

Sparsity in Geometry Processing[☆]

Abstract

Sparse representation has been an active research area in recent years.

Keywords: sparse regularization, dictionary learning, low rank, TV model

1. Introduction

Because of the fast development of Internet and other electronic equipments, the size of dataset is becoming incredibly massive. How to extract compact knowledge from such massive datasets is yet to be resolved. At the same time, the dimension of data becomes much higher than before. Thus how to extract low-dimensional structures from high-dimensional data is another serious problem in modern signal processing.

To solve these two challenging problems, sparsity-based approaches have been successfully introduced in many applications. Sparse representation which models data vectors as sparse linear combinations of basis elements, is widely used in machine learning, signal processing, neuroscience and statistics. Dictionary learning learns an over-complete dictionary which owns the ability to represent given signals. Low rank representation which decomposes a given matrix into a low rank matrix and residual with certain property. So far sparse techniques have become state-of-art tools in many fields like machine learning, data mining, computer vision, pattern recognition etc.

In geometric processing and computer graphics, people start to find out the advantages of sparse techniques. Better results are obtained with sparse techniques. At the same time, most formulations cannot be directly applied on geometric problems. Thus many non-trivial problems must be solved while applying sparse techniques. We would like to show how sparse technique, a strong tool in machine learning is brought into a fresh filed, geometric processing.

In the rest of this paper, we first introduce traditional sparse models used in machine learning and computer vision. Then we illustrate how people in

geometric processing use sparse techniques in different applications.

2. Preliminaries

The goal of sparse coding is to represent a given input vector \mathbf{y} in \mathbb{R}^k approximately as a linear combination of a small number of basis vectors $\{\mathbf{d}_i\}_{i=1}^n$ where $\mathbf{d}_i \in \mathbb{R}^k$. More precisely, any given input vector \mathbf{y} is compactly represented using basis vectors $\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_n$ and a sparse vector of weights or coefficients $\mathbf{a} = (a_1, a_2, \dots, a_n)^T \in \mathbb{R}^n$ such that $\mathbf{y} \approx \sum_i a_i \mathbf{d}_i$. Here the basis vectors can be overcomplete ($n > k$) and thus sparse coding can capture high-level patterns in the input.

To make this paper self contained, here we introduce some basic notations. Let $\mathbf{x} = (x_1, x_2, \dots, x_k)^T$ be any vector in Euclidean space \mathbb{R}^k , $\|\mathbf{x}\|_p$ is 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 $\|\mathbf{x}_0\| = \#\{i|x_i \neq 0\} = \sum_{i=1}^k |x_i|^0$. $M = (m_{ij})$ is denoted as a matrix in space $\mathbb{R}^{m \times n}$. Frobenius norm of M is given by $\|M\|_F = (\sum_{i=1}^m \sum_{j=1}^n m_{ij}^2)^{1/2}$. Another important norm is the nuclear norm for a matrix: $\|M\|_* = \sum_i \sigma_i(M)$ where $\sigma_i(M)$ is the i -th eigenvalue of M . Nuclear norm is the convex envelope of $\text{rank}(M)$, which makes that $\|M\|_*$ can be considered the relaxation of the rank of M .

3. The Problems of Sparse Coding

Generally, there are some basic assumptions in sparse techniques as an input signal can be represented by a sparse linear combination of basis functions or a matrix can be approximated by a low rank matrix. So there are several essential issues in this

⁷¹ field for instance finding the weights or coefficients
⁷² of the combination, learning the basis functions or
⁷³ vectors, rank minimization etc.

⁷⁴ 3.1. Sparse Coding

At the very first, we briefly introduce some popular sparse methods for machine learning. Recall that the target of sparse coding is to find as small number of basis vectors as possible to represent an input vector. A common formulation for this problem is that:

$$\begin{aligned} \min_{\mathbf{a}} \quad & \|\mathbf{a}\|_0 \\ \text{s.t.} \quad & \|\mathbf{y} - D\mathbf{a}\|_2 \leq \epsilon. \end{aligned} \quad (1)$$

Here $D = (\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_n)$ denotes the dictionary matrix collecting all basis vectors. Unfortunately, $\|\mathbf{d}\|_0$ is not differentiable or even continuous. Thus this optimization problem is a NP-hard problem and cannot be easily solved. Generally, we are not able to obtain the optimal solution of Problem 1. But many approximated solvers have been created to solve this problem. Matching pursuit (MP) [1], orthogonal matching pursuit (OMP) [2, 3] iteratively add the best basis vector to represent \mathbf{x} which is an approximating solution of problem:

$$\begin{aligned} \min_{\mathbf{a}} \quad & \|\mathbf{y} - D\mathbf{a}\|_2^2 \\ \text{s.t.} \quad & \|\mathbf{a}\|_0 \leq s \end{aligned} \quad (2)$$

⁷⁵ Here s denotes the number of chosen basis vectors
⁷⁶ which stands for the sparsity of this problem. Under certain assumptions [2], the solution of OMP
⁷⁷ algorithm converges to the real solution.

Another widely-used technique for solving sparse problem is to use ℓ_1 norm to replace ℓ_0 norm in Problem 1. The least absolute shrinkage and selection operator (LASSO) [4] method uses the constraint that $\|\mathbf{a}\|_1$ is no greater than a given value ϵ

$$\begin{aligned} \min_{\mathbf{a}} \quad & \|\mathbf{y} - D\mathbf{a}\|_2^2 \\ \text{s.t.} \quad & \|\mathbf{a}\|_1 \leq \epsilon \end{aligned} \quad (3)$$

This problem is also equivalent to an unconstrained minimization with a specific β of:

$$\min_{\mathbf{a}} \quad \|\mathbf{y} - D\mathbf{a}\|_2^2 + \beta \|\mathbf{a}\|_1 \quad (4)$$

⁷⁹ The problem is solved using general convex optimization methods like quadratic programming, as
⁸⁰ well as by specific approaches like the least angle regression [5] algorithm.

⁸³ Sparse coding or sparse representation, on the other way has been applied on many kinds of problems [6] as face recognition [7], image super-resolution [8], image classification [9].

⁸⁷ 3.2. Dictionary Learning.

Sparse coding or representation aims at finding the sparse coefficient while assuming the dictionary is comprised of given basis. On the other hand, dictionary learning algorithms want to optimize the dictionary at the same time given a set of training data. Classical dictionary learning techniques for sparse coding [10, 11] consider a training set of signals or data $Y = [\mathbf{y}_1, \dots, \mathbf{y}_n]$ and optimize the empirical cost function

$$f_n(D) \triangleq \frac{1}{n} \sum_{i=1}^n c(\mathbf{y}_i, D) \quad (5)$$

where D in $\mathbb{R}^{k \times m}$ is the dictionary, each column representing a basis, and $c(\mathbf{y}, D)$ is a loss function that $c(\mathbf{y}, D)$ should be small if D represents the signal \mathbf{y} well in a sparse fashion. [12] introduces a sparse constraint assuming that each signal is represented by a certain number of basis vectors. The problem is specified as:

$$\begin{aligned} \min_{D, A} \quad & \|Y - DA\|_F^2 \\ \text{s.t.} \quad & \|\mathbf{a}_i\|_0 \leq s, \forall i = 1, 2, \dots, m \end{aligned} \quad (6)$$

⁸⁸ where the dictionary $D = (\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_m)$ is also obtained as the coefficient matrix A is obtained.
⁸⁹ As we can see, it is very difficult to optimize D and A at the same time. [12] introduces the K-SVD algorithm, an iterative method that alternates between sparse coding and dictionary updating.

⁹⁴ 3.3. Compressed Sensing

⁹⁵ 3.4. Low rank representation and robust principle component analysis

Sparse-based matrix factorization technique is another hot field in machine learning. A typical technique is low rank representation which attempts to decompose any given matrix M into a low rank matrix L and a residual matrix S . The residual matrix S may have some specific properties like the input is corrupted by Gaussian noise or sparse noise. Under the assumption that M is corrupted by Gaussian noise, a low rank problem is formulated as:

$$\begin{aligned} \min_{L, S} \quad & \|S\|_F \\ \text{s.t.} \quad & \text{rank}(L) \leq r \\ & M = L + S \end{aligned} \quad (7)$$

⁹⁷ here $r \ll \min(m, n)$ where M is a $m \times n$ matrix. The above problem is equivalent to principle component analysis (PCA) according to [13, 14]. This problem is easily solved by first computing the singular value decomposition (SVD) of M and then projecting the

¹⁰² columns of M onto the subspace spanned by the r
¹⁰³ principle left singular vectors.

However, PCA assumes that corruption is caused by Gaussian noise. The result of PCA can be arbitrarily far from M if only a few entry of M is corrupted. [7, 15] show that under some conditions that S is rather sparse, one can exactly recover M by solving

$$\begin{aligned} \min_{L,S} \quad & \|L\|_* + \lambda \|S\|_1 \\ \text{s.t.} \quad & M = L + S \end{aligned} \quad (8)$$

The formulation is obtained by relaxing following problem replacing the ℓ_0 norm with the ℓ_1 norm:

$$\begin{aligned} \min_{L,S} \quad & \text{rank}(L) + \lambda \|S\|_0 \\ \text{s.t.} \quad & M = L + S \end{aligned} \quad (9)$$

¹⁰⁴ [16, 7] shows the uniqueness of the solution
¹⁰⁵ and [17] discusses efficient algorithm for solving low
¹⁰⁶ rank problem. Generally, problem 8 can be treated
¹⁰⁷ as a general convex optimization problem and
¹⁰⁸ solved by any off-the-shelf interior point solver (like
¹⁰⁹ CVX [18]).

¹¹⁰ *Numerical solution.* There have been several popular ways to solve low-rank (robust PCA) problem (8): The iterative thresholding approach introduced in [7] solves a relaxed convex problem of (8); the accelerated proximal gradient approach [19] is applied to a relaxed version of RPCA problem; [20] tackles the problem via its dual; the methods of augmented Lagrange multipliers [17] is introduced to efficiently solve RPCA problem.

Here we take a closer look at how augmented Lagrange multipliers (ALM) is applied on solving (8). According to [21], the general approach of augmented Lagrange multipliers is utilized to solve constrained optimization problems:

$$\begin{aligned} \min \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & h(\mathbf{x}) = 0 \end{aligned} \quad (10)$$

Here $f : R^n \rightarrow R$ and $h : R^n \rightarrow R^m$. the augmented Lagrangian function is defined as:

$$Lag(\mathbf{x}, \Lambda, \mu) = f(\mathbf{x}) + \langle \Lambda, h(\mathbf{x}) \rangle + \frac{\mu}{2} \|h(\mathbf{x})\|_F^2 \quad (11)$$

Then \mathbf{x} and Λ are updated iteratively solving subproblems (see [22] for more details). Thus the augmented Lagrangian function of (8) is:

$$\begin{aligned} Lag(L, S, \Lambda, \mu) = & \|L\|_* + \lambda \|S\|_1 \\ & + \langle \Lambda, M - L - S \rangle \\ & + \frac{\mu}{2} \|M - L - S\|_2^2 \end{aligned} \quad (12)$$

Table 1: Comparison of applications between image and geometry.

Domain	Image	Geometry
Smoothness	Regular	Irregular
	Piecewise constant	Piecewise Smooth

¹¹⁹ Then L , S and Λ are separately updated by fixing
¹²⁰ other variables and minimizing augmented La-
¹²¹ grangian function [17].

¹²² Low rank representation or robust principle com-
¹²³ ponent analysis has been successfully applied onto
¹²⁴ different problems like face recognition [23], latent
¹²⁵ object detection [24], video denoising [25].

¹²⁶ Before reviewing the usage of sparsity in geom-
¹²⁷ etry modeling and processing, Table 2 gives an
¹²⁸ overview about the effectiveness of sparsity in all
¹²⁹ the papers.

4. Sparse Regularization

4.1. Point Cloud Consolidation

¹³⁰ Point cloud consolidation, known as reconstruct-
¹³¹ ing the geometry of a shape from scanned data, is
¹³² a convenient and direct way to obtain 3D models.
¹³³ It can be a preprocessing phase for some geom-
¹³⁴ etry problem, e.g., surface reconstruction whose re-
¹³⁵ sult is a mesh object, with functionalities such as
¹³⁶ denoising, outlier removal, orientation, and redis-
¹³⁷ tribution of the input points. However, even with
¹³⁸ high-fidelity scanners, a variety of acquisition er-
¹³⁹ rors, like noise, outliers, missing data(holes) or reg-
¹⁴⁰ istration artifacts, are inevitable in the produced
¹⁴¹ large amount of raw, dense point sets. Then finding
¹⁴² a robust consolidation technique has always been an
¹⁴³ active researching area.

