Sparse Modeling and Sparse Optimization

Zhouwang Yang

University of Science and Technology of China

April 25, 2021

Outline I

Outline II

Outline I

Outline II

Big data era

Electronic commerce data

Social network data

Financial data

Multimedia data

Bioinformatics data

Geometry data

...

High dimensional data and sparsity

Techniques:

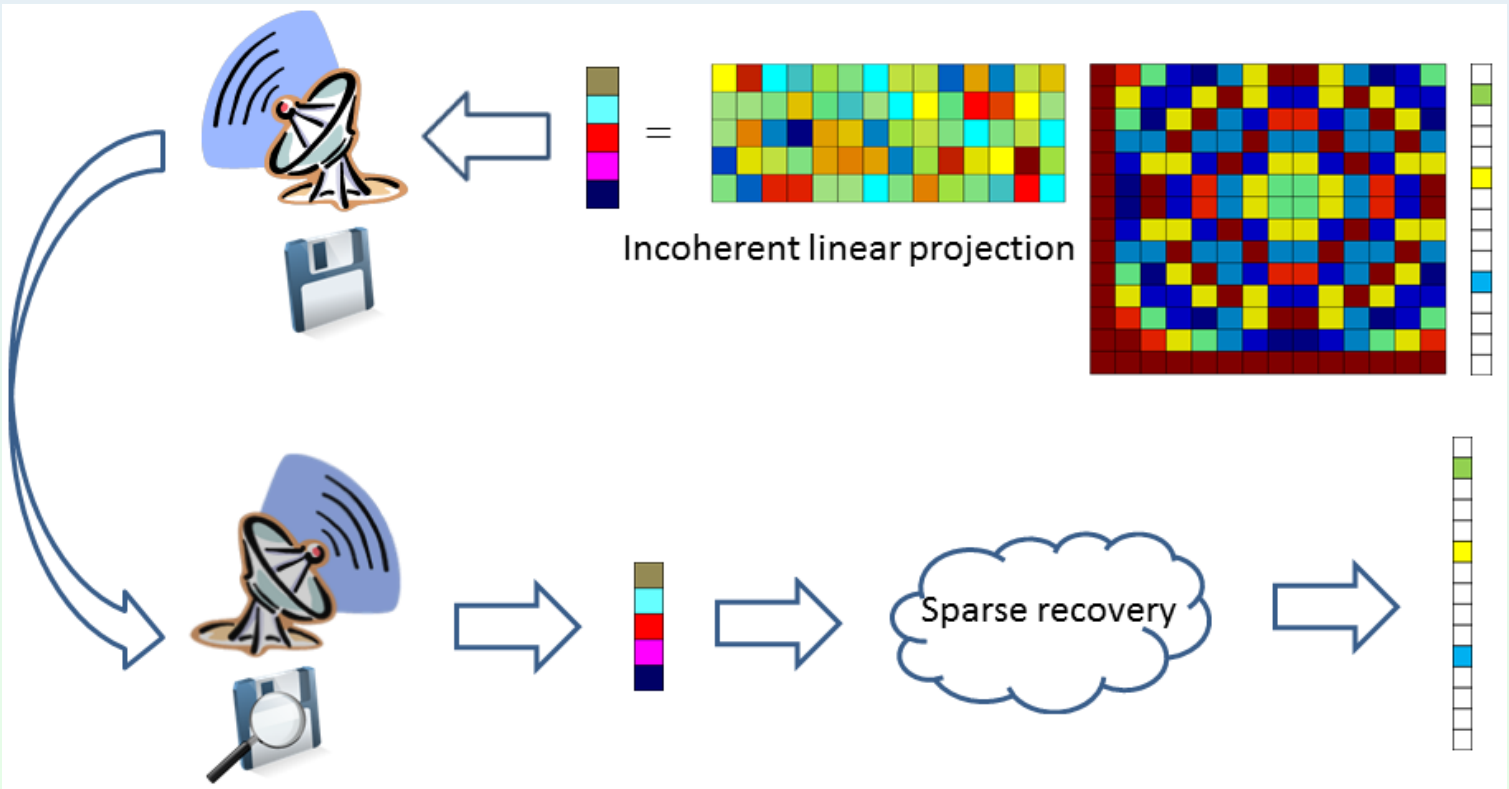
- Statistics (Bayesian/Lasso)
- Optimization (OMP/BP)
- Priors and Transforms
- Sparse and Redundant Representations
- Low Rank Representations
- ...

Compressed Sensing

In recent years, Compressed Sensing (CS) has attracted considerable attention in areas of applied mathematics, computer science, and signal processing [Candes and Tao 2005; Donoho 2006; Bruckstein et al. 2009].

Compressed Sensing

The central insight of CS is that many signals are sparse, i.e., represented using only a few non-zero coefficients in a suitable basis or dictionary and such signals can be recovered from very few measurements (undersampled data) by an optimization algorithm.



The Sparsest Solution of $Ax = b$

$$(P_0) \quad \min_{x} \|x\|_0 \quad \text{s.t.} \quad Ax = b. \tag{1}$$

For the underdetermined linear system of equations $Ax = b$ (a full-rank matrix $A \in \mathbb{R}^{m \times n}$ with $m << n$), the following questions are posed:

Q1: When can uniqueness of the sparsest solution be claimed?

Q2: Can a candidate solution be tested to verify its (global) optimality?

Q3: Can the solution be reliably and efficiently found in practice?

Q4: What performance guarantees can be given for various approximate and practical solvers?

Uniqueness via the Spark

DEFINITION 01. The spark of a given matrix A is the smallest number of columns from A that are linearly dependent.

THEOREM 02. If a system of linear equations $Ax = b$ has a solution x obeying $\|x\|_0 < spark(A)/2$, this solution is necessarily the sparsest possible.

Uniqueness via the Mutual Coherence

DEFINITION 03. The mutual coherence of a given matrix A is the largest absolute normalized inner product between different columns from A. Denoting the k-th column in A by a_k, the mutual coherence is given by

$$\mu(A) = \max_{1 \le i \neq j \le n} \frac{|\mathrm{a}_i^T \mathrm{a}_j|}{\|\mathrm{a}_i\|_2 \|\mathrm{a}_j\|_2}.$$

LEMMA 04. For any matrix $A \in \mathbb{R}^{m \times n}$, the following relationship holds:

$$spark(A) \geqslant 1 + \frac{1}{\mu(A)}.$$

THEOREM 05. If a system of linear equations $A\mathrm{x} = \mathrm{b}$ has a solution x obeying $\|\mathrm{x}\|_0 < (1 + 1/\mu(A))/2$, this solution is necessarily the sparsest possible.

Pursuit Algorithms

Greedy strategies are usually adopted in solving the problem (P_0).

The following algorithm is known in the literature of signal processing by the name *Orthogonal Matching Pursuit* (OMP).

Task: Approximate the solution of (P_0): $\min_{\mathbf{x}} \|\mathbf{x}\|_0$ subject to $\mathbf{A}\mathbf{x} = \mathbf{b}$.

Parameters: We are given the matrix $\mathbf{A}$, the vector $\mathbf{b}$, and the error threshold ϵ_0.

Initialization: Initialize $k = 0$, and set

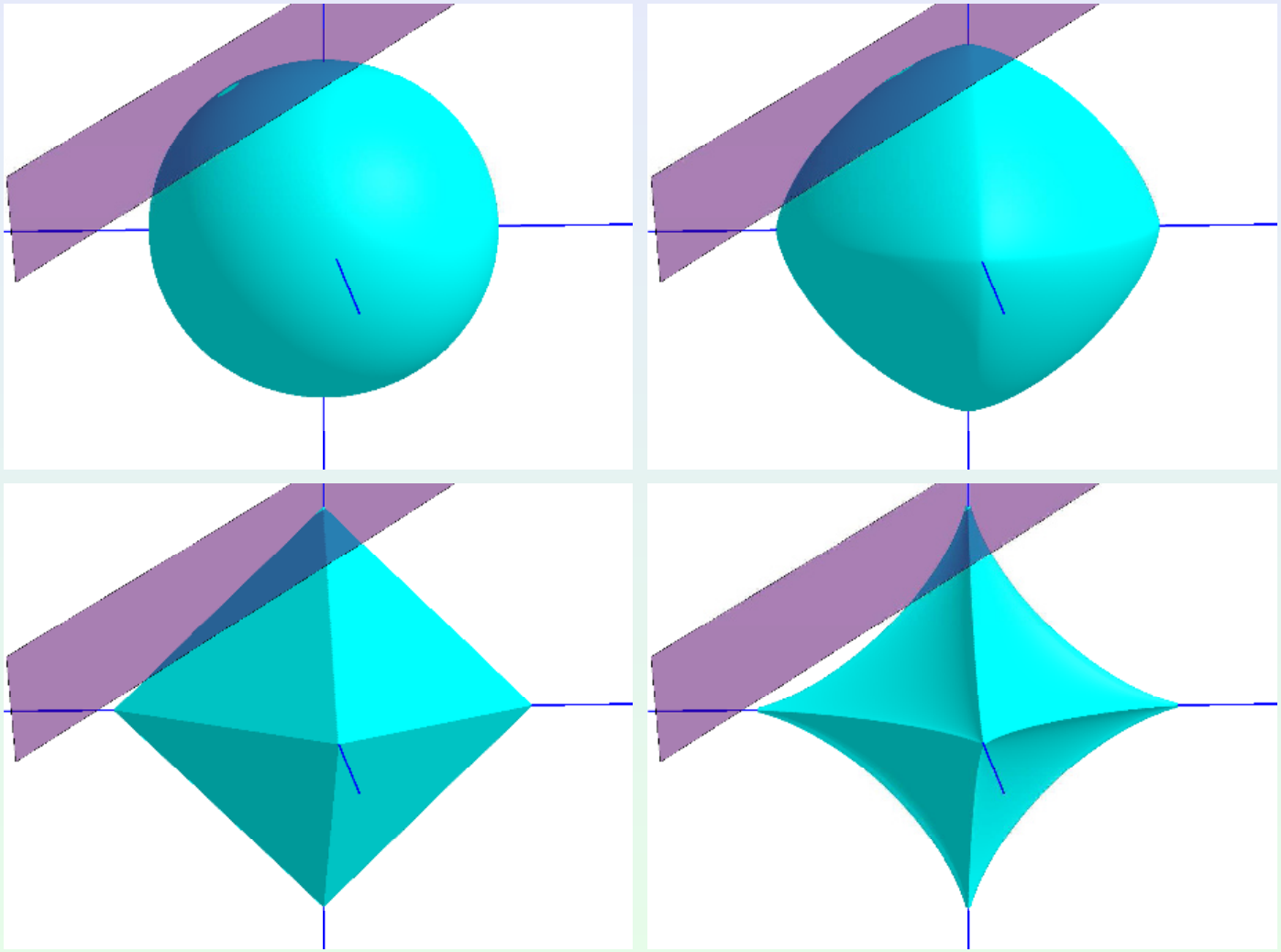
- The initial solution $\mathbf{x}^0 = 0$.
- The initial residual $\mathbf{r}^0 = \mathbf{b} - \mathbf{A}\mathbf{x}^0 = \mathbf{b}$.
- The initial solution support $\mathcal{S}^0 = Support\{\mathbf{x}^0\} = \emptyset$.

Main Iteration: Increment k by 1 and perform the following steps:

- **Sweep:** Compute the errors $\epsilon(j) = \min_{z_j} \|\mathbf{a}_j z_j - \mathbf{r}^{k-1}\|_2^2$ for all j using the optimal choice $z_j^* = \mathbf{a}_j^T \mathbf{r}^{k-1} / \|\mathbf{a}_j\|_2^2$.
- **Update Support:** Find a minimizer j_0 of $\epsilon(j)$: $\forall\ j \notin \mathcal{S}^{k-1}$, $\epsilon(j_0) \leq \epsilon(j)$, and update $\mathcal{S}^k = \mathcal{S}^{k-1} \cup \{j_0\}$.
- **Update Provisional Solution:** Compute $\mathbf{x}^k$, the minimizer of $\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$ subject to $Support\{\mathbf{x}\} = \mathcal{S}^k$.
- **Update Residual:** Compute $\mathbf{r}^k = \mathbf{b} - \mathbf{A}\mathbf{x}^k$.
- **Stopping Rule:** If $\|\mathbf{r}^k\|_2 < \epsilon_0$, stop. Otherwise, apply another iteration.

Output: The proposed solution is $\mathbf{x}^k$ obtained after k iterations.

Geometry of ℓ_p-Norm



Pursuit Algorithms

Convex relaxation technique is a second way to render (P_0) more tractable.

Convexifying with the ℓ_1 norm, we come to the new optimization problem

$$(P_1) \quad \min_{\mathsf{x}} \|W\mathsf{x}\|_1 \quad \text{s.t.} \quad A\mathsf{x} = \mathsf{b} \tag{2}$$

where W is a diagonal positive-definite matrix that introduces the precompensating weights.

It was named *Basis Pursuit* (BP) when all the columns of A are normalized (and thus $W = I$).

Pursuit Algorithms

THEOREM 06. For a system of linear equations $A\mathrm{x} = \mathrm{b}$, if a solution x exists obeying $\|\mathrm{x}\|_0 < (1 + 1/\mu(A))/2$, then an OMP algorithm run with threshold parameter $\epsilon_0 = 0$ is guaranteed to find it exactly.

THEOREM 07. For a system of linear equations $A\mathrm{x} = \mathrm{b}$, if a solution x exists obeying $\|\mathrm{x}\|_0 < (1 + 1/\mu(A))/2$, that solution is both the unique solution of (P_1) and the unique solution of (P_0).

From Exact to Approximate Solutions

An error-tolerant version of (P_0) is defined by

$$(P_0^\epsilon) \quad \min_{\mathrm{x}} \|\mathrm{x}\|_0 \quad \text{s.t.} \quad \|\mathrm{b} - A\mathrm{x}\| \leqslant \epsilon. \tag{3}$$

THEOREM 08. Consider the instance of problem (P_0^ϵ) defined by the triplet $(A; \mathrm{b}; \epsilon)$. Suppose that a sparse vector x_0 satisfies the sparsity constraint $\|\mathrm{x}_0\|_0 < (1 + 1/\mu(A))/2$, and gives a representation of b to within error tolerance ϵ (i.e., $\|\mathrm{b} - A\mathrm{x}_0\| \leqslant \epsilon$). Every solution x_0^ϵ of (P_0^ϵ) must obey

$$\|\mathrm{x}_0^\epsilon - \mathrm{x}_0\|_2^2 \leqslant \frac{4\epsilon^2}{1 - \mu(A)(2\|\mathrm{x}_0\|_0 - 1)}.$$

From Exact to Approximate Solutions

An error-tolerant version of (P_1) is defined by

$$(P_1^\epsilon) \quad \min_{\mathrm{x}} \|\mathrm{x}\|_1 \quad \text{s.t.} \quad \|\mathrm{b} - A\mathrm{x}\| \leqslant \epsilon. \tag{4}$$

THEOREM 09. Consider the instance of problem (P_1^ϵ) defined by the triplet $(A; \mathrm{b}; \epsilon)$. Suppose that a sparse vector x_0 is a feasible solution to (P_1^ϵ) satisfying the sparsity constraint $\|\mathrm{x}_0\|_0 < (1 + 1/\mu(A))/4$. The solution x_1^ϵ of (P_1^ϵ) must obey

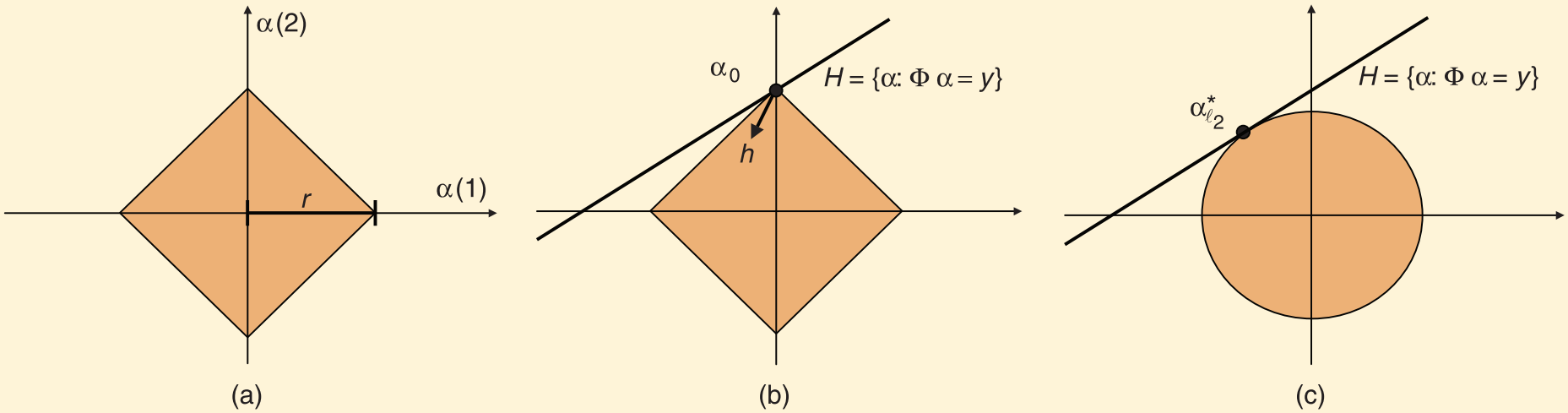
$$\|\mathrm{x}_1^\epsilon - \mathrm{x}_0\|_2^2 \leqslant \frac{4\epsilon^2}{1 - \mu(A)(4\|\mathrm{x}_0\|_0 - 1)}.$$

Restricted Isometry Property

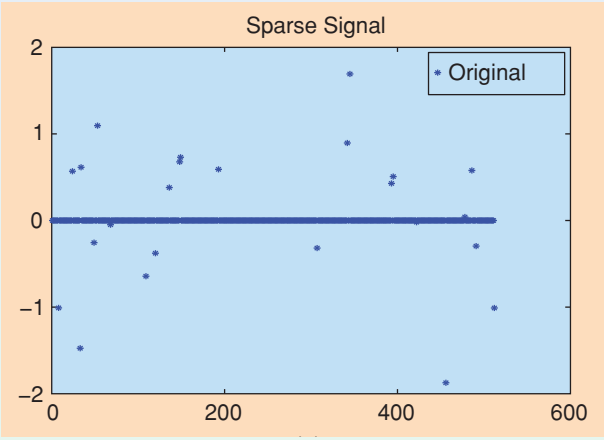
DEFINITION 10. A matrix $A \in \mathbb{R}^{m \times n}$ is said to have the restricted isometry property $RIP(\delta; s)$ if each submatrix A_s formed by combining at most s columns of A has its nonzero singular values bounded above by $1 + \delta$ and below by $1 - \delta$.

