathbb{R}^2$.

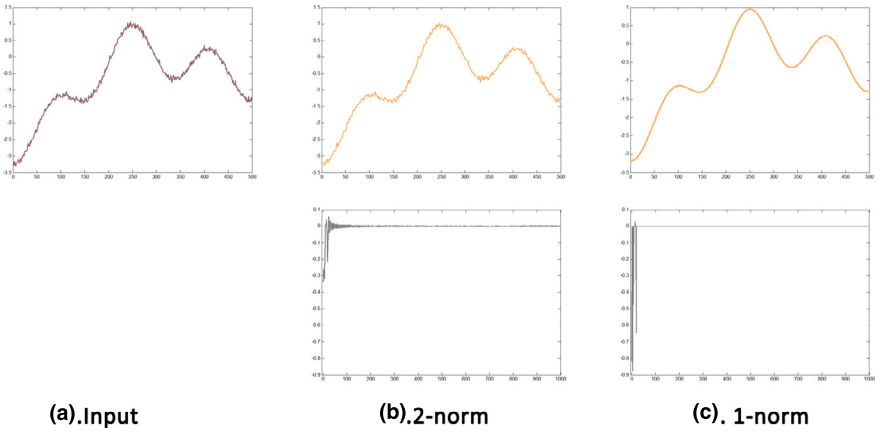


Fig. 2. Signal denoising via sparse coding formulation Eq. (2). The second column is the result with $p = 2$, and the third column is the result with $p = 1$. The second row is the distribution of coefficients $\mathbf{x}$.

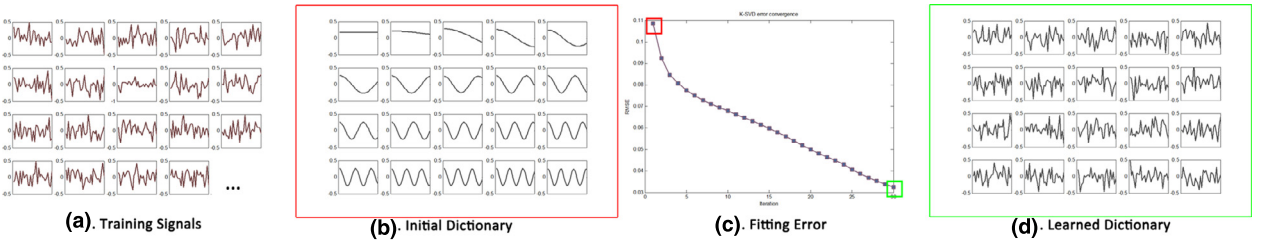


Fig. 3. Dictionary learning method (3) results in the most representative dictionary (d) with satisfactory fitting error shown with the green rectangle in (c). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

considered like Wavelet [5], DCT [34], and Fourier transform [35]. Every given input vector is solved separately and the underlying pattern of these vectors is ignored. Therefore a natural idea is whether it is possible to construct a special dictionary that consists of basis vectors and then algorithms of dictionary learning are invented ever since [36]. These algorithms attempt to optimize the dictionary and the coefficients at the same time. Here dictionary $\mathbf{D}$ stands for the set of dictionary elements which is $\{\mathbf{d}_i\}_{i=1}^m$ (Fig. 3(b)(d)) and $\mathbf{D}$ is represented as a matrix $\mathbf{D} = (\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_m)$. Assume that we are given a set of input signals $\{\mathbf{f}_j\}_{j=1}^k$ which are represented by $\mathbf{D}$ with coefficient $\{\mathbf{x}_j\}_{j=1}^k$ and energy function for these vectors is specified as

$$E(\mathbf{X}, \mathbf{D}) = \frac{\lambda}{2} \sum_{j=1}^{k} \|\mathbf{f}_j - \mathbf{D}\mathbf{x}_j\|_2^2 + \|\mathbf{x}_j\|_p = \frac{\lambda}{2} \|\mathbf{F} - \mathbf{D}\mathbf{X}\|_F^2 + \|\mathbf{X}\|_{p,1}, \quad (3)$$

where $\mathbf{F} = (\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_k)$ and $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k)$. Compared to sparse coding problems, these models consider a learning step of dictionary, so sparse coding can be regarded as a special case of dictionary learning. Given an initial dictionary (Fig. 3(b)) whose fitting error is relative large as shown with the red rectangle in Fig. 3(c), the most representative dictionary (Fig. 3(d)) with satisfactory error in the green rectangle will be learned with some iterative algorithm.

Sparse regularization and fitting. In both situations discussed above, we assume that the input signal can be sparsely represented by a few dictionary elements. Except sparse representation, another kind of sparsity is that the signal is sparse under certain kinds of transformation. Mathematically $\mathbf{u}$ may not be sparse but $\mathbf{Tu}$ is sparse where $\mathbf{T}$ corresponds to some transformation T. As shown in Fig. 4(a), the values of the original signal is not sparse. However, as this signal is piecewise constant, its gradient (Fig. 4(b)) is sparse. Based on this observation, we aim at recovering the signal $\mathbf{u}$ from the

Please cite this article as: L. Xu et al., Survey on sparsity in geometry modeling and processing, Graphical Models (2015), http://dx.doi.org/10.1016/j.gmod.2015.06.012

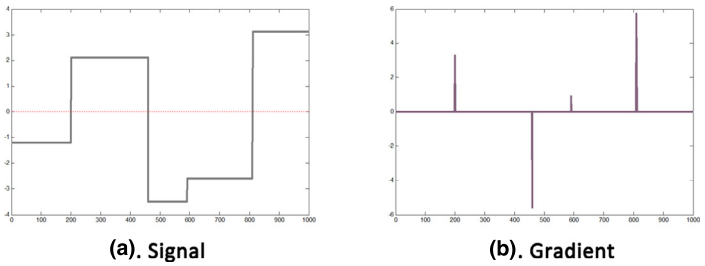


Fig. 4. Original signal and its gradient.

observed (input) signal $\mathbf{f}$ via the following model:

$$\min_{\mathbf{u}} \frac{\lambda}{2} F(\mathbf{u}, \mathbf{f}) + \|\mathbf{Tu}\|_p, \tag{4}$$

where $F(\mathbf{u}, \mathbf{f})$ is the fitting term which measures the distance between the restored signal and the input signal. As for the transform T, it depends on the type of signals. For example, in total variation model [37,38], $T = \nabla$ as the gradient of natural image is sparse

$$\min_{\mathbf{u}} \frac{\lambda}{2} F(\mathbf{u}, \mathbf{f}) + R(\nabla \mathbf{u}),$$

where $R(\nabla \mathbf{u}) = \sum_{i,j} |\nabla \mathbf{u}(i, j)|$ demonstrates that the ℓ_1 norm of the gradient of output image. So far many variations of TV model have been introduced like adding mixed regularization [39] directly using the ℓ_0 norm of gradient[40] or using penalty method to solve the problem.

If the input signal is corrupted by impulsive noises or outliers, the least square formulation of fitting term always returns unsatisfactory results. To avoid the influence of impulsive or outliers, a better way is to introduce sparsity measurement into the fitting term like

$$F(\mathbf{u}, \mathbf{f}) = \|\mathbf{u} - \mathbf{f}\|_p, \quad p \in [0, 1].$$

It has been successfully applied in image restoration with noise types like impulsive noise, salt-and-pepper noise [41]. As shown in Fig. 5(a), the sampled points from a circle contains a lot of noises and outliers. To restore the signal, we let regularization term be $\|\mathbf{Lu}\|_2^2$ where $\mathbf{L}$ represents the discrete Laplacian operator. The reconstructed curve by ℓ_2 norm fitting term is very jaggy because of the influence of outliers, while the reconstructed result by ℓ_1 norm fitting term is quite close to the clean circle benefitted by its sparse property.

Numerical solution. So far we have introduced the sparse models that have been widely used and now we would like to show general approaches for solving them. For sparse coding problem that $p = 0$ in (2)

$$\min_{\mathbf{x}} \frac{\lambda}{2} \|\mathbf{u} - \mathbf{Dx}\|_2^2 + \|\mathbf{x}\|_0. \tag{5}$$

This problem is NP-hard and optimal solution is unable to be obtained in polynomial time. Approximating approaches like matching pursuit (MP) [27], orthogonal matching pursuit (OMP) [28,42] greedily and iteratively add the currently best basis vector to represent $\mathbf{u}$. And Pati el al. [28] prove that the solution of OMP algorithm converges under certain conditions. If $p = 1$, problem (2) is a convex problem which can be easily solved by convex optimization algorithms like quadratic programming [43]. However, specially designed method is also widely used, for instance [44] is used for solving LASSO models.

Classical dictionary learning techniques [36,45] consider a training set of signals or data $\mathbf{F} = [\mathbf{f}_1, \ldots, \mathbf{f}_n]$ and optimize the empirical cost function

$$f_n(\mathbf{D}) \triangleq \frac{1}{n} \sum_{i=1}^{n} c(\mathbf{f}_i, \mathbf{D}), \tag{6}$$

where $\mathbf{D}$ in $\mathbb{R}^{k \times m}$ is the dictionary, and $c(\mathbf{f}, \mathbf{D})$ is a loss function that $c(\mathbf{f}, \mathbf{D})$ should be small if $\mathbf{D}$ represents the signal $\mathbf{f}$ well in a sparse fashion. However K-SVD algorithm [46] specifies the dictionary learning problem as

$$\min_{\mathbf{D},\mathbf{X}} \|\mathbf{F} - \mathbf{DX}\|_F^2,$$
$$\text{s.t.} \|\mathbf{x}_i\|_0 \le s, \quad \forall i = 1, 2, \ldots, m, \tag{7}$$

and iteratively alternates between a process of updating dictionary $\mathbf{D}$ to better fit the training data and sparse coding of $\mathbf{X}$ based on current dictionary, just like the traditional processing method of separating difficult problems into easier parts.

The problem of sparse regularization (4) when $p = 1$ is convex and a lot of approaches have been invented to solve it. Alternating direction method of multipliers [47,48] introduces one auxiliary variable $\mathbf{v}$ to represent $\mathbf{Tu}$ with constraint that $\mathbf{v} = \mathbf{Tu}$. Then an iterative strategy is used to update each variable with a Lagrangian multiplier. For $p = 0$, Xu et al. [40] introduce a variation of penalty method [43] to solve (4) without any convergency guarantee.

2.2.2. Low rank

Low rank representation attempts to decompose an input matrix $\mathbf{M} \in \mathbb{R}^{m \times n}$ into a low rank matrix $\mathbf{L}$ and a residual matrix $\mathbf{S}$. The residual matrix $\mathbf{S}$ may have some specific properties like the input is corrupted by Gaussian noise or sparse noise. Under the assumption that $\mathbf{M}$ is corrupted by Gaussian noise, a low rank problem is formulated as:

$$\min_{\mathbf{L},\mathbf{S}} \|\mathbf{S}\|_F$$
$$\text{s.t. } \text{rank}(\mathbf{L}) \le r$$
$$\mathbf{M} = \mathbf{L} + \mathbf{S}, \tag{8}$$

where $r \ll \min(m, n)$. The above problem is equivalent to principle component analysis (PCA) according to [49,50]. This problem is easily solved by first computing the singular value decomposition (SVD) of $\mathbf{M}$ and then projecting the columns of $\mathbf{M}$ onto the subspace spanned by the r principle left singular vectors.

Please cite this article as: L. Xu et al., Survey on sparsity in geometry modeling and processing, Graphical Models (2015), http://dx.doi.org/10.1016/j.gmod.2015.06.012

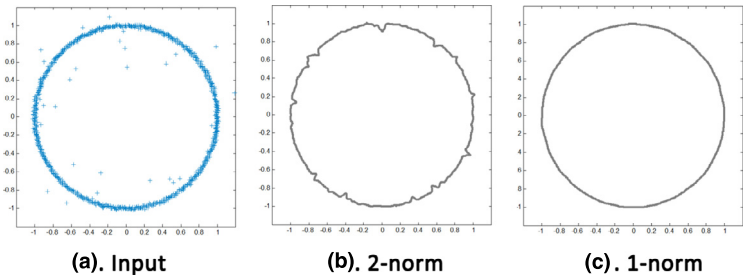


Fig. 5. Curve fitting with different type fitting term.

However, PCA assumes that corruption is caused by Gaussian noise. The result of PCA might be far from $\mathbf{M}$ if only a few entries of $\mathbf{M}$ is corrupted. Refs. [31,51] show that under some conditions that the residual matrix $\mathbf{S}$ is sparse, one can exactly recover $\mathbf{M}$ by solving

$$\min_{\mathbf{L},\mathbf{S}} \|\mathbf{L}\|_* + \lambda\|\mathbf{S}\|_1, \quad \text{s.t. } \mathbf{M} = \mathbf{L} + \mathbf{S}. \tag{9}$$

The formulation is obtained by relaxing following problem replacing the matrix rank with nuclear norm and the ℓ_0 norm with the ℓ_1 norm:

$$\min_{\mathbf{L},\mathbf{S}} \text{rank}(\mathbf{L}) + \lambda\|\mathbf{S}\|_0, \quad \text{s.t. } \mathbf{M} = \mathbf{L} + \mathbf{S}. \tag{10}$$

Refs. [31,52] show the uniqueness of the solution and [53] discusses efficient algorithm for solving low rank problem. Generally, problem (9) can be treated as a general convex optimization problem and solved by any off-the-shelf interior point solver (like CVX [54]).

Numerical solution. There have been several popular ways to solve low-rank (robust PCA) problem (9): The iterative thresholding approach introduced in [31] solves a relaxed convex problem of (9); the accelerated proximal gradient approach [55] is applied to a relaxed version of RPCA problem; Lin et al. [56] tackle the problem via its dual; the method of augmented Lagrange multipliers [53] is introduced to efficiently solve RPCA problem.