4.1.1. ℓ_1 median based

¹⁴⁴ Reconstruction by a projection operator, as
¹⁴⁵ shown in Figure 1, is to approximate the origin
¹⁴⁶ point set(green) by iteratively projecting an ar-
¹⁴⁷ bitrary point set(red) onto itself while removing
¹⁴⁸ the noises or outliers. It has an important virtue:
¹⁴⁹ it defines a consistent geometry based on the da-
¹⁵⁰ ta points, and provides constructive means to up-
¹⁵¹ sample it.

¹⁵² ℓ_1 median[51, 52], closely related to projection
¹⁵³ operator, is a statistical tool applied globally to
¹⁵⁴ multivariate non-parametric point-samples in the p-
¹⁵⁵ resence of noises and outliers. Briefly, it is a robust
¹⁵⁶ global center of an arbitrary set of points. Given
¹⁵⁷ a data set $P = \{p_j\}_{j \in J} \subset \mathbb{R}^3$, the ℓ_1 median is

Table 2: An overview about the effectiveness of sparsity in all the papers.

Application	Method			Effectiveness of Sparsity
	Sparse Regularization	Dictionary Learning	Low Rank	
Point Cloud Consolidation				
LOP[26] WLOP[27] CLOP[28] TV(ℓ_1) based[29] Subdivision[30] LRSCPK[31]	★ ★ ★ ★ ★		★	robust to noises, outliers robust to noises, outliers robust to noises, outliers robust to noises, outliers robust to noises, outliers sharp feature preserving
Mesh Denoising				
ℓ_0 -norm of Edge Operator[32] ℓ_1 -analysis Compressed Sensing[33] TV ℓ_1 based[34]	★ ★ ★			sharp feature preserving sharp feature preserving sharp feature preserving
Shape Matching				
Rigid[35] Non-rigid[34] Co-segmentation[36]	★ ★ ★			robust to noises, outliers constraining invariance of intrinsic properties dimension reduction
Compression				
Point Cloud[37] Rendering[38]		★ ★		high compression ratio, robust to noise real-time rendering, small reconstruction error, low memory
Decomposition				
Mesh[39] Mesh Sequence[40]	★	★		improving smoothness of decomposition boundaries versatile with the local controllability
Deformation				
SSDR[41] Blend Skinning Compression[42] Skeletal Rigging[43] Constrained Mesh Editing[44]		★ ★ ★		low memory, robust to noise large reduction of computational cost low memory local controllability
Upright Orientation				
Low Rank[45] Tensor Rank[46]		★ ★		intuitive, relatively robust capturing global symmetry, relatively robust
Other Applications				
CMM[47] LBC[48] Skeleton Extraction[49] Reconstruction[50]	★ ★ ★	★		local controllability local controllability robust to noise and outlier sharp feature preserving, robust to noise and outlier, unifying geometry and connectivity

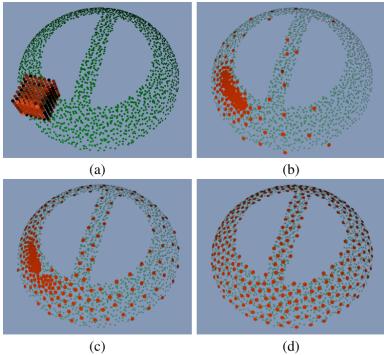


Figure 1: Reconstruction by projection operation. (a). noisy point-set P(green) and an arbitrary point-set Q(red) that will be projected to P to approximate P. (b),(c) are two iterative projection results. (d) is the final projection.

defined as the point q obtained by minimizing the sum of Euclidean distances to the data points

$$q = \arg \min_x \left\{ \sum_{j \in J} \|p_j - x\| \right\} \quad (13)$$

(1). [26] applies this tool locally to constitute a parameterization-free local projection operator(LOP).

Starting with an arbitrary initial point set $X^0 = \{x_i^0\}_{i \in I} \subset \mathbb{R}^3$ (typically $|X| \ll |P|$, $|\cdot|$ is the number of point set), LOP computes the target point positions X by performing a fixed-point iteration

$$X^{k+1} = \operatorname{argmin}_{X=\{x_i\}_{i \in I}} \{E_1(X^k, P) + E_2(X^k)\}, \quad (14)$$

where,

$$\begin{aligned} E_1(X^k, P) &= \sum_{i \in I} \sum_{j \in J} \|x_i - p_j\| \theta(\|x_i^k - p_j\|), \\ E_2(X^k) &= \sum_{i' \in I} \lambda_{i'} \sum_{i \in I \setminus \{i'\}} \eta(\|x_i - x_{i'}^k\|) \theta(\|x_i^k - x_{i'}^k\|). \end{aligned} \quad (15)$$

The term E_1 is in fact a *localized* version of (13) 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 . The term E_2 keeps the distribution of the points X fair by incorporating local repulsion forces.

To be convenient for the following works, now we give the expression of the solution. Let $\xi_{ij}^k = x_i^k - p_j$ and $\delta_{ii'}^k = x_i^k - x_{i'}^k$, solving (14), the projection for point x_i^{k+1} is obtained as

$$x_i^{k+1} = F_1(x_i^k, P) + F_2(x_i^k, X'_i) \quad (16)$$

where,

$$\begin{aligned} F_1(x_i^k, P) &= \sum_{j \in J} p_j \frac{\alpha_{ij}^k}{\sum_{j \in J} \alpha_{ij}^k}, \\ F_2(x_i^k, X'_i) &= \mu \sum_{i' \in I \setminus \{i\}} \delta_{ii'}^k \frac{\beta_{ii'}^k}{\sum_{i' \in I \setminus \{i\}} \beta_{ii'}^k}, \\ \alpha_{ij}^k &= \frac{\theta(\|\xi_{ij}^k\|)}{\|\xi_{ij}^k\|}, \quad \beta_{ii'}^k = \frac{\theta(\|\delta_{ii'}^k\|) |\eta'(\|\delta_{ii'}^k\|)|}{\|\delta_{ii'}^k\|}. \end{aligned} \quad (17)$$

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.

(2). Like[26], many consolidation methods try to obtain the resulted geometry 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, [27] incorporates locally adaptive density weights into LOP, resulting in a new consolidation technique WLOP, to address the non-uniform distribution problem while taking advantage of the success of LOP in denoising and outlier removal.

They define the weighted local densities for each point p_j in P and x_i in X during the k th iteration by $v_j = 1 + \sum_{j' \in J \setminus \{j\}} \theta(\|p_j - p_{j'}\|)$ and $w_i^k = 1 + \sum_{i' \in I \setminus \{i\}} \theta(\|\delta_{ii'}^k\|)$, the term F_1 and F_2 in (17) finally becomes

$$\begin{aligned} F_1(x_i^k, P) &= \sum_{j \in J} p_j \frac{\alpha_{ij}^k / v_j}{\sum_{j \in J} (\alpha_{ij}^k / v_j)} \\ F_2(x_i^k, X'_i) &= \mu \sum_{i' \in I \setminus \{i\}} \delta_{ii'}^k \frac{w_{i'}^k \beta_{ii'}^k}{\sum_{i' \in I \setminus \{i\}} (w_{i'}^k \beta_{ii'}^k)}, \end{aligned} \quad (18)$$

The weighted local density v in F_1 relaxes the attraction of point clusters and repulsion force in dense areas is strengthened by the local density w in F_2 .

Here, 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 , so [28] 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

$$f(\mathbf{x}|\mathcal{M}) = \sum_s w_s g(\mathbf{x}|\Theta_s), \quad (19)$$

where the $\Theta_s = (\mu_s, \Sigma_s)$ are the Gaussian parameters, w_s are their corresponding convex weights, and g denotes the d -variate Gaussian pdf.

They define a *continuous* \mathcal{F}_1 corresponding to F_1 in (17) by the convex sum over the internal attraction of each single Gaussian, with convex weights w_s accounting for the Gaussian's relative point mass:

$$\mathcal{F}_1(q, \mathcal{M}) = \sum_s w_s \int_{\mathbb{R}^3} \frac{\mathbf{x} g(\mathbf{x}|\Theta_s) \alpha(\mathbf{x})}{\sum_{s'}^{w_{s'}} \int_{\mathbb{R}^3} g(\mathbf{x}'|\Theta_{s'}) \alpha(x') d\mathbf{x}'} d\mathbf{x}, \quad (20)$$

and the final **closed form** is expressed as

$$\mathcal{F}_1(q, \mathcal{M}) = \frac{\sum_s \sum_k \int_{\mathbb{R}^3} \mathbf{x} \widehat{\Omega}_{sk}(\mathbf{x}) d\mathbf{x}}{\sum_s \sum_k \int_{\mathbb{R}^3} \widehat{\Omega}_{sk}(\mathbf{x}) d\mathbf{x}} = \frac{\sum_{s,k} w_{sk} \mu_{sk}}{\sum_{s,k} w_{sk}} \quad (21)$$

changing the convex sum of 3D points p_j (15) into a convex combination of the product Gaussians' means μ_{sk} with weights w_{sk} . Figure 2 shows the results of these three methods.

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 can not automatically get the best choice of the parameters for different point set.

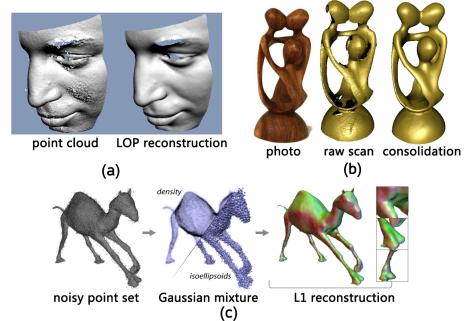


Figure 2: Sparse regularization: point cloud consolidation. (a): LOP[26]. (b): WOLP[27]. (c): continuous WLOP[28].

243 4.1.2. ℓ_1 regression based

244 Due to the robustness to noises and outliers of ℓ_1
 245 norm, [30] develops an ℓ_1 regression based subdivision
 246 algorithm for curve and surface fitting, where the size
 247 of target point cloud is largely more than that of origin
 248 data in contrast to the previous consolidation works.

249 For curve fitting, they try to find the best fit straight
 250 line $f(x) = \beta_1 + \beta_2 x$ with observations $(x_r = r, f_r), r =$
 251 $-n+1, \dots, n$. The ℓ_1 regression optimization is simply
 252 formulated as

$$\begin{aligned} \beta_1, \beta_2 &= \arg \min_{\beta_1, \beta_2 \in \mathbb{R}} \sum_{r=-n+1}^n |f_r - (\beta_1 + \beta_2 r)| \\ &= \arg \min_{\beta_1, \beta_2 \in \mathbb{R}} F(\beta_1, \beta_2), \end{aligned} \quad (22)$$

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

$$\begin{aligned} F_\delta(\beta_1, \beta_2) &= \sum_{r=-n+1}^n h_\delta(f_r - \beta_1 - \beta_2 r), \text{ where} \\ h_\delta(f_r - \beta_1 - \beta_2 r) &= [(f_r - \beta_1 - \beta_2 r)^2 + \delta]^{1/2} \end{aligned} \quad (23)$$

255 then for a given δ , the solution of (22) is approximated
 256 by $\beta_{1,\delta}$ and $\beta_{2,\delta}$. By substituting optimum $\beta_{1,\delta}, \beta_{2,\delta}$
 257 into $f(x)$ and evaluating this function at $1/4$ and $3/4$,
 258 the closed form of ℓ_1 scheme for curve fitting is obtained.

259 With the closed form, ℓ_1 scheme D_{2n} firstly iteratively
 260 assigns weights to only $2n$ local initial points, then
 261 gets the final fitting result(e.g.,Figure 3) through sub-
 262 division rule for locations of vertices of the new mesh
 263 and topological rule for size of added vertices and their
 264 connectivity.

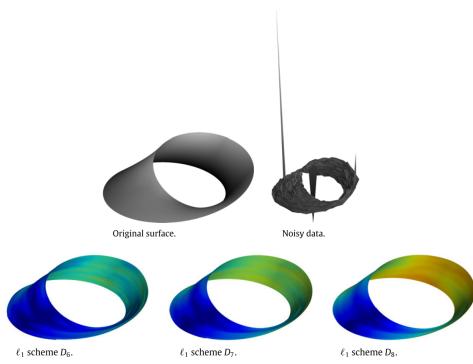


Figure 3: Sparse regularization: ℓ_1 based subdivision[30]. Parametric surface reconstructed by ℓ_1 scheme from highly noisy parametric data with outliers.