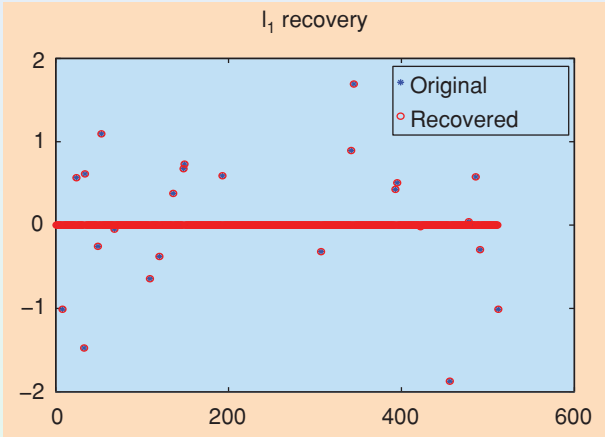
THEOREM 11. Candès and Tao have shown that $A \in RIP(\sqrt{2} - 1; 2s)$ implies that (P_1) and (P_0) have identical solutions on all s-sparse vectors and, moreover, that (P_1^ϵ) stably approximates the sparsest near-solution of $\mathrm{b} = A\mathrm{x} + \mathrm{v}$ with a reasonable stability coefficient.

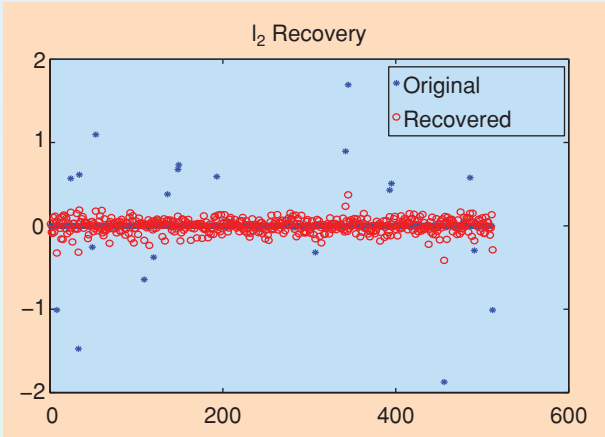
ℓ_1 and ℓ_2 Recovery



ℓ_1 and ℓ_2 Recovery







Compressed Sensing

Designing measurement/sensing matrices with favorable properties and constructing suitable transforms/dictionaries are the important research topics in Compressed Sensing.

Outline I

Outline II

Sparse Modeling

All the previous theorems have shown us that the problem of finding a sparse solution to an under-determined linear system can be given a meaningful definition and can also be computationally tractable.

We now turn to discuss the applicability of these ideas to signal, image, and geometric processing.

Priors and transforms for signals

The Bayesian framework imposes a Probability-Density-Function (PDF) on the signals – a prior distribution $P(\mathrm{y})$.

Priors are extensively used in signal processing, serving in inverse problems, compression, anomaly detection, and more.

Priors and transforms for signals

Consider the denoising problem: a given image b is known to be a noisy version of a clean image y, contaminated by an additive perturbation vector v, known to have a finite energy $\|\mathrm{v}\|_2 \leqslant \epsilon$, i.e., $\mathrm{b} = \mathrm{y} + \mathrm{v}$.

The optimization problem

$$\max_{\mathrm{y}} P(\mathrm{y}) \quad \text{s.t.} \quad \|\mathrm{y} - \mathrm{b}\|_2 \leqslant \epsilon$$

leads to the most probable image $\hat{\mathrm{y}}$ that is an effective estimate of y.

This way the prior is exploited for solving the noise cleaning problem. The above formulation of the denoising problem is in fact that Maximum-A-posteriori-Probability (MAP) estimator.

Priors and transforms for signals

Much effort has been allocated in the signal and image processing communities for forming priors as closed-form expressions.

One very common way to construct $P(\mathrm{y})$ is to guess its structure based on intuitive expectations from the data content. For example, the Gibbs distribution $P(\mathrm{y}) = Const \cdot exp\{-\lambda \|L\mathrm{y}\|_2^2\}$ uses a Laplacian matrix to give an evaluation of the probability of the image y.

In such a prior, smoothness, measured by the Laplacian operator, is used for judging the probability of the signal.

Priors and transforms for signals

This prior is well-known and extensively used in signal processing, and is known to be related to both Tikhonov regularization and Wiener filtering.

The prior leads to an optimization problem of the form

$$\min \|L\mathrm{y}\|_2^2 \quad \text{s.t.} \quad \|\mathrm{y}-\mathrm{b}\|_2 \leqslant \epsilon$$

which can be converted to

$$\min \lambda\|L\mathrm{y}\|_2^2 + \|\mathrm{y}-\mathrm{b}\|_2^2$$

where we have replaced the constraint by an equivalent penalty.

The closed-form solution is easily obtained as

$$\hat{\mathrm{y}} = (I + \lambda L^T L)^{-1}\mathrm{b}.$$

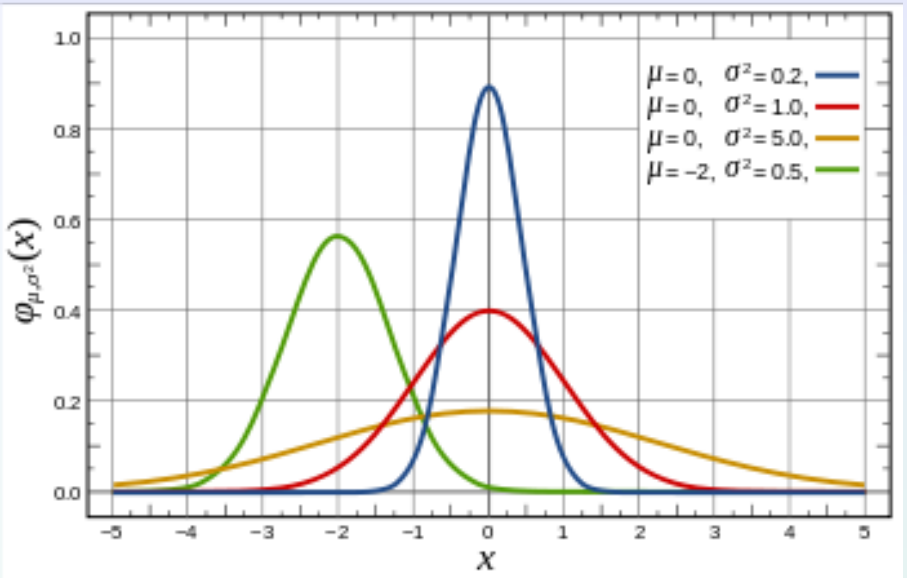
Priors and transforms for signals

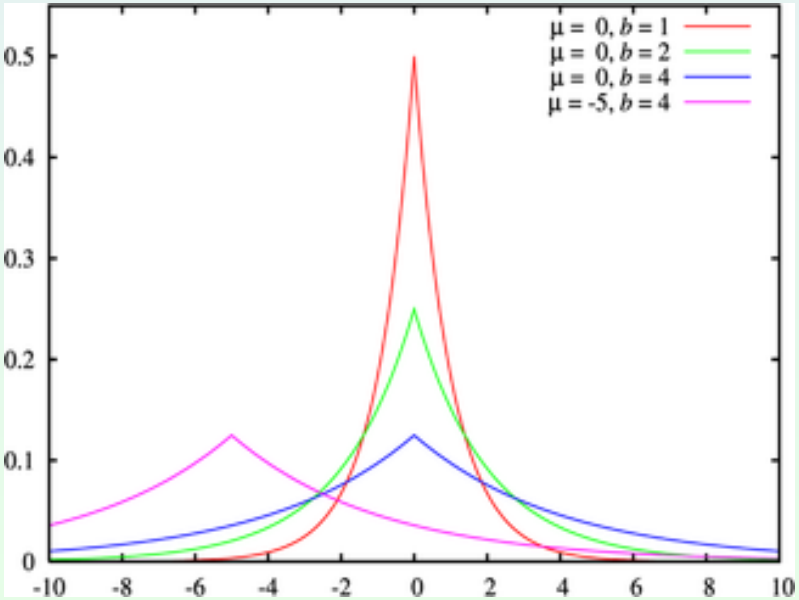
The above specific prior stressing smoothness is known to cause blurring of the image when used in various restoration tasks. The remedy for this problem was found to be the replacement of the ℓ_2-norm by a more robust measure, such as an ℓ_1-norm, that allows heavy tails for the distribution of the values of $L\mathrm{y}$.

A prior of the form $P(\mathrm{y}) = Const \cdot exp\{-\lambda \|L\mathrm{y}\|_1\}$ is far more versatile and thus became popular in recent years.

Similar to this option is the Total-Variation (TV) prior $P(\mathrm{y}) = Const \cdot exp\{-\lambda \|\mathrm{y}\|_{TV}\}$ [Rudin, Osher, and Fatemi, 1993] that also promotes smoothness, but differently, by replacing the Laplacian with gradient norms.

Priors and transforms for signals





Priors and transforms for signals

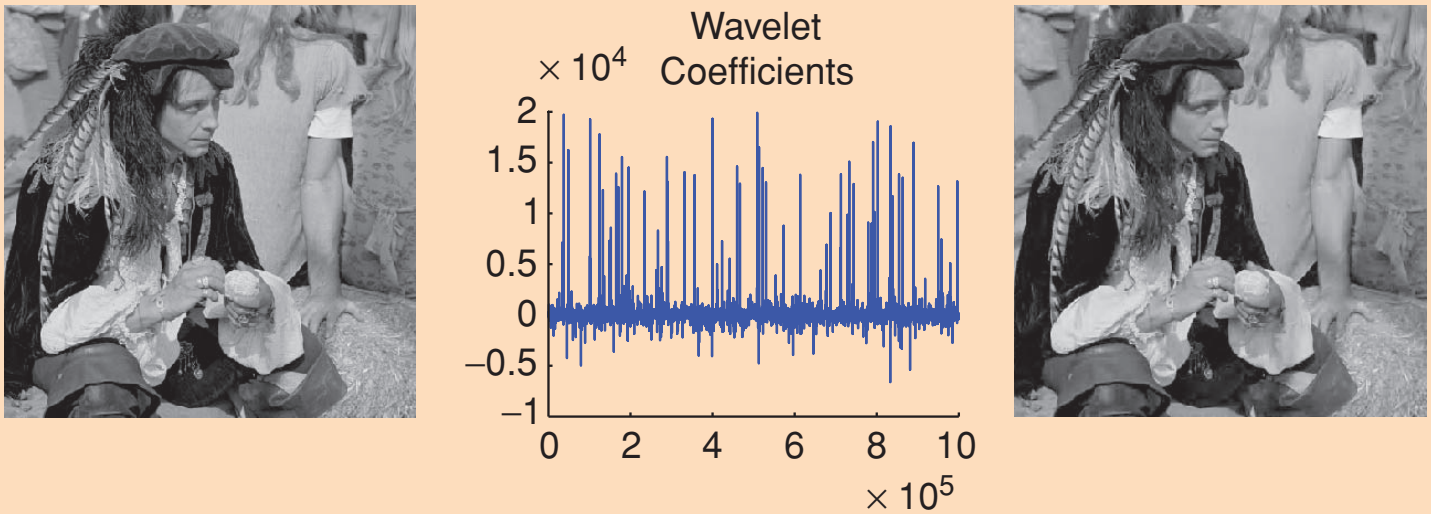
A different property that can be used for constructing a prior is assuming a structure on the signals transform-coefficients.

One such example is the JPEG compression algorithm, which relies on the fact that 2D-DCT coefficients of small image patches tend to behave in a predicted way (being concentrated around the origin).

Another well-known example refers to the wavelet transform of signals and images, where the coefficients are expected to be sparse, most of them tending to zero while few remain active.

Priors and transforms for signals

For a signal y, the wavelet transform is given by $T\mathrm{y}$ where the matrix T is a specially designed orthogonal matrix that contains in its rows spatial derivatives of varying scale, thereby providing what is known as multi-scale analysis of the signal.



Therefore, the prior in this case becomes
$P(\mathrm{y}) = Const \cdot exp\{-\lambda \| T\mathrm{y} \|_p^p\}$ with $p \leqslant 1$ to promote sparsity.

Priors and transforms for signals

A rich family of signal priors assign likelihood for an image based on the behavior of its transform coefficients $T\mathrm{y}$. In the signal and image processing literature, such priors were postulated in conjunction with a variety of transforms, such as

- the Discrete-Fourier-Transform (DFT)
- the Discrete-Cosine-Transform (DCT)
- the Hadamard-Transform (HT)
- the Principal-Component-Analysis (PCA)

Bayesian perspective

- Using Bayes' rule, the posterior probability of y given the measurements is formulated by

$$P(\mathrm{y} \mid \mathrm{z}) = \frac{P(\mathrm{z} \mid \mathrm{y})P(\mathrm{y})}{P(\mathrm{z})}.$$

- Considering the fact that the denominator $P(\mathrm{z})$ is not a function of the unknown y, and as such it can be disregarded, the MAP estimation amounts to

$$\hat{\mathrm{y}}_{\mathrm{MAP}} = \arg\max_{\mathrm{y}} P(\mathrm{y} \mid \mathrm{z}) = \arg\max_{\mathrm{y}} P(\mathrm{z} \mid \mathrm{y})P(\mathrm{y}).$$

- The probability $P(\mathrm{z} \mid \mathrm{y})$ is known as the likelihood function, and the probability $P(\mathrm{y})$ is the known/unknown's prior.

The Sparse-Land model

The linear system $D\mathrm{x} = \mathrm{y}$ can be interpreted as a way of constructing signals y. Every column in D is a possible signal in $\mathbb{R}^n$ – we refer to these m columns as atomic signals, and the matrix D displays a dictionary of atoms.

The multiplication of D by a sparse vector x with $\|\mathrm{x}\|_0 = k_0 \ll n$ produces a linear combination of k_0 atoms with varying portions, generating the signal y. The vector x that generates y will be called its representation.

The Sparse-Land model

The Sparse-Land model $\mathcal{M}(D, k_0, \alpha, \epsilon)$:

$$\mathrm{y} = D\mathrm{x} + \mathrm{v}$$

- Consider all the possible sparse representation vectors with cardinality $\|\mathrm{x}\|_0 = k_0 \ll n$, and assume that this set of $C_m^{k_0}$ possible cardinalities are drawn with uniform probability.
- Assume further that the non-zero entries in x are drawn from the zero-mean Gaussian distribution $Const \cdot exp\{-\alpha x_i^2\}$.
- Postulate that the observations are contaminated by a random perturbation (noise) vector $\mathrm{v} \in \mathbb{R}^n$ with bounded power $\|\mathrm{v}\|_2 \leqslant \epsilon$.

The quest for a dictionary

In the quest for the proper dictionary to use in applications, one line of work considers choosing pre-constructed dictionaries, such as undecimated wavelets, steerable wavelets, contourlets, curvelets, and more.

Some of these proposed dictionaries (which are often referred to also as transforms) are accompanied by a detailed theoretical analysis establishing the sparsity of the representation coefficients for such simplified content of signals.

The quest for a dictionary

While pre-constructed or adapted dictionaries typically lead to fast transforms, they are typically limited in their ability to sparsify the signals they are designed to handle. Furthermore, most of those dictionaries are restricted to signals/images of a certain type, and cannot be used for a new and arbitrary family of signals of interest.

This leads us to yet another approach for obtaining dictionaries that overcomes these limitations – by adopting a learning point-of-view.

As opposed to the pre-constructed and adapted dictionaries, the learning method is able to adapt to any family of signals that complies with the Sparse-Land model.