Here we take a closer look at how augmented Lagrange multipliers (ALM) is applied on solving (9). According to [57], the general approach of augmented Lagrange multipliers is utilized to solve constrained optimization problems:

$$\min f(\mathbf{x}), \quad \text{s.t.} h(\mathbf{x}) = 0, \tag{11}$$

here f: $R^n \to R$ and h: $R^n \to R^m$. The augmented Lagrangian function is defined as:

$$\text{Lag}(\mathbf{x}, \Lambda, \mu) = f(\mathbf{x}) + < \Lambda, h(\mathbf{x}) > + \frac{\mu}{2}\|h(\mathbf{x})\|_F^2. \tag{12}$$

Then $\mathbf{x}$ and Λ are updated iteratively solving sub-problems (see [58] for more details). Thus the augmented Lagrangian function of (9) is:

$$\text{Lag}(\mathbf{L}, \mathbf{S}, \Lambda, \mu) = \|\mathbf{L}\|_* + \lambda\|\mathbf{S}\|_1 + < \Lambda, \mathbf{M} - \mathbf{L} - \mathbf{S} > + \frac{\mu}{2}\|\mathbf{M} - \mathbf{L} - \mathbf{S}\|_2^2. \tag{13}$$

Then $\mathbf{L}$, $\mathbf{S}$ and Λ are separately updated by fixing other variables and minimizing augmented Lagrangian function [53].

Low rank representation or robust principle component analysis has been successfully applied onto different problems like face recognition [59], latent object detection [60], video denoising [61].

In summary, there are mainly three types of sparse techniques: sparse regularization and fitting, dictionary learning and low rank, according to which we organized our paper. Certainly, not all the papers have the same formulations as models (3) (4) (8), we will give some directions making all the papers align with them. After introducing the notations and with the basic knowledge of the general models, we will review the usage of sparsity in geometric modeling and processing. Table 1 summaries the effectiveness of sparsity in all the papers. We can see that many geometric processing problems achieve state-of-art performance with sparsity tools.

3. Sparse regularization and fitting

From (4), it is clear that sparse regularization and fitting can be classified into two types:

- Imposing sparsity induced norm on the regularization term. The main goal is to design the suitable transformation T for the problems. This type of sparse regularization is widely used in mesh denoising to preserve sharp features, shape matching and deformation to obtain locality.
- Imposing sparsity based measurement in the fitting term. This type of fitting term works quite well for the dataset with noises and outliers.

In the following, we will give the details on how sparse regularization and fitting are applied to geometric processing problems.

3.1. Mesh denoising

Mesh denoising, just as the name implies, attempts to remove noises from the 3D mesh objects. How to distinguish features from noises has been a challenging problem all the time.

(1). In image processing, Xu et al. [40], aiming to smooth images, provide an algorithm for directly optimizing the ℓ_0 norm of gradients of image colors to create piecewise constant images corresponding to the case that $p = 0$ and $T = \nabla$ in (4). Let $\mathbf{c}$ be a vector of pixel colors and $\nabla\mathbf{c}$ be a vector of

Table 1
An overview about the effectiveness of sparsity in all the papers. (SR-Sparse Regularization, DL-Dictionary Learning, LR-Low Rank).

Applications	Methods			Effectiveness of sparsity
	SR	DL	LR	
Point cloud consolidation				
LOP [17]	⋆			Robust to noises, outliers
WLOP [18]	⋆			Robust to noises, outliers
CLOP [19]	⋆			Robust to noises, outliers
TV(ℓ_1) based/ [20]	⋆			Robust to noises, outliers
Subdivision [21]	⋆			Robust to noises, outliers
Mesh denoising				
ℓ_0-norm of edge operator [13]	⋆			Sharp feature preserving
ℓ_1-analysis compressed sensing [14]	⋆			Sharp feature preserving
TV(ℓ_1) based [15]	⋆			Sharp feature preserving
Shape matching				
Rigid [26]	⋆			Robust to noises, outliers
Non-rigid [62]	⋆			Constraining invariance of intrinsic properties
Co-matching [63]	⋆			Cycle-consistent with theoretical guarantee
Segmentation				
Single mesh [22]	⋆			Improving smoothness of decomposition boundaries
Mesh animation [23]	⋆			Versatile with the local controllability
Co-segmentation [24]	⋆			Dimension reduction
Labeling [25]			⋆	Dimension reduction, correctness improvement
Deformation				
SSDR [64]		⋆		Low memory, robust to noise
Blend skinning compression [65]		⋆		Large reduction of computational cost
Skeletal rigging [66]		⋆		Low memory
Constrained mesh editing [67]	⋆			Local controllability
ℓ_p Deformation [68]	⋆			Control distortion distribution/local controllability
Upright orientation				
Low rank [69]			⋆	Intuitive, relatively robust
Tensor rank [70]			⋆	Capturing global symmetry, relatively robust
Other applications				
CMM [71]	⋆			Local controllability
LBC [72]	⋆			Local controllability
Skeleton extraction [73]	⋆			Robust to noise and outlier
3D Printing [74]	⋆			Reduce the material largely
LRSCPK [75]			⋆	Sharp feature preserving
Point cloud compression [76]		⋆		High compression ratio, robust to noise
Reconstruction [16]		⋆		Sharp feature preserving, robust to noise and outlier, and unifying geometry and connectivity

gradients of these colors. They formulate the smooth problem as

$$\min_{\mathbf{c}} |\mathbf{c} - \mathbf{c}^*|^2 + |\nabla \mathbf{c}|_0,$$

where $\mathbf{c}^*$ represents the original image colors to provide a data fidelity term.

A natural extension to triangulated meshes is to design a discrete differential operator to replace $\nabla \mathbf{c}$ that is zero when the surface is flat for arbitrary triangulations. So some form of second order information rather than the first order information provided by $\nabla \mathbf{c}$ is needed. He et al. [13] generalize the construction of the vertex-based cotan operator in discrete Laplacian operator [77] to an operator that acts directly on an edge

$$T(e) := \begin{bmatrix} -\cot(\theta_{2,3,1}) - \cot(\theta_{1,3,4}) \\ \cot(\theta_{2,3,1}) + \cot(\theta_{3,1,2}) \\ -\cot(\theta_{3,1,2}) - \cot(\theta_{4,1,3}) \\ \cot(\theta_{1,3,4}) + \cot(\theta_{4,1,3}) \end{bmatrix}^T \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix}, \tag{14}$$

where the notation is shown in Fig. 6(a) and this operator is also expanded to get a new area-based edge operator. Fig. 6(b) gives one denoised result with sharp features.

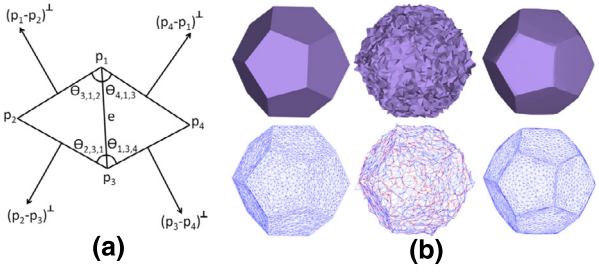


Fig. 6. Sparse regularization: mesh denoising [13]. (a) The notation for the one-ring of an edge. (b) Left: initial surface. Center: surface corrupted by Gaussian noise in random directions. Right: denoising result. The wireframe shows folded triangles as red edges. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

(2). Instead, Zhang et al. [15] adopt the sparsity of face normals differences and propose a two-phase method including *face* normal filtering and vertex updating. They filter face normals with a TV based variational denoising method based on another kind of differential operators on triangulated

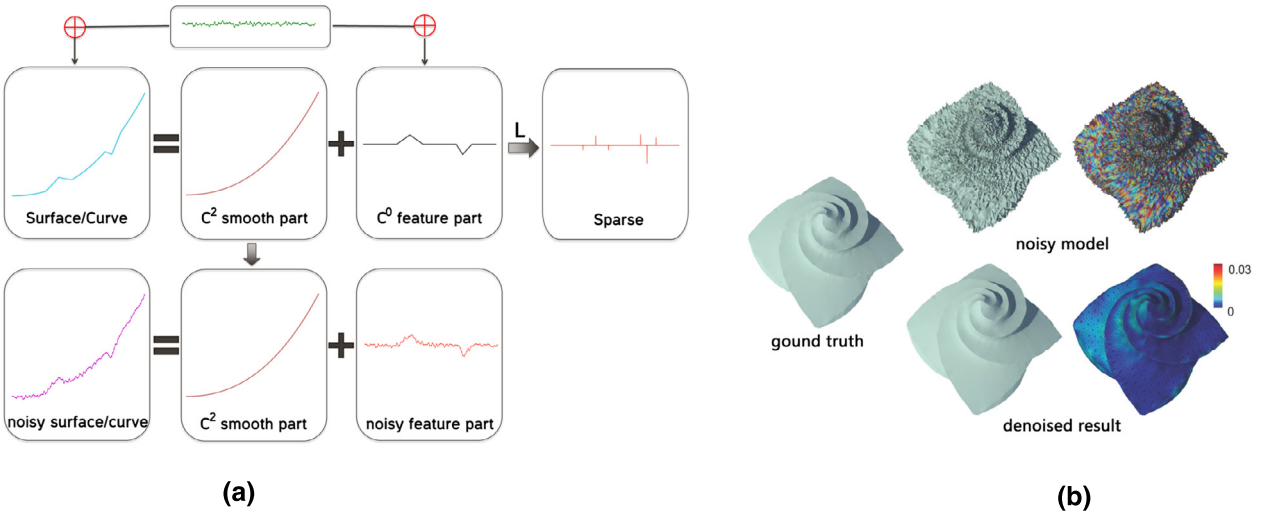


Fig. 7. Sparse regularization: mesh denoising [14]. (a) is the two-dimensional illustration for their key observation. (b) is a denoising example.

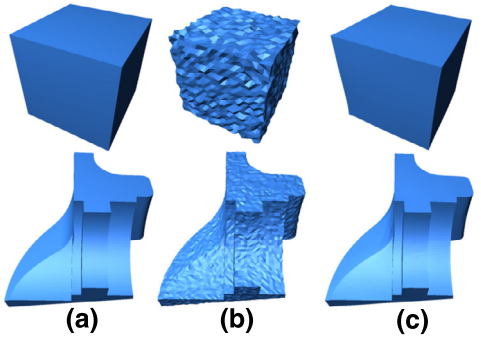


Fig. 8. Sparse regularization: TV based mesh denoising [15]. (a) Clean meshes. (b) Noisy mesh(Gaussian noise). (c) Denoising result.

surfaces

$$T(\mathbf{n}|_e) = w_e l_e \sqrt{\sum_{i=1}^{3} |[n_i]_e|^2}, \tag{15}$$

where, w_e and l_e are scalar parameters, $[n_i]_e$ is the gradient operator of normals $\{\mathbf{n}\}$ defined on edge e

$$[n_i]_e = \begin{cases} \sum_{e \prec \tau} n_i|_\tau \mathrm{sgn}(e,\tau), & e \not\subseteq \partial M, \\ 0, & e \subseteq \partial M. \end{cases}$$

Here, $e \prec \tau$ indicates that e is an edge of triangle τ, M is the triangulated surface and $\mathrm{sgn}(e,\tau) = 1$ indicates that the orientation of e is consistent with the orientation of τ, otherwise -1.

With $\{\mathbf{n}\}$ as the optimization object, minimizing the ℓ_1 norm of the new operator (4) along all edges iteratively and updating vertex using previous method, the denoising results can be obtained as shown in Fig. 8.

Like many other optimization problem, the optimal values of the parameters like λ in (4), are given by the experimental data and there is of course no theoretical convergence guarantee. Thus how to tune this kind of parameters is an interesting and meaningful work that has not been well addressed. So the next work of Wang et al. [14] tries to improve this situation also with a two-phase approach for decoupling features and noises on discrete surfaces.

(3). Fig. 7 (a) gives a planar curve as the illustration for their key observation: any surface is piecewise C^2, that is, a surface consists of two parts: C^2 smooth part and C^0 feature part which can be transformed into a sparse signal by applying the Laplacian operator. Here C^2 smooth part means the part is at least C^2 smooth. As such, the denoising problem is divided into two phase: smooth part (base mesh) estimation and features recovering from the corrupted feature part.

They first get a base mesh by denoising the input data using a global Laplacian regularization smoothing optimization, in which the smoothness parameter is automatically chosen by adopting the generalized cross-validation scheme, then decouple the features $\mathbf{u}$ and noises simultaneously from the noisy feature part $\mathbf{f}$ via the ℓ_1 analysis compressed sensing optimization

$$\min_{\mathbf{u}} \|\mathbf{L}\mathbf{u}\|_1, \text{ s.t.} \|\mathbf{f} - \mathbf{u}\|_2 \le \epsilon, \tag{16}$$

that is, $\mathbf{T} = \mathbf{L}$, $p = 1$ in Eq. (4), here $\mathbf{L}$ is the Laplacian matrix. Finally, combining the denoised feature part and the obtained base mesh reduces the final denoising result.

Note that it is the first time noise and features are analyzed and separated in such an elegant manner with guarantees by statistical theory which is much exciting and sightworthy in the smoothing optimization.

3.2. Shape matching

Shape matching aims at finding the set of corresponding points on source and target point set as shown in Fig. 9. Rigid registration is a fundamental task in geometric processing where the two data sets are transformed globally with a rigid transformation, generally the main goal is to deal with the data artifacts like outliers and noises. But in non-rigid registration, the intrinsic information of geometric data should be considered which directly brings more challenges.

Please cite this article as: L. Xu et al., Survey on sparsity in geometry modeling and processing, Graphical Models (2015), http://dx.doi.org/10.1016/j.gmod.2015.06.012

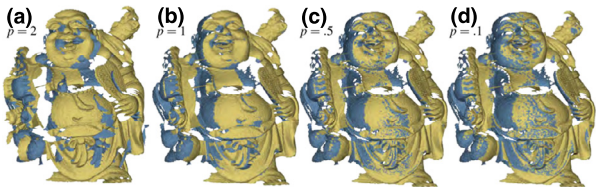


Fig. 9. Sparse regularization: rigid registration results using sparse ICP [26] under different ℓ_p norms.

3.2.1. Rigid registration

For rigid registration, the classical algorithm *Iterative Closest Point* (ICP) addresses it by assuming the input data to be in coarse alignment. Under this assumption, a set of correspondences can be obtained by querying closest points on the target geometry. Given two surfaces $\mathcal{X}$, $\mathcal{Y}$, it is formulated as

$$\underset{\mathbf{R},\mathbf{t}}{\operatorname{argmin}} \int_{\mathcal{X}} F(\mathbf{R}\mathbf{x}+\mathbf{t}, \mathcal{Y})d\mathbf{x} + I_{\mathcal{SO}(k)}(\mathbf{R}),$$

where $\mathbf{x}$ is the point on the source geometry, $\mathbf{t}$ is a translation vector, $\mathbf{R}$ is a rotation matrix which is constrained using $I_{\mathcal{SO}(k)}(\mathbf{R})$ to get the rigidity of the transformation. Classical ICP is in a least-square sense with the metric $F(\mathbf{u},\mathbf{f}) = \|\mathbf{u}-\mathbf{f}\|_2^2$ which would fail with outliers.