267 4.2. Mesh Denoising

268 Different from denoising in point cloud, for mesh sur-
 269 faces, there are vertex connectivity and triangle quality
 270 that can be used or considered. But how to distinguish
 271 features from noise is still a challenging problem.

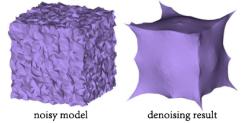
272 (1). In image processing, [53], aiming to smooth im-
 273 ages, provides an algorithm for directly optimizing the
 274 ℓ_0 norm of gradients of image colors to create piece-
 275 wise constant images. Let \mathbf{c} be a vector of pixel colors
 276 and $\nabla \mathbf{c}$ be a vector of gradients of these colors. They
 277 formulates the smooth problem as

$$\min_{\mathbf{c}} |\mathbf{c} - \mathbf{c}^*|^2 + |\nabla \mathbf{c}|_0 \quad (24)$$

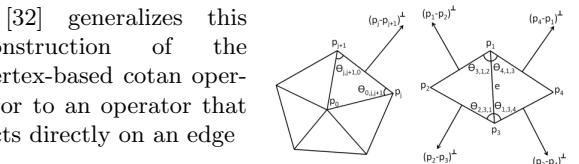
278 where \mathbf{c}^* represents the original image colors to provide
 279 a data fidelity term.

280 A natural extension to triangulated meshes is to de-
 281 sign a discrete differential operator to replace ∇c that is
 282 zero when the surface is flat for arbitrary triangulations
 283 irrespective of the rotation or translation of the surface.
 284 This constraint implies that some form of second order
 285 information rather than the first order information
 286 provided by ∇c is needed, e.g., the discrete Laplacian
 287 operator[54] which is computed as a weighted combina-
 288 tion of a vertex and its one-ring where the weights are
 289 given by cotangents of angles of the triangles. However,

290 the vertex-based Laplacian
 291 only constrains the mean curvature vector as opposed to
 292 a metric that should directly
 293 measure sharpness per edge,
 294 then the optimization fails to reproduce sharp features
 295 well shown as the right figure.



[32] generalizes this
 construction of the
 vertex-based cotan operator
 to an operator that
 acts directly on an edge



$$D(e) := \begin{bmatrix} \frac{\Delta_{1,2,3}((p_4-p_3)\cdot(p_3-p_1))}{\Delta_{1,3,4}} \\ \frac{\Delta_{1,3,4}((p_1-p_3)\cdot(p_3-p_2))}{\Delta_{1,3,4}} \\ \frac{\Delta_{1,2,3}+\Delta_{1,3,4}}{\Delta_{1,2,3,4}} \\ \frac{\Delta_{1,2,3}((p_3-p_1)\cdot(p_1-p_4))}{\Delta_{1,3,4}((p_2-p_1)\cdot(p_1-p_3))} \\ \frac{\Delta_{1,2,3}}{\Delta_{1,2,3}+\Delta_{1,3,4}} \end{bmatrix}^T \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix} \quad (25)$$

Then the extended optimization problem is to make
 the edge operator sparse formulated as following

$$\min_{p, \delta} |p - p^*|^2 + \alpha |R(p)|^2 + \beta |D(p) - \delta|^2 + \lambda |\delta|_0 \quad (26)$$

where p are the vertices of the shape, p^* are their initial positions, $D(p)$ is a vector where the i^{th} entry corresponds to the area-based edge operator applied to

the i^{th} edge, and $R(p)$ is a regularization term. Figure 4 gives one denoised result with sharp features.

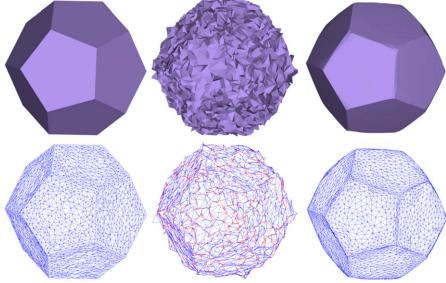


Figure 4: Sparse regularization: mesh denoising[32]. Left: initial surface. Center: surface corrupted by Gaussian noise in random directions with standard deviation $\sigma = 0.4l_e$ (l_e is the mean edge length). Right: denoising result. The wireframe shows folded triangles as red edges.

(2). Like most previous methods, how to tune the parameters shown in (26) has not theoretical guarantee and the computation of differential properties for distinguishing noise from feature is unreliable and unstable.

To address these problems, [33] presents a two-phase approach for decoupling features and noise on discrete surfaces. Figure 5(a) gives a two-dimensional 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. As such, the denoising problem is divided into two phase: smooth part(base mesh) estimation and recovering features from the corrupted feature part.

They firstly 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 x and noises simultaneously from the noisy feature part y via the ℓ_1 analysis compressed sensing optimization

$$\min_x \|Lx\|_1 \quad s.t. \quad \|y - x\|_2 \leq \epsilon \quad (27)$$

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.

4.3. Shape Matching

Here, we give shape matching a more extensive definition: finding the correspondence(point-wise, pair-wise)

between two rigid or non-rigid deformable geometric data sets.

282 4.3.1. Rigid registration

Rigid registration aims at finding a suitable set of corresponding points on source and target point set. The *Iterative Closest Point*(ICP) addresses this problem 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

$$\operatorname{argmin}_{R, t} \int_{\mathcal{X}} \varphi(R\mathbf{x} + t, \mathcal{Y}) d\mathbf{x} + I_{SO(k)}(R) \quad (28)$$

where R is a rotation matrix, t is a translation vector, \mathbf{x} is a point on the source geometry. The quality of a registration is evaluated by the metric $\varphi(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_2^2$, i.e., classical ICP is in a least-square sense which would fail with outliers.

Now that sparse regularization methods excels in processing data set with noises or outliers, [35] tries to formulate the local alignment problem as recovering rigid transformation that minimizes the number of zero distances between two correspondences. They adopt ℓ_p ($0 \leq p \leq 1$) norm based sparse regularizer to obtain an heuristic-free, robust rigid registration algorithm by modifying

$$\varphi(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_2^p \quad (29)$$

About ℓ_p norm, [55] shows that ℓ_p norms with $p < 1$ outperform the ℓ_1 norm in inducing sparsity and [56] also illustrates the tendency of ℓ_p ($0 < p < 1$) norms to drive results to become sparse. Figure 6 is the registration results of sparse ICP under different values of p among which it can be found that $0 < p < 1$ reduces 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.

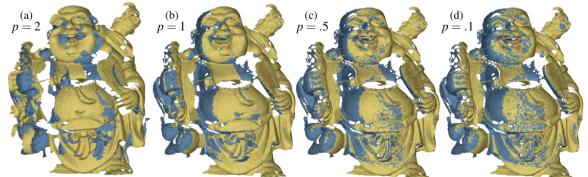


Figure 6: Sparse regularization: rigid registration results using sparse ICP[55] under different l_p norms.

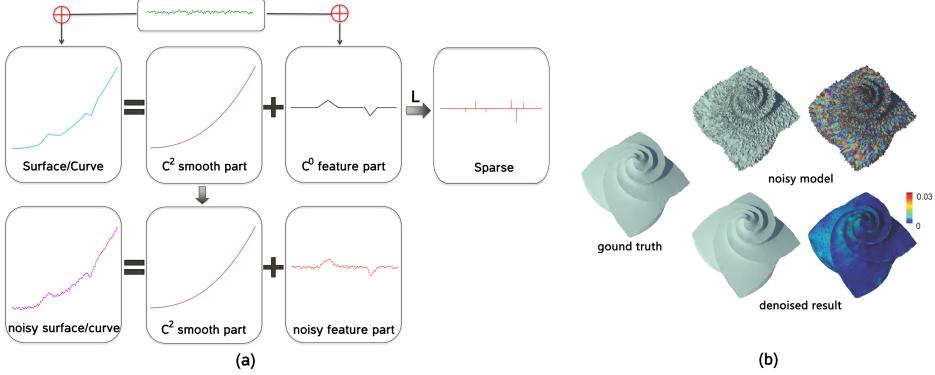


Figure 5: Sparse regularization: mesh denoising[33]. (a) is the two-dimensional illustration for their key observation. (b) is a denoising example.

283 4.3.2. 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, 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, which is closely related to the following non-rigid shape matching algorithm.

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

$$-\Delta\phi_k = \lambda_k\phi_k, \quad k \in \mathbb{N}, \lambda_k \in \mathbb{R}, \quad (30)$$

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

But as we have known that many times only locality can reduce a good result, like deformation([48] mentioned above), correspondence. So, to produce an intrinsic shape basis with local spatial support while taking advantage of MHB simultaneously, [47] proposes the *compressed manifold basis(CMB)*, whose individual basis functions are called *compressed manifold modes(CMMs)*, by adding a sparsity inducing ℓ_1 norm into (30)

$$\begin{aligned} & \min_{\phi_k} \sum_{k=1}^K \langle \phi_k, \Delta\phi_k \rangle + \mu|\phi_k|_1, \\ & \text{s.t. } \langle \phi_k, \phi_j \rangle = \delta_{kj}, \end{aligned} \quad (31)$$

where δ_{kj} is the Kronecker delta used to enforce orthogonality of the eigenfunctions and μ is used to control the sparsity. For a triangle mesh with N vertices, discretizing the Laplacian Δ using a sparse matrix $L \in \mathbb{R}^{N \times N}$

with cotangent weights in previous work, and incorporating a lumped mass matrix M , containing the vertex areas along its diagonal making the eigenbasis independent of the mesh resolution, the discretization of (31) becomes

$$\begin{aligned} & \min_{\Phi} \text{Tr}(\Phi^T L \Phi) + \mu\|\Phi\|_1, \\ & \text{s.t. } \Phi^T M \Phi = I. \end{aligned} \quad (32)$$

here, $\Phi \in \mathbb{R}^{N \times K}$ contains the first K eigenvectors corresponding to the matrix columns. Solving (32), the obtained orthogonal *compressed manifold modes(CMMs)* could automatically identify key shape features of the underlying mesh, as shown in Figure 7. As such, it can be used for shape matching which involves robust feature detection.

(2) *Non-rigid shape 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, [57] introduces a functional representation for correspondences which are modeled as the correspondences between functions on two shapes rather than points. Mathematically, let X and Y be two shapes equipped with bases $\{\phi_i\}_{i \geq 1}$ and $\{\psi_j\}_{j \geq 1}$ respectively, any real function $f : X \rightarrow \mathbb{R}$ and $g = T(f) : Y \rightarrow \mathbb{R}$ can be represented as $f = \sum_{i \geq 1} a_i \phi_i$ and $g = \sum_{j \geq 1} b_j \psi_j$. Taking discretized functions ϕ_i and ψ_j as the columns of bases matrices Φ and Ψ , the function vectors can be represented as $\mathbf{f} = \Phi \mathbf{a}$ and $\mathbf{g} = \Psi \mathbf{b}$, and then from $\Psi \mathbf{b} = T(\Phi \mathbf{a}) = \Psi C^T \mathbf{a}$, the relationship between two coefficients is clear that $\mathbf{b}^T = \mathbf{a}^T C$. Thus, the matrix C fully encodes the linear map T between the functional spaces.

In case the shapes X and Y are isometric and the corresponding Laplace-Beltrami operators have simple

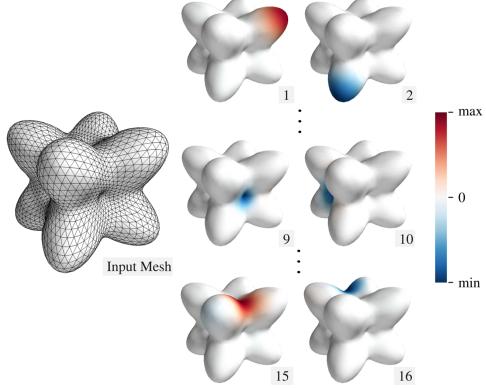


Figure 7: Sparse regularization: local functional basis[31]. The proposed compressed manifold modes(CMMs) have local support and are confined to specific local features like protrusions and ridges. 8 of the CMMs were found for the 8 protrusion at the corner(2 shown here), 6 concentrate at each of the dents(2 shown here), and 12 CMMs automatically form at the valleys between the protrusions.

spectra mentioned above, the harmonic bases(Laplacian eigenfunctions) have a compatible behavior, $\psi_i = T(\phi_i)$ such that $c_{ij} = \delta_{ij}$. Choosing the discretized eigenfunctions of the Laplace-Beltrami operator as Φ and Ψ causes every low-distortion correspondence being represented by a nearly diagonal, and therefore very sparse matrix C .