Dictionary learning

Assume that a training database $\{\mathrm{y}_j\}_{j=1}^N$ is given, and thought to have been generated by some fixed but unknown model $\mathcal{M}(D, k_0, \alpha, \epsilon)$.

- Control the deviation:

$$\min_{D,\{\mathrm{x}_j\}_{j=1}^N} \sum_{j=1}^N \|\mathrm{x}_j\|_0 \quad \text{s.t.} \quad \|\mathrm{y}_j - D\mathrm{x}_j\|_2 \leqslant \epsilon,\ j = 1, \cdots, N$$

- Control the sparsity:

$$\min_{D,\{\mathrm{x}_j\}_{j=1}^N} \sum_{j=1}^N \|\mathrm{y}_j - D\mathrm{x}_j\|_2^2 \quad \text{s.t.} \quad \|\mathrm{x}_j\|_0 \leqslant k_0,\ j = 1, \cdots, N$$

Analysis versus Synthesis

- Synthesis based modeling

$$(P_s) \quad \hat{\mathrm{y}}_s = D \cdot \arg\min_{\mathrm{x}} \|\mathrm{x}\|_p \quad \text{s.t.} \quad \|\mathrm{z} - D\mathrm{x}\| \leqslant \epsilon. \tag{5}$$

- Analysis based modeling

$$(P_a) \quad \hat{\mathrm{y}}_a = \arg\min_{\mathrm{y}} \|T\mathrm{y}\|_p \quad \text{s.t.} \quad \|\mathrm{z} - \mathrm{y}\| \leqslant \epsilon. \tag{6}$$

Dictionary learning algorithms

The optimization problem for dictionary learning (sparse representations and coding):

$$\min_{D,X} \|Y - DX\|_{Frob} \quad \text{s.t.} \quad \|x_j\|_0 \leqslant k_0,\ j = 1, \cdots, N \tag{7}$$

where

$$Y = (y_1, \cdots, y_N) \in \mathbb{R}^{n \times N},$$

$$D = (d_1, \cdots, d_m) \in \mathbb{R}^{n \times m},$$

$$X = (x_1, \cdots, x_N) \in \mathbb{R}^{m \times N}.$$

Dictionary learning algorithms

There are two training mechanisms, the first named Method of Optimal Directions (MOD) by Engan et al., and the second named K-SVD, by Aharon et al..

- MOD
- K-SVD
- .....

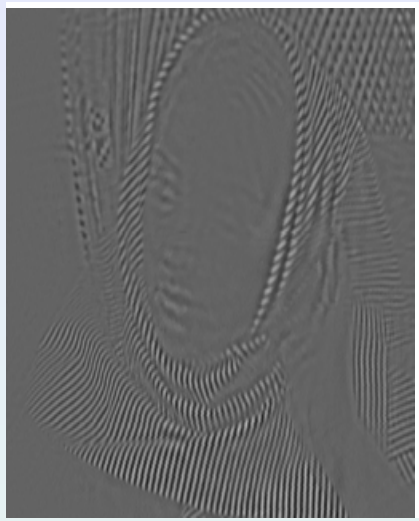
Applications

- Image deblurring
- Facial image compression
- Image denoising
- Image inpainting
- .....

Applications

Original image

Cartoon part



Texture part

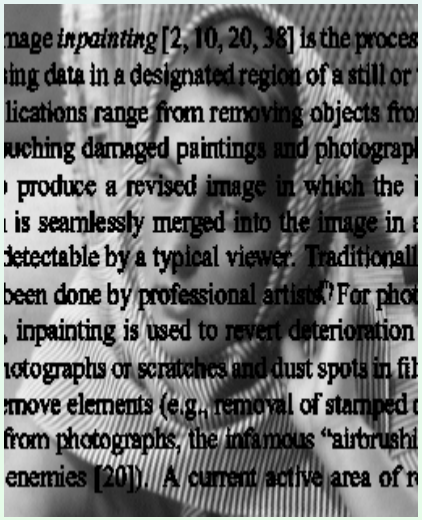
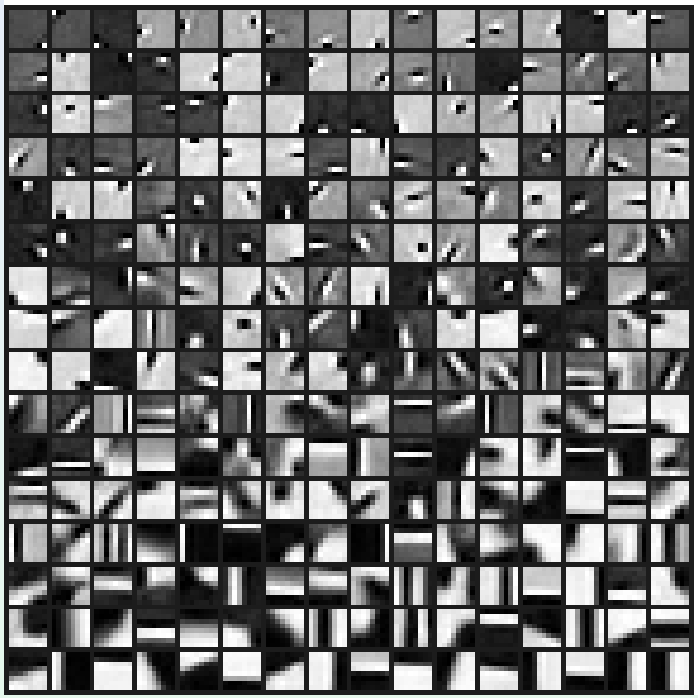
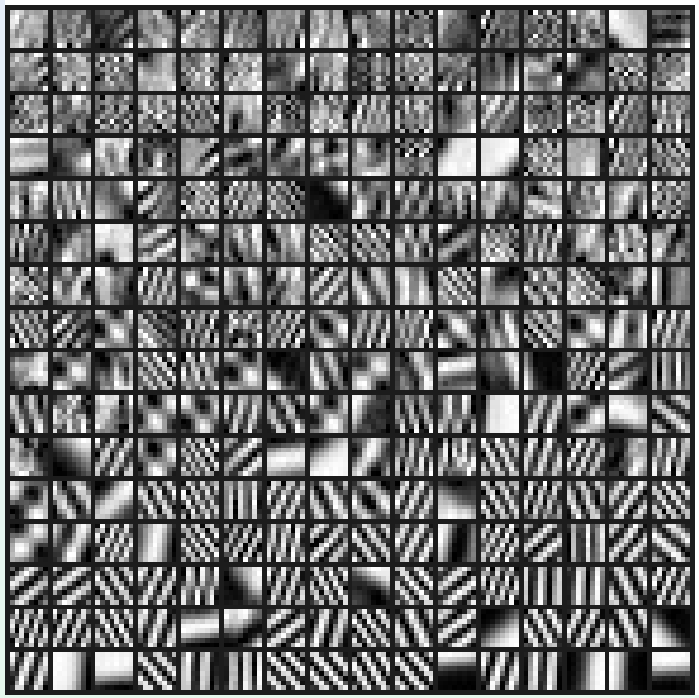


Image with missing data

Inpainting result

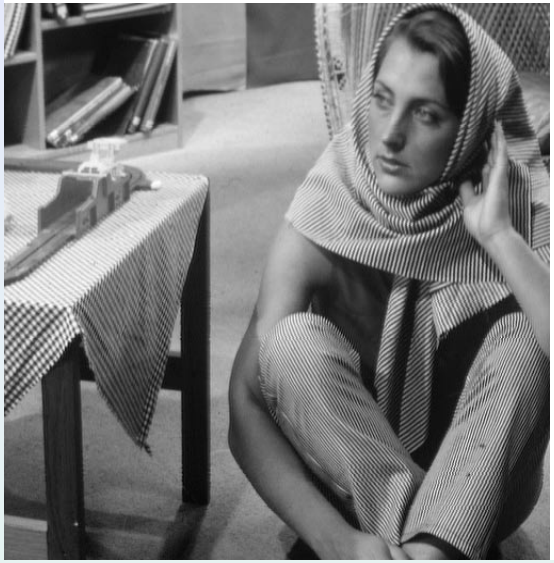
Applications



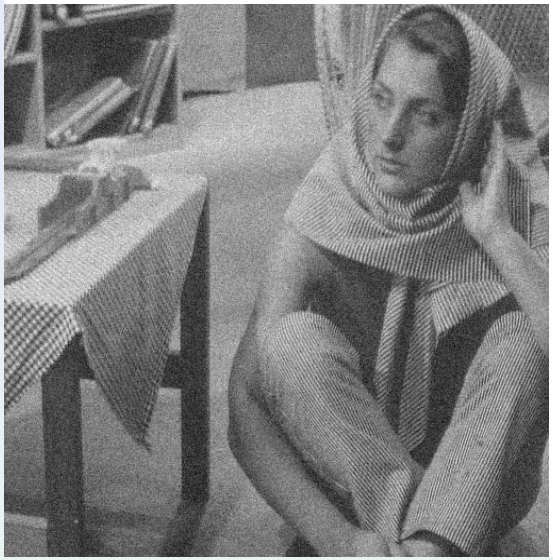


Applications

Original Image



Noisy Image (22.1307 dB, σ=20)



Denoised Image Using
Global Trained Dictionary (28.8528 dB)

Denoised Image Using
Adaptive Dictionary (30.8295 dB)

References

1. Emmanuel J. Candès, Terence Tao. Decoding by linear programming. Information Theory, IEEE Transactions on 51.12 (2005): 4203-4215.
2. Emmanuel J. Candès, Justin Romberg, Terence Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. Information Theory, IEEE Transactions on 52.2 (2006): 489-509.
3. Emmanuel J. Candès, Michael B. Wakin. An introduction to compressive sampling. Signal Processing Magazine, IEEE 25.2 (2008): 21-30.
4. Alfred M. Bruckstein, David L. Donoho, Michael Elad. From sparse solutions of systems of equations to sparse modeling of signals and images. SIAM review 51.1 (2009): 34-81.
5. Michael Elad. Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing. Springer, 2010.

Outline I

Outline II

Sparse Optimization

Many problems of recent interest in statistics and related areas can be posed in the framework of sparse optimization. Due to the explosion in size and complexity of modern data analysis (BigData), it is increasingly important to be able to solve problems with a very large number of features, training examples, or both.

0-norm optimization

$$(P_0) \quad \min_{x} \|x\|_0 \quad \text{s.t.} \quad Ax = b. \tag{8}$$

$$(P_0^\epsilon) \quad \min_{x} \|x\|_0 \quad \text{s.t.} \quad \|b - Ax\| \leqslant \epsilon. \tag{9}$$

Greedy algorithms

Greedy strategies are usually adopted in solving the 0-norm problems. The following algorithm is known in the literature of signal processing by the name *Orthogonal Matching Pursuit* (OMP).

Task: Approximate the solution of (P_0): $\min_{\mathbf{x}} \|\mathbf{x}\|_0$ subject to $\mathbf{A}\mathbf{x} = \mathbf{b}$.

Parameters: We are given the matrix $\mathbf{A}$, the vector $\mathbf{b}$, and the error threshold ϵ_0.

Initialization: Initialize $k = 0$, and set

- The initial solution $\mathbf{x}^0 = 0$.
- The initial residual $\mathbf{r}^0 = \mathbf{b} - \mathbf{A}\mathbf{x}^0 = \mathbf{b}$.
- The initial solution support $\mathcal{S}^0 = Support\{\mathbf{x}^0\} = \emptyset$.

Main Iteration: Increment k by 1 and perform the following steps:

- **Sweep:** Compute the errors $\epsilon(j) = \min_{z_j} \|\mathbf{a}_j z_j - \mathbf{r}^{k-1}\|_2^2$ for all j using the optimal choice $z_j^* = \mathbf{a}_j^T \mathbf{r}^{k-1} / \|\mathbf{a}_j\|_2^2$.
- **Update Support:** Find a minimizer j_0 of $\epsilon(j)$: $\forall\ j \notin \mathcal{S}^{k-1}$, $\epsilon(j_0) \leq \epsilon(j)$, and update $\mathcal{S}^k = \mathcal{S}^{k-1} \cup \{j_0\}$.
- **Update Provisional Solution:** Compute $\mathbf{x}^k$, the minimizer of $\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$ subject to $Support\{\mathbf{x}\} = \mathcal{S}^k$.
- **Update Residual:** Compute $\mathbf{r}^k = \mathbf{b} - \mathbf{A}\mathbf{x}^k$.
- **Stopping Rule:** If $\|\mathbf{r}^k\|_2 < \epsilon_0$, stop. Otherwise, apply another iteration.

Output: The proposed solution is $\mathbf{x}^k$ obtained after k iterations.

Dictionary learning

The optimization model of dictionary learning for sparse and redundant representations:

$$\min_{D,X} \|Y - DX\|_{Frob} \quad \text{s.t.} \quad \|x_j\|_0 \leqslant k_0,\ j = 1, \cdots, N \tag{10}$$

where

$$Y = (y_1, \cdots, y_N) \in \mathbb{R}^{n \times N},$$

$$D = (d_1, \cdots, d_m) \in \mathbb{R}^{n \times m},$$

$$X = (x_1, \cdots, x_N) \in \mathbb{R}^{m \times N}.$$

Dictionary learning

There are two training mechanisms, the first named Method of Optimal Directions (MOD) by Engan et al., and the second named K-SVD, by Aharon et al..

- MOD
- K-SVD
- .....

Convex relaxation

Convex relaxation technique is a way to render 0-norm more tractable.

Convexifying with the ℓ_1 norm, we come to the new optimization problem

$$(P_1) \quad \min_{\mathrm{x}} \|W\mathrm{x}\|_1 \quad \text{s.t.} \quad A\mathrm{x} = \mathrm{b} \tag{11}$$

where W is a diagonal positive-definite matrix that introduces the precompensating weights.

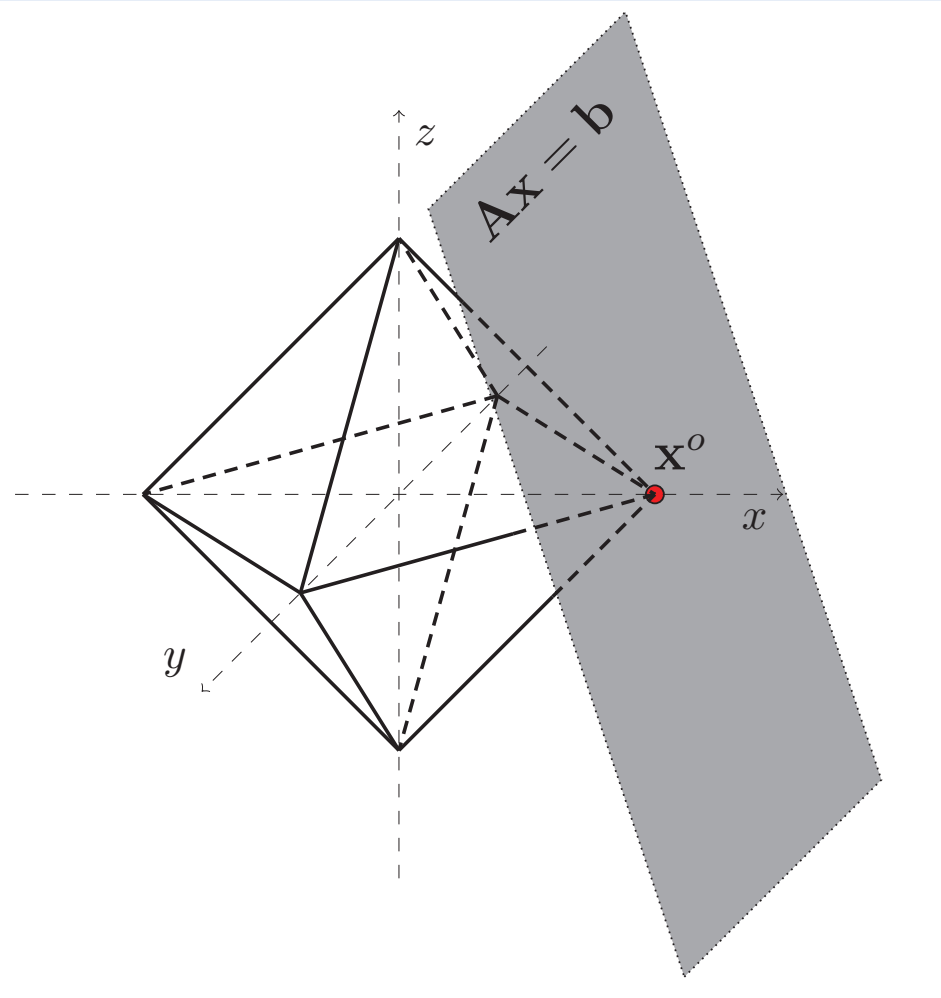
An error-tolerant version of (P_1) is defined by

$$(P_1^\epsilon) \quad \min_{\mathrm{x}} \|W\mathrm{x}\|_1 \quad \text{s.t.} \quad \|\mathrm{b} - A\mathrm{x}\| \leqslant \epsilon. \tag{12}$$

It was named *Basis Pursuit* (BP) when all the columns of A are normalized (and thus $W = I$).