Now that sparse regularization methods excel in processing data set with noises or outliers, Bouaziz et al. [26] try to formulate the local alignment problem as recovering rigid transformation that minimizes the number of zero distances between two correspondences. Just like the fitting term $F(\mathbf{u},\mathbf{f})$ in (4), they adopt $\ell_p(0 \le p \le 1)$ norm to obtain an heuristic-free, robust rigid registration algorithm by modifying

$$F(\mathbf{u},\mathbf{f}) = \|\mathbf{u}-\mathbf{f}\|_2^p. \tag{17}$$

About ℓ_p norm, Chartrand et al. [78] show that ℓ_p norms with $p < 1$ outperform the ℓ_1 norm in inducing sparsity and Elad [1] also illustrates the tendency of ℓ_p $(0 < p < 1)$ norm to drive results to become sparse. Fig. 9 is the registration results of sparse ICP under different values of p among which it can be found that $0 < p < 1$ induces better results, but the value of p is selected according to the experiments to offer a trade-off between performance and robustness which may make the sparse ICP unpractical.

3.2.2. Non-rigid shape matching

(1) Local functional basis. For a 3D surface, the invariance of intrinsic properties to extrinsic transformations should always be handled. The eigenfunctions of the Laplace–Beltrami operator just define this kind of basis called manifold harmonic basis (MHB), which is unique and characteristic of the geometric and topological properties of the shape. Now we first have a look at one work about it, followed by a closely related non-rigid shape matching algorithm.

The Laplace–Beltrami operator $\triangle$ on a 2D manifold surface embedded in 3D space induces the eigenfunctions $\{\phi_k\}$ satisfying the equations

$$-\triangle\phi_k = \lambda_k\phi_k, k \in \mathbb{N}, \quad \lambda_k \in \mathbb{R}, \tag{18}$$

where λ_k are the eigenvalues of the operator. With their *global* spatial support, MHB has been used for many applications.

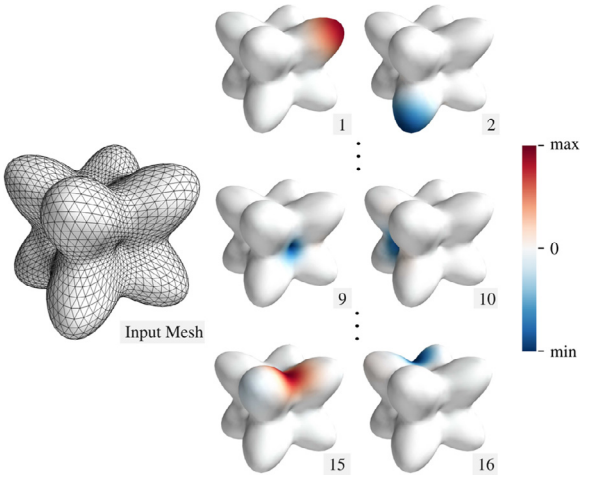


Fig. 10. Sparse regularization: local functional basis [71]. The proposed compressed manifold modes (CMMs) have local support and are confined to specific local features like protrusions and ridges. 8 of the CMMs were found for the 8 protrusion at the corner(2 shown here), 6 concentrate at each of the dents (2 shown here), and 12 CMMs automatically form at the valleys between the protrusions.

But as we have known that many times *locality* may induce better result, like deformation [72] in Section 3.5.1. To produce an intrinsic shape basis with local spatial support and take advantage of MHB, Neumann et al. [71] propose the *compressed manifold basis* (CMB) obtained with a sparsity inducing ℓ_1 norm of the individual basis functions $\{\phi_k\}$ as following:

$$\min_{\phi_k} \sum_{k=1} \langle\phi_k, \triangle\phi_k\rangle + \mu|\phi_k|_1,$$
$$\text{s.t. } \langle\phi_k, \phi_j\rangle = \delta_{kj}. \tag{19}$$

Here, with the same optimization problem (4), the sparsity is directly defined on the basis functions.

For a triangle mesh, discretizing the Laplacian $\triangle$ using a sparse matrix $\mathbf{L}$ with cotangent weights defined in previous work, and incorporating a lumped mass matrix $\mathbf{M}$ that contains the vertex areas along its diagonal making the eigenbasis independent of the mesh resolution, the discretization of (19) becomes

$$\min_{\mathbf{\Phi}} \operatorname{Tr}(\mathbf{\Phi}^T\mathbf{L}\mathbf{\Phi}) + \mu\|\mathbf{\Phi}\|_1,$$
$$\text{s.t. } \mathbf{\Phi}^T\mathbf{M}\mathbf{\Phi} = \mathbf{I}, \tag{20}$$

where $\mathbf{\Phi}$ contains the first several eigenvectors. Solving (20), the obtained orthogonal CMMs could automatically identify key shape features of the underlying mesh as shown in Fig. 10. As such, it can be used for shape matching which involves robust feature detection.

(2) Non-rigid matching. Matching of deformable shapes is a notoriously difficult problem which results in the number of degrees of freedom growing exponentially with the number of matched points.

Recently, Ovsjanikov et al. [79] introduce a functional representation for correspondences which are modeled as the correspondences between functions on two shapes rather

Please cite this article as: L. Xu et al., Survey on sparsity in geometry modeling and processing, Graphical Models (2015), http://dx.doi.org/10.1016/j.gmod.2015.06.012

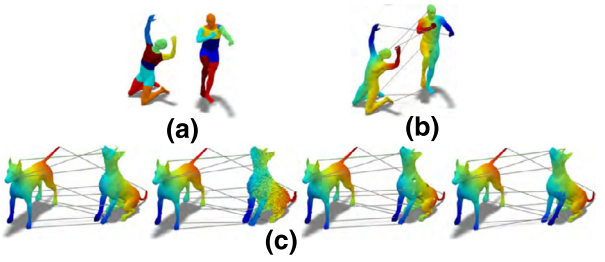


Fig. 11. Sparse regularization: non-rigid shape matching [62]. (a) The regions detected for two non-isometric shapes. (b) The final point-to-point correspondences result of shapes in (a). (c) More correspondence results between SHREC shapes undergoing nearly isometric deformations and noise.

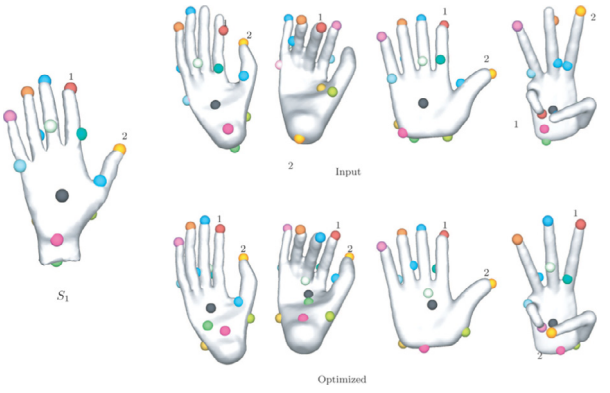


Fig. 12. Sparse regularization: co-matching [63]. Comparison between the input maps and the optimized maps from a source shape to selected target shapes shows the consistent correspondences across the shapes.

than points. They encode the correspondences with one matrix $\mathbf{C}$ that satisfies $\mathbf{b}^{\mathbf{T}} = \mathbf{a}^{\mathbf{T}}\mathbf{C}$, where $\mathbf{a}, \mathbf{b}$ are the linear combination coefficient vectors for the functions under the respective bases.

If two shapes are isometric and the bases are constructed with the discretized eigenfunctions of the Laplace–Beltrami operator introduced in above work, due to the intrinsic property, every low-distortion correspondence can then be represented by a nearly diagonal, and very sparse matrix $\mathbf{C}$.

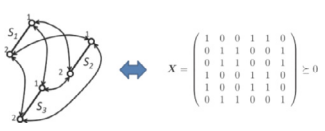
Based on this observation, Pokrass et al. [62] first get two collections of similar functions $\{f_i\}$ and $\{g_j\}$ for shape X and Y using some region detection process like [80]. As shown in Fig. 11(a), different colors represent different functions, but it is unknown to which g_j in Y a f_i in X corresponds. Thus the unknown permutation matrix $\mathbf{\Pi}$ is adopted to express this ordering, finally the robust permuted sparse coding is formulated as following:

$$\min_{\mathbf{C},\mathbf{O},\mathbf{\Pi}} \frac{1}{2}\|\mathbf{\Pi B} - \mathbf{AC} - \mathbf{O}\|_F^2 + \lambda\|\mathbf{W}\odot\mathbf{C}\|_1 + \mu\|\mathbf{O}\|_{2,1}, \qquad (21)$$

where $\mathbf{T} = \mathbf{W}$ with $p = 1$ is assigned with larger weights in off-diagonal part and small weights in diagonal part to promote diagonal solutions, $\odot$ denotes element-wise multiplication, $\|\mathbf{O}\|_{2,1}$ is for the case that the rows of $\mathbf{A}$ having no corresponding rows in $\mathbf{B}$. Solving (21), the final point-to-point correspondence result will be obtained (Fig. 11(b)) and Fig. 11(c) shows more results undergoing nearly isometric deformations and noise. From the formulation we know that this method relies on the region detection technique and assumption: near-isometric shapes.

3.2.3. Co-matching

Here, we use *co-matching* to denote the consistent shape maps in a collection of shapes. Based on the fact that compositions of maps along a cycle of shapes could approximately the identity map, Huang et al. [63] represent the point-to-point map as a binary matrix $\mathbf{X}$ which stores all the maps in blocks between two shapes as shown in the right figure



from the paper where S_i is an input shape, then they formulate the problem of estimating the cycle-consistent maps as finding s closest positive semidefinite matrix $\mathbf{X}$ to an input initial map matrix $\mathbf{X}^{in}$

$$\min_{\mathbf{X}} \sum_{(i,j)\in\mathcal{E}} \|\mathbf{X}_{ij}^{in} - \mathbf{X}_{ij}\|_1, \qquad (22)$$

with some constraints for the cycle-consistency, and $\mathcal{E}$ specifies a collection of input maps $\{\phi_{ij}^{in} : S_i \to S_j | (i,j) \in \mathcal{E}\}$. Here they only use the fitting term of (4) with $p = 1$. It is worth noting that there is theoretical guarantee to recover the ground-truth maps by solving (22) with its convex relaxation. Fig. 12 shows some representative results on the Hand datasets.

3.3. Segmentation

Segmentation means segmenting one mesh or a group of meshes (co-segmentation) into meaningful or semantic parts that consistent with user intension, geometric mesh attributes, and human shape perception. Generally, the elements within the same segment should have high similarity, the segment boundary should be tight and smooth as well as matching human perception, and the segmentation should reflect significant features.

3.3.1. Single object segmentation

Motivated by the preceding observation, Zhang et al. [22] attempt to find a partition $M = \bigcup_{i=1}^{k} M_i$, where M_i are pairwise disjoint by convexifying the Mumford-Shah model (M-S model) [81] that has been proven successful in image segmentation to 3D meshes,

$$\min_{\mathbf{u}\in K,\chi_i} \left\{ \int_M \langle \mathbf{u}(x), \mathbf{s}(x)\rangle + \mu g(x)|\nabla_M \mathbf{u}(x)|d\sigma \right\}, \qquad (23)$$

here K is the set of vector functions $\mathbf{u}(x) = (u_1(x), \ldots, u_k(x))^T : M \to \mathbb{R}^k$ which just corresponds to each segment, the second regularization term uses ∇ as T in (4) to constrain the segment boundary quality, $\mathbf{s}(x)$ is a k-dimensional vector with $s_i(x)$ indicating the affinity of x with segment M_i and $g(x)$ is an edge detection function. By defining $\mathbf{u}(x) = \sum_{v_i\in V} \mathbf{u}(v_i)\varphi_i(x)$ as a piecewise linear function where φ_i is a hat function that is linear on each triangle, the energy functional in (23) will be discretized to be the final optimization problem for solving $\mathbf{u}$.

Please cite this article as: L. Xu et al., Survey on sparsity in geometry modeling and processing, Graphical Models (2015), http://dx.doi.org/10.1016/j.gmod.2015.06.012

Fig. 13. Sparse regularization: mesh segmentation [22]. The segmentation results match human perception well in not only the cutting boundaries but also the number of segments.

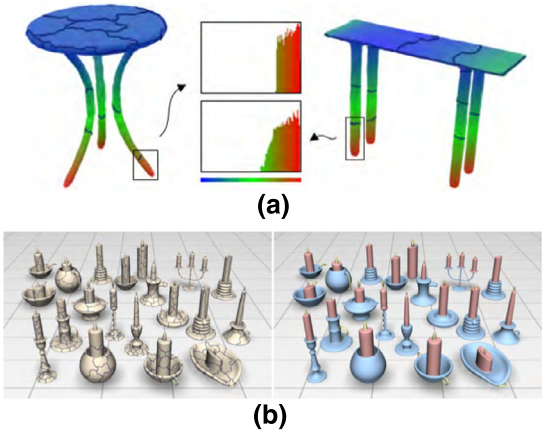


Fig. 14. Sparse regularization: co-segmentation [24]. (a) Colormaps of AGD features of two tables with over-segmented patches. The AGD feature vectors of the two patches (marked in rectangles) from each table's leg have similar distribution, as shown in histograms in the middle. (b) Left shows the over segmented patches that will be clustered to get the right co-segmentation result. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Then with some following processing work, final segmentation can be obtained. Different kind of segmentation results are shown in Fig. 13. Since the regularization term is for constraining the boundary with some geometric difference information between segments, this optimization may fail for the relative smooth models.

3.3.2. Co-segmentation

As Fig. 14(b) shows, co-segmentation consistently segments a group of shapes, where corresponding parts are labeled in the same colors. To be more intuitive and efficient, Hu et al. [24] process co-segmentation on patch-level instead of face-level.