Based on the above theory, [58] firstly gets two collections of similar functions $\{f_i : X \rightarrow \mathbb{R}\}$ and $\{g_j : Y \rightarrow \mathbb{R}\}$ using some region detection process like[59].

As shown in the right figure, different colors represent different functions and the correspondence of these two collections of functions is unknown, i.e., we do not know to which g_j in Y a f_i in X corresponds. [58] adopts an unknown permutation matrix Π to express this ordering. Finally, the robust permuted sparse coding is formulated as following

$$\min_{C, O, \Pi} \frac{1}{2} \|\Pi B - AC - O\|_F^2 + \lambda \|W \odot C\|_1 + \mu \|O\|_{2,1} \quad (33)$$

where W is assigned with larger weights in off-diagonal part and small weights in diagonal part to promote diagonal solutions, $\|O\|_{2,1}$ promotes row-wise sparsity allowing to absorb the errors in the data term corresponding to the rows of A having no corresponding rows in B . From the formulation we know that this method relies on the region detection technique and assumption: near-isometric shapes. Figure 8 shows the correspondences between non-isometric shapes.

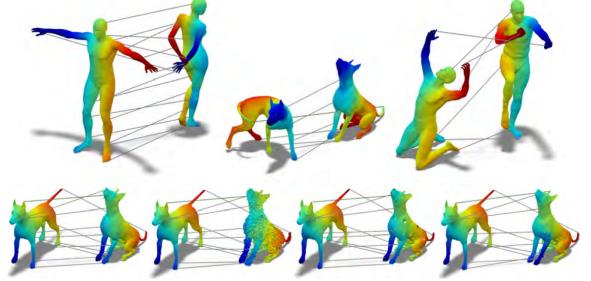


Figure 8: Sparse regularization: non-rigid shape matching [33]. First row: point-to-point correspondences between different non-isometric shapes. Second row: point-to-point correspondence between SHREC shapes undergoing nearly isometric deformations and noise.

4.3.3. Co-segmentation

Co-segmentation aims to consistently segment a group of shapes and obtain the correspondence between resulted segments simultaneously, as the right figure in Figure 10 shows, corresponding parts are labeled in the same colors. To be more intuitive and efficient, [36] processes co-segmentation on patch-level instead of face-level.

Thus they firstly over-segment all the models(left in Figure 10) followed by calculating their feature vectors using some feature descriptors(in this paper, they adopt $H = 5$ feature descriptors). For example, Figure 9 shows the colormaps of average geodesic distance(AGD) features of two tables with over-segmented patches. They define the feature vector as a histogram of the feature measurement on the triangles of that patch. It is obvious that two corresponding patches have similar distributions, that is, their feature vectors lie in a common subspace generated by standard basis corresponding to these nonzero entries. Based on this observation, they regard co-segmentation as a subspace clustering problem since the final segments are all clustering of over-segmented patches.

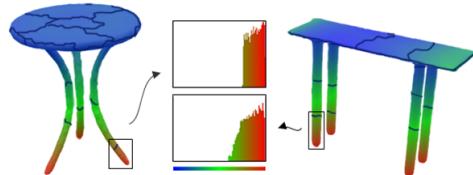


Figure 9: Sparse regularization: co-segmentation[36]. Colormaps of AGD features of two tables with over-segmented patches. The AGD feature vectors of the two patches(marked in rectangles) from each table's leg have similar distribution, as shown in histograms in the middle. It can be seen that these two feature vectors lie in the common subspace generated by standard basis corresponding to the nonzero entries.

Since each data point(here is the feature vector) in a union of linear subspaces can always be represented as a linear combination of the points belonging to the same linear subspace, the combination will be sparse if the point is written as a linear combination of all other points. Following[60, 61], finding the sparse combination matrix for the single-feature co-segmentation is formulated as

$$\begin{aligned} \min_{W_h} & \|X_h W_h - X_h\|_F^2 + \lambda \|W_h^T W_h\|_{1,1} \\ \text{s.t. } & W_h \geq 0, \text{ diag}(W_h) = 0 \end{aligned} \quad (34)$$

where h corresponds to the h -th feature descriptor, the feature matrix $X_h = [x_{h1}, x_{h2}, \dots, x_{hN}]$ is constructed with the feature vector x_{hi} of the i -th patch($i = 1, 2, \dots, N$). $\|W_h^T W_h\|_{1,1}$, as a penalty item, favors the sparsity of the optimal solution W_h of which each entry measures the linear correlation between two points in the meshes. After defining the affinity matrix $S = (s_{ij})$ as $s_{ij} = |\bar{w}_{hij}| + |\bar{w}_{hji}|$, the NCut method[62] is applied to get the co-segmentation results.

In fact, single one feature is not enough for co-segmenting different categories of models. To find the most similar patch pairs considering all selected features some of which the corresponding patches may not be similar , [36] adds the consistent multi-feature penalty to ensure the co-segmentation results consistent with different feature spaces by combing H feature descriptors

$$\begin{aligned} \min_{W_1, \dots, W_H} & \sum_{h=1}^H \mathcal{F}(W_h) + \mathcal{P}_{cons}(W_1, W_2, \dots, W_H) \\ \text{s.t. } & W_h \geq 0, \text{ diag}(W_h) = 0, h = 1, 2, \dots, H. \end{aligned} \quad (35)$$

where \mathcal{P}_{cons} is the penalty on the matrices W_1, W_2, \dots, W_H

$$\mathcal{P}_{cons}(W_1, W_2, \dots, W_H) = \alpha \|W\|_{2,1} + \beta \|W\|_{1,1} \quad (36)$$

here the $H \times N^2$ matrix W is formed by concatenating W_1, W_2, \dots, W_H (each matrix in one row) together:

$$W = \begin{bmatrix} (W_1)_{11} & (W_1)_{12} & \cdots & (W_1)_{N^2} \\ (W_2)_{11} & (W_2)_{12} & \cdots & (W_2)_{N^2} \\ \vdots & \vdots & \ddots & \vdots \\ (W_H)_{11} & (W_H)_{12} & \cdots & (W_H)_{N^2} \end{bmatrix} \quad (37)$$

the $\ell_{2,1}$ penalty induces column sparsity of W such that most columns of W are shrunken to be entirely zero, which means 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 W_1, W_2, \dots, W_H .

Notice that without \mathcal{P}_{cons} , the formulation (35) will reduce to a naive solution which is exactly the same as applying subspace clustering to each feature matrix X_h independently.

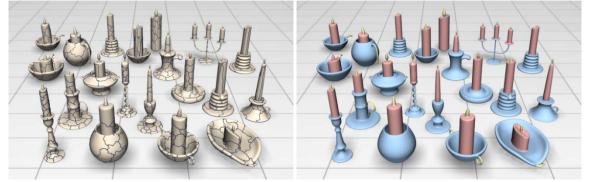


Figure 10: Sparse regularization: co-segmentation[36]. Left shows the over segmented patches that will be clustered to get the co-segmentation result.

4.4. Skeleton Extraction

In section 4.1.1, we have introduced much information about ℓ_1 median and its success in point cloud consolidation. Except for reducing 2D surface that approximate origin point set, [49] observed that adapting ℓ_1 medians *locally* to a point set representing a geometric shape also gives rise to a *one dimensional* structure which can be seen as a localized center of the shape, i.e., a medial curve skeleton, which can be used for shape abstraction and consequently an effective tool for shape analysis and manipulation[63].

Without building any point connectivity or estimating point normals, by modifying the repulsion term E_2 in (15) and proposing a different weighted density parameter that can also be named WLOP[27], 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. Figure 11 shows an example.

4.5. Deformation-Constrained Modeling

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, [44] gives a novel framework using $\ell_{2,1}$ sparse regularization penalty.

For a mesh with vertices $\{v_i\}_1^n$, $\mathbf{p} = [\mathbf{p}_1^T, \dots, \mathbf{p}_n^T]^T \in \mathbb{R}^{3n}$ is the position vector. $E_j(\mathbf{p}) = 0 (j = 1, \dots, m)$

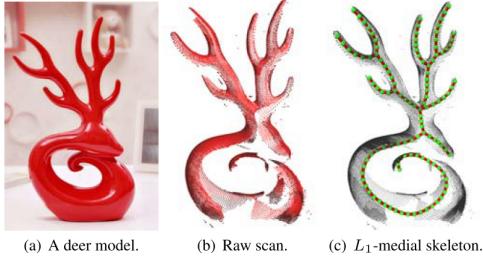


Figure 11: Sparse regularization: skeleton extraction[49]. Given an unorganized, unoriented, and incomplete raw scan with noise and outliers(b), a complete and quality curve skeleton is extracted(c).

denotes the constraints satisfactory. Then from a given mesh \mathbf{p}^0 and the target positions of vertices(handles) $\{v_i | i \in \Gamma\}$ specified by a user, the deformation is achieved by computing the displacement $\mathbf{d} = [\mathbf{d}_1^T, \dots, \mathbf{d}_n^T]^T \in \mathbb{R}^{3n}$, with $\mathbf{d}_k \in \mathbb{R}^3$ corresponding to vertex v_k , by the following optimization problem

$$\begin{aligned} & \min_{\mathbf{d}} \frac{\omega_h}{2} \sum_{i \in \Gamma} \|\mathbf{d}_i - \tilde{\mathbf{d}}_i\|_2^2 + \frac{\omega_s}{2} \sum_{i \notin \Gamma} \|\mathbf{d}_i\|_2 + \frac{\omega_f}{2} \|\mathbf{E}\mathbf{d}\|_2^2, \\ & \text{s.t. } E_j(\mathbf{p}^0 + \mathbf{d}) = 0, \quad j = 1, \dots, m, \end{aligned} \quad (38)$$

where $\tilde{\mathbf{d}}_i$ in the first term is the target displacement of the handle vertex v_i , the second $\ell_{2,1}$ term minimizing the ℓ_1 norm of vector $[\|\mathbf{d}_1\|_2, \dots, \|\mathbf{d}_n\|_2]$ reduces the sparsity of \mathbf{d} , and last term is for a smooth displacement for a nice shape. With fixed weight $\omega = (\omega_h, \omega_s, \omega_f)$, the resulted single solution \mathbf{d} may not satisfy the user's intent.

Then to enrich the solution while preserving nice shapes of meshes, based on the modified mesh $\mathbf{p}^0 + \mathbf{d}$ denoted as \mathbf{p}^+ , they compute a local modification space $S_{\mathbf{p}^+} = \{\mathbf{p}^+ + \mathbf{t} \mid \mathbf{t} \in \mathcal{S}_{\mathbf{p}^+}\}$ spanned by an orthonormal basis $\mathbf{t}_1, \dots, \mathbf{s}_1$, where $\mathcal{S}_{\mathbf{p}^+}$ is a linear subspace of $\mathcal{T}_{\mathbf{p}^+} = \{\mathbf{t} \mid \nabla E_j(\mathbf{p}) \cdot \mathbf{t} = 0, j = 1, \dots, m\}$ representing displacements from \mathbf{p} that satisfy the constraints up to first order. Also taking shape quality and sparsity into consideration, this problem is finally formulated as

$$\begin{aligned} & \min_{\mathbf{t}} \frac{\beta_f}{2} \|\mathbf{E}\mathbf{d}\|_2^2 + \frac{\beta_h}{2} \sum_{i=1}^s \|\mathbf{t}_i\|_2^2 + \frac{\beta_s}{2} \|\mathbf{t}\|_{2,1} - \frac{\beta_c}{2} C(\mathbf{t}), \\ & \text{s.t. } \|\mathbf{t}\|_2 = 1, \quad \mathbf{J}\mathbf{t} = \mathbf{0}, \quad \mathbf{B}^T \mathbf{t} = 0. \end{aligned} \quad (39)$$

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

After the interactive exploration, the final result is optimized to fully satisfy the set of constraints(Figure 12).

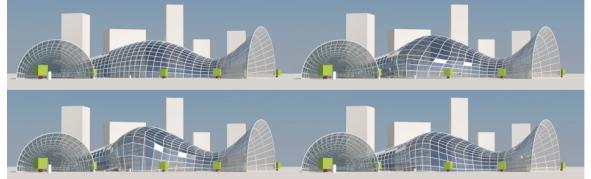


Figure 12: Sparse regularization: constraint modeling[44]. Local modifications of a constrained mesh. A glass structure composed of planar quads is locally deformed by exploring a subspace encoding local planar modifications of its central zone.