Basic pursuit

$$(BP) \quad \min_{x} \|x\|_1 \quad \text{s.t.} \quad Ax = b.$$



BP denoising and LASSO

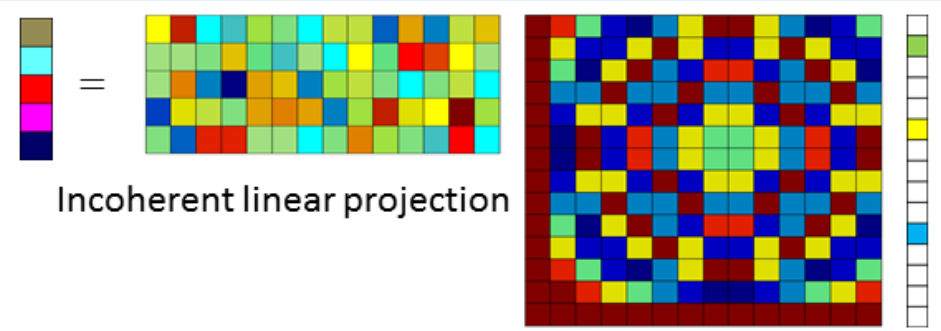
$$
\begin{aligned}
(BP_\tau) \quad & \min_{\mathsf{x}} \|A\mathsf{x}-\mathsf{b}\|_2^2 \quad \text{s.t.} \quad \|\mathsf{x}\|_1 \leqslant \tau, \\
(BP_\mu) \quad & \min_{\mathsf{x}} \|\mathsf{x}\|_1 + \tfrac{\mu}{2}\|A\mathsf{x}-\mathsf{b}\|_2^2, \\
(BP_\delta) \quad & \min_{\mathsf{x}} \|\mathsf{x}\|_1 \quad \text{s.t.} \quad \|A\mathsf{x}-\mathsf{b}\|_2 \leqslant \delta.
\end{aligned}
$$

Questions:

- Are they equivalent? and in what sense?
- How to choose parameters?

Sparse under basis Ψ

$$\min_{s}\{\|s\|_1 : A\Psi s = b\}$$



If Ψ is orthogonal, the problem is equivalent to

$$\min_{x}\{\|\Psi^* x\|_1 : Ax = b\}.$$

Sparse after transform $\mathcal{L}$

$$\min_{x}\{\|\mathcal{L}x\|_1 : Ax = b\}$$

Examples of $\mathcal{L}$:

- DCT, wavelets, curvelets, ridgelets, ...
- tight frames, Gabor, ...
- total (generalized) variation

Ref: E. J. Cands, Y. Eldar, D. Needell and P. Randall. Compressed sensing with coherent and redundant dictionaries. Applied and Computational Harmonic Analysis, 31(1): 59-73.

Joint/group sparsity

Decompose $\{1, 2, \cdots, n\} = \mathcal{G}_1 \bigcup \mathcal{G}_2 \bigcup \cdots \bigcup \mathcal{G}_S$, and $\mathcal{G}_i \bigcap \mathcal{G}_j = \emptyset, i \neq j$.

Joint/group sparse recovery model:

$$\min_{x} \{\|x\|_{\mathcal{G},2,1} : Ax = b\}$$

where

$$\|x\|_{\mathcal{G},2,1} = \sum_{s=1}^{S} w_s \|x_{\mathcal{G}_s}\|_2.$$

Side constraints

- Nonnegativity: $x \geq 0$
- Box constraints: $lb \leq x \leq ub$
- Linear inequalities: $Qx \leq c$

They generate "corners" and can be very effective in practice.

Shrinkage

- Shrinkage is popular in sparse optimization algorithms
- In optimization, non-smooth functions like ℓ_1 has difficulty using general smooth optimization methods.
- But, ℓ_1 is component-wise separable, so it does get along well with separable (smooth or non-smooth) functions.
- For example,

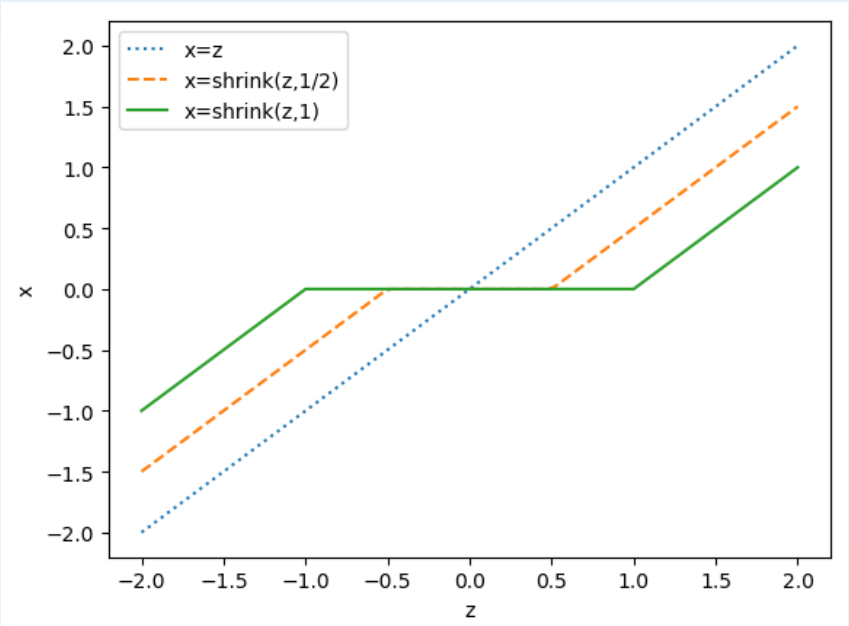
$$\min_{\mathrm{x}} \|\mathrm{x}\|_1 + \frac{1}{2\tau}\|\mathrm{x} - \mathrm{z}\|_2^2$$

is equivalent to solving $\min_{x_i} |x_i| + \frac{1}{2\tau}|x_i - z_i|^2$ over each i.

Soft-thresholding shrinkage

The problem is separable and has an explicit solution

$$(\text{shrink}(\mathbf{z},\tau))_i = \begin{cases} z_i - \tau & z_i > \tau, \\ 0 & -\tau \le z_i \le \tau, \\ z_i + \tau & z_i < -\tau. \end{cases}$$



The shrinkage operator can be written in Matlab code as:

$$\mathbf{x} = \max(\text{abs}(\mathbf{z})\text{-tau},0).*\text{sign}(\mathbf{z}).$$

Soft-thresholding shrinkage

- The following problem is called Moreau-Yosida regularization

$$\min_{x} r(x) + \frac{1}{2\tau}\|x - z\|_2^2.$$

- For example $r(x) = \|x\|_2$, the solution to

$$\min_{x} \|x\|_2 + \frac{1}{2\tau}\|x - z\|_2^2$$

is, if we treat $0/0 = 0$,

$$x_{opt} = \max\{\|z\|_2 - \tau, 0\} \cdot (z/\|z\|_2).$$

- Used in joint/group-sparse recovery algorithms.

Soft-thresholding shrinkage

- Consider the following nuclear norm optimization

$$\min_{X} \|X\|_* + \frac{1}{2\tau}\|X - Z\|_F^2.$$

  Let $Z = U\Sigma V^T$ be the singular value decomposition of Z.

- Let $\hat{\Sigma}$ be the diagonal matrix with diagonal entries

$$\mathrm{diag}(\hat{\Sigma}) = \mathrm{shrink}(\mathrm{diag}(\Sigma), \tau)),$$

  then

$$X_{opt} = U\hat{\Sigma}V^T.$$

- In general, matrix problems with only unitary-invariant functions (e.g., $\|\cdot\|_*$, $\|\cdot\|_F$, spectral norm, trace) and constraints (e.g., positive or negative semi-definiteness) typically reduce to vector problems regarding singular values.

Prox-linear algorithm

Consider the general form

$$\min_{x} r(x) + f(x).$$

where r is the regularization function and f is the data fidelity function.

The prox-linear algorithm is:

$$x^{k+1} = \arg\min_{x} r(x) + f(x^k) + < \nabla f(x^k), x - x^k > + \frac{1}{2\delta_k}\|x - x^k\|_2^2.$$

The last term keeps x^{k+1} close to x^k, and the parameter δ_k determines the step size. It is equivalent to

$$x^{k+1} = \arg\min_{x} r(x) + \frac{1}{2\delta_k}\|x - (x^k - \delta_k \nabla f(x^k))\|_2^2.$$

Alternating direction method of multipliers (ADMM)

The Alternating Direction Method of Multipliers (ADMM) was developed in the 1970s, with roots in the 1950s, and is equivalent or closely related to many other algorithms, such as dual decomposition, the method of multipliers, Douglas-Rachford splitting, Spingarns method of partial inverses, Dykstras alternating projections, Bregman iterative algorithms for 1-norm problems, proximal methods, and others.

ADMM

The ADMM can be applied to a wide variety of statistical and machine learning problems of recent interest, including the lasso, sparse logistic regression, basis pursuit, covariance selection, support vector machines, and many others.

ADMM

$$\min_{X \in C^{n \times T}} \mu \|X\|_p + \|AX - B\|_q \tag{13}$$

Let $p := \{2, 1\}, q := \{1, 1\}$ which denote joint convex norm, we have

$$\min_{X \in C^{n \times T}} \mu \|X\|_{2,1} + \|AX - B\|_{1,1}$$

where $\|X\|_{2,1} = \sum_{i=1}^{n} \sqrt{\sum_{j=1}^{T} x_{ij}^2}$, $\|X\|_{1,1} = \sum_{i=1}^{n} \sum_{j=1}^{T} |x_{ij}|$.

For example $T = 1$,

$$\min_{x \in C^n} \mu \|x\|_p + \|Ax - b\|_q.$$

ADMM

$$\min_{X \in C^{n \times T}} \mu \|X\|_p + \|AX - B\|_q \tag{13}$$

Let $p := \{2, 1\}, q := \{1, 1\}$ which denote joint convex norm, we have

$$\min_{X \in C^{n \times T}} \mu \|X\|_{2,1} + \|AX - B\|_{1,1}$$

where $\|X\|_{2,1} = \sum_{i=1}^{n} \sqrt{\sum_{j=1}^{T} x_{ij}^2}$, $\|X\|_{1,1} = \sum_{i=1}^{n} \sum_{j=1}^{T} |x_{ij}|$.

For example $T = 1$,

$$\min_{x \in C^n} \mu \|x\|_p + \|Ax - b\|_q.$$

ADMM

$$\min_{X \in C^{n \times T}} \mu \|X\|_p + \|AX - B\|_q \tag{13}$$

Let $p := \{2, 1\}, q := \{1, 1\}$ which denote joint convex norm, we have

$$\min_{X \in C^{n \times T}} \mu \|X\|_{2,1} + \|AX - B\|_{1,1}$$

where $\|X\|_{2,1} = \sum_{i=1}^{n} \sqrt{\sum_{j=1}^{T} x_{ij}^2}$, $\|X\|_{1,1} = \sum_{i=1}^{n} \sum_{j=1}^{T} |x_{ij}|$.

For example $T = 1$,

$$\min_{x \in C^{n}} \mu \|x\|_p + \|Ax - b\|_q.$$

ADMM

$$\begin{aligned} \min\ & \mu\|z\|_p + \|y\|_q \\ \text{s.t.}\ & x - z = 0 \\ & Ax - y = b \end{aligned} \tag{14}$$

$$\begin{aligned} L(x, y, z, \lambda_y, \lambda_z, \rho) =& \mu\|z\|_p + \|y\|_q + \mathrm{Re}(\lambda_z^T(x - z) + \lambda_y^T(Ax - y - b)) \\ &+ \frac{\rho}{2}(\|x - z\|_2^2 + \|Ax - y - b\|_2^2) \end{aligned} \tag{15}$$

where $\lambda_y \in C^n, \lambda_z \in C^m$ are the Lagrangian multipliers and $\rho > 0$ is a penalty parameter.

ADMM

$$\begin{aligned} \min\ & \mu\|\mathsf{z}\|_p + \|\mathsf{y}\|_q \\ \text{s.t.}\ & \mathsf{x} - \mathsf{z} = 0 \\ & \mathsf{A}\mathsf{x} - \mathsf{y} = \mathsf{b} \end{aligned} \tag{14}$$

$$\begin{aligned} L(\mathsf{x}, \mathsf{y}, \mathsf{z}, \lambda_y, \lambda_z, \rho) =& \mu\|\mathsf{z}\|_p + \|\mathsf{y}\|_q + \mathrm{Re}(\lambda_z^T(\mathsf{x} - \mathsf{z}) + \lambda_y^T(\mathsf{A}\mathsf{x} - \mathsf{y} - \mathsf{b})) \\ & + \frac{\rho}{2}(\|\mathsf{x} - \mathsf{z}\|_2^2 + \|\mathsf{A}\mathsf{x} - \mathsf{y} - \mathsf{b}\|_2^2) \end{aligned} \tag{15}$$

where $\lambda_y \in C^n, \lambda_z \in C^m$ are the Lagrangian multipliers and $\rho > 0$ is a penalty parameter.

ADMM

$$\begin{cases} x^{k+1} := \arg\min \frac{1}{2}(\|x - z^k + u_z^k\|_2^2 + \|Ax - y^k - b + u_y^k\|_2^2) \\ y^{k+1} := \arg\min \|y\|_q + \frac{\rho}{2}\|y - (Ax^{k+1} - b) - u_y^k\|_2^2 \\ z^{k+1} := \arg\min \mu\|z\|_p + \frac{\rho}{2}\|z - x^{k+1} - u_z^k\|_2^2 \end{cases} \tag{16}$$

After solving three subproblems, we update the Lagrangian multipliers as follows:

$$\begin{cases} u_z^{k+1} = u_z^k + \gamma(x^{k+1} - z^{k+1}) \\ u_y^{k+1} = u_y^k + \gamma(Ax^{k+1} - y^{k+1} - b) \end{cases} \tag{17}$$

where $u_y = \frac{1}{\rho}\lambda_y, u_z = \frac{1}{\rho}\lambda_z, \gamma > 0$ is the step size.

Outline I

Outline II

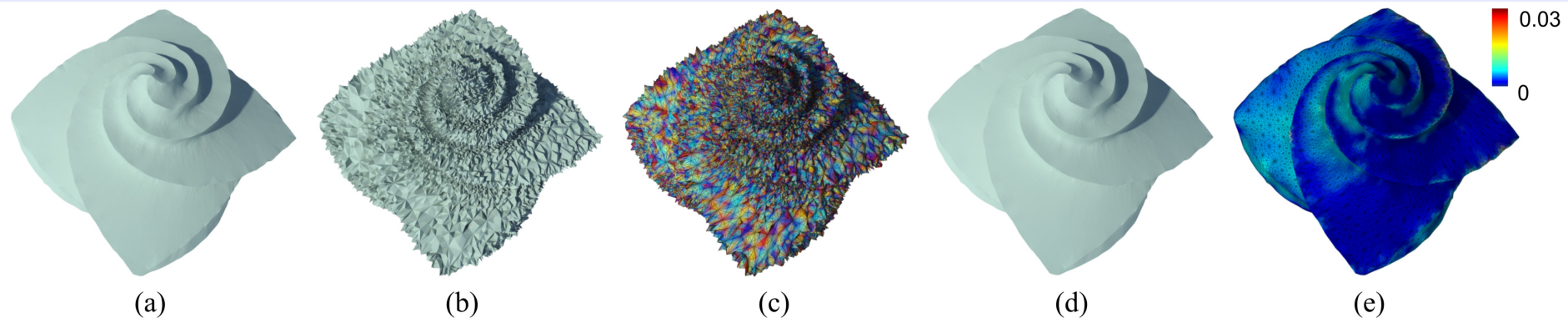


Figure: Our approach is able to faithfully recover sharp features corrupted by noise. (a) The octa-flower model as a ground truth; (b) the model artificially corrupted by independent and identically distributed (i.i.d.) noise (with zero mean and standard deviation $\sigma = 0.01$ of the diagonal of the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

Feature preserving surface denoising

- Surface fairing [Taubin 1995; Desbrun 1999]
- Anisotropic geometric diffusion [Clarenz 2000; Bajaj and Xu 2003]
- Bilateral filtering [Fleishman et al. 2003]
- Non-iterative smoothing [Jones et al. 2003]
- Anisotropic filtering [Hildebrandt and Polthier 2004]
- Normal-based bilateral filtering [Zheng et al. 2010]
- ...

Statistics

- Splines [Duchon 1977; Utreras 1988; Wahba 1990]
- Nonparametric regression [Stone 1982]
- Generalized cross validation (GCV) [Wahba and Wendelberger 1980]
- ...

Problem formulation

Assume that $P = \{\mathrm{p}_i\}_1^n$ are sampled possibly with noise from a C^2-smooth surface S, i.e.,

$$\mathrm{p}_i = \mathrm{s}_i + \varepsilon_i \mathrm{n}_i, \ \ i = 1, \cdots, n, \tag{18}$$

where n_i is the unit normal of surface at $\mathrm{s}_i \in S$ and $\varepsilon_i, i = 1, \cdots, n$, are i.i.d. random variables with zero mean and finite variance σ^2.