They first over-segment all the models (left in Fig. 14(b)) followed by calculating their feature vectors using some feature descriptors. For example, Fig. 14(a) shows the colormaps of average geodesic distance (AGD) features of two tables with over-segmented patches. By defining the feature vector as a histogram of the feature measurement on the triangles of one patch (middle), it is obvious that two corresponding patches have similar distributions, that is, their feature vectors lie in a common subspace. So they regard co-segmentation as a subspace clustering problem since the final segments are all clustering of over-segmented patches.

Since each feature vector in a union of linear subspaces can always be represented as a linear combination of the feature vectors belonging to the same linear subspace, the combination will be sparse if the feature is written as a linear combination of all other feature vectors. Following [82,83], for the hth feature descriptor, finding the sparse combination matrix for the single-feature co-segmentation is formulated as

$$\min_{\mathbf{W}_h} \|\mathbf{X}_h\mathbf{W}_h - \mathbf{X}_h\|_F^2 + \lambda\|\mathbf{W}_h^T\mathbf{W}_h\|_{1,1},$$

with some constraints for $\mathbf{W}_h$. The feature matrix $\mathbf{X}_h$ is constructed with the feature vector. The $\ell_{1,1}$ penalty item favors the sparsity of the optimal solution $\overline{\mathbf{W}_h}$ of which each entry measures the linear correlation between two points in the meshes. After defining the affinity matrix $\mathbf{S} = (s_{ij})$ as $s_{ij} = |\overline{w_{hij}}| + |\overline{w_{hji}}|$, the NCut method [84] is applied to get the co-segmentation results.

Considering the universality for different categories of models, they add a consistent multi-feature penalty by combing some selected feature descriptors

$$\min_{\{\mathbf{W}_h\}} \sum_{h=1} \mathcal{F}(\mathbf{W}_h) + \alpha\|\mathbf{W}\|_{2,1} + \beta\|\mathbf{W}\|_{1,1}, \tag{24}$$

where the matrix $\mathbf{W}$ is formed by concatenating $\{\mathbf{W}_h\}$(each matrix in one row) together:

$$\mathbf{W} = \begin{bmatrix} (\mathbf{W}_1)_{11} & (\mathbf{W}_1)_{12} & \cdots \\ (\mathbf{W}_2)_{11} & (\mathbf{W}_2)_{12} & \cdots \\ \vdots & \vdots & \vdots \end{bmatrix},$$

in two regularization terms, $\mathbf{T}$ in (4) equals to $\mathbf{I}$, the $\ell_{2,1}$ penalty induces column sparsity of W such that the corresponding pairs of patches will likely not be in the same cluster. The $\ell_{1,1}$ penalty induces the sparsity within each column, then for each similar patch pair, only a subset of features are actually used to measure their similarity. Combining these two penalties enables the prominent features to pop up and guarantees the sparsity consistency of the matrices $\{\mathbf{W}_h\}$.

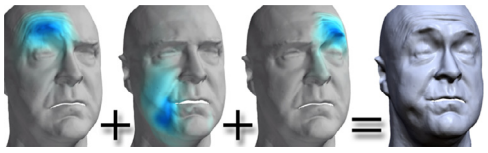


Fig. 15. Sparse regularization: mesh sequence decomposition [23]. A new facial expression is generated by summing deformation components, the method automatically separates spatially confined effects like separate eyebrow motions from the data.

3.3.3. Time varying mesh segmentation

In Fig. 15, a new facial expression is generated by summing deformation components. To decompose any mesh animations like performance faces into sparse and localized deformation modes (shown in blue), Neumann et al. [23] propose an efficient, easy-to-implement, and versatile data-driven approach inspired by matrix decomposition methods like sparse PCA [85].

Given a mesh animation with several frames each of which consists of the same vertex size, a single animation matrix $\mathbf{X}$ is constructed by assembling the vertices of all frames in a row-wise fashion

$$\mathbf{X} = \begin{bmatrix} (\mathbf{p}_1^{(1)})^T & (\mathbf{p}_2^{(1)})^T & \cdots \\ (\mathbf{p}_1^{(2)})^T & (\mathbf{p}_2^{(2)})^T & \cdots \\ \vdots & \vdots & \vdots \end{bmatrix}.$$

After some preprocessing for $\mathbf{X}$, following the framework of Zou et al. [85], they formulate the matrix factorization into K deformation components $\mathbf{C}$ with weights $\mathbf{W}$ as a joint regularized minimization problem

$$\underset{\mathbf{W},\mathbf{C}}{\operatorname{argmin}} \|\mathbf{X} - \mathbf{W}\cdot\mathbf{C}\|_F^2 + \Omega(\mathbf{C}), \quad \text{s.t. } \mathcal{V}(\mathbf{W}). \tag{25}$$

Observing that each triplet in the rows of C forms a three-dimensional vector $\mathbf{c}_k^{(i)} = [x, y, z]_k^{(i)}$, every such triplet corresponds to the x, y, and z displacement of vertex i in component k. To make the dimensions vanish simultaneously to get *sparsity*, $\Omega(\mathbf{C})$ is formulated by acting ℓ_1 norm on the lengths of the displacement vectors

$$\Omega(\mathbf{C}) = \sum_{k=1}^{K}\sum_{i=1}^{N} \Lambda_{ki}\|\mathbf{c}_k^{(i)}\|_2.$$

The spatially-varying regularization parameters Λ_{ki} make it possible to enforce local support for the deformation components which is and exciting innovation. As a result, this sparse localized deformation components for space-time mesh animation data is applicable to many settings: editing, control, scan alignment, construction of static and parametric shape models, etc.

However, it is mentioned in this paper that several parameters in the formulation are specified by users, we are still not clear that whether the users should have knowledge of graphics or whether it is easy for the users to give the suitable values.

3.4. Point cloud consolidation

Point cloud consolidation, known as reconstructing the geometry of a shape from scanned data, is a convenient and direct way to obtain 3D models. It can be a preprocessing phase for some geometric problem, e.g., surface reconstruction whose result is a mesh object, with functionalities such as denoising, outlier removal, orientation, and redistribution of the input points. However, even with high-fidelity scanners, a variety of acquisition errors, like noise, outliers, missing data (holes) or registration artifacts, are inevitable in the produced large amount of raw, dense point sets. And different from mesh denoising, there are not vertex connectivity and triangle quality which can be used or considered. Then finding a robust consolidation technique has always been an active researching area.

3.4.1. Projection operator based

Reconstruction by a projection operator, as shown in Fig. 16 from [17], is to approximate the origin point set (green) by iteratively projecting an arbitrary point set (red) onto itself while removing the noises or outliers. One important virtue is that it defines a consistent geometry based on the data points, and provides constructive means to upsample it.

ℓ_1 median [86,87], closely related to projection operator, is a statistical tool applied globally to multivariate nonparametric point-samples in the presence of noises and outliers. Briefly, it is a robust global center of an arbitrary set of points. Given a data set $P = \{\mathbf{p}_j\}_{j\in J} \subset \mathbb{R}^3$ where J denotes the indices set, the ℓ_1 median is defined as the point q obtained by minimizing the sum of Euclidean distances to the

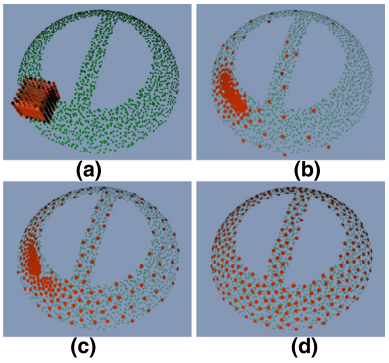


Fig. 16. Reconstruction by projection operator [17]. (a) Noisy point-set P (green) and an arbitrary point-set Q (red) that will be projected to P to approximate P. (b),(c) are two iterative projection results. (d) is the final projection. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

data points

$$\mathbf{q} = \underset{x}{\operatorname{argmin}} \left\{ \sum_{j\in J} \|\mathbf{p}_j - \mathbf{x}\| \right\}. \tag{26}$$

It is just the fitting term in (4) with ℓ_1 norm.

(1). Lipman et al. [17] apply this tool locally to constitute a parameterization-free local projection operator (LOP). Starting with an arbitrary initial point set $X^0 = \{\mathbf{x}_i^0\}_{i\in I} \subset \mathbb{R}^3$(typically $|X| \ll |P|$, $|\cdot|$ is the number of point set) where I is also the indices set, LOP computes the target point positions X by performing a fixed-point iteration

$$X^{k+1} = \underset{X=\{\mathbf{x}_i\}_{i\in I}}{\operatorname{argmin}} \{E_1(X^k, P) + E_2(X^k)\}, \tag{27}$$

where,

$$E_1(X^k, P) = \sum_{i\in I}\sum_{j\in J} \|\mathbf{x}_i - \mathbf{p}_j\|\theta(\|\mathbf{x}_i^k - \mathbf{p}_j\|),$$

$$E_2(X^k) = \sum_{i'\in I} \lambda_{i'} \sum_{i\in I\setminus\{i'\}} \eta(\|\mathbf{x}_i - \mathbf{x}_{i'}^k\|)\theta(\|\mathbf{x}_i^k - \mathbf{x}_{i'}^k\|). \tag{28}$$

The term E_2 which is unrelated with sparsity keeps the distribution of the points X fair by incorporating local repulsion forces. The term E_1 is in fact a *localized* version of (26) by using a fast-decaying weight function $\theta(r) = e^{-r^2/(h/4)^2}$ with the finite support radius h, and thus it is just E_1 that drives the projected points X to approximate the geometry of P.

Actually, the projection for point $\mathbf{x}_i^{k+1}$ has an explicit representation:

$$\mathbf{x}_i^{k+1} = F_1(\mathbf{x}_i^k, P) + F_2(\mathbf{x}_i^k, X_i'). \tag{29}$$

Intuitively, LOP distributes the points by approximating their ℓ_1 median to achieve robustness to outliers and data noises without any local orientation information nor a local manifold assumption. But, also because of the use of the local density parameter h, it may not work well when the distribution of the input points is highly non-uniform and can fail to converge. By making some improvements on these two terms, the next two works obtain better consolidation results.

Please cite this article as: L. Xu et al., Survey on sparsity in geometry modeling and processing, Graphical Models (2015), http://dx.doi.org/10.1016/j.gmod.2015.06.012

(2). Like LOP, many consolidation methods try to obtain the resulted geometric object without estimation of normals due to the unreliability resulting from the noisy data as oppose to the fact that oriented normals at the points play a critical role in geometry reconstruction.

To achieve a better normal estimation that requires the sampling points to be uniformly distributed, Huang et al. [18] propose the WLOP method by incorporating locally adaptive density weights (scalar parameters) into the explicit representation (29).

The obtained uniformly distributed point set can largely improve the reliability of normal initialization for a second normal estimation phase. Practically, due to the high computational effort, it may not be a preferable choice to use this consolidation technique as a preprocessing method for surface reconstruction, even though some high quality surface can be reconstructed.

(3). In LOP/WLOP, the majority of the time is spent on the evaluation of the attractive forces from all points in P, Preiner et al. [19] efficiently reduce the set P of unordered input points to a much more compact mixture of Gaussians $\mathcal{M} = \{w_s, \Theta_s\}$ that reflects the density distribution of the points. That is, $\mathcal{M}$ defines a probability density function (pdf) as a weighted sum of $|\mathcal{M}|$ Gaussian components

$$f(\mathbf{x}|\mathcal{M}) = \sum_s w_s g(\mathbf{x}|\Theta_s),$$

where the $\Theta_s = (\mu_s, \sum_s)$ are the Gaussian parameters, w_s are their corresponding convex weights, and g denotes the d-variate Gaussian pdf. Then they define a *continuous* $\mathcal{F}_1$ corresponding to F_1 in (29) changing the convex sum of 3D points $\mathbf{p}_j$ (28) into a convex combination of the product Gaussians means μ_{sk} with weights w_{sk}.

This continuous method is up to 7 times faster than an optimized GPU implementation of LOP/WLOP, and achieves interactive frame rates for moderately sized point clouds though it cannot automatically get the best choice of the parameters for different point set.

(4). Similar to the sparse gradient minimization, and based on the observation that the gradients (normal differences) of smooth surface normals are sparse, Avron et al. [20] formulate the piecewise smoothness reconstruction problem as a sparse minimization of orientation differences and position projections as following:

$$N^{\text{out}} = \underset{N}{\text{argmin}} \sum_{(\mathbf{p}_i,\mathbf{p}_j)\in E} w_{i,j}\|\mathbf{n}_i - \mathbf{n}_j\|_2,$$

$$X^{\text{out}} = \underset{X}{\text{argmin}} \sum_{(\mathbf{p}_i,\mathbf{p}_j)\in E} w_{i,j}|\mathbf{n}_{i,j}\cdot(\mathbf{x}_i - \mathbf{x}_j)|, \tag{30}$$

where $\{\mathbf{n}_i\}$ denote the surface normals, $\{\mathbf{x}_i\}$ denote the point positions and $\{w_{i,j}\}$ is a set of the weight whose role is to achieve lower-than-ℓ_1 sparsity.

Convexity of these two problems allows for finding a global optimum and deriving efficient solvers. Due to the global nature, this algorithm is extremely slow. And it may fail for the point set with severe noises and outliers.

Fig. 17 shows the results of these ℓ_1 based methods.

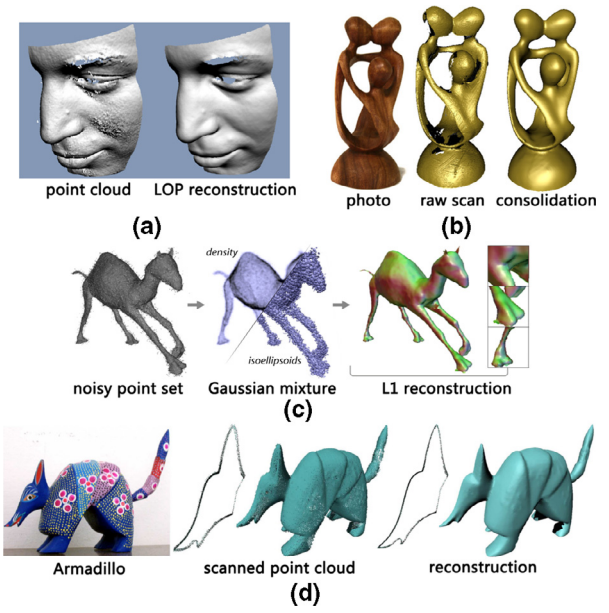


Fig. 17. Sparse regularization: point cloud consolidation. (a) LOP [17]. (b) WOLP [18]. (c) CLOP [19]. (d) Avron et al. [20].