4.6. Total Variation(TV) Based Applications

Total variation(TV) has been a popular tool for image processing tasks,such as denoising, reconstruction, and segmentation[64]. the underlying model for TV methods aims at exploiting the sparsity of the gradient of image pixel values. The discrete variant yields the following convex objective function

$$TV(\mathbf{u}) = \sum_{i,j} \|\mathcal{D}_{i,j} \mathbf{u}\|_2 \quad (40)$$

where $\mathcal{D}_{i,j}$ is the discrete gradient operator at pixel (i, j) and \mathbf{u} is a vector containing the gray-level pixel values. TV methods filter the image by minimizing $TV(\mathbf{u})$ which is in fact the ℓ_1 norm of the vector $[\dots \|\mathcal{D}_{i,j} \mathbf{u}\|_2 \dots]$. Since TV is designed for images, it is not directly applicable to geometry processing problem. As we have stated, the key point is to find some form of second order information.

Note that some of the following applications have been reviewed in previous sections, but considering the good development of TV in geometry processing, we take one subsection to introduce its applications attempting to help readers understands it better.

4.6.1. Point cloud consolidation

In section 4.1, the consolidation works are all ℓ_1 median based. Here we review one more well-known work. Similar to the sparse gradient minimization, and based on the observation that the gradients(normal differences) of smooth surface normals(normal differences) are sparse, [29] formulates the piecewise smoothness reconstruction problem as a sparse minimization of orientation differences and position projections as following

$$\begin{aligned} N^{out} = \operatorname{argmin}_N \sum_{(p_i, p_j) \in E} w_{i,j} \|n_i - n_j\|_2 \\ \text{s.t. } \forall i \|n_i - n_i^{in}\|_2 \leq \gamma_n \end{aligned} \quad (41)$$

$$X^{out} = \operatorname{argmin}_X \sum_{(p_i, p_j) \in E} w_{i,j} |n_{i,j} \cdot (x_i - x_j)| \quad (42)$$

where $\{n_i\}$ denote the surface normals, $\{x_i\}$ 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. Figure 13 shows a well reconstructed example with sharp features. Due to the global nature, this algorithm is extremely slow. And it may fail for the point set with severe noises and outliers.



Figure 13: Sparse regularization: TV based point cloud consolidation[26]. The Armadillo statue(left) is scanned generating a noisy point-cloud(middle). The right figure shows the consolidation result preserving the sharp features.

289 4.6.2. Mesh Denoising

Unlike[32] achieving denoising with a new proposed edge operator, [34], adopting the sparsity of face normals differences, proposes a two-phase method including *face* normal filtering and vertex updating. They filter face normals with a new variational denoising method based on the definition of piecewise constant function spaces and associated differential operators on triangulated surfaces.

Piecewise constant function spaces and operators. For a space $V_M = \mathbb{R}^T$ equipped with well-defined inner product and norm, it is isomorphic to the piecewise constant function space over surface M , i.e., $u = (u_0, u_1, \dots, u_{T-1}) \in V_M$, T is the triangle number in M and u_τ denotes the value of u restricted on the triangle τ . After defining the jump of $u \in V_M$ over an edge e as

$$[u]_e := \begin{cases} \sum_{e \prec \tau} u|_\tau \operatorname{sgn}(e, \tau), & e \not\subseteq \partial M \\ 0, & e \subseteq \partial M \end{cases} \quad (43)$$

the gradient operator is defined as

$$\nabla : u \rightarrow \nabla u, \nabla u|_e = [u]_e, \forall e, \text{ for } u \in V_M. \quad (44)$$

here, $e \prec \tau$ indicates that e is an edge of the triangle τ , then if the orientation of e is consistent with the orientation of τ , $\operatorname{sgn}(e, \tau) = 1$; otherwise $\operatorname{sgn}(e, \tau) = -1$.

Due to the ambiguity of the tangent space at an edge, they define another space $Q_M = \operatorname{Range}(\nabla)$. The divergence operator, $\operatorname{div}: Q_M \rightarrow V_M$, as the adjoint operator of $-\nabla$, is formulated as

$$(\operatorname{div} p|_\tau) = \frac{-1}{s_\tau} \sum_{e \prec \tau, e \not\subseteq \partial M} l_e p|_e \operatorname{sgn}(e, \tau), \forall p \in Q_M. \quad (45)$$

where, s_τ is the area of triangle τ , l_e is the length of edge e . Thus for $u \in V_M$, the total variation is

$$R_{\operatorname{tv}}(\nabla u) = (\operatorname{TV})(u) = \sum_e l_e |(\nabla u)|_e = \sum_e l_e |[u]_e|. \quad (46)$$

here, l_e just meets the perimeter formulae defined using total variation of the characteristic function. To be more suitable to practical applications, they extend the above definition to vectorial case, for \mathfrak{N} -channel data

$$\mathbf{V}_M = \underbrace{V_M \times \dots \times V_M}_{\mathfrak{N}}, \mathbf{Q}_M = \underbrace{Q_M \times \dots \times Q_M}_{\mathfrak{N}} \quad (47)$$

and the vectorial total variation of $\mathbf{u} \in \mathbf{V}_M$ is

$$R_{\operatorname{vvtv}}(\nabla \mathbf{u}) = (\operatorname{TV})(\mathbf{u}) = \sum_e \sqrt{\sum_{i=1}^{\mathfrak{N}} l_e |[u_i]|_e^2}. \quad (48)$$

Variational model. Based on these definitions, they give a new variational face normals(\mathbf{N}) filtering method formulated as

$$\min_{\mathbf{N} \in C_{\mathbf{N}}} \left\{ R_{\operatorname{wvtv}}(\nabla \mathbf{N}) + \frac{\alpha}{2} \|\mathbf{N} - \mathbf{N}^{in}\|_{\mathbf{V}_M}^2 \right\}, \quad (49)$$

where

$$R_{\operatorname{wvtv}}(\nabla \mathbf{N}) = \sum_e w_e \sqrt{\sum_{i=1}^3 l_e |[N_i]|_e^2}, \quad (50)$$

$$C_{\mathbf{N}} = \{\mathbf{N} \in \mathbf{V}_M : \|\mathbf{N}_\tau\|_2 = 1, \forall \tau\}$$

with w_e as a weight aiming for improving preserving sharp features. Iteratively solving this variational model (49) and updating vertex using previous method, the denoising results preserving sharp features can be obtained as shown in Figure 14. Like many other optimization problem, the optimal values of the parameters, like α , are given by the experimental data and there is of course no theoretical convergence guarantee.

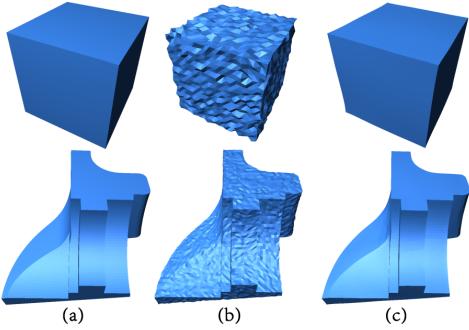


Figure 14: Sparse regularization: TV based mesh denoising[34]. (a): clean meshes. (b): noisy mesh(Gaussian noise, standard deviation=0.2 mean edge length for Cube; standard deviation=0.1 mean edge length for Fandisk). (c): denoising result.

290 4.6.3. Decomposition

Mesh decomposition means segmenting a mesh into meaningful parts that are consistent with user intention, 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 obviously the segmentation should reflect significant features.

Motivated by the preceding observation, [39] proposes a new method based on the Mumford-Shah model(M-S model)[65] that has proven successful in image segmentation, i.e., this method is also an extension from 2D images to 3D meshes.

In 2D image $I : \Omega \rightarrow \mathbb{R}^2$, the Mumford-Shah image segmentation is to find a partition $\Omega = \bigcup_{i=1}^k \Omega_i$, where Ω_i are pairwise disjoint, and numbers c_i for Ω_i formulated as

$$\inf_{\Omega_i, c_i} \sum_{i=1}^k \left(\int_{\Omega_i} (I(x) - c_i)^2 dx + \frac{\mu}{2} |\partial \Omega_i| \right), \quad (51)$$

where μ is a constant, $\partial \Omega$ and $|\partial \Omega|$ represent the boundary and the boundary length of segment Ω , respectively. The first data term measures the consistency of each segment and the second regularization term measures the boundary length.

To segment a 3D triangulated surface M , [39] convexifies this difficult nonconvex problem (51) based on TV to get a new version of M-S model

$$\min_{\mathbf{u} \in K, \chi_i} \left\{ \int_M \langle \mathbf{u}(x), \mathbf{s}(x) \rangle + \mu g(x) |\nabla_M \mathbf{u}(x)| d\sigma \right\}, \quad (52)$$

where K is the set of vector functions $\mathbf{u} = (u_1, \dots, u_k)^T : M \rightarrow \mathbb{R}^k$ satisfying that for all $x \in M$ and $i \in [1, \dots, k]$, $u_i(x) \geq 0$ and $\sum_{i=1}^k u_i(x) =$

1 ; $\mathbf{s}(x) = (s_1(x), \dots, s_k(x))^T$ is a k -dimensional vector with $s_i(x) = (\mathbf{f}(x) - \chi_i)^T (\mathbf{f}(x) - \chi_i)$ indicating the affinity vector χ_i that is associated with M_i which is a segment. The $\mathbf{f}(x)$ is constructed with the eigenvectors of the Laplacian matrix of the dual graph of M to represent some attributions of x over mesh M similar to the RGB function for an image. Since the regularization term is to constrain the boundary with some geometric difference information between segments, this optimization may fail for the relative smooth models.

In (52), if the affinity of x with segment M_i is large, $u_i(x)$ will tend to be small in order to reach the minimization, thus $u_i(x)$ can be viewed as the probability of x being assigned to segment M_i and so $\mathbf{u}(x)$ is used as a classification function for the segmentation achieved with some following processing work. Different kind of segmentation results are shown in Figure 15.

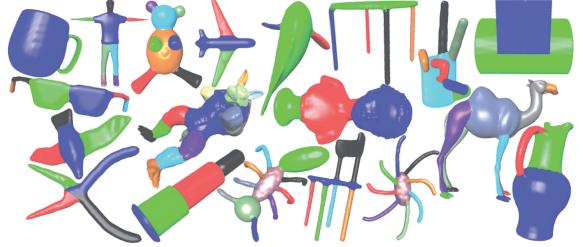


Figure 15: Sparse regularization: TV based mesh decomposition[39]. Decomposition results where the models are taken from the Princeton Segmentation Benchmark[66]. One mesh is shown for each category. The segmentation results match results match human perception well in not only the cutting boundaries but also the number of segments.

291 4.6.4. Barycentric coordinates

Barycentric coordinates provide a simple and convenient way of interpolating values from a set of control points over the interior of a domain, using weighted combinations of values associated with different control points. Many barycentric coordinates typically get a interpolated value depends on many, potentially *all*, control points. Then locality, scalability, and moderate computation cost can not be achieved.

[48] introduces a novel method to derive *local barycentric coordinates*(LBC) that depend only on a small number of control points. Given a set of control points $\mathbf{c}_1, \dots, \mathbf{c}_n$ in \mathbb{R}^2 or \mathbb{R}^3 which are the vertices of a closed control cage that bounds a domain. The goal is to find a function $w_i : \Omega \rightarrow \mathbb{R}$ for each \mathbf{c}_i , such that $[w_1(\mathbf{x}), \dots, w_n(\mathbf{x})]$ 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 $f(\mathbf{c}_1), \dots, f(\mathbf{c}_n)$ at control points on the interior of Ω by

$$f(\mathbf{x}) = \sum_{i=1}^n w_i(\mathbf{x}) f(\mathbf{c}_i), \quad (53)$$

except for the properties satisfied in many barycentric coordinate schemes, like reproduction and partition of unity, they prefer a target *convex* functional that also reflects *locality* and *smoothness* for the coordinate functions.

For a function w_i and a given value s , denote by $\{w_i > s\} := \{\mathbf{x}|w_i(\mathbf{x}) > s\}$ and $\{w_i = s\} := \{\mathbf{x}|w_i(\mathbf{x}) = s\}$ the *superlevel set* and the *levelset* of s , respectively. Locality requires the area of the superlevel set $\{w_i > 0\}$ to be small which means that the vector $[w_1(\mathbf{x}), \dots, w_n(\mathbf{x})]$ is sparse, while for smoothness it is necessary that all curves/surfaces $w_i = \text{const}$ are smooth.