The goal of denoising is to produce a smooth surface $\hat{S} = \{\hat{\mathrm{s}}_i\}$ to approximate the true underlying surface S as much as possible.

Problem formulation

To find a C^2-smooth surface $\hat{S}$ which is a reasonable estimate of the true underlying surface S, we formulate it as the following variational minimization problem

$$\hat{S} = \arg\min_{S} \frac{1}{n} \sum_{i=1}^{n} d^2(\mathrm{p}_i, S) + \lambda \int_S (2H)^2, \tag{19}$$

where $d(\mathrm{p}_i, S)$ is the geometric distance from p_i to S, H is the mean curvature of S, and λ is a smoothness parameter.

Problem formulation

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

$$\begin{aligned} \mathcal{J}(S) &= \frac{1}{n}\sum_{i=1}^{n} 4H^2(\mathrm{s}_i) \\ &= \frac{1}{n}\sum_{i=1}^{n} \|\Delta_S \mathrm{s}_i\|^2 \simeq \frac{1}{n}\sum_{i=1}^{n} \|L_i S\|^2, \end{aligned} \tag{20}$$

where $S = (\mathrm{s}_1, \cdots, \mathrm{s}_n)^T \in \mathbb{R}^{n\times 3}$, Δ_S is the Laplace-Beltrami operator on surface S, and $L = (L_1^T, \cdots, L_n^T)^T$ is the discrete Laplacian matrix. The second equality in (20) is derived by $\Delta_S \mathrm{s}_i = 2H\mathrm{n}_i$.

DLRS model

Thus we arrive at the following denoising model:

$$\hat{S} = \arg\min_{S} \sum_{i=1}^{n} \|\mathrm{p}_i - \mathrm{s}_i\|^2 + \lambda \sum_{i=1}^{n} \|L_i S\|^2. \tag{21}$$

We call it the discrete Laplacian regularization smoothing (DLRS) model.

DLRS model

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

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

where I_n is the $n \times n$ identity matrix and $M = L^T L = \sum_{i=1}^{n} L_i^T L_i$.

Thus, given a specific value of the smoothness parameter λ, we have the solution

$$\hat{S}_n(\lambda) = (I_n + \lambda M)^{-1} P \tag{22}$$

as the estimated base mesh of the true underlying surface S.

Choice of the smoothness parameter λ

We adopt the *generalized cross validation* (GCV) to determine the smoothness parameter λ in our DLRS model. Specifically, the merit function of GCV is defined as

$$\mathrm{GCV}_n(\lambda) = \frac{\frac{1}{n}\|P - \hat{S}_n(\lambda)\|_F^2}{(1 - \frac{1}{n}\mathrm{tr}[A_n(\lambda)])^2}, \tag{23}$$

where $A_n(\lambda) = (I_n + \lambda M)^{-1}$. The optimal value of λ can be computed by minimizing the above GCV function, i.e.,

$$\hat{\lambda}_G = \arg\min_{\lambda>0} \mathrm{GCV}_n(\lambda). \tag{24}$$

Asymptotic properties

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

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

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

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

Asymptotic properties

Theorem 2. If the smoothness parameter $\hat{\lambda}_G$ is the GCV choice according to (24), then the estimated base surface $\hat{S}_n(\hat{\lambda}_G)$ from our DLRS model is asymptotically optimal, i.e.,

$$\frac{r_n(\hat{\lambda}_G)}{\inf\limits_{\lambda \in \mathbb{R}_+} r_n(\lambda)} \to_p 1,$$

where $\to_p$ means the convergence in probability.

Asymptotic properties

From the above theorems, it is seen that the estimated base surface $\hat{S}$ asymptotically converges to the ground truth surface S with probability one as the sample size goes to infinity.

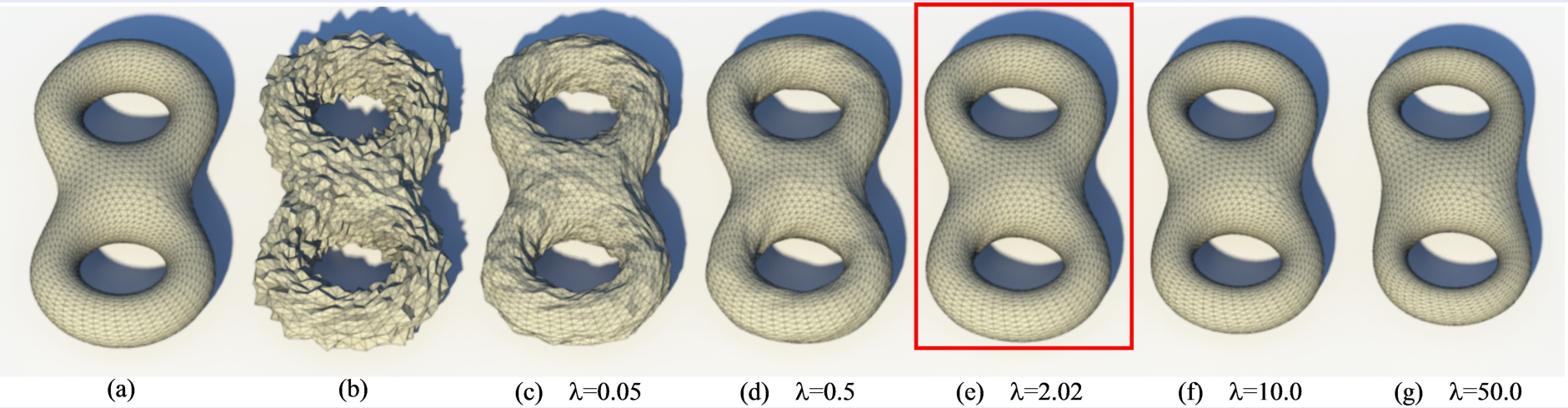


Figure: Our approach computes the optimal smoothness parameter to denoise the mesh. (a) the ground truth 8-like model which is a C^2-smooth surface; (b) the model artificially corrupted by severe synthetic noise; (c)-(g) denoised results by the global smoothing approach with various parameters λ in (21). Our approach obtains the best smoothing result (e) using the optimal parameter $\lambda = 2.02$.

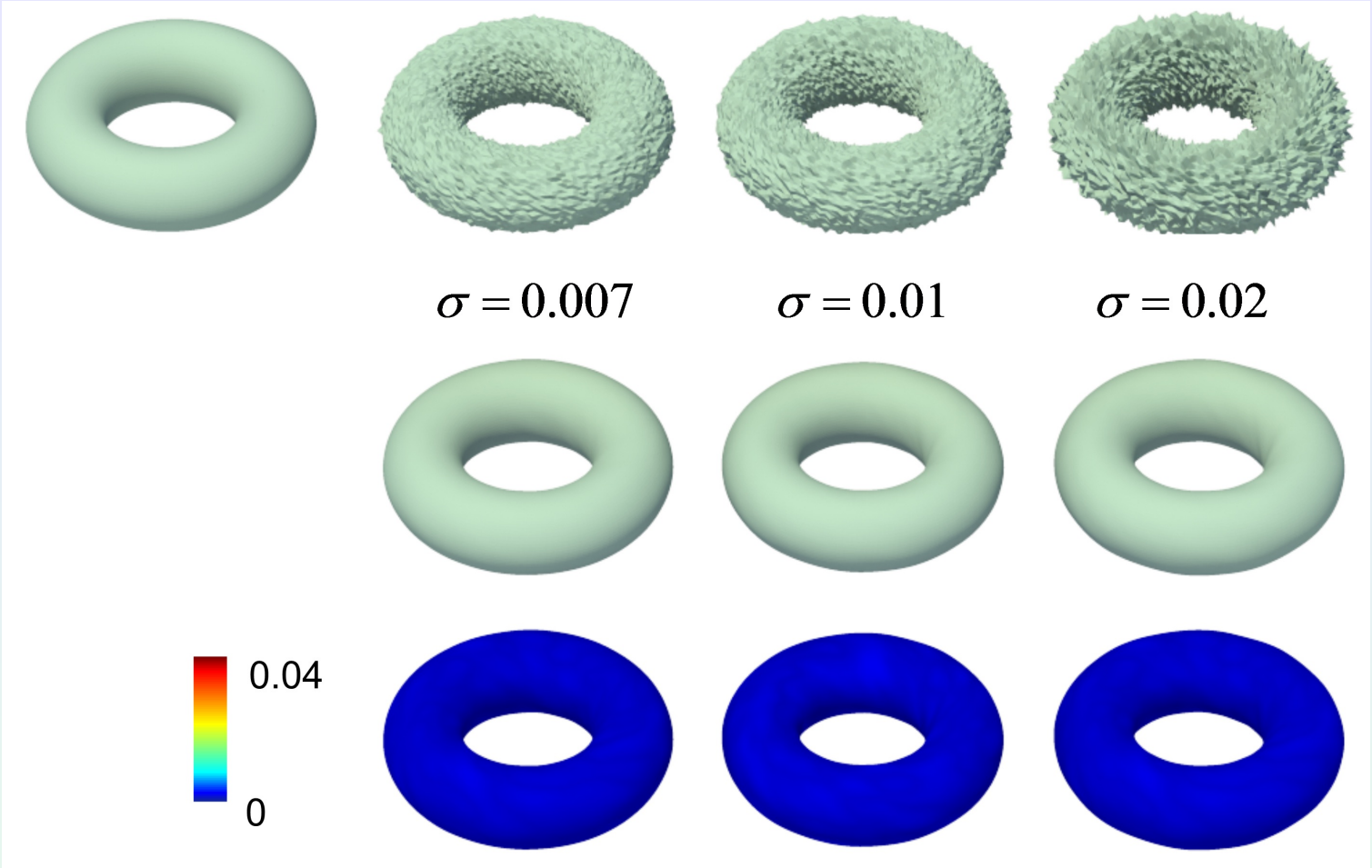


Figure: Our approach is robust to different amount of noises. Example: a C^2-smooth torus shape. The first row shows the ground truth model followed by the noisy models with different amount of noise (the variance is shown below each model), the second row shows the denoised models, and the third row visualizes the differences between the denoised models and the ground truth model.

Residual

By the denoising phase we now have the estimated base surface $\hat{S} = \{\hat{\mathrm{s}}_i\}_1^n$ from the input noisy mesh $P = \{\mathrm{p}_i\}_1^n$. The residual between $\hat{S}$ and P is defined as

$$b_i = (\mathrm{p}_i - \hat{\mathrm{s}}_i)^T \hat{\mathrm{n}}_i, \quad i = 1, \cdots, n, \tag{25}$$

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

Residual

As an inference of asymptotic properties, the residual signal b is essentially i.i.d. noise when the true underlying surface of input mesh P is at least C^2-continuous as a whole.

But in case the underlying surface containing sharp features, the residual b inevitably mixes the information from features and the noise, as illustrated in Figure 4 (left), and is decomposed as

$$\mathrm{b} = \mathrm{h} + \mathrm{z}, \tag{26}$$

where $\mathrm{h} = (h_1, \cdots, h_n)^T$ is the unknown signal of the features and $\mathrm{z} = (z_1, \cdots, z_n)^T$ is the measurement errors.

Residual

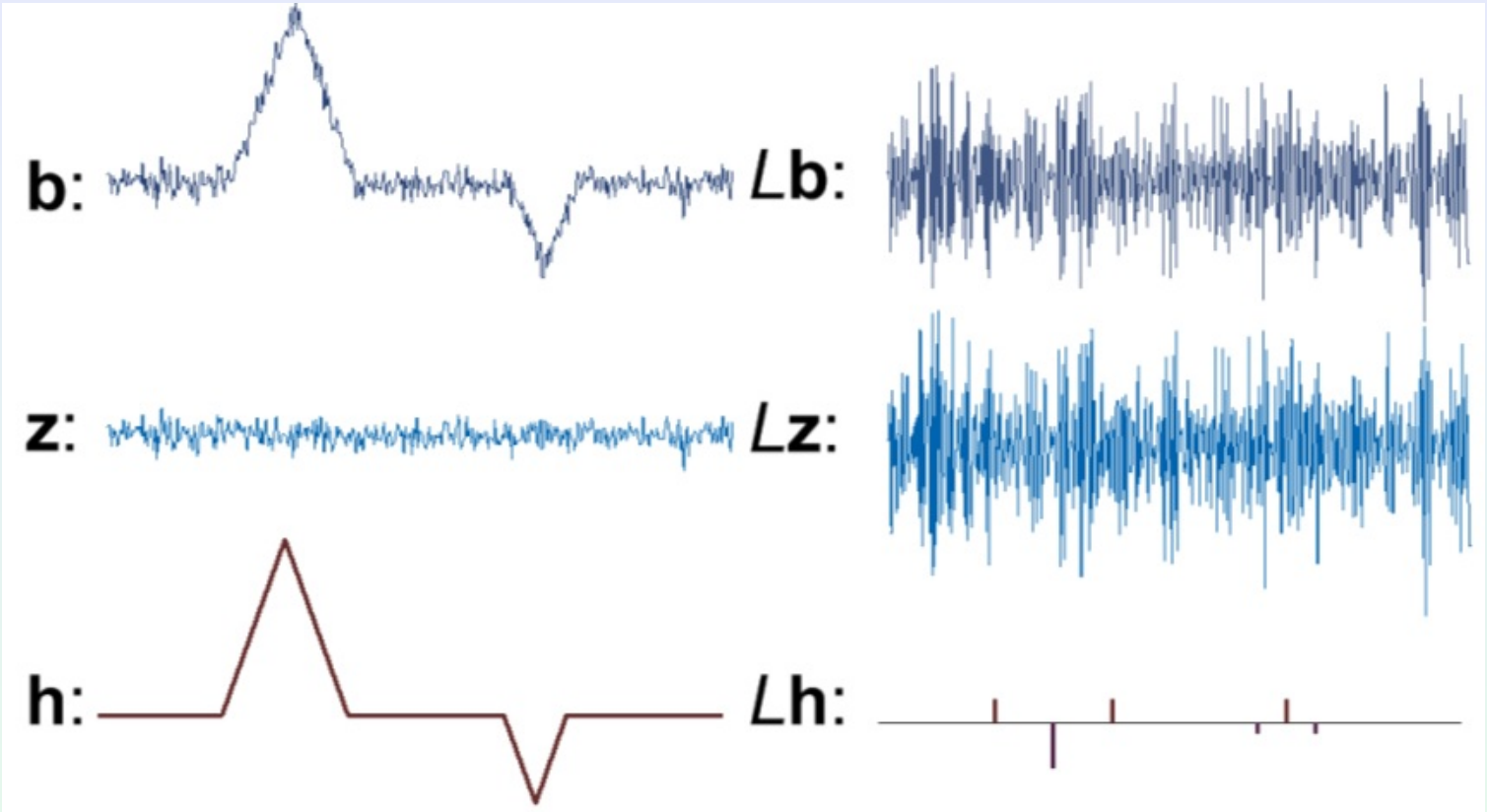


Figure: 1D illustration of signals and their Laplacians. Left: the residual b (upper) is a mixture of noises z (middle) and features h (lower). Right: the corresponding Laplacians of signals on the left.

Coherent dictionary for shape features

Considering the signal of shape feature h in our case, we discover that shape features can be represented as sparse in some coherent dictionary which is constructed by the pseudo-inverse matrix L^+ of the Laplacian matrix L of the shape.

It is not surprising to see that the Laplacian of feature Lh is indeed a sparse signal and has quickly decaying coefficients, as illustrated in Figure 4 (right).