3.4.2. Subdivision

Due to the robustness to noises and outliers of ℓ_1 norm, Mustafa et al. [21] develop an ℓ_1 regression based subdivision algorithm for curve and surface fitting, where the size of target point cloud is largely more than that of origin data in contrast to the previous consolidation works.

For curve fitting corresponding to fitting term in (4), they try to find the best fit straight line $f(x) = \beta_1 + \beta_2 x$ with observations $(x_r = r, f_r), r = -n+1, \ldots, n$. The ℓ_1 regression optimization is simply formulated as

$$\begin{aligned}\beta_1, \beta_2 &= \underset{\beta_1,\beta_2\in\mathbb{R}}{\text{argmin}} \sum_{r=-n+1}^{n} |f_r - (\beta_1 + \beta_2 r)| \\ &= \underset{\beta_1,\beta_2\in\mathbb{R}}{\text{argmin}}\, F(\beta_1, \beta_2),\end{aligned} \tag{31}$$

because of the lack of differentiability, they regularize F with a family of convex functional $F_\delta, \delta > 0$,

$$F_\delta(\beta_1, \beta_2) = \sum_{r=-n+1}^{n} h_\delta(f_r - \beta_1 - \beta_2 r), \quad \text{where}$$

$$h_\delta(f_r - \beta_1 - \beta_2 r) = [(f_r - \beta_1 - \beta_2 r)^2 + \delta]^{1/2}.$$

Then for a given δ, the solution of (31) is approximated by $\beta_{1,\delta}$ and $\beta_{2,\delta}$. By substituting optimum $\beta_{1,\delta}$, $\beta_{2,\delta}$ into $f(x)$ and evaluating this function at 1/4 and 3/4, the closed form of ℓ_1 scheme for curve fitting is obtained.

With the closed form, ℓ_1 scheme D_{2n} first iteratively assigns weights to only $2n$ local initial points, then gets the final fitting result through subdivision rule for locations of vertices of the new mesh and topological rule for size of added vertices and their connectivity. Fig. 18 gives an curve fitting result. By generalizing this ℓ_1 scheme to 3-dimensional case, it can be used for surface fitting.

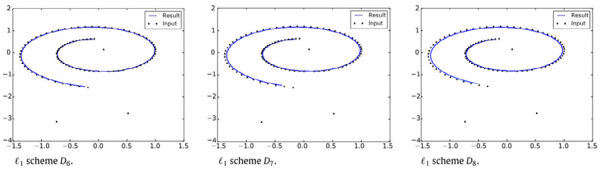


Fig. 18. Sparse regularization: ℓ_1 based subdivision [21]. Parametric curve reconstructed by ℓ_1 scheme from highly corrupted data with three outliers.

3.5. Other applications

3.5.1. Deformation

Constrained modeling is an important tool for the modification of 3D geometric models. Local control, in contrast to some global algorithms, is designed for adjusting *as few vertices as possible* in order not to influence the regions that are already satisfactory. To automatically explore a local deformation which satisfies all constraints, Deng et al. [67] give a novel framework using $\ell_{2,1}$ sparse regularization penalty as following:

$$\min_{\mathbf{d}} \frac{\omega_h}{2} \sum_{i\in\Gamma} \|\mathbf{d}_i - \widetilde{\mathbf{d}}_i\|_2^2 + \frac{\omega_s}{2} \sum_{i\notin\Gamma} \|\mathbf{d}_i\|_2 + \frac{\omega_f}{2} \|\mathbf{E}\mathbf{d}\|_2^2,$$
$$\text{s.t. } E_j(\mathbf{p}^0 + \mathbf{d}) = 0,\ j = 1, \ldots, m, \tag{32}$$

where $\mathbf{p} = [\cdots, \mathbf{p}_i^T, \cdots]^T$ is the position vector, $E_j(\mathbf{p}) = 0 (j = 1, \ldots, m)$ denotes the constraints satisfactory, $\mathbf{d} = [\cdots, \mathbf{d}_i^T, \cdots]^T$ is the displacement with $\mathbf{d}_k$ corresponding to vertex v_k. The second $\ell_{2,1}$ term minimizing the ℓ_1 norm of vector $[\cdots, \|\mathbf{d}_i\|_2, \ldots,]$ induces the sparsity of $\mathbf{d}$, that is, $\mathbf{T} = \mathbf{I}$. $\widetilde{\mathbf{d}}_i$ is the target displacement of the handle vertex v_i and the last term is for a smooth displacement for a nice shape.

To enrich the single solution with fixed weight $\omega = (\omega_h, \omega_s, \omega_f)$ to give more choices for users, based on the modified mesh $\mathbf{p}^0 + \mathbf{d}$, they compute a local modification space spanned by an orthonormal basis $\{\phi_1, \ldots, \phi_s\}$:

$$\min_{\phi} \frac{\beta_f}{2} \|\mathbf{E}\mathbf{d}\|_2^2 + \frac{\beta_h}{2} \sum_{i=1}^{s} \|\phi_i\|_2^2 + \frac{\beta_s}{2} \|\phi\|_{2,1} - \frac{\beta_c}{2} C(\phi). \tag{33}$$

Obviously, the third term is for the sparse displacement ϕ with the first two term for shape quality and the last term for the sparsity of $\mathbf{d} + \phi$.

After the interactive exploration, the final result is optimized to fully satisfy the set of constraints. Fig. 19 shows the advantages of $\ell_{2,1}$ norm over manually fixing vertices or using an ℓ_2 norm closeness energy term.

In addition, Gao et al. [68] give another surface deformation method based on ℓ_p norms. It aims at controlling the distribution of unavoidable distortions (Fig. 20) which also can be explained as local controllability.

3.5.2. Barycentric coordinates

Barycentric coordinates are designed for interpolating values from a set of control points over the interior of a domain, using weighted combinations of values associated with different control points.

Mathematically, given a set of control points $\mathbf{c}_1, \ldots, \mathbf{c}_n$ which are the vertices of a closed control cage that bounds a domain Ω. The goal is to find a function w_i: $\Omega \to \mathbb{R}$ for each $\mathbf{c}_i$, such that $[w_1(\mathbf{x}), \ldots, w_n(\mathbf{x})]$ is a set of generalized barycentric coordinates of $\mathbf{x} \in \Omega$ with respect to the control points $\{\mathbf{c}_i\}$ and is used for interpolating function values $f(\mathbf{c}_1), \ldots, f(\mathbf{c}_n)$ at control points on the interior of Ω by

$$f(\mathbf{x}) = \sum_{i=1}^{n} w_i(\mathbf{x}) f(\mathbf{c}_i),$$

many barycentric coordinates typically get a interpolated value depends on many, potentially *all*, control points.

Except for the properties satisfied in many barycentric coordinate schemes, like reproduction and partition of unity, Zhang et al. [72] prefer a target *convex* functional that also reflects *locality* and *smoothness* for the coordinate functions, so a novel method to derive *local barycentric coordinates* (LBC), which depend only on a small number of control points, is proposed.

For the functions w_i, using the perimeter of each superlevel set $\{w_i > s\} := \{\mathbf{x} | w_i(\mathbf{x}) > s\}$ to regularize the smoothness of its boundary level curve/surface while the perimeter of $\{w_i > 0\}$ to penalize the area of the influence region which results in the locality, the convex functional is exactly the total variation of $\{w_i\}$ where $T = \nabla$, $p = 1$ for (4):

$$\min_{w_1,\ldots,w_2} \sum_{i=1}^{n} \int_{\Omega} |\nabla w_i|, \tag{34}$$

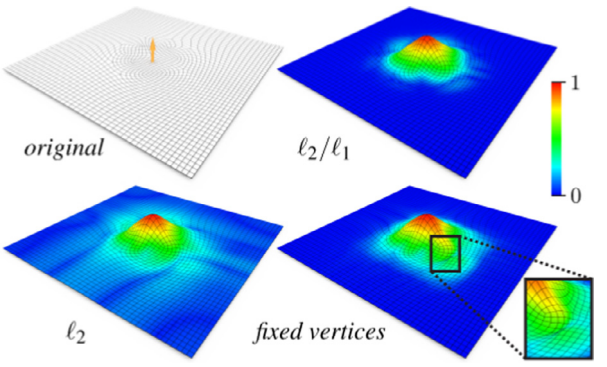


Fig. 19. Sparse regularization: constraint modeling [67]. Comparison of $\ell_{2,1}$ regularization with classical local editing approaches. $\ell_{2,1}$ regularizer provides a more local edit. The colormap denotes the lengths of vertices displacements. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

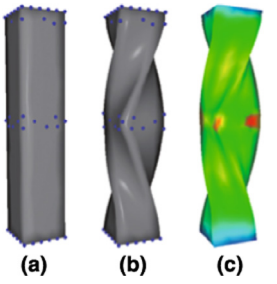


Fig. 20. Sparse regularization: L_p shape deformation [68]. (a) The input surface, the blue points are handles to control the deformation. (b) Deformation result. (c) Color map of distortion distribution(distortion increase from blue to red). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

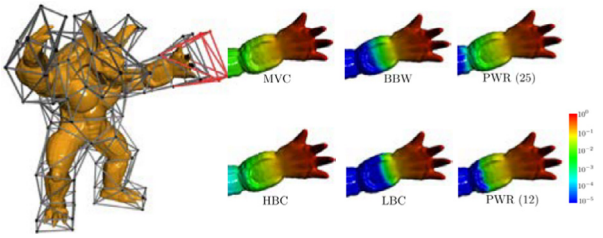


Fig. 21. Sparse regularization: Local barycentric coordinates [72]. Local 3D deformation. A set of control points near the left hand of armadillo model are moved with the deformed cage edges shown in red. The color-coding shows the magnitudes of mesh vertex deformations. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

with some barycentric qualities as the constraints, such as partition of unity, non-negativity. Discretely, after triangulating the domain Ω, each w_i is represented a function that is linear within each cell (triangle in 2D or tetrahedron in 3D) and then the gradient of w_i is constant on each cell. Solving an induced discrete energy function, $\{w_i\}$ will be solved.

Fig. 21 shows a cage-based deformation example with lower computational and storage requirement since each point on the target shape is only determined by a small number of control points. Whatever, from the observation, we can see that there is a trade-off between locality and smoothness which is a common troubling issue for so many existing works.

3.5.3. Skeleton extraction

In Section 3.4.1, we have introduced much information about ℓ_1 median and its success in point cloud consolidation. Except for inducing geometry surface that approximates origin point set, Huang et al. [73] find that adapting ℓ_1 medians *locally* to a point set which represents a geometric shape also gives rise to a *one dimensional* structure, named medial curve skeleton, which can be used for shape analysis and manipulation [88]. Intrinsically, it is a localized center of the shape.

Without building any point connectivity or estimating point normals, by modifying the repulsion term E_2 in Eq. (28) and proposing a different weighted density parameter that can also be named WLOP [18], they project point samples onto their local centers with growing neighborhood and push the projected samples via conditional regularization to obtain a uniform distribution of samples along skeleton branches. To deal with some data errors like holes, they also do more processing which is out of our scope. Fig. 22 shows an example.

3.5.4. Cost-effective 3D printing

3D printing enables fabrication of physical objects from digital models with the expensive cost of the material which is generally much too high. To reduce the material used in printing is then an important and practical problem.

Wang et al. [74] design a skin-frame structure by an multi-objective optimization scheme which significantly reduces material volume

$$\min_{\mathbf{r},\mathbf{V}_{\text{int}},E_{\text{int}}} \text{Vol}(\mathbf{r},\mathbf{V},E)\&|E_{\text{int}}|,$$

with the constraints making the printed objects physically stable, geometrically approximate, and printable. Here $\mathbf{r}$ denotes strut radii whose ℓ_0 norm $|E_{\text{int}}|$ is used to reduce the number of struts

$$\min_{\mathbf{r}} |E_{\text{int}}| = \|\mathbf{r}_{\text{int}}\|_0, \tag{35}$$

that is, they directly use the sparsity of signals to be the regularizer.

Fig. 23 shows the printed objects with the comparison with other method. It is so clear that this automatic method largely reduces the material and maintains physical stability. To be more applicable, the computation cost of the iteratively optimization of the multi-objective scheme cannot be ignored, as well as the size of 3D objects.

(a) A deer model. **(b)** Raw scan. **(c)** L_1-medial skeleton.

Fig. 22. Sparse regularization: skeleton extraction [73]. Given an unorganized, unoriented, and incomplete raw scan with noise and outliers (b), a complete and quality curve skeleton is extracted (c).

Fig. 23. Sparse regularization: cost-effective 3D printing [74]. Left: results produced by the method of Stava et al. [89] by adding external struts; Middle: the new skin-frame (with half-naked rendering); Right: printed objects.

4. Dictionary learning

From the definition of sparse representation, it is obvious that the choice of the dictionary will directly affect the signal processing result. As mentioned in Section 2.2.1, dictionary can either be chosen as a prespecified set of functions (e.g., Wavelet) or designed by adapting its content to fit a given set of signal examples. From the performance of existing dictionary learning based works, the learned dictionaries used to outperform predefined dictionaries. Actually, dictionary learning techniques can be directly used to deal with some geometric problems, such as reconstruction, compression of point cloud and rendering.

Please cite this article as: L. Xu et al., Survey on sparsity in geometry modeling and processing, Graphical Models (2015), http://dx.doi.org/10.1016/j.gmod.2015.06.012

In this section, in addition to (37), (39) which have some variations compared with (3), the others have the same formulation. Here, we also regard sparse matrix decomposition, which decomposes a dense matrix into the multiplication of a simple matrix (e.g., transformation matrix [64]) and the correspondent coefficient matrix that is as sparse as possible, as dictionary learning. Generally, this decomposition is achieved with some iterative algorithm just like the dictionary learning algorithm, so we also call the resulted simple matrix dictionary though it may not be overcomplete. Here, it is all about the Blending Skinning.

4.1. Blend skinning

In this section, the deformation methods are all about blend skinning. Among many proposed techniques, Linear Blend Skinning (LBS), driving skin deformation by a set of bones, is the most popular skinning computational model due to its efficiency, simplicity, and effectiveness. In the LBS model, every vertex is associated with the bones via a bone-vertex weight map which quantifies the influence of each bone to the vertices. The skin is deformed by transforming each vertex through a weighted combination of bone transformations from the rest pose.