Then the locality and smoothness of w_i can be obtained using a functional that measures the sum of the perimeters of superlevel sets $\{w_i > s\}$ for all s . That is, the perimeter of each superlevel set regularizes the smoothness of its boundary level curve/surface, while the perimeter of $\{w_i > 0\}$ penalizes the area of the influence region. This functional is exactly the total variation of $\{w_i\}$ formulated as

$$\begin{aligned} & \min_{w_1, \dots, w_n} \sum_{i=1}^n \int_{\Omega} |\nabla w_i| \\ \text{s.t. } & \sum_{i=1}^n w_i(\mathbf{x}) \mathbf{c}_i = \mathbf{x}, \sum_{i=1}^n w_i = 1, w_i \geq 0, \forall \mathbf{x} \in \Omega, \\ & w_i(\mathbf{c}_j) = \delta_{ij}, \text{ for all } i, j, \\ & w_i \text{ is linear on cage edges and faces } \forall i. \end{aligned} \quad (54)$$

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. Let \mathcal{C} be the set of cells in the triangulation, the target functional (54) finally becomes

$$\sum_{T \in \mathcal{C}} \sum_{i=1}^n \phi_i^T A_T \|\nabla T w_i\|_2, \quad (55)$$

where $\nabla T w_i$ is the gradient of w_i in cell T , A_T is the area(volume) of T , and ϕ_i^T is the value of the weighting function ϕ_i at the centroid of T .

Figure 16 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.



Figure 16: Sparse regularization: TV based local barycentric coordinates[48]. Using LBC for 3D cage-based manipulation allows for local, smooth and shape-aware deformations. Only parts near the manipulated control points are deformed, as indicated by the color-coding.

5. Dictionary Learning

Until now, we haven't given any discussion about the overcomplete dictionary D that exactly leads to the sparse representations of signals. From the definition of sparse representation, it is obvious that the choice of the dictionary will directly affect the signal processing result.

In general, this dictionary can either be chosen as a prespecified set of functions(e.g., wavelet dictionary) or designed by adapting its content to fit a given set of signal examples(e.g., [12]). From the performance of existing dictionary learning based works, the learned dictionaries used to outperform predefined dictionaries, so there have been many techniques aiming to get a expressive dictionary with less computational cost. And these techniques can be directly used to deal with some geometric problems, such as compression of point cloud[37].

In this section, we also regard sparse matrix decomposition as dictionary learning. As the name implies, sparse matrix decomposition attempts to decompose a dense matrix into the multiplication of a simple matrix(e.g., transformation matrix[41]) and the correspondent coefficient matrix which is as sparse as possible. 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.

The following works show the success of dictionary learning based method in different applications.

5.1. Deformation-Blend Skinning

Skinning mesh animations has been an active area. 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 j -th bone to the i -th vertex, p_i is the position of the i -th vertex at the rest

pose, $|B|$ is the number of bones, and R_j^t and T_j^t are the rotation matrix and translation vector of the j -th bone at the t -th configuration, respectively, then the deformed i -th vertex, v_j^t , can be computed as follows:

$$v_j^t = \sum_{j=1}^{|B|} w_{ij}(R_j^t p_i + T_j^t) \quad (56)$$

By posing sparseness constraint on the weight map, the number of non-zero bone weights per vertex can be limited. The orthogonal constraint on R_j^t 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). [41] introduces 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 $|t|$ example poses of a $|V|$ -vertices model, taking $\{v_j^t : t = 1..|t|, i = 1..|V|\}$ as input, SSDR decomposes them to bone transformations(R_j^t, T_j^t) and the bone-vertex weight map(as Figure 17 left shows)

$$\begin{aligned} \min_{w,R,T} E &= \min_{w,R,T} \sum_{t=1}^{|t|} \sum_{i=1}^{|V|} \|v_j^t - \sum_{j=1}^{|B|} w_{ij}(R_j^t p_i + T_j^t)\|^2 \\ \text{s.t. } w_{ij} &\geq 0, \forall i, j \\ \sum_{j=1}^{|B|} w_{ij} &= 1, \forall i \\ |\{w_{ij} | w_{ij} \neq 0\}| &\leq |K|, \forall i \\ R_j^{tT} R_j^t &= I, \det R_j^t = 1, \forall t \end{aligned} \quad (57)$$

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. Figure 17 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.

(2). To address these above limitations, [42] introduces an efficient two-layer compression technique formulated as a sparse dictionary learning problem. The left figure in Figure 17(b) gives a clear explanation about the new technique.

Let $W \in \mathbb{R}^{k \times n}$ be the weight matrix of an input skinning model with n vertices and k bones, as illustrated as

the top left. At *master bone blending* layer, they calculate and cache the transformations of m virtual bones by blending the transformations of k 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

$$\begin{aligned} \min_{D,A} \Delta_W^2 &= \min_{D,A} \frac{1}{kn} \|DA - W\|_F^2 \\ \text{s.t. } \mathbf{card}(\alpha_i) &\leq 2, \forall i \\ \mathbf{card}(d_i) &\leq c, \forall i \end{aligned} \quad (58)$$

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 Figure 17(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. Setting up the skeleton-based animation(also known as rigging) generally consists of two main steps: building a hierarchical skeleton with rigid bones connected by joints, and skinning the 3D model to define how joint rotations and translations would propagate to the surface during animation. A best result may be achieved by manually or semi-automatically repeating the two steps many times which make it costly and time-consuming. Fortunately, the concept of using example poses can ease this problem[67].

(3). Attempting to take advantage of example poses, taking a set of example poses as input, [43] introduces 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

$$E = E_D + wE_S + \lambda E_J \quad (59)$$

subject to the same set of constraints (57) 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

$$E_D = \frac{1}{|t||V|} \sum_{t=1}^{|t|} \sum_{i=1}^{|V|} \|v_j^t - \sum_{j=1}^{|B|} w_{ij}(R_j^t p_i + T_j^t)\|^2. \quad (60)$$

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[43] 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. 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.

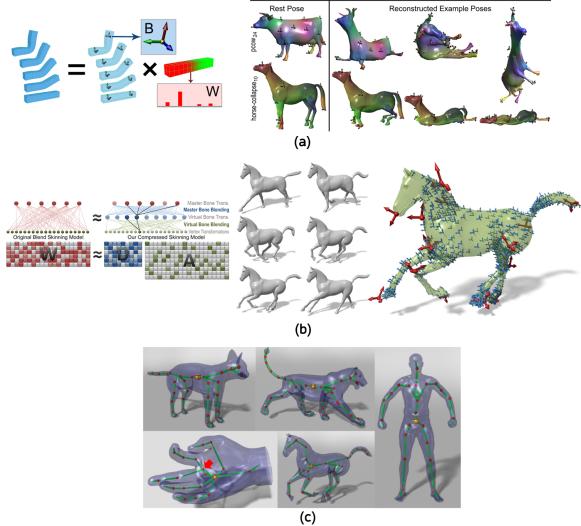


Figure 17: Dictionary learning: skinning results. (a): [41], left: a set of example poses are decomposed into rigid bone transformation B and a sparse, convex bone-vertex weight map W . right: results of SSDR on elastic models. (b): [42], left: two-layer scheme. right: an animated mesh sequence and its corresponding compressed skinning model. (c): [43], result of rigging various models such as quadruped animals, humans, and highly deformable models.

294 5.2. Decomposition

Time-varying dynamic geometry with very fine dynamic shape detail can be generated and rendered at very high visual fidelity. When creating such content, artists usually rely on a low-dimensional control parameterization. Despite increasing expensive power of such

parameterizations and simulations, producing such realistic animations from scratch is a labor-intensive process.

As Figure 18 shows, 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), [40] proposes a new efficient, easy-to-implement, and versatile data-driven approach inspired by matrix decomposition methods like sparse PCA[68]. Given a mesh animation with F frames, each frame f consists of N vertices positions $\mathbf{v}_i^{(f)}$, constructing a single animation matrix $X \in \mathbb{R}^{F \times 3N}$ by assembling the vertices of all frames in a row-wise fashion

$$\mathbf{X} = \begin{bmatrix} (\mathbf{v}_1^{(1)})^T & (\mathbf{v}_2^{(1)})^T & \dots & (\mathbf{v}_N^{(1)})^T \\ (\mathbf{v}_1^{(2)})^T & (\mathbf{v}_2^{(2)})^T & \dots & (\mathbf{v}_N^{(2)})^T \\ \vdots & \vdots & \ddots & \vdots \\ (\mathbf{v}_1^{(F)})^T & (\mathbf{v}_2^{(F)})^T & \dots & (\mathbf{v}_N^{(F)})^T \end{bmatrix} \quad (61)$$

After some preprocessing for X , and following the framework of [68], [40] formulates the matrix factorization into K deformation components $C \in \mathbb{R}^{k \times 3N}$ with weights $W \in \mathbb{R}^{F \times K}$ as a joint regularized minimization problem

$$\underset{W, C}{\operatorname{argmin}} \|X - W \cdot C\|_F^2 + \Omega(C) \text{ s.t. } \mathcal{V}(W) \quad (62)$$

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*, $\Omega(C)$ is formulated by acting ℓ_1 norm on the lengths of the displacement vectors

$$\Omega(C) = \sum_{k=1}^K \sum_{i=1}^N \Lambda_{ki} \|\mathbf{c}_k^{(i)}\|_2. \quad (63)$$

The spatially-varying regularization parameters Λ_{ki} makes it possible to enforce local support for the deformation components which is an exciting innovation.

Sparse Localized Deformation components present a decomposition method for space-time mesh animation data that is applicable to many settings: editing, control, scan alignment, construction of static and parametric shape models etc. It is mentioned in this paper that several parameters in the formulation are specified by users, but it is not clear that whether the users should have knowledge of graphics or whether it is easy for the users to give the suitable values.

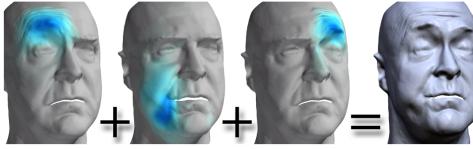


Figure 18: Dictionary learning: decomposition[40]. A new facial expression is generated by summing deformation components, the method automatically separates spatially confined effects like separate eyebrow motions from the data.

295 5.3. 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 4.1) 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, [50] proposes a unified framework that treats geometry and connectivity construction as one joint optimization problem.

As Figure 19(a) shows, given a point set $\mathbb{P} = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n\}$ (blue) sampled from a piecewise smooth surface S , they attempt to find a triangular mesh $M = \{\mathbb{V}, \mathbb{F}\}$ with vertex set $\mathbb{V} = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m\}$ (red) and triangle set \mathbb{F} to approximate the underlying surface S so that the approximation error is as small as possible

$$\begin{aligned} & \min_{\mathbf{B}, \mathbf{V}} \frac{1}{n} \sum_{i=1}^n \|\mathbf{p}_i - \mathbf{V}\mathbf{b}_i\|_2^q + E_{reg} \\ & \text{s.t. } \|\mathbf{b}_i\|_0 \leq 3, \quad \|\mathbf{b}_i\|_1 = 1, \quad b_i \geq 0, \quad \forall i \\ & \quad \mathbf{B} \in \mathbb{R}^{m \times n} \end{aligned} \quad (64)$$

where E_{reg} regularizes the reconstructed mesh to produce good mesh quality, each column of sparse coding matrix 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.

Figure 19(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.

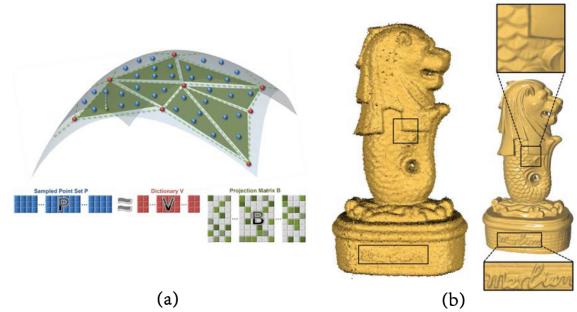


Figure 19: Dictionary learning: reconstruction[50]. Left: (Top)an illustration of the reconstruction problem. Given point set \mathbb{P} (blue) sampled from surface S , they approximate S with piecewise linear surface M with vertices \mathbb{V} (red) and triangles \mathbb{F} . (Bottom) The reconstruction problem where P is the position of sample point set. V is the dictionary and B (green) is the sparse coding matrix that encodes triangles \mathbb{F} . Right: reconstruction result of the Merlion model.

296 5.4. Compression

(1). The compression of unorganized point clouds have also attracted much attention due to the drastic improvement in scanner acquisition devices yielding point sets of tens of millions of points at high precision. But the counterpart of this development are datasets requiring ever higher storage capacity which results in the expensive cost in point cloud processing.