Coherent dictionary for shape features

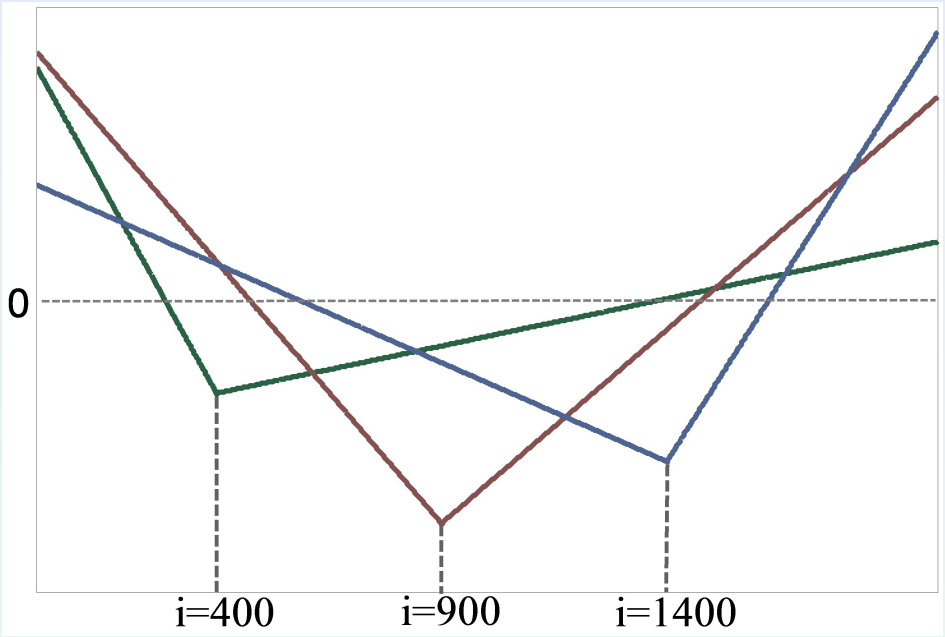
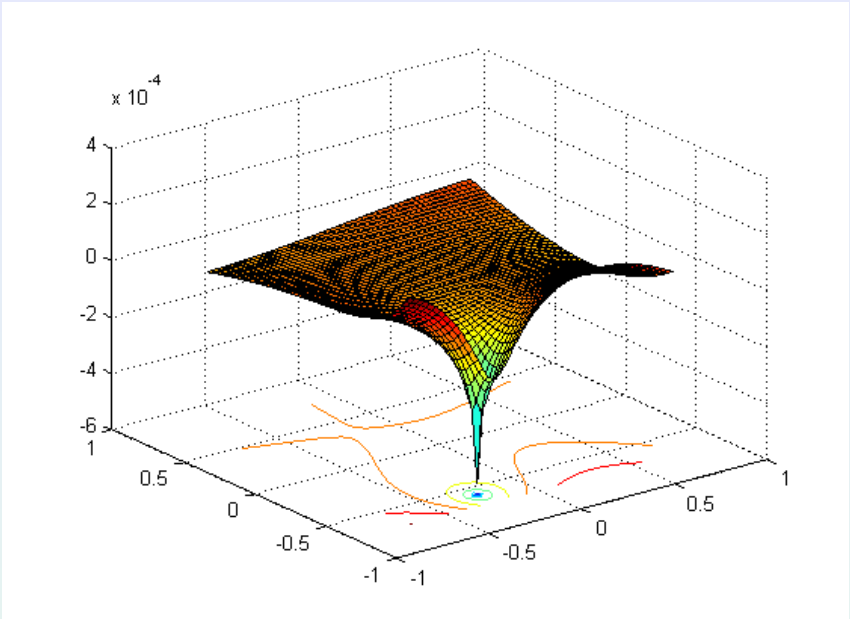
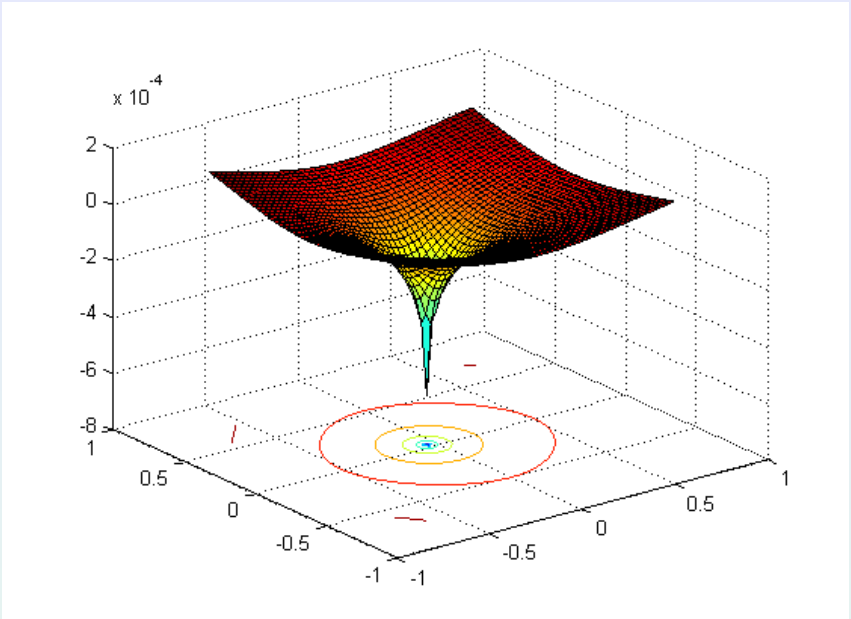
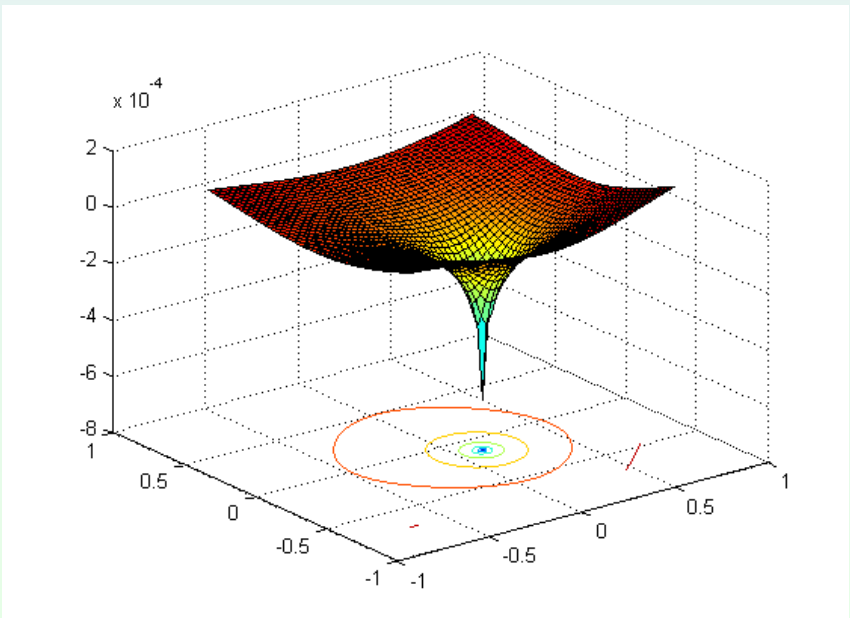


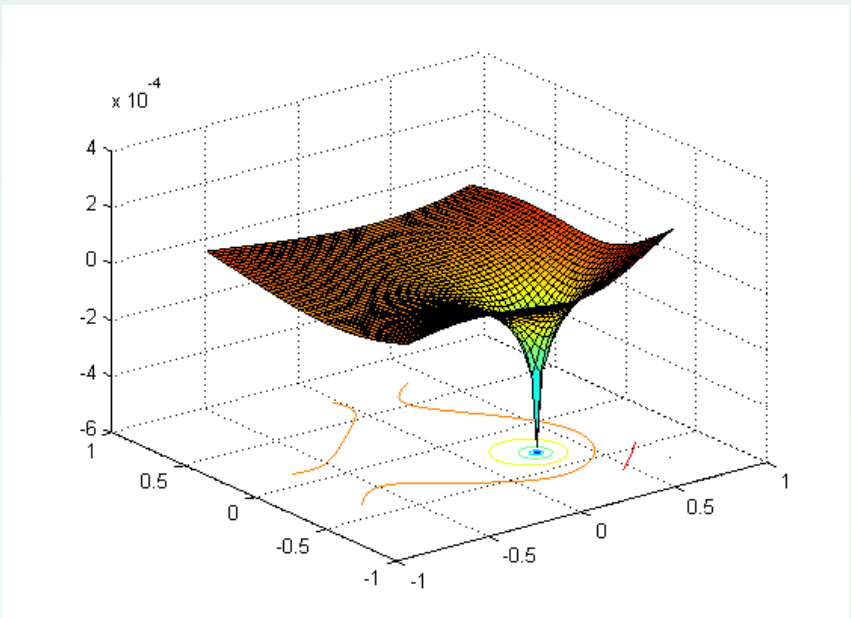
Figure: Three basis functions corresponding to the 400-th, 900-th, and 1400-th columns of L^+ respectively. This evidence suggests that L^+ is a coherent dictionary for representing the feature signal on the shape. We use 1D case as an illustration.

Coherent dictionary for shape features









ℓ_1-analysis on residual

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

In our setting, the sensing matrix is the identity matrix, and the coherent dictionary is $D = L^+$. Thus we have the following ℓ_1-analysis compressed sensing optimization:

$$\min_{\mathsf{h}} \|L\mathsf{h}\|_1 \quad \text{s.t.} \quad \|\mathsf{h} - \mathsf{b}\|_2 \leq \epsilon, \tag{27}$$

where ϵ^2 is a likely upper bound on the noise power $\|\mathsf{z}\|_2^2$.

ℓ_1-analysis on residual

The roles of the penalty and the constraint in (27) might also be reversed if we choose to constrain the sparsity and obtain the best fit for that sparsity. Here we prefer to solve an equivalent optimization:

$$\min_{\mathsf{h}} \|\mathsf{h}-\mathsf{b}\|_2^2 \quad \text{s.t.} \quad \|L\mathsf{h}\|_1 \leq \tau, \tag{28}$$

where τ is a tune parameter controlling the sparsity. We call τ the sparsity parameter.

Weighted ℓ_1-analysis

Consider the weighted ℓ_1-analysis on the residual

$$\begin{aligned} \min_{\mathsf{h}} \quad & \|\mathsf{h}-\mathsf{b}\|_2^2 \\ \text{s.t.} \quad & \|W(L\mathsf{h})\|_1 = \sum_{i=1}^{n} w_i |L_i \mathsf{h}| \le \tau \end{aligned} \tag{29}$$

where $W = \mathrm{diag}(w_1, \cdots, w_n)$ and $w_1, \cdots, w_n$ are positive weights. The weighted ℓ_1-analysis optimization (29) can be regarded as a relaxation of an ℓ_0-minimization problem.

Weighted ℓ_1-analysis

It is desired that the weights could be to counteract the influence of the signal magnitude on the ℓ_1-penalty function. Ideally, the weights are inversely proportional to the true signal magnitude, i.e.,

$$w_i = \begin{cases} \frac{1}{|L_i \mathrm{h}|} & L_i \mathrm{h} \neq 0, \\ \infty & L_i \mathrm{h} = 0. \end{cases} \tag{30}$$

The large entries in W force the Lapalcian of feature $L\mathrm{h}$ to concentrate on the indices where w_i is small. These constructed weights precisely correspond to the indices where $L\mathrm{h}$ is nonzero.

It is of course impossible to construct the precise weights (30) without knowing the feature signal h itself, but this suggests more generally that large weights could be used to discourage nonzero entries in the recovered $L\mathrm{h}$, while small weights could be used to encourage nonzero entries.

Weighted ℓ_1-analysis

Based on the geometric information of the estimated base surface $\hat{S}$, we design the weights as follows

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

where ρ is a small number ($\rho = 10^{-7}$ by default) that provides numerical stability and should be set slightly smaller than the expected nonzero magnitudes of Lh. With these well designed weights, we then perform a weighted ℓ_1-analysis (29) on the residual for recovering the feature signal.

Weighted ℓ_1-analysis

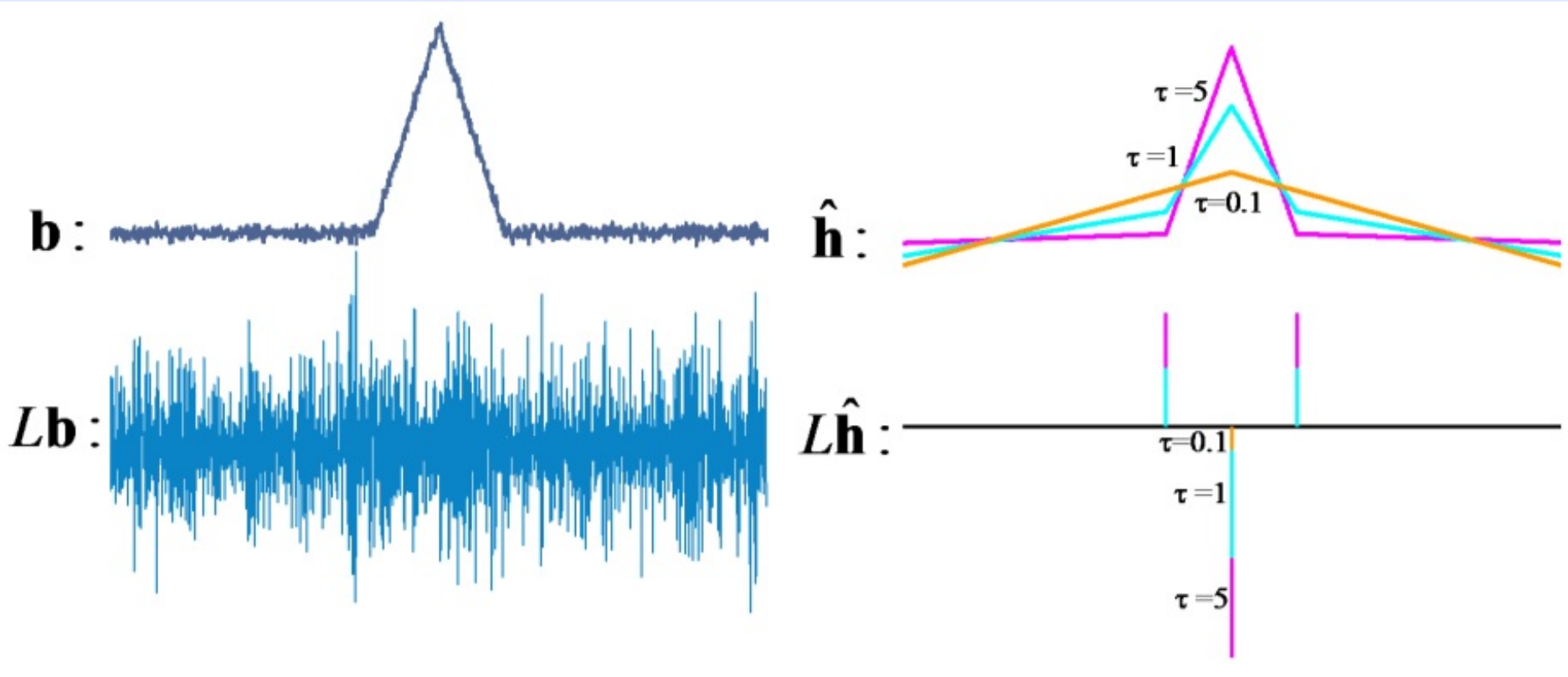


Figure: 1D illustration of results using different sparsity parameters τ in the ℓ_1-analysis optimization. Left: the residual signal b (upper) and its Laplacian (lower); Right: the recovered feature signals $\hat{h}$ corresponding to different values 0.1 (orange), 1 (blue), and 5 (magenta) of τ (upper)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

Iterative feature recovering

If we set large value of the sparsity parameter τ, the optimization (29) will recover features but may introduce some non-feature points in the result. Thus we prefer to choose a small value of τ. For some models with large portion of features, the solution to (29) returns only the most prominent (sharpest) features.

To recover the other features, our idea is to modify the rows corresponding to the identified features in the Laplacian matrix L and perform the weighted ℓ_1-analysis optimization (29) in an iterative manner. In each iteration, we identify a part of features. After marking the identified features, we find more features in the next iteration. This is reasonable as features become sparser in the latter iterations as the identified features in previous iterations have already marked and do not contribute to the feature recovering.

Feature types and classifications

Generally there are two types of sharp features, corners and creases, as shown in Figure 7, on 3D shapes. A corner point is the one at which the tangent of any passing curve on the surface is discontinuous. A crease curve introduces the discontinuities of first derivatives across it, but preserves C^2-continuity along it. Corners can also be the intersections of several creases.

After identifying locations of the features by the weighnted ℓ_1-analysis (29) on the residual, we adopt a simple scheme to classify their types. If a feature is isolated from others, it is identified as a corner. If a feature point has a few feature points in its neighbor, we compute a dominant direction by PCA and classify it as a crease along this direction, see Figure 7.

Feature types and classifications

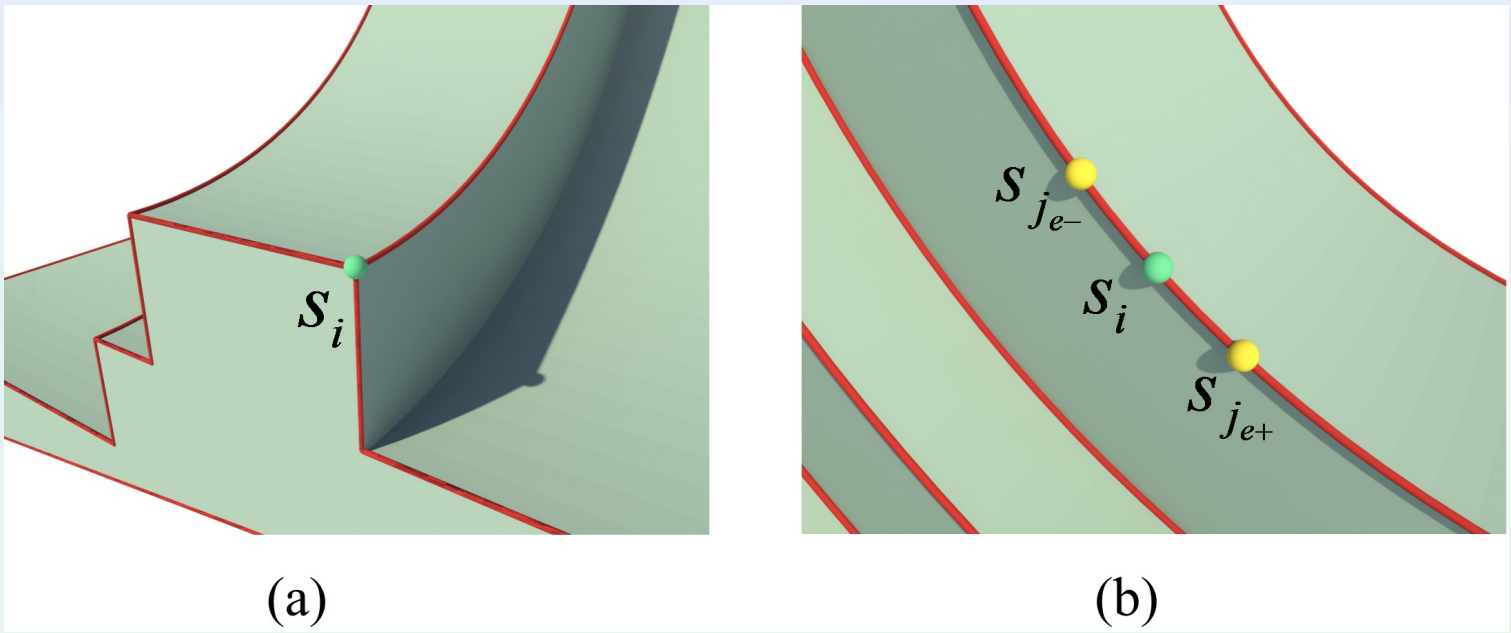


Figure: Two types of feature points: (a) corner; (b) crease.