Assume w_{ij} is the influence of jth bone to the ith vertex, $\mathbf{p}_i$ is the position of the ith vertex at the rest pose, $|B|$ is the number of bones, and $\mathbf{R}_j^k$ and $\mathbf{t}_j^k$ are the rotation matrix and translation vector of the jth bone at the kth configuration, respectively, then the deformed ith vertex, $\mathbf{x}_i^k$, can be computed as follows:

$$\mathbf{x}_i^k = \sum_{j=1}^{|B|} w_{ij}(\mathbf{R}_j^k \mathbf{p}_i + \mathbf{t}_j^k). \tag{36}$$

By posing sparseness constraint on the weight map, the number of non-zero bone weights per vertex can be limited. The orthogonal constraint on $\mathbf{R}_j^k$ avoids any shearing or scaling effect on the bone transformations, thus put the transformation into rigid group. Thus the bone transformation with orthogonal rotation matrix is called the "rigid bone".

(1). Le et al. [64] introduce Smooth Skinning Decomposition with Rigid Bones (SSDR), an automated algorithm to extract the linear blending skinning, i.e., it aims to solve the inverse problem of the LBS model.

Suppose there are $|k|$ example poses of a $|X|$-vertices model, taking $\{\mathbf{x}_i^k : k = 1..|k|, i = 1..|X|\}$ as input, SSDR decomposes them to bone transformations $(\mathbf{R}_j^k,\ \mathbf{t}_j^k)$ and the bone-vertex weight map (as Fig. 24(a) left shows)

$$\min_{\mathbf{w},\mathbf{R},\mathbf{t}} E = \min_{\mathbf{w},\mathbf{R},\mathbf{t}} \sum_{k=1}^{|k|}\sum_{i=1}^{|X|} \|\mathbf{x}_i^k - \sum_{j=1}^{|B|} w_{ij}(\mathbf{R}_j^k \mathbf{p}_i + \mathbf{t}_j^k)\|^2,$$
$$\text{s.t.}|\{w_{ij}|w_{ij} \neq 0\}| \leq |K|, \quad \forall i \& \mathcal{C}(w, R), \tag{37}$$

where $\mathcal{C}(w, R)$ contains some other constraints about w and R.

With the sparseness constraint on the weight map, SSDR can be used for traditional skinning decomposition tasks such as animation compression and hardware-accelerated rendering. Fig. 24(a) right shows an approximation result of highly deformable model. However, the sparseness constraint also poses certain limitations to skinning models, e.g., it is difficult to handle exceptional vertices that are naturally associated with more than $|K|$ bones or control points. And the relatively high computational cost makes it impractical to some degree.

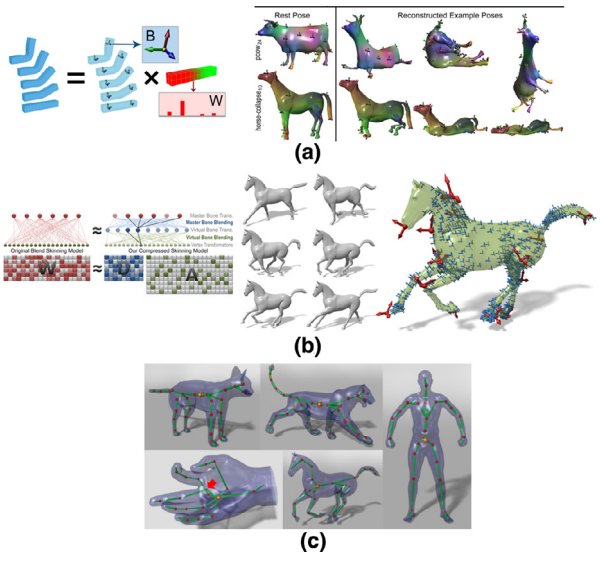


Fig. 24. Dictionary learning: skinning results. (a) [64], Left: a set of example poses are decomposed into rigid bone transformation B and a sparse, convex bone-vertex weight map W. Right: results of SSDR on elastic models. (b) [65], Left: two-layer scheme. Right: an animated mesh sequence and its corresponding compressed skinning model. (c) [66], Result of rigging various models such as quadrupled animals, humans, and highly deformable models.

(2). To address these above limitations, Le et al. [65] introduce an efficient two-layer compression technique that is clearly explained using the left figure of Fig. 24(b). This gives a clear explanation about the new technique.

Let $\mathbf{W} \in \mathbb{R}^{|B|\times|X|}$ be the weight matrix of an input skinning model with $|X|$ vertices and $|B|$ bones, as illustrated in the top left. At *master bone blending* layer, they calculate and cache the transformations of m virtual bones by blending the transformations of $|B|$ original bones (called *master bones*). At *virtual bone blending* layer, they calculate the position of each vertex by blending the transformations of the virtual bones and applying the resultant transformation to the vertex. Imposing a sparseness constraint on each blending layer, the optimization problem is formulated as

$$\min_{\mathbf{D},\mathbf{A}} \Delta_W^2 = \min_{\mathbf{D},\mathbf{A}} \frac{1}{|B||X|}\|\mathbf{DA} - \mathbf{W}\|_F^2,$$
$$\text{s.t.}\mathbf{card}(\mathbf{a}_i) \leq 2, \quad \mathbf{card}(\mathbf{d}_i) \leq c, \forall i, \tag{38}$$

where $\mathbf{a}_i$ and $\mathbf{d}_i$ are the columns of $\mathbf{A}$ and $\mathbf{D}$ respectively. By employing virtual bones to cache transformation blending of master bones, this approach significantly reduces computation of LBS with dense weights, with insignificant loss of accuracy of the original skinning model. But it requires additional storage space for caching virtual bone transformations. Intrinsically, the transformation blending cannot go beyond certain limitations of the LBS model, among which

sophisticated deformation effects such as muscle bulges or skin wrinkles cannot be captured well. The right figure in Fig. 24(b) shows an expressed skinning model.

Actually, these two above methods are both not quite suitable for animation editing purposes since their extracted bone transformations are not organized in any skeletal structures based on which the mesh deformation is a widely-used method for animating articulated creatures such as humans and animals.

(3). Attempting to take advantage of example poses for computation reduction, taking a set of example poses as input, Le et al. [66] introduce a robust and accurate rigging framework producing its corresponding *Skeleton-based* LBS model including skeletal structure, skinning weights, joint locations, and bone transformations corresponding to all the example poses.

After initializing bone transformations and determining the skeleton topology, they get the optimized LBS model by minimizing function

$$E = E_D + wE_S + \lambda E_J,$$

with the same set of constraints (37) including the *sparseness constraints* (no more than 4 non-zero weights per vertex) there and the term E_D is also similar to their work

$$E_D = \frac{1}{|k||X|}\sum_{i=1}^{|X|}\sum_{k=1}^{|k|}\|\mathbf{x}_i^k - \sum_{j=1}^{|B|} w_{ij}(\mathbf{R}_j^k\mathbf{p}_i + \mathbf{t}_j^k)\|^2. \tag{39}$$

The term E_S favors the smoothness of skinning weights and drives the removal of redundant bones and E_J keeps any two connected transformations rotate around their common joint (refer to [66] for the formulation).

The output can be directly used to set up skeleton-based animation in various 3D modeling and animation software as well as game engines (Fig. 24(c)). Despite the achieved accuracy and robustness, this approach has several limitations including the aforementioned low computational efficiency, example data dependency, and limited approximation power of the LBS model.

4.2. Reconstruction

Surface reconstruction takes a set of dense unorganized points, which are sampled from a subjacent, piecewise smooth surface, as input and outputs a triangular mesh to approximate the surface. Existing methods often realize reconstruction via a few phases with respective goals, e.g., point cloud consolidation (Section 3.4) can be a preprocessing phase to denoise, remove outlier and thus reduce more reliable normal estimation. However, integration of processing phases may not give an optimal solution. To avoid the inherent limitations of multi-phase processing, Xiong et al. [16] propose a unified framework that treats geometry and connectivity construction as one joint optimization problem.

As Fig. 25(a) shows, given a point set $P = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n\}$(blue) sampled from a piecewise smooth surface S, they attempt to find a triangular mesh $M = \{V, F\}$ with vertex set $V = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m\}$(red) and triangle set F to approximate the underlying surface S such that the approximation error is as small as possible

$$\min_{\mathbf{B},\mathbf{V}} \frac{1}{n}\sum_{i=1}^{n}\|\mathbf{p}_i - \mathbf{V}\mathbf{b}_i\|_2^q + E_{\text{reg}},$$
$$\text{s.t.}\|\mathbf{b}_i\|_0 \le 3, \quad \|\mathbf{b}_i\|_1 = 1, \quad \mathbf{b}_i \ge 0, \quad \forall i, \tag{40}$$

where $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m]$ is the vertex positions matrix, then E_{reg} regularizes the reconstructed mesh to produce good mesh quality, each column of sparse coding matrix $\mathbf{B}$ corresponds to a triangle in mesh. Finally, all the points sampled from the region approximated by a triangle can be represented as a convex combination of the same three vertices.

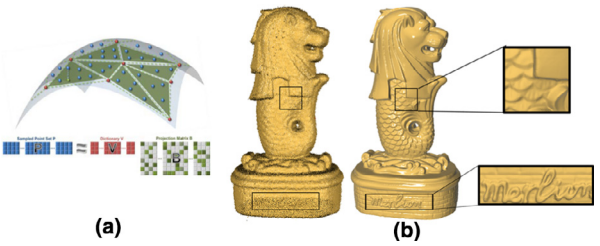


Fig. 25. Dictionary learning: reconstruction [16]. Left: (Top) an illustration of the reconstruction problem. Given point set P (blue) sampled from surface S, they approximate S with piecewise linear surface M with vertices V(red) and triangles F. (Bottom) The reconstruction problem where P is the position of sample point set. $\mathbf{V}$ is the dictionary and $\mathbf{B}$ (green) is the sparse coding matrix that encodes triangles F. Right: reconstruction result of the Merlion model. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fig. 25 (b) shows the reconstruction result, with high triangle quality, of the Merlion model with various geometric features such as sharp and semi-sharp features and different levels of surface details. Despite these high quality results, the nonconvex optimization model makes it difficult for the solver to theoretically guarantee convergence or produce a global optimal solution. And it can fail when the point cloud has large holes or is highly non-uniform due to the current sampling method.

4.3. Compression

Also due to the drastic improvement in scanner acquisition devices yielding point sets of tens of millions of points at high precision, point cloud processing problems generally require much higher storage capacity which results in the expensive cost.

In the work of Digne et al. [76] for point cloud compression, after selecting a subset of points (the seeds) that will serve as center points to cover the surface with local patches, they compute patch descriptions which are the observations Y in Eq. (7) using a new neighborhood descriptor (Fig. 26(a)), then directly using the K-SVD algorithm [46] to solve (7): it is easy to exploit the self-similarity of the descriptions and build a custom dictionary D (Fig. 26(b)) over which all descriptors will be decomposed sparsely with A. Here Y corresponds to the patch descriptions.

Briefly, selected patch descriptions deduce the dictionary. Thus a new seed selection strategies or patch descriptors may result in higher performance, this just tells the unrobustness of heuristic methods. This compression is done at the resolution of the scanner enabling improved control of the point

Please cite this article as: L. Xu et al., Survey on sparsity in geometry modeling and processing, Graphical Models (2015), http://dx.doi.org/10.1016/j.gmod.2015.06.012

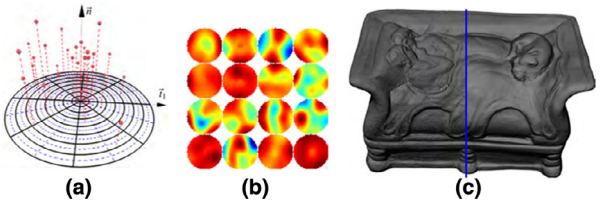


Fig. 26. Dictionary learning: point cloud compression [76]. (a) The local neighborhood description: a height map over a radial grid. (b) Dictionary built for the Lovers model shown in (c). (c) The Lovers model containing 15.8 million points is compressed down to 1.15 MB, the left side of the blue line is the original data while the right side is the decompressed result. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

cloud resolution. It achieves a filtering of noise whose magnitude is smaller than the scanner precision. Fig. 26(c) gives one compression result.

5. Low rank

So far, low rank is mainly used in subspace segmentation or clustering for which there have been much works. Based on it, Zhang et al. [75] successfully estimate the point cloud normals by segmenting the neighborhoods of the points around the sharp features into some smooth subneighborhoods, and thus it can well detect sharp features. Upright orientation is actually an intrinsic application due to the image projections which can be regarded as matrices as Fig. 27 shows.

5.1. Upright orientation

Most man-made models can be posed at a unique upright orientation which is consistent to human sense. Given a 3D digital model, finding its upright orientation and posing it at the right orientation is vital for users to recognize it.

Fig. 27 shows the axis-aligned projections of an input man-made model with arbitrary (a) and axis-aligned orientation (b) onto the *y*-*z*, *z*-*x*, and *x*-*y* plane in the *x*-*y*-*z* coordinate system. Regarding these projections as two-dimensional matrices, it is clear that the ranks of projection matrices in (b) are significantly *lower* than those in (a). And, the upright orientation (c) should be one of the six orientations determined by the six axis-aligned candidate bases, i.e., top, bottom, left, right, front and back surface of the bounding box of the model. Briefly, ranks of projection matrices at axis-aligned orientations are lower than their counterparts at other orientations, since man-made models are mainly composed by horizonal and vertical edges and shapes.

(1). Based on this observation, Jin et al. [69] present an unsupervised approach for finding the upright orientation of man-made models. Taking the *x*-*y* plane projection as an example, they binarize the projection as black and white to generate the projection image $\beth$ with fixed resolution which can also be referred as a two-dimensional matrix. To avoid affect of noise, $\beth$ is modeled as a low-rank version $\mathbf{L}$ with sparse-error matrix $\mathbf{S}$. And the problem is formulated as (9) with a little difference in constraint $\beth \circ \mathbf{R} = \mathbf{L} + \mathbf{S}$ where $\mathbf{R}$ is a rigid rotation transformation matrix used to rectify $\beth$ to recover the optimal low-rank representation of *x*-*y* plane projection from an arbitrary orientation.