In [37], 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 using a new neighborhood descriptor(Figure 20(a)), finally directly using the K-SVD algorithm [12]

$$\begin{aligned} & \min_{D, A} \|Y - DA\| \\ & \text{s.t. } \|\mathbf{a}_i\|_0 \leq s, \quad \forall i \end{aligned} \quad (65)$$

to exploit the self-similarity of the descriptions and build a custom dictionary D (Figure 20(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 cloud resolution. It achieves a filtering of noise whose magnitude is smaller than the scanner precision. Figure 20(c) gives one compression result.

(2). Real-time rendering of complex scenes with full global illumination is one of the main goals of computer graphics. A popular approach is to pre-compute

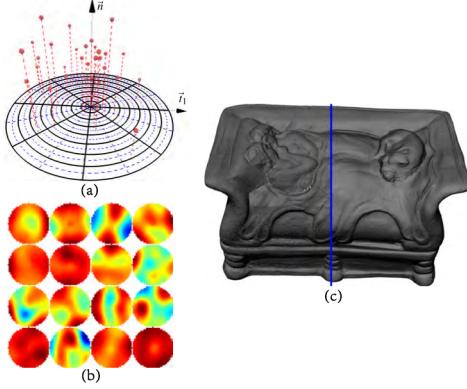


Figure 20: Dictionary learning: point cloud compression[37]. (a): the local neighborhood description: a height map over a radial grid. (b): dictionary built for the Lovers(c) and the atoms are shown by order of importance(total absolute weight in the linear decompositions). (c): the Lovers(15.8 million points) is compressed down to 1.15 MB. The resulting model(right) is very close to the original one(left), as the reconstruction error is less than the laser scanner precision(0.02mm) for 99.14% of the input points.

detailed surface light fields(SLF) describing the appearance of the objects which often exhibit a very large memory footprint and does not easily meet the real-time requirement.

To handle arbitrary light source configurations, general high frequency scenes and materials, and reduce the complexity of off-line pre-computations as well as support real-time rendering, [38] presents a learning based algorithm for efficient compression of appearance information encoded as SLFs which are 4D functions $f(u, v, \phi, \theta)$ represented as a *hemispherical radiance distribution function*(HRDF)(Figure 21(a)).

After analyzing the spatial correlation in the data by clustering points with similar HRDFs, they firstly *training* a set of exemplar(basis) pairs $\{(\bar{U}_c, \bar{V}_c)\}$ by minimizing the following energy function

$$\begin{aligned}
& E(\{\bar{U}_a^c, \bar{V}_a^c, \bar{S}_{ia}, M_{ia}^c\}) \\
&= \sum_{i=1} \sum_{a=1} M_{ia}^c \|H_i^c - (\bar{U}_a^c)^T \bar{S}_{ia} (\bar{V}_a^c)^T\|^2 \\
&\text{s.t.} \\
& (\bar{U}_a^c)^T \bar{U}_a^c = (\bar{V}_a^c)^T \bar{V}_a^c = I, \forall a, \\
& \|\bar{S}_{ia}\|_0 \leq t \text{ and } \sum_a M_{ia}^c = 1, \forall i,
\end{aligned} \tag{66}$$

where \bar{S}_{ia} contains the coefficients of the i th HRDF when projected onto the a th exemplar, M is a binary matrix associating each HRDF to its corresponding exemplar pair $(\bar{U}_a^c, \bar{V}_a^c)$. Finally, let (ξ_1, ξ_2) be an element of an HRDF matrix, the sparsity of \bar{S}_i allows for

a compact reconstruction formula for Clustered Exemplar Orthogonal Bases(CEOB)

$$H_i^c(\xi_1, \xi_2) = \sum_{i=1} \bar{U}_a^c(\bar{S}_i(t, 1), \xi_1) \bar{S}_i(t, 3) \bar{V}_a^c(\bar{S}_i(t, 1), \xi_2), \tag{67}$$

where \bar{S}_i is a matrix of size $t \times 3$: the first two columns describe the index of a non-zero element and the third stores its value. Figure 21(b) shows three scenes with different materials.

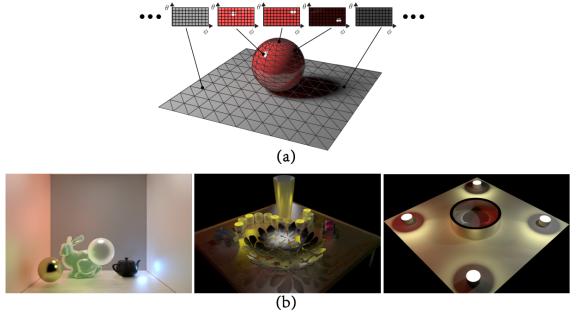


Figure 21: Sparse decomposition: rendering[38]. (a): the 4D SLF function $f(u, v, \phi, \theta)$ is represented as a hemispherical radiance distribution function, HRDF. (b): rendering results using CEOB for three scenes with different materials.

6. Low Rank

6.1. 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.

Figure 22 shows the axis-aligned projections of an input man-made model with arbitrary(a) and axis-aligned orientation(b) onto the $y-z$, $y-z$ and $y-z$ plane(left to right) in the $x-y-z$ (red-green-blue) 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 horizontal and vertical edges and shapes.

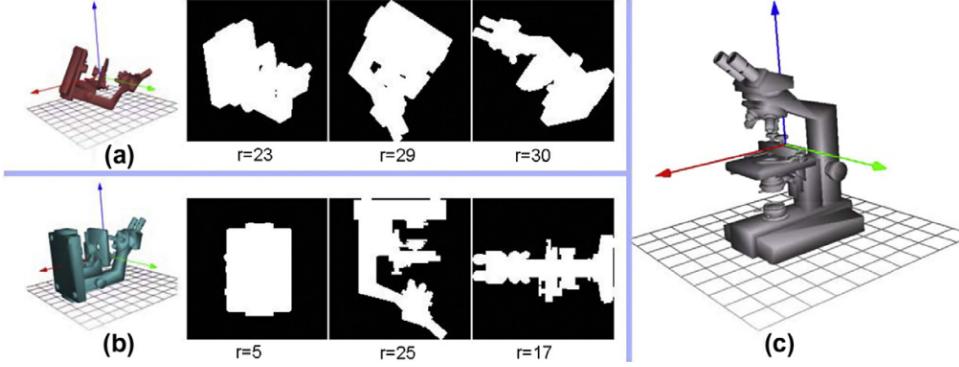


Figure 22: Low rank: observation. (a): local method[45]. (b): global method[46].

(1). Based on this observation, [45] presents 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 I with fixed resolution which can also be referred as a two-dimensional matrix. To avoid affect of noise, I is modeled as a low-rank version L with sparse-error matrix E : $I = L + E$. Then the problem is formulated as

$$\min_{L, E, R} \|L\|_* + \lambda \|E\|_1, \text{ s.t. } I \circ R = L + E \quad (68)$$

where $\|\cdot\|_*$ and $\|\cdot\|_1$ are the nuclear norm and the L_1 norm, which are closely related to rank and sparsity of matrix respectively. R is a rigid rotation transformation matrix used to rectify I 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 y - z , y - z and y - z , using (69) 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). It is very natural to generalize this method in 3D space to construct three-order tensor(multidimensional array) with volume of the 3D model, i.e., the three-order tensor ought to have a "low rank" behavior. [46] constructs 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 R to align the model with three axes by following optimization model

$$R_* = \operatorname{argmin}_R (\|\chi(V \circ R)\|_*) \quad (69)$$

where V and $V \circ R$ respectively indicate the point coordinates of input model and the rotated model, $\chi(\cdot)$ is the three-order tensor. Similar to [45], after aligning the model with three axes, they select the upright orientation from six orientations by analyzing the geometric properties. Figure 23 shows the results of these two methods.

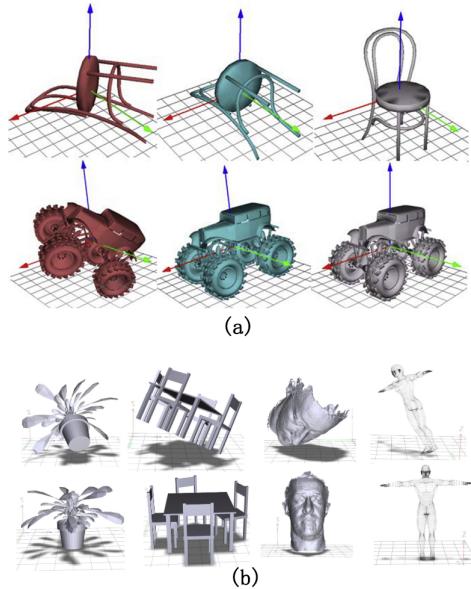


Figure 23: Low rank: unsupervised upright orientation. (a): local method[45]. (b): global method[46].

299 6.2. Point cloud normal estimation

Now after reviewing almost all the sparsity based geometric problems, it is clear to us that good normal estimation from noisy data has been always focused since it induces better geometry 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 easy to get a robust normal estimation. Thus using the robust results in smooth regions as prior knowledge, [31] estimates the point normals around the sharp regions by low-rank clustering(LRSCP).

In section 4.3.3, 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

$$\min \|Z\|_*, \text{ s.t. } X = XZ \quad (70)$$

With an input noisy point cloud $\mathcal{P} = \{p_i\}_{i=1}^N$, they firstly detect the candidate feature points(Figure 24(b)) by covariance analysis of the point neighborhoods of size S . Since their neighborhoods are all anisotropic, to segment each neighborhood into several isotropic subneighborhoods, for each candidate point p_i , they select a larger neighborhood of size S^* , i.e., $S^* > S$. In the local coordinates with p_i as the origin, the neighbor point p_{ij} of p_i is represented as $p_{ij} = [x_j, y_j, z_j, n_{xj}, n_{yj}, n_{zj}]'$, $j = 1, \dots, S^*$, where $[x_j, y_j, z_j]$ is the coordinate of p_{ij} and $[n_{xj}, n_{yj}, n_{zj}]$ is its normal computed by PCA. The sampling matrix is defined as $X = [p_{i1}, p_{i2}, \dots, p_{iS^*}]$. The optimal coefficient matrix $Z \in \mathbb{R}^{S^* \times S^*}$ is computed by solving

$$\begin{aligned} \min \|Z\|_* + \beta \|\mathcal{P}_\Omega(Z)\|_1 + \gamma \|E\|_{2,1}, \\ \text{s.t. } X = XZ + E, \end{aligned} \quad (71)$$

where, Ω , ($0 \leq \Omega(i, j) \leq 1$), is a guiding matrix constructed according to the distance relation between each two neighbor points for current candidate point. \mathcal{P}_Ω is defined as $\mathcal{P}_\Omega(Z_{i,j}) = \Omega(i, j) \times Z(i, j)$. After getting Z , by defining the affinity matrix like section 4.3.3, this larger neighborhood is segmented into several subneighborhoods in which a consistent subneighborhood is used to estimate the current point normal.

Now an obvious conclusion is that the extensions from images to geometries is non-trivial, thus it is much difficult to achieve some applications that have high performance in images(Figure 3). Then how to resolve the unresolved problems gives much challenges as future works.

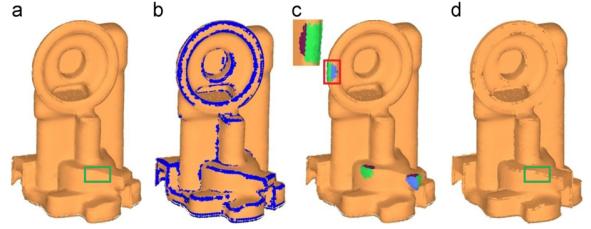


Figure 24: Low rank: normal estimation[31]. (a): the oil pump module with normal computed by PCA. (b): initial detected candidate feature points. (c): the classified subneighborhoods. The neighborhood within the red box contains three subneighborhoods rendered in blue, green and brown and the zoomed view is from left. (d): estimated normals.

Table 3: Comparison of applications between image and geometry.

	Image	Geometry
Denoising	★	★
Retrieval	★	
Alignment	★	★
Inpainting	★	
Separation	★	
Compression	★	★
Deformation		★
Segmentation	★	★
Classification	★	
Super-resolution	★	★
Skeleton Extraction	★	★

7. Discussion

Sparse representation shows its dominance in many geometry processing due to the robustness to noises and outliers as well as features preserving. Actually, in summary, it can be classified into two main groups:

- (1). From the geometry itself, finding the intrinsic sparse characteristic like curvature;
- (2). If the intrinsic characteristic is not sparse, then using some differential operator, like Laplacian, to transform it into another characteristic space in which the transformed characteristic is sparse;

In this case, paying more attention to analyze some kind of geometric information carefully, such as Interaction Bisector Surface(IBS) and Beltrami number, may lead to a new exciting sparse problem.