Feature aware modification of Laplacian matrix

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

$$\mathcal{L}(s_i) = L_i S = \sum_{j \in \mathcal{N}(i) \cap \mathcal{E}} (s_j - s_i) = s_{j_{e-}} + s_{j_{e+}} - 2s_i,$$

where $s_{j_{e-}}$ and $s_{j_{e+}}$ are the adjacent neighbors of s_i along the crease $\mathcal{E}$, as illustrated in Figure 7(b).

Termination condition of iterations

We adopt a statistical method of nonparametric test for checking whether the residual b contains more features. Specifically, residuals were randomly divided into two sets b_1 and b_2. We use a two-sample Kolmogorov-Smirnov test to compare the distributions of the values in the two sets b_1 and b_2. We state that

- The null hypothesis H_0: b_1 and b_2 are from the same continuous distribution.
- The alternative hypothesis H_1: they are from different continuous distributions.

This hypothesis does not specify what that common distribution is (e.g. normal or not normal). The result is 1 if the test rejects the null hypothesis at the α significance level; 0 otherwise. We use the significance level $\alpha = 0.05$ in our implementation.

Pseudo-code

The pseudo-code of the iterative feature recovering via ℓ_1-analysis can be found in the following algorithm.

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

Output: denoised mesh with features

1. Call $\hat{S} = \text{DLRS}(P, L)$.
2. Calculate the residual b according to (25).
3. If the result of the two-sample Kolmogorov-Smirnov test on residual b is 1, go to Step 4; Otherwise, exit and output the current $\hat{S}$.
4. Recover the features $\hat{\text{h}}$ by the weighted ℓ_1-analysis (29), and get the reliable locations of $\hat{\text{h}}$ indicated by $L\hat{\text{h}}$.
5. Classify the features $\hat{\text{h}}$ and modify accordingly the Lapalcian matrix L based on their feature types.
6. Go to Step 1.

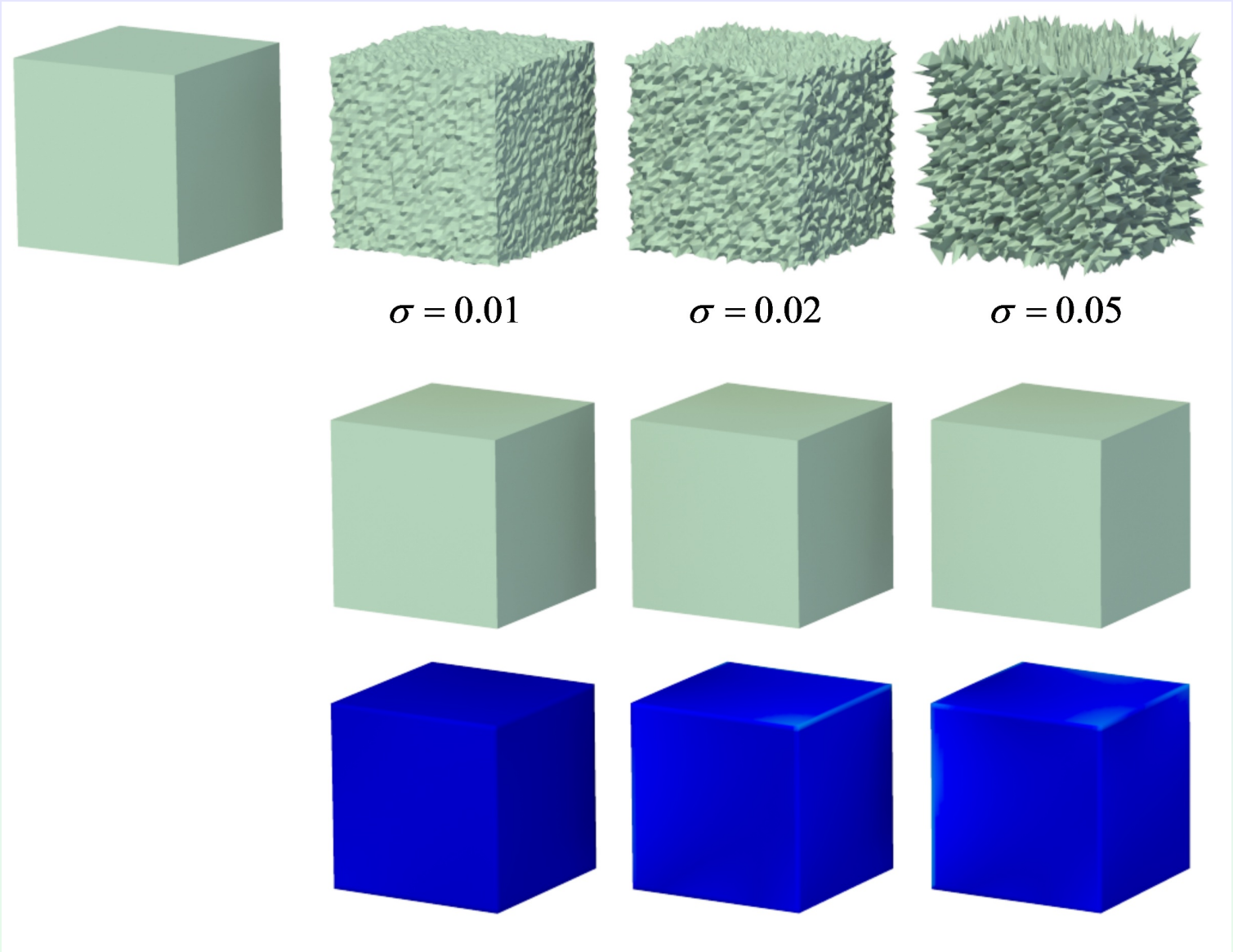


Figure: Our approach is robust to different amount of noises. Example: a cube model with sharp features.

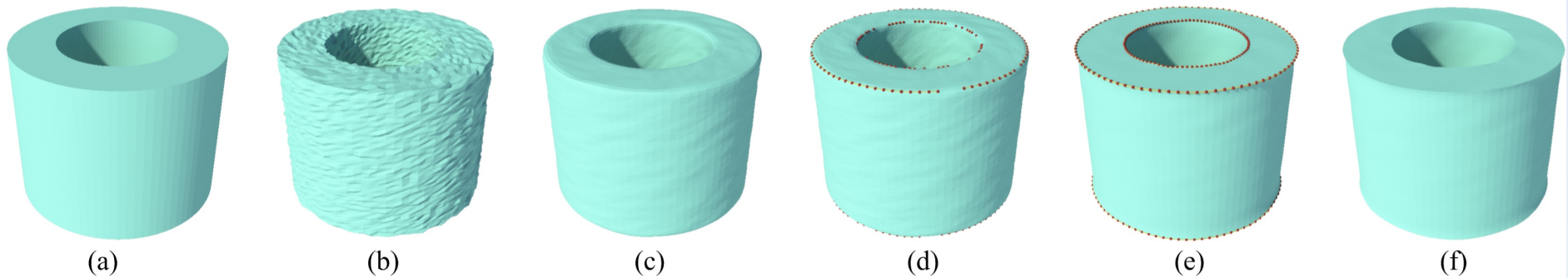


Figure: Our algorithm progressively recovers sharp features. (a) the ground truth model; (b) the noisy model; (c) the denoised model using our approach; (d)-(e) progressively recovered features shown in red dots with two iterations; (f) the final result.

Figure: Main phases of our approach. (a) The input noisy mesh model; (b) phase I: denoising the model using the DLRS scheme; (c) phase II: recovering sharp features progressively based on the ℓ_1-analysis optimization; (d) the final denoised result with recovered features.

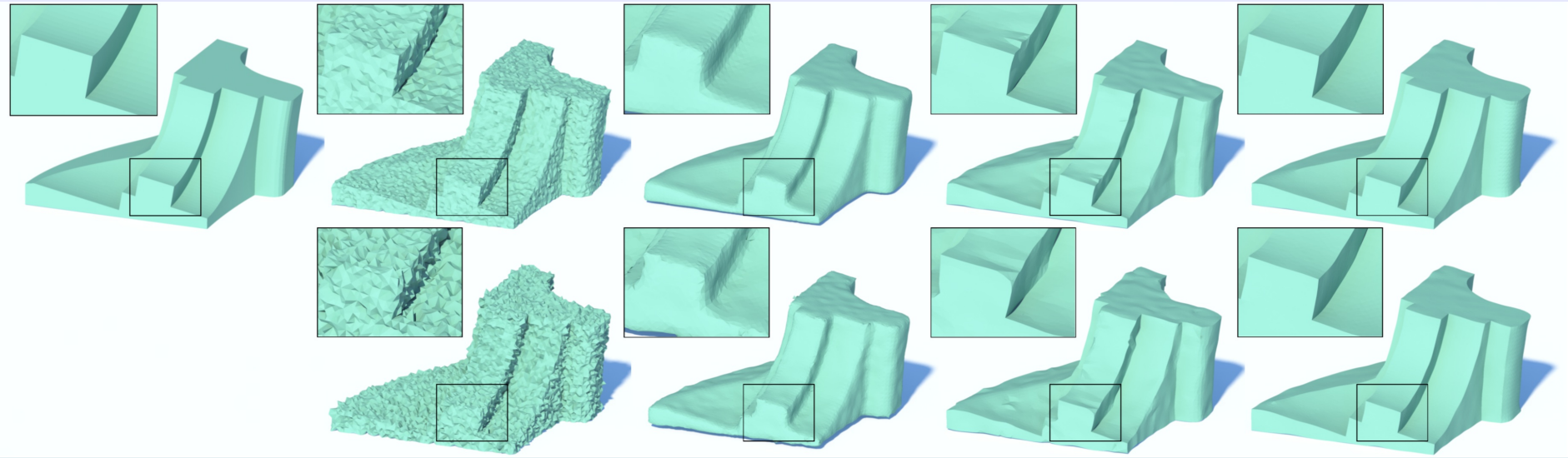


Figure: Comparisons with previous approaches. From left to right: the ground truth Fandisk model, the models corrupted by different levels of artificial noise ($\sigma = 0.005$ in the upper row and $\sigma = 0.01$ in the lower row), denoised results with vertex based bilateral mesh filtering [Fleishman et al. 2003], normal based bilateral mesh filtering [Zheng et al. 2010], and our approach. The small regions with the frames are magnified to clearly show the differences. All meshes are flat-shaded to show faceting.

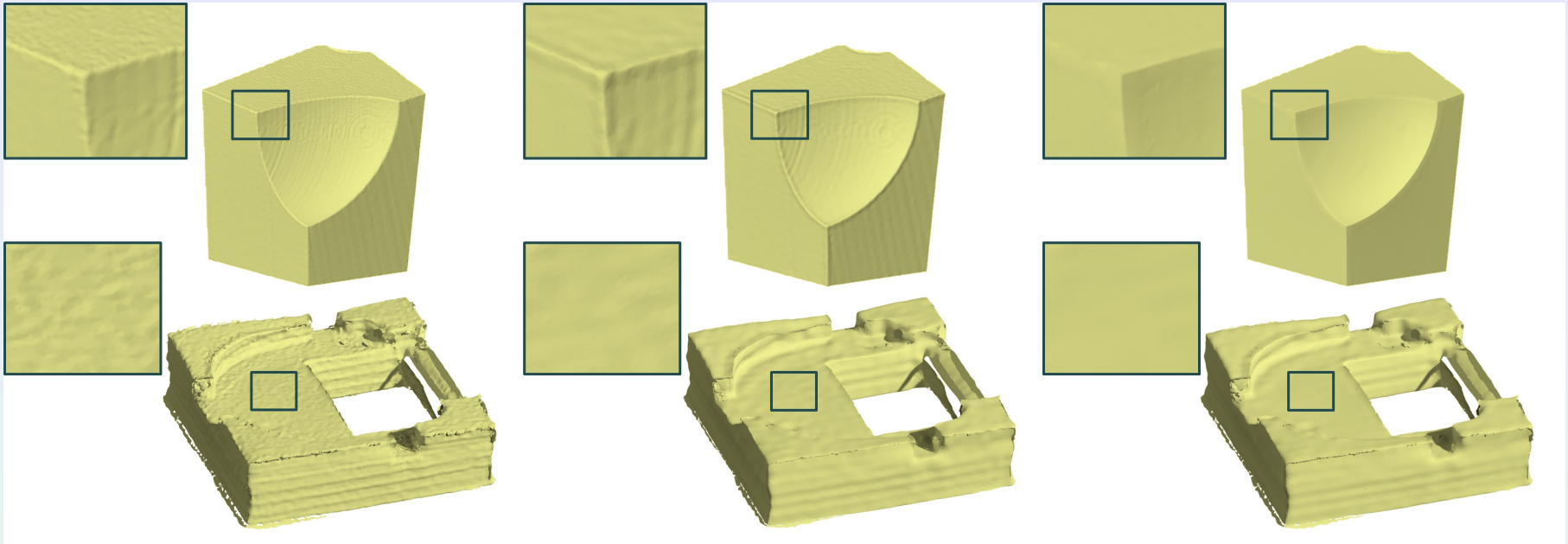


Figure: Applying our approach to scanning data of real objects. From left to right: the input scanning raw data; the denoised results using the non-iterative smoothing method [Jones et al. 2003]; the denoised results using our approach. The close-up views show the details in framed regions.

Figure: Applying our approach to real point clouds. The second row is the

Summary

A two-phase approach for decoupling features and noises on discrete surfaces:

- The first phase generates a base mesh which is obtained by denoising the input data using the DLRS estimator.
- The second phase recovers sharp features from the residual between the base mesh and the input mesh based on an ℓ_1-analysis compressed sensing optimization.

Summary

If the true underlying surface of input data is C^2 smooth, Step 1 obtains a smoothed data which is theoretically guaranteed to asymptotically converge to the true underlying surface (Theorem 2).

When the underlying surface is not a C^2-smooth surface, e.g., a piecewise C^2 surface with C^0 features, Step 2 is performed to extract features from residual between original and smoothed data. By removing the unnecessary smoothness penalty at locations of features, Step 1 performs equivalently as denoising each C^2 patch individually and thus eventually obtains the optimal approximation to the underlying surface with theoretical guaranty.

Contributions

- **Asymptotically optimal surface denoising:** Our denoising approach is fully automatic without tuning any parameter. The smoothness parameter is automatically computed. The denoised surface is guaranteed to asymptotically converge to the true underlying surface with probability one if the underlying surface is C^2-smooth.
- **Faithful feature recovering by ℓ_1-analysis:** We successfully apply the ℓ_1-analysis compressed sensing technique to identify and recover sharp features from the residual between the estimated base surface and the input surface. This is based on our discovery that the pseudo-inverse matrix of the Laplacian matrix of the mesh is a coherent dictionary for sparsely representing the sharp feature signal on the shape.