For the whole algorithm, after selecting which projection should be rectified from *x*-*y*, *y*-*z* and *z*-*x*, using the low rank formulation the man-made model will be aligned with some axes followed by final upright orientation selection from six orientations as mentioned above. However, whether a model fits for this algorithm depends on if the model contains dominant parts parallel to the supporting base, then it will fail if the model is composed by several equivalently main parts which have their own low-rank observation in different orientations.

(2). The above method uses the information of projected images on three axes planes which contains not only the boundary but also the internal of the projected model. It is very natural to generalize this method in 3D space to construct

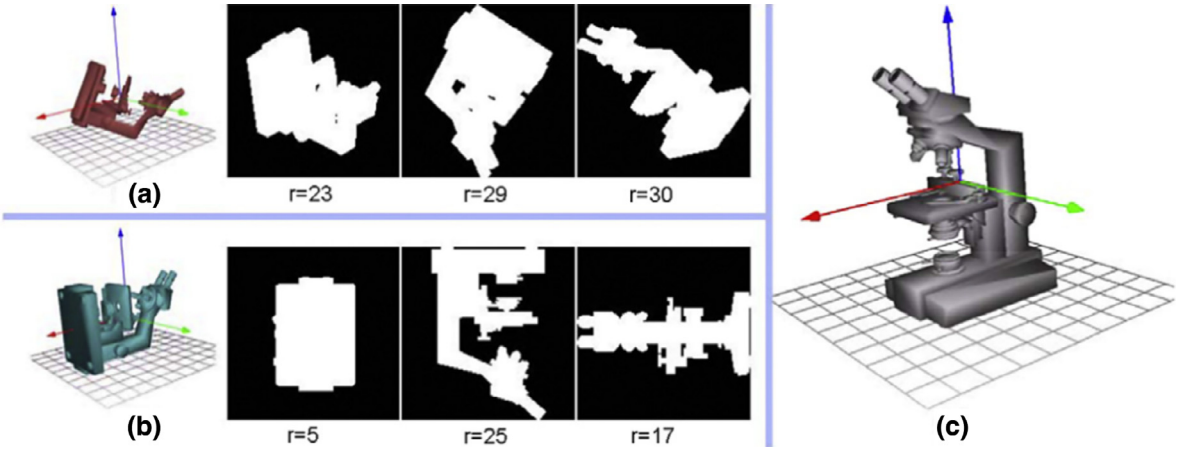


Fig. 27. Low rank: upright orientation observation [69]. (a) Input model (left) at arbitrary orientation and its axis-aligned projections in the *x*-*y*-*z*(red-green-blue) coordinate system. Projections from left to right *y*-*z*, *z*-*x*, and *x*-*y* plane projection, accompanied with their corresponding matrix ranks r. (b) Axis-aligned model (left) and its projections with matrix ranks. (c) The model (gray) posed at the upright orientation. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Please cite this article as: L. Xu et al., Survey on sparsity in geometry modeling and processing, Graphical Models (2015), http://dx.doi.org/10.1016/j.gmod.2015.06.012

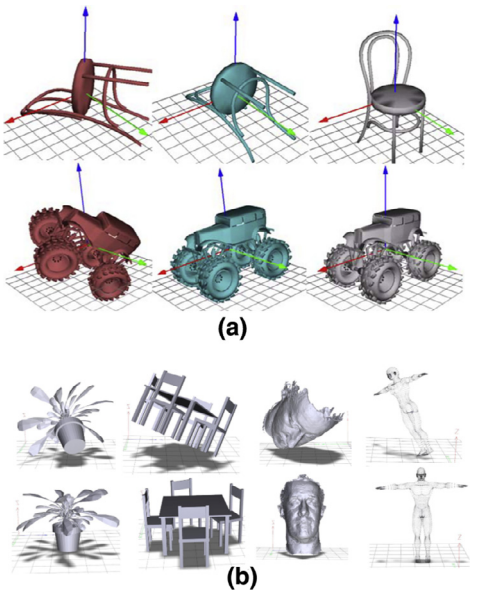


Fig. 28. Low rank: unsupervised upright orientation. (a) Local method [69]. (b) Global method [70].

three-order tensor (multidimensional array) with volume of the 3D model, i.e., the three-order tensor ought to have a low rank behavior. Wang et al. [70] construct this three-order tensor using the bounding box of the 3D model since the bounding box parallels the coordinate planes and contains the whole model. By translating the barycenter of the input model to the origin of the coordinate system, they just need to find and optimal rotation matrix $\mathbf{R}$ to align the model with three axes by following optimization model:

$$\mathbf{R}_* = \underset{\mathbf{R}}{\operatorname{argmin}} \left(\| \chi (\mathbf{V} \circ \mathbf{R}) \|_* \right), \tag{41}$$

where $\mathbf{V}$ and $\mathbf{V}\circ\mathbf{R}$ respectively indicate the point coordinates of input model and the rotated model, $\chi(\cdot)$ is the three-order tensor. Similar to [69], after aligning the model with three axes, they select the upright orientation from six orientations by analyzing the geometric properties. Fig. 28 shows the results of these two methods.

5.2. Point cloud normal estimation

As mentioned above, good normal estimation from noisy data would induce better geometric processing results, like reconstruction, rendering.

Considering that, the neighborhood of a point in a smooth region can be well approximated by a plane, it is then easy to get a robust normal estimation. Thus using the robust results in smooth regions as prior knowledge, Zhang et al. [75] estimate the point normals around the sharp regions by low-rank clustering (LRSCPK).

In Section 3.3.2, we have given an overview about sparse subspace clustering. Low-rank subspace clustering, for capturing the global structure of the whole data, is a modification as

$$\min \|\mathbf{Z}\|_*, \text{ s.t. } \mathbf{X} = \mathbf{XZ}.$$

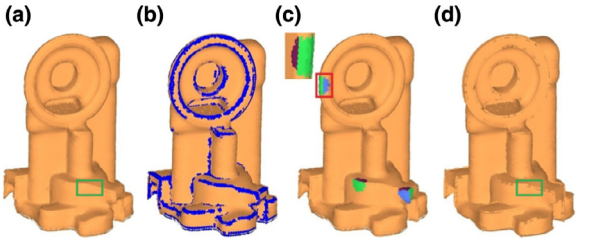


Fig. 29. Low rank: normal estimation [75]. (a) The oil pump module with normal computed by PCA. (b) Initial detected candidate feature points. (c) The classified subneighborhoods. The neighborhood within the red box contains three subneighborhoods rendered in blue, green and brown and the zoomed view is from left. (d) Estimated normals. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

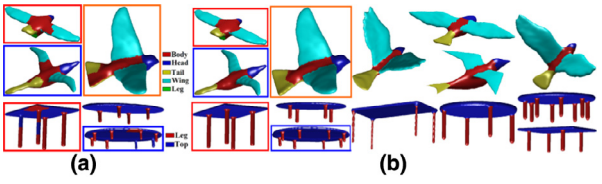


Fig. 30. Low rank: segmentation and labeling [25]. (a) The labeled examples with mislabeled meshes marked by rectangle. (b) The results of the low rank based method where the meshes marked by rectangles are the relabeling results for the mislabeled examples.

With an input noisy point cloud $P = \{\mathbf{p}_i\}_{i=1}^n$, they first detect the candidate feature points (Fig. 29(b)) by covariance analysis of the point neighborhoods. To segment each neighborhood into several isotropic subneighborhoods, for each candidate point $\mathbf{p}_i$, they select a larger neighborhood. By defining sampling matrix $\mathbf{X} = [\cdots, \mathbf{p}_{ij}, \cdots]$, where $\mathbf{p}_{ij}$ is the neighbor point of $\mathbf{p}_i$ containing point coordinates and normal computed by PCA in the local coordinates with $\mathbf{p}_i$ as the origin, the optimal coefficient matrix $\mathbf{Z}$ is computed by solving

$$\begin{aligned} &\min \|\mathbf{Z}\|_* + \beta \|\mathbf{P}_\Omega(\mathbf{Z})\|_1 + \gamma \|\mathbf{S}\|_{2,1}, \\ &\text{s.t. } \mathbf{X} = \mathbf{XZ} + \mathbf{S}, \end{aligned} \tag{42}$$

where, $\mathbf{P}_\Omega$ is related to $\Omega(0 \le \Omega(i,j) \le 1)$ which is a guiding matrix constructed according to the distance relation between each two neighbor points for current candidate point. After getting $\mathbf{Z}$, by defining the affinity matrix like Section 3.3.2, this larger neighborhood is segmented into several subneighborhoods (Fig. 29(c)) in which a consistent subneighborhood is used to estimate the current point normal.

5.3. Segmentation

With the same optimization model as (42), Liu et al. [25] propose a low rank and example based method for mesh segmentation and labeling.

With the labeled example models and like the co-segmentation work [24], they first oversegment all the labeled and unlabeled meshes followed by the feature vector construction through accumulating the histograms of several selected feature descriptors, then each feature vector is used to be one column of the feature matrix $\mathbf{X}$ in (42). Here,

Please cite this article as: L. Xu et al., Survey on sparsity in geometry modeling and processing, Graphical Models (2015), http://dx.doi.org/10.1016/j.gmod.2015.06.012

Table 2
Some applications in geometric processing with problems to adapt sparsity.

Applications	Effectiveness in image	Problems in geometry
Classification	Group images into meaningful categories with robustness to noises	What should be the proper dictionaries? What should be the proper atom size for capturing geometry features? How to deal with the orientation?
Inpainting	Fill in missing pixels in known locations in the image with robustness to noises	What should be the proper dictionaries? What should be the proper atom size for capturing geometry features? What is the essence of linearly combining the learned dictionary atoms? Whether the symmetry information can be captured?
Separation	Cartoon+texture	What should be the proper dictionaries?

$\mathbf{P}_\Omega(\mathbf{Z}) = \Omega \odot \mathbf{Z}$ where Ω is a guiding matrix, and it is estimated according to the labels of the labeled examples and the geometric similarity between the labeled and the unlabeled patches through the affinity of another feature descriptor different from those mentioned above. After solving Eq. (42), the relationship matrix $\mathbf{Z}$ would be used to achieve the final labeling processing which is also out of our scope. Fig. 30 shows some labeling examples.

6. Discussion

Sparse techniques, as the effective tools for signal processing such as denoising, reconstruction and segmentation, have shown great potentials in geometric processing related applications. Its robustness to outliers and the ability to preserve sharp features are attracting more and more attentions from geometric processing community. As a summary, we want to discuss the usage of sparsity in geometric processing and propose some open questions for future study.

Summary. The effectiveness of sparsity in geometric processing can be categorized into the following parts:

- Point cloud processing. As the point cloud scanned by sensors like Kinect and PrimeSense always contains a lot of noises and outliers, sparsity based fitting method can be employed to detect the outliers and reconstruct a clean point cloud [16–21,76].
- Local control. Mesh editing is a quite popular method in geometric modeling. Local deformation is strongly needed to preserve last editing, and thus sparsity regularization is a very suitable tool to achieve this purpose [67,68,71,72].
- Feature preserving. In many mesh reconstruction and editing applications, sharp features should be preserved after processing. As sharp feature is always sparse, sparsity formulation catches this observation well and achieves state-of-art performance [13–15,75].
- Variable selection. Variable selection is a classical problem in statistics and can be formulated as a sparsity based problem. Variable selection can be applied in many geometric processing problems. For example, the co-segmentation problem [24] selects the suitable feature for different part.
- Repeated pattern. The 3D shape models always contain a lot of repeated pattern, for example, street building contains the same shape of windows. By concentrating the similar repeated pattern together, we can get a low-rank matrix, and thus low-rank method [25,69,70,75] can effectively deal with this type of 3D model well.

Inspirations. Although the concept of sparsity has attracted a lot of attention in recent years, many problems still remain to be solved and many other applications can be applied on:

- Sparse coding or representation, that use basis to represent given input signal requires that basis and signal have the same expression form or simply have the same dimension. This is the main reason that this approach is hardly used in geometry and only [76] adopts a simple strategy to resample input point cloud. Therefore it is a basic and remaining problem needed to be addressed of applying sparse coding.
- Another important problem is that the design of $\mathbf{T}$ in (4). Total variation model is a widely used and respected model for computer vision where T is the gradient operator. However in geometry related problems the computation of such an operator is a basic problem and the operator should be suitable for current application.
- Finally, there are a huge amount of applications with high performance in images [1] but with much difficulties to achieve in geometric processing except for the three items mentioned in Section 1. Table 2 lists some of them we are interested in and the corresponding problems. And it can be seen that what should be the proper dictionary is the common problem among them as oppose to the low dimensional images where the pixel colors can be used directly. In addition, the computation and memory cost should also be considered when selecting the dictionary. How to resolve these problems may give much chances as future works.

Acknowledgements

The authors thank the reviewers for providing useful comments and suggestion. The work is supported by the Natural Science Foundation of China (61222206, 61303148, 11371341, 11171322), the 973 Program (2011CB302400), the 111 Project (b07033), the NSF of Anhui Province of China (1408085QF119), Specialized Research Fund for the Doctoral Program of Higher Education (20133402120002), One Hundred Talent Project of the Chinese Academy of Sciences, and the Fundamental Research Funds for the Central Universities.

References

[1] M. Elad, Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing, Springer, 2010.