However, due to the nature of sparsity modeling, the high computational cost is a common problem in most works with the iterative solving process. Though they are designed for performance, they will be less meaningful if they cannot be applied to the practical applications. So how to solve the sparse optimization problem

while preserving the existing good properties is an worthy researching direction which will also maximize the values of many previous works.

- [1] S. G. Mallat, Z. Zhang, Matching pursuits with time-frequency dictionaries, *Signal Processing, IEEE Transactions on* 41 (12) (1993) 3397–3415.

[2] 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.

[3] J. A. Tropp, A. C. Gilbert, Signal recovery from random measurements via orthogonal matching pursuit, *Information Theory, IEEE Transactions on* 53 (12) (2007) 4655–4666.

[4] R. Tibshirani, Regression shrinkage and selection via the lasso, *Journal of the Royal Statistical Society. Series B (Methodological)* (1996) 267–288.

[5] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, et al., Least angle regression, *The Annals of statistics* 32 (2) (2004) 407–499.

[6] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. S. Huang, S. Yan, Sparse representation for computer vision and pattern recognition, *Proceedings of the IEEE* 98 (6) (2010) 1031–1044.

[7] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, Y. Ma, Robust face recognition via sparse representation, *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 31 (2) (2009) 210–227.

[8] 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.

[9] 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.

[10] B. A. Olshausen, D. J. Field, Sparse coding with an overcomplete basis set: A strategy employed by v1?, *Vision research* 37 (23) (1997) 3311–3325.

[11] 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*, Vol. 4, IEEE, 1999, pp. 1–4.

[12] M. Aharon, M. Elad, A. Bruckstein, K-svd: An algorithm for designing overcomplete dictionaries for sparse representation, *Signal Processing, IEEE Transactions on* 54 (11) (2006) 4311–4322.

[13] I. Jolliffe, Principal component analysis, Wiley Online Library, 2005.

[14] S. Wold, K. Esbensen, P. Geladi, Principal component analysis, *Chemometrics and intelligent laboratory systems* 2 (1) (1987) 37–52.

[15] E. J. Candès, X. Li, Y. Ma, J. Wright, Robust principal component analysis?, *Journal of the ACM (JACM)* 58 (3) (2011) 11.

[16] E. J. Candès, B. Recht, Exact matrix completion via a convex optimization, *Foundations of Computational mathematics* 9 (6) (2009) 717–772.

[17] Z. Lin, M. Chen, Y. Ma, The augmented lagrange multiplier method for exact recovery of corrupted low-rank thresholding algorithm for linear inverse problems, *SIAM Journal on Imaging Sciences* 2 (1) (2009) 183–202.

[18] 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.

[19] D. P. Bertsekas, Constrained optimization and lagrange multiplier methods, *Computer Science and Applied Mathematics*, Boston: Academic Press, 1982 1.

[20] D. P. Bertsekas, Nonlinear programming.

[21] 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.

[22] 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.

[23] 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.

[24] Y. Lipman, D. Cohen-Or, D. Levin, H. Tal-Ezer, Parameterization-free projection for geometry reconstruction, in: *ACM Transactions on Graphics (TOG)*, Vol. 26, ACM, 2007, p. 22.

[25] H. Huang, D. Li, H. Zhang, U. Ascher, D. Cohen-Or, Consolidation of unorganized point clouds for surface reconstruction, in: *ACM Transactions on Graphics (TOG)*, Vol. 28, ACM, 2009, p. 176.

[26] 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. doi:10.1145/2601097.2601172. URL <http://doi.acm.org/10.1145/2601097.2601172>

[27] H. Avron, A. Sharf, C. Greif, D. Cohen-Or, L1-sparse reconstruction of sharp point set surfaces, *ACM Transactions on Graphics (TOG)* 29 (5) (2010) 135.

[28] G. Mustafa, H. Li, J. Zhang, J. Deng, ℓ_1 -regression based subdivision schemes for noisy data, *Computer-Aided Design*.

[29] J. Zhang, J. Cao, X. Liu, J. Wang, J. Liu, X. Shi, Point cloud normal estimation via low-rank subspace clustering, *Computers & Graphics* 37 (6) (2013) 697–706.

[30] L. He, S. Schaefer, Mesh denoising via ℓ_0 minimization, *ACM Transactions on Graphics (TOG)* 32 (4) (2013) 64.

[31] R. Wang, Z. Yang, L. Liu, J. Deng, F. Chen, Decoupling noise and features via weighted ℓ_1 -analysis compressed sensing, *ACM Transactions on Graphics (TOG)* 33 (2) (2014) 18.

[32] H. Zhang, C. Wu, J. Zhang, J. Deng, Variational mesh denoising using total variation and piecewise constant function space, *Visualization and Computer Graphics, IEEE Transactions on*.

[33] S. Bouaziz, A. Tagliasacchi, M. Pauly, Sparse iterative closest point, in: *Proceedings of the Eleventh Euromicro/ACMSIGGRAPH Symposium on Geometry*

- 427 Processing, Eurographics Association, 2013, pp. 113– 492
 428 123. 493
- 429 [36] R. Hu, L. Fan, L. Liu, Co-segmentation of 3d shapes 494
 430 via subspace clustering, in: Computer graphics forum, 495
 431 Vol. 31, Wiley Online Library, 2012, pp. 1703–1713. 496
- 432 [37] J. Digne, R. Chaine, S. Valette, Self-similarity for 497
 433 accurate compression of point sampled surfaces, month. 498
- 434 [38] E. Miandji, J. Kronander, J. Unger, Learning based 499
 435 compression of surface light fields for real-time render- 500
 436 ing of global illumination scenes, in: SIGGRAPH Asia 501
 437 2013 Technical Briefs, ACM, 2013, p. 24. 502
- 438 [39] J. Zhang, J. Zheng, C. Wu, J. Cai, Variational mesh 503
 439 decomposition, ACM Transactions on Graphics (TOG) 504
 440 31 (3) (2012) 21. 505
- 441 [40] T. Neumann, K. Varanasi, S. Wenger, M. Wacker, 506
 442 M. Magnor, C. Theobalt, Sparse localized deformation 507
 443 components, ACM Transactions on Graphics (TOG) 508
 444 32 (6) (2013) 179. 509
- 445 [41] B. H. Le, Z. Deng, Smooth skinning decomposition with 510
 446 rigid bones, ACM Transactions on Graphics (TOG) 511
 447 31 (6) (2012) 199. 512
- 448 [42] B. H. Le, Z. Deng, Two-layer sparse compression of 513
 449 dense-weight blend skinning, ACM Transactions on 514
 450 Graphics (TOG) 32 (4) (2013) 124. 515
- 451 [43] B. H. Le, Z. Deng, Robust and accurate skeletal rigging 516
 452 from mesh sequences, ACM Trans. Graph. 33 (4) (2014) 517
 453 84:1–84:10. doi:10.1145/2601097.2601161. 518
- 454 URL <http://doi.acm.org/10.1145/2601097.2601161> 519
- 455 [44] B. Deng, S. Bouaziz, M. Deuss, J. Zhang, 520
 456 Y. Schwartzburg, M. Pauly, Exploring local mod- 521
 457 ifications for constrained meshes, Computer Graphics 522
 458 Forum (Proceedings of Eurographics 2013) 32 (2). 523
- 459 [45] Y. Jin, Q. Wu, L. Liu, Unsupervised upright orientation 524
 460 of man-made models, Graphical Models 74 (4) (2012) 525
 461 99–108. 526
- 462 [46] W. Wang, X. Liu, L. Liu, Upright orientation of 3d 527
 463 shapes via tensor rank minimization, Journal of Me- 528
 464 chanical Science and Technology (2014) 40–53. 529
- 465 [47] T. Neumann, K. Varanasi, C. Theobalt, M. Magnor, 530
 466 M. Wacker, Compressed manifold modes for mesh 531
 467 processing, in: Computer Graphics Forum, Vol. 33, Wiley 532
 468 Online Library, 2014, pp. 35–44. 533
- 469 [48] J. Zhang, B. Deng, Z. Liu, G. Patane, S. Bouaziz, 534
 470 K. Hormann, L. Liu, Local barycentric coordinates, 535
 471 ACM Transactions on Graphics (Proc. SIGGRAPH 536
 472 Aisa) 33 (6). 537
- 473 [49] H. Huang, S. Wu, D. Cohen-Or, M. Gong, H. Zhang, 538
 474 G. Li, B. Chen, ℓ_1 -medial skeleton of point cloud., ACM 539
 475 Trans. Graph. 32 (4) (2013) 65.
- 476 [50] S. Xiong, J. Zhang, J. Zheng, J. Cai, L. Liu, Robust sur- 540
 477 face reconstruction via dictionary learning, ACM Trans- 541
 478 actions on Graphics (Proc. SIGGRAPH Aisa) 33 (6).
- 479 [51] B. Brown, Statistical uses of the spatial median, Journal 542
 480 of the Royal Statistical Society. Series B (Methodolog- 543
 481 ical) (1983) 25–30.
- 482 [52] C. G. Small, A survey of multidimensional medians, 544
 483 International Statistical Review/Revue Internationale 545
 484 de Statistique (1990) 263–277.
- 485 [53] L. Xu, C. Lu, Y. Xu, J. Jia, Image smoothing via ℓ_0 546
 486 gradient minimization, ACM Transactions on Graphics 547
 487 (TOG) 30 (6) (2011) 174.
- 488 [54] U. Pinkall, K. Polthier, Computing discrete minimal 548
 489 surfaces and their conjugates, Experimental mathemat- 549
 490 ics 2 (1) (1993) 15–36.
- 491 [55] R. Chartrand, Exact reconstruction of sparse signals 550
 via nonconvex minimization, Signal Processing Letters, 551
 IEEE 14 (10) (2007) 707–710.
- [56] M. Elad, Sparse and redundant representations: from 552
 theory to applications in signal and image processing, 553
 Springer, 2010.
- [57] M. Ovsjanikov, M. Ben-Chen, J. Solomon, A. Butscher, 554
 L. Guibas, Functional maps: a flexible representation of 555
 maps between shapes, ACM Transactions on Graphics 556
 (TOG) 31 (4) (2012) 30.
- [58] J. Pokrass, A. M. Bronstein, M. M. Bronstein, P. Sprechmann, G. Sapiro, Sparse modeling of intrinsic 557
 correspondences, in: Computer Graphics Forum, 558
 Vol. 32, Wiley Online Library, 2013, pp. 459–468.
- [59] R. Litman, A. M. Bronstein, M. M. Bronstein, Diffusion-geometric maximally stable component 559
 detection in deformable shapes, Computers & Graphics 560
 35 (3) (2011) 549–560.
- [60] E. Elhamifar, R. Vidal, Sparse subspace clustering, 561
 in: Computer Vision and Pattern Recognition, 2009. 562
 CVPR 2009. IEEE Conference on, IEEE, 2009, pp. 563
 2790–2797.
- [61] S. Wang, X. Yuan, T. Yao, S. Yan, J. Shen, Efficient 564
 subspace segmentation via quadratic programming., in: 565
 AAAI, 2011.
- [62] J. Shi, J. Malik, Normalized cuts and image segmenta- 566
 tion, Pattern Analysis and Machine Intelligence, IEEE 567
 Transactions on 22 (8) (2000) 888–905.
- [63] N. D. Cornea, D. Silver, P. Min, Curve-skeleton prop- 568
 erties, applications, and algorithms, Visualization and 569
 Computer Graphics, IEEE Transactions on 13 (3) 570
 (2007) 530–548.
- [64] A. Chambolle, V. Caselles, D. Cremers, M. Novaga, T. Pock, An introduction to total variation for image 571
 analysis, Theoretical foundations and numerical meth- 572
 ods for sparse recovery 9 (2010) 263–340.
- [65] D. Mumford, J. Shah, Optimal approximations by 573
 piecewise smooth functions and associated variational 574
 problems, Communications on pure and applied math- 575
 ematics 42 (5) (1989) 577–685.
- [66] X. Chen, A. Golovinskiy, T. Funkhouser, A benchmark 576
 for 3d mesh segmentation, in: ACM Transactions on 577
 Graphics (TOG), Vol. 28, ACM, 2009, p. 73.
- [67] S. Schaefer, C. Yuksel, Example-based skeleton extrac- 578
 tion, in: Symposium on Geometry Processing, Citeseer, 579
 2007, pp. 153–162.
- [68] H. Zou, T. Hastie, R. Tibshirani, Sparse principal com- 580
 ponent analysis, Journal of computational and graphi- 581
 cal statistics 15 (2) (2006) 265–286.