Outline I

Outline II

Motivation

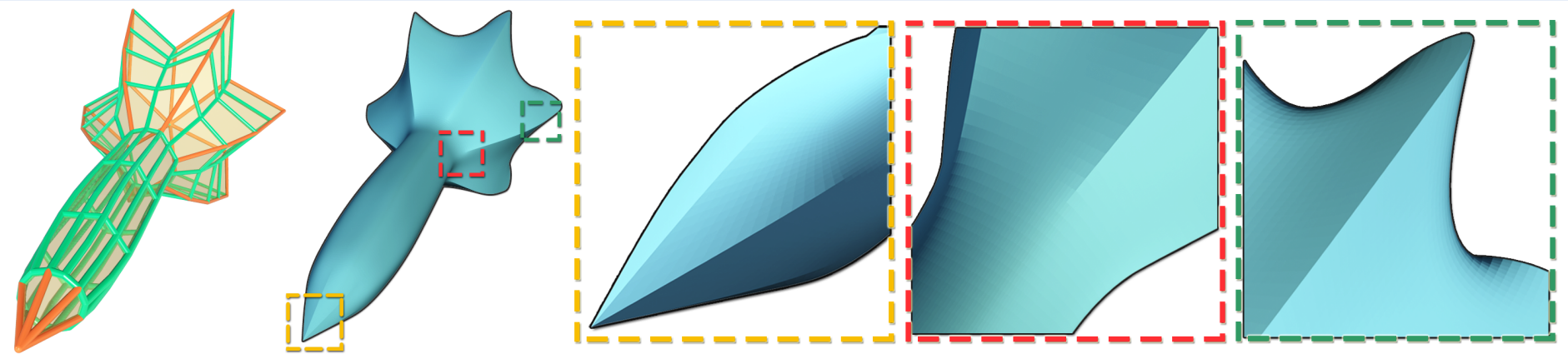
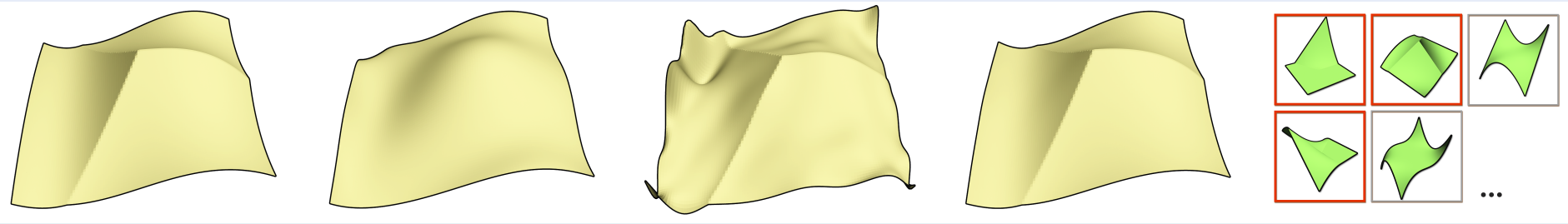


Figure: To construct a manifold that approximates various features like crease, dart, and cusp well.

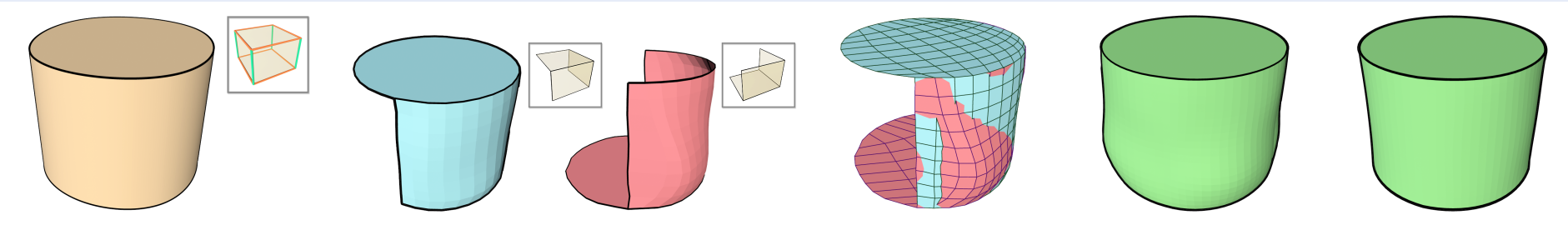
Motivation



(a) (b) (c) (d) (e)

Figure: Given a set of sampling points from a C^0 surface (a), the obtained fitting surface via polynomials (b), redundant atom functions (c), and sparse representation (d). (e) The first five atom functions (C^0 shape functions in red boxes and polynomials in gray boxes) that are selected in the sparse representation.

Motivation



(a) (b) (c) (d) (e)

Figure: Given subdivision surface with control mesh (a), the results of sparse representations on individual charts (b), the surface's appearance in overlapping regions (c), the result of a manifold (d) by simple blending, and the obtained surface (e) by consistent sparse representation.

Manifold

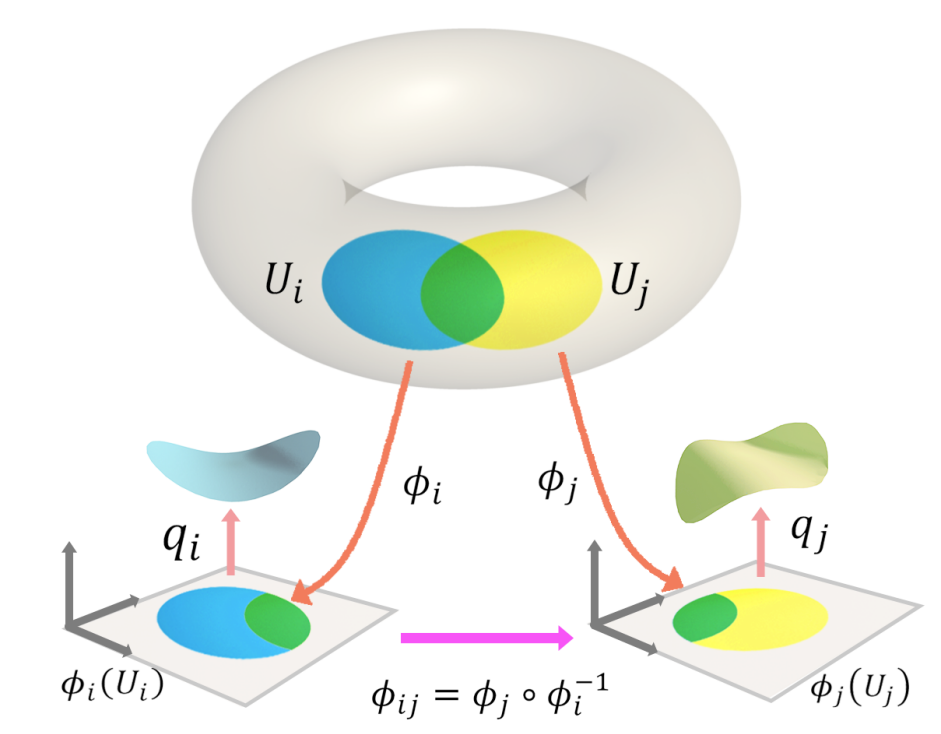


Figure: The definition of a manifold.

The C^0 atom functions

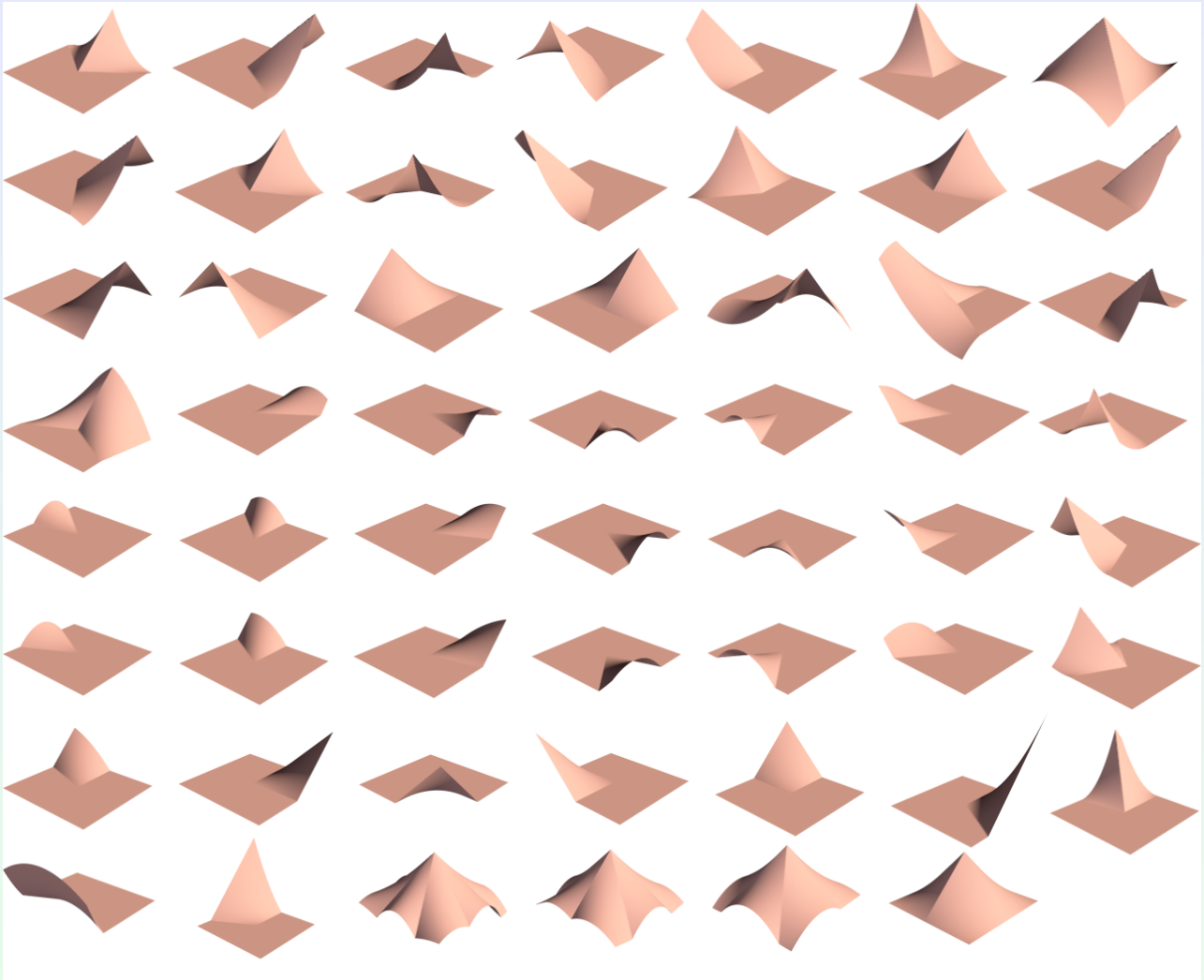


Figure: 55 shape functions with valence of 3 to 7 (which is suitable for most domain manifold).

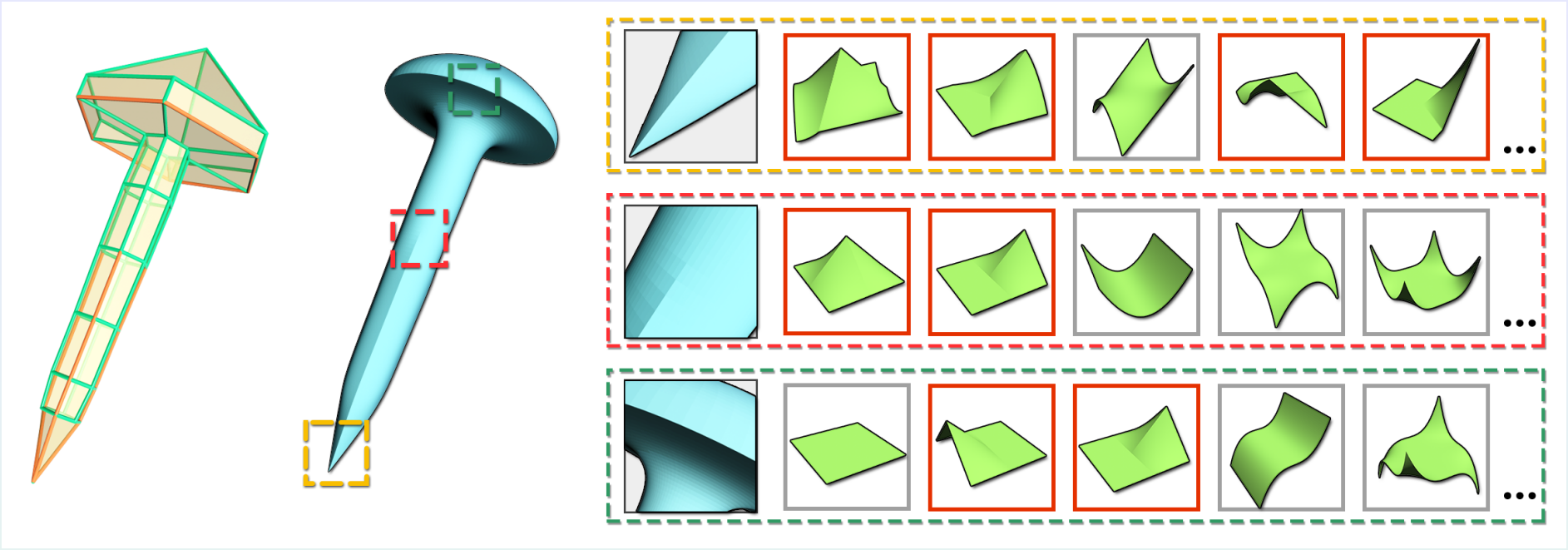
Compatible sparse representations

$$E_{\text{fit}}(C) = \sum_{k=1}^{K} \left(\sum_{j:\, \mathrm{p}_k \in U_j} \left(w_j(u_k^j, v_k^j) \sum_{l=1}^{L} c_{lj} a_l(u_k^j, v_k^j) \right) - h_k \right)^2 \tag{31}$$

where $\mathrm{c}_j = (c_{1j}, \cdots, c_{Lj})^T$ is the coefficient vector of the local function on the chart (U_j, ϕ_j) and $C = (\mathrm{c}_1, \cdots, \mathrm{c}_n)$ is the matrix of decision variables.

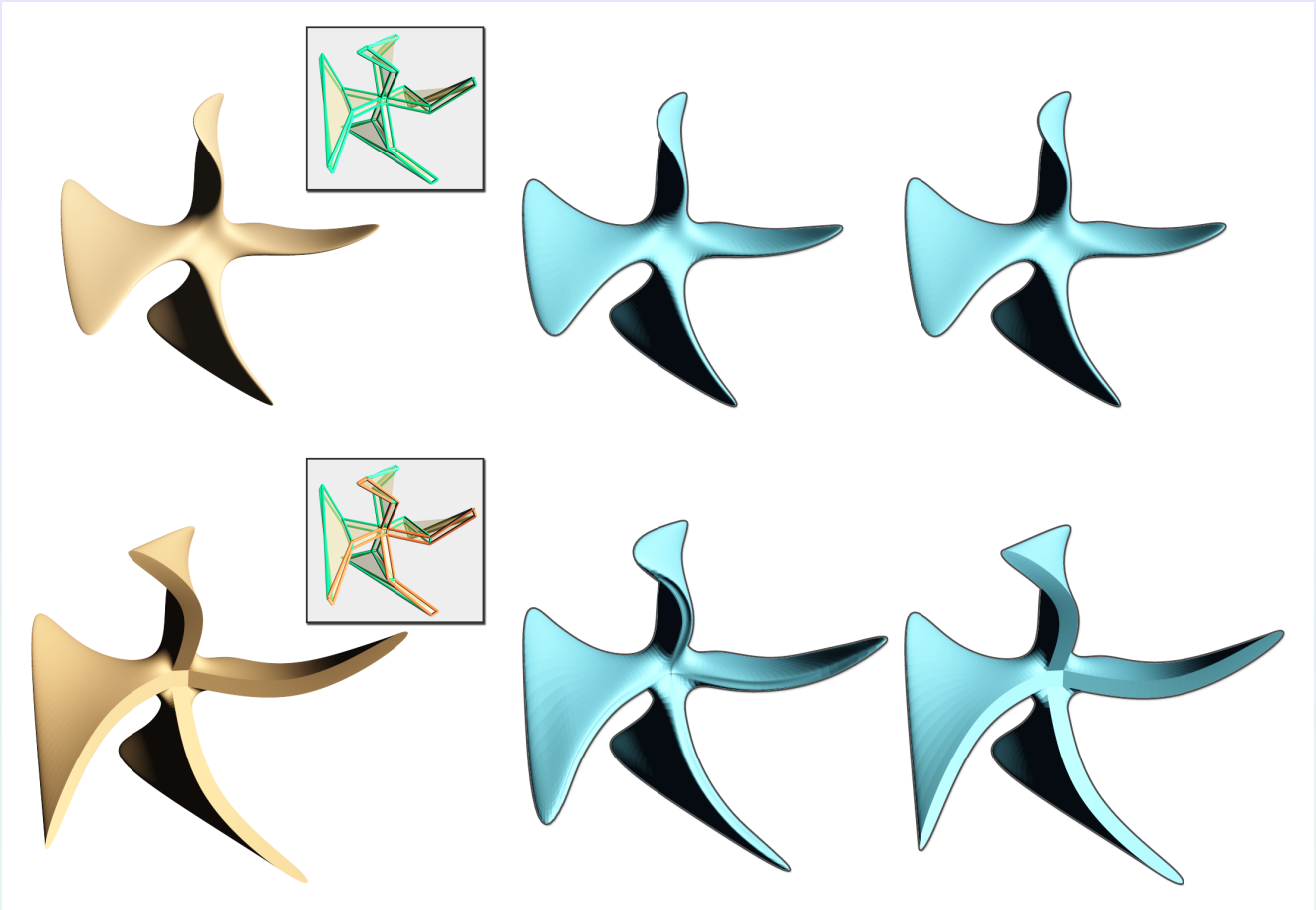
To obtain the sparse coefficients of the local function on each chart, we formulate the sparse representation of global geometry function as

$$\begin{aligned} \min_{C} \quad & E_{\text{fit}}(C) \\ \text{s.t.} \quad & \|\mathrm{c}_j\|_0 \leq \delta,\ j = 1, 2, \cdots, n. \end{aligned} \tag{32}$$



(a) (b) (c)

Figure: Constructing manifold