[2] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T.S. Huang, S. Yan, Sparse representation for computer vision and pattern recognition, Proc. IEEE 98 (6) (2010) 1031–1044.
[3] C.M. Bishop, et al., Pattern Recognition and Machine Learning, 1, Springer, New York, 2006.
[4] I.H. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufmann, 2005.
[5] S. Mallat, A Wavelet Tour of Signal Processing: The Sparse Way, Academic Press, 2008.
[6] A.M. Bruckstein, D.L. Donoho, M. Elad, From sparse solutions of systems of equations to sparse modeling of signals and images, SIAM Rev. 51 (1) (2009) 34–81.
[7] S. Waydo, A. Kraskov, R.Q. Quiroga, I. Fried, C. Koch, Sparse representation in the human medial temporal lobe, J. Neurosci. 26 (40) (2006) 10232–10234.
[8] D. Rinberg, A. Koulakov, A. Gelperin, Sparse odor coding in awake behaving mice, J. Neurosci. 26 (34) (2006) 8857–8865.
[9] V.N. Vapnik, V. Vapnik, Statistical Learning Theory, 2, Wiley, New York, 1998.
[10] V. Vapnik, The Nature of Statistical Learning Theory, Springer, 2000.
[11] T. Hastie, R. Tibshirani, J. Friedman, T. Hastie, J. Friedman, R. Tibshirani, The Elements of Statistical Learning, 2, Springer, 2009.
[12] A. Chambolle, V. Caselles, D. Cremers, M. Novaga, T. Pock, An introduction to total variation for image analysis, in: Theoretical Foundations and Numerical Methods for Sparse Recovery, De Gruyter, 2010.
[13] L. He, S. Schaefer, Mesh denoising via ℓ_0 minimization, ACM Trans. Graph. 32 (4) (2013) 64.
[14] R. Wang, Z. Yang, L. Liu, J. Deng, F. Chen, Decoupling noise and features via weighted ℓ_1-analysis compressed sensing, ACM Trans. Graph. 33 (2) (2014) 18.
[15] H. Zhang, C. Wu, J. Zhang, J. Deng, Variational mesh denoising using total variation and piecewise constant function space, IEEE Trans. Visual. Comput. Graph. 21 (7) (2015) 873–886.
[16] S. Xiong, J. Zhang, J. Zheng, J. Cai, L. Liu, Robust surface reconstruction via dictionary learning, ACM Trans. Graph. 33 (6) (2014) 201.
[17] Y. Lipman, D. Cohen-Or, D. Levin, H. Tal-Ezer, Parameterization-free projection for geometry reconstruction, ACM Trans. Graph. 26 (2007) 22.
[18] H. Huang, D. Li, H. Zhang, U. Ascher, D. Cohen-Or, Consolidation of unorganized point clouds for surface reconstruction, ACM Trans. Graph. 28 (2009) 176.
[19] R. Preiner, O. Mattausch, M. Arikan, R. Pajarola, M. Wimmer, Continuous projection for fast l1 reconstruction, ACM Trans. Graph. 33 (4) (2014) 47:1–47:13.
[20] H. Avron, A. Sharf, C. Greif, D. Cohen-Or, L1-sparse reconstruction of sharp point set surfaces, ACM Trans. Graph. 29 (5) (2010) 135.
[21] G. Mustafa, H. Li, J. Zhang, J. Deng, ℓ_1-regression based subdivision schemes for noisy data, Comput.-Aided Design 58 (2014) 189–199.
[22] J. Zhang, J. Zheng, C. Wu, J. Cai, Variational mesh decomposition, ACM Trans. Graph. 31 (3) (2012) 21.
[23] T. Neumann, K. Varanasi, S. Wenger, M. Wacker, M. Magnor, C. Theobalt, Sparse localized deformation components, ACM Trans. Graph. 32 (6) (2013) 179.
[24] R. Hu, L. Fan, L. Liu, Co-segmentation of 3D shapes via subspace clustering, in: Computer Graphics Forum, 31, Wiley Online Library, 2012, pp. 1703–1713.
[25] X. Liu, J. Zhang, R. Liu, B. Li, J. Wang, J. Cao, Low-rank 3D mesh segmentation and labeling with structure guiding, Comput. Graph. 46 (2014) 99–109.
[26] S. Bouaziz, A. Tagliasacchi, M. Pauly, Sparse iterative closest point, in: Proceedings of the Eleventh Eurographics/ACMSIGGRAPH Symposium on Geometry Processing, Eurographics Association, 2013, pp. 113–123.
[27] S.G. Mallat, Z. Zhang, Matching pursuits with time-frequency dictionaries, IEEE Trans. Signal Process. 41 (12) (1993) 3397–3415.
[28] Y.C. Pati, R. Rezaiifar, P. Krishnaprasad, Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition, in: Signals, Systems and Computers, 1993. 1993 Conference Record of The Twenty-Seventh Asilomar Conference on, IEEE, 1993, pp. 40–44.
[29] R. Tibshirani, Regression shrinkage and selection via the lasso, J. R. Stat. Soc. Ser. B (Methodological) (1996) 267–288.
[30] D.L. Donoho, Compressed sensing, IEEE Trans. Inform. Theory 52 (4) (2006) 1289–1306.
[31] J. Wright, A.Y. Yang, A. Ganesh, S.S. Sastry, Y. Ma, Robust face recognition via sparse representation, IEEE Trans. Pattern Anal. Mach. Intell. 31 (2) (2009) 210–227.
[32] J. Yang, J. Wright, T. Huang, Y. Ma, Image super-resolution as sparse representation of raw image patches, in: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, IEEE, 2008, pp. 1–8.
[33] J. Mairal, F. Bach, J. Ponce, G. Sapiro, A. Zisserman, Discriminative learned dictionaries for local image analysis, in: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, IEEE, 2008, pp. 1–8.
[34] N. Ahmed, T. Natarajan, K.R. Rao, Discrete cosine transform, IEEE Trans. Comput. 100 (1) (1974) 90–93.
[35] R.N. Bracewell, R. Bracewell, The Fourier Transform and Its Applications, 31999, McGraw-Hill, New York, 1986.
[36] B.A. Olshausen, D.J. Field, Sparse coding with an overcomplete basis set: a strategy employed by v1? Vis. Res. 37 (23) (1997) 3311–3325.
[37] L.I. Rudin, S. Osher, E. Fatemi, Nonlinear total variation based noise removal algorithms, Physica D 60 (1) (1992) 259–268.
[38] A. Chambolle, V. Caselles, D. Cremers, M. Novaga, T. Pock, An introduction to total variation for image analysis, Theor. Found. Numer. Methods Sparse Recov. 9 (2010) 263–340.
[39] J. Yang, Y. Zhang, W. Yin, A fast alternating direction method for tvl1-l2 signal reconstruction from partial fourier data, IEEE J Selected Topics Signal Process. 4 (2) (2010) 288–297.
[40] L. Xu, C. Lu, Y. Xu, J. Jia, Image smoothing via ℓ_0 gradient minimization, ACM Trans. Graph. 30 (6) (2011) 174.
[41] C. Wu, J. Zhang, X.-C. Tai, Augmented lagrangian method for total variation restoration with non-quadratic fidelity, Inverse Problems Imaging 5 (1) (2011) 237261.
[42] J.A. Tropp, A.C. Gilbert, Signal recovery from random measurements via orthogonal matching pursuit, IEEE Trans. Inform. Theory 53 (12) (2007) 4655–4666.
[43] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, 2009.
[44] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, et al., Least angle regression, Ann. Stat. 32 (2) (2004) 407–499.
[45] K. Engan, S.O. Aase, J. Husoy, Frame based signal compression using method of optimal directions (mod), in: Circuits and Systems, 1999. ISCAS'99. Proceedings of the 1999 IEEE International Symposium on, 4, IEEE, 1999, pp. 1–4.
[46] M. Aharon, M. Elad, A. Bruckstein, K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation, IEEE Trans. Signal Process. 54 (11) (2006) 4311–4322.
[47] M. Fukushima, Application of the alternating direction method of multipliers to separable convex programming problems, Comput. Optim. Appl. 1 (1) (1992) 93–111.
[48] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, Found. Trends® Mach. Learn. 3 (1) (2011) 1–122.
[49] I. Jolliffe, Principal Component Analysis, Wiley Online Library, 2005.
[50] S. Wold, K. Esbensen, P. Geladi, Principal component analysis, Chemomet. Intell. Lab. Syst. 2 (1) (1987) 37–52.
[51] E.J. Candès, X. Li, Y. Ma, J. Wright, Robust principal component analysis? J. ACM (JACM) 58 (3) (2011) 11.
[52] E.J. Candès, B. Recht, Exact matrix completion via convex optimization, Found. Comput. Math. 9 (6) (2009) 717–772.
[53] Z. Lin, M. Chen, Y. Ma, The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices, preprint arXiv:1009.5055 (2010).
[54] M. Grant, S. Boyd, Y. Ye, Cvx: Matlab software for disciplined convex programming, 2008.
[55] A. Beck, M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, SIAM J. Imaging Sci. 2 (1) (2009) 183–202.
[56] Z. Lin, A. Ganesh, J. Wright, L. Wu, M. Chen, Y. Ma, Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix, Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP) 61 (2009).
[57] D.P. Bertsekas, Constrained optimization and lagrange multiplier methods, in: Computer Science and Applied Mathematics, 1, Boston: Academic Press, 1982.
[58] D.P. Bertsekas, Nonlinear Programming, Athena Scientific, 1999.
[59] L. Ma, C. Wang, B. Xiao, W. Zhou, Sparse representation for face recognition based on discriminative low-rank dictionary learning, in: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE, 2012, pp. 2586–2593.
[60] X. Shen, Y. Wu, A unified approach to salient object detection via low rank matrix recovery, in: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE, 2012, pp. 853–860.
[61] H. Ji, C. Liu, Z. Shen, Y. Xu, Robust video denoising using low rank matrix completion, in: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, IEEE, 2010, pp. 1791–1798.
[62] J. Pokrass, A.M. Bronstein, M.M. Bronstein, P. Sprechmann, G. Sapiro, Sparse modeling of intrinsic correspondences, in: Computer Graphics Forum, 32, Wiley Online Library, 2013, pp. 459–468.

Please cite this article as: L. Xu et al., Survey on sparsity in geometry modeling and processing, Graphical Models (2015), http://dx.doi.org/10.1016/j.gmod.2015.06.012

[63] Q.-X. Huang, L. Guibas, Consistent shape maps via semidefinite programming, in: Computer Graphics Forum, 32, Wiley Online Library, 2013, pp. 177–186.
[64] B.H. Le, Z. Deng, Smooth skinning decomposition with rigid bones, ACM Trans. Graph. 31 (6) (2012) 199.
[65] B.H. Le, Z. Deng, Two-layer sparse compression of dense-weight blend skinning, ACM Trans. Graph. 32 (4) (2013) 124.
[66] B.H. Le, Z. Deng, Robust and accurate skeletal rigging from mesh sequences, ACM Trans. Graph. 33 (4) (2014) 84:1–84:10.
[67] B. Deng, S. Bouaziz, M. Deuss, J. Zhang, Y. Schwartzburg, M. Pauly, Exploring local modifications for constrained meshes, in: Computer Graphics Forum, 32, Wiley Online Library, 2013, pp. 11–20.
[68] L. Gao, G.-X. Zhang, Y.-K. Lai, L_p shape deformation, Sci. China Inform. Sci. 55 (5) (2012) 983–993.
[69] Y. Jin, Q. Wu, L. Liu, Unsupervised upright orientation of man-made models, Graph. Models 74 (4) (2012) 99–108.
[70] W. Wang, X. Liu, L. Liu, Upright orientation of 3D shapes via tensor rank minimization, J. Mech. Sci. Technol. (2014) 40–53.
[71] T. Neumann, K. Varanasi, C. Theobalt, M. Magnor, M. Wacker, Compressed manifold modes for mesh processing, in: Computer Graphics Forum, 33, Wiley Online Library, 2014, pp. 35–44.
[72] J. Zhang, B. Deng, Z. Liu, G. Patanè, S. Bouaziz, K. Hormann, L. Liu, Local barycentric coordinates, ACM Trans. Graph. 33 (6) (2014) 188.
[73] H. Huang, S. Wu, D. Cohen-Or, M. Gong, H. Zhang, G. Li, B. Chen, ℓ_1-medial skeleton of point cloud., ACM Trans. Graph. 32 (4) (2013) 65.
[74] W. Wang, T.Y. Wang, Z. Yang, L. Liu, X. Tong, W. Tong, J. Deng, F. Chen, X. Liu, Cost-effective printing of 3D objects with skin-frame structures, ACM Trans. Graph. 32 (6) (2013) 177.
[75] J. Zhang, J. Cao, X. Liu, J. Wang, J. Liu, X. Shi, Point cloud normal estimation via low-rank subspace clustering, Comput. Graph. 37 (6) (2013) 697–706.
[76] J. Digne, R. Chaine, S. Valette, Self-similarity for accurate compression of point sampled surfaces, in: Computer Graphics Forum, 33, Wiley Online Library, 2014, pp. 155–164.
[77] U. Pinkall, K. Polthier, Computing discrete minimal surfaces and their conjugates, Exp. Math. 2 (1) (1993) 15–36.
[78] R. Chartrand, Exact reconstruction of sparse signals via nonconvex minimization, IEEE Signal Process. Lett. 14 (10) (2007) 707–710.
[79] M. Ovsjanikov, M. Ben-Chen, J. Solomon, A. Butscher, L. Guibas, Functional maps: a flexible representation of maps between shapes, ACM Trans. Graph. 31 (4) (2012) 30.
[80] R. Litman, A.M. Bronstein, M.M. Bronstein, Diffusion-geometric maximally stable component detection in deformable shapes, Computers & Graphics 35 (3) (2011) 549–560.
[81] D. Mumford, J. Shah, Optimal approximations by piecewise smooth functions and associated variational problems, Commun. Pure Appl. Math. 42 (5) (1989) 577–685.
[82] E. Elhamifar, R. Vidal, Sparse subspace clustering, in: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE, 2009, pp. 2790–2797.
[83] S. Wang, X. Yuan, T. Yao, S. Yan, J. Shen, Efficient subspace segmentation via quadratic programming., in: AAAI, 1, 2011, pp. 519–524.
[84] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 22 (8) (2000) 888–905.
[85] H. Zou, T. Hastie, R. Tibshirani, Sparse principal component analysis, J. Comput. Graph. Stat. 15 (2) (2006) 265–286.
[86] B. Brown, Statistical uses of the spatial median, J. R. Stat. Soc. Ser. B (1983) 25–30.
[87] C.G. Small, A survey of multidimensional medians, Int. Stat. Rev. (1990) 263–277.
[88] N.D. Cornea, D. Silver, P. Min, Curve-skeleton properties, applications, and algorithms, IEEE Trans. Visual. Comput. Graph. 13 (3) (2007) 530–548.
[89] O. Stava, J. Vanek, B. Benes, N. Carr, R. Měch, Stress relief: improving structural strength of 3D printable objects, ACM Trans. Graph. 31 (4) (2012) 48.

Please cite this article as: L. Xu et al., Survey on sparsity in geometry modeling and processing, Graphical Models (2015), http://dx.doi.org/10.1016/j.gmod.2015.06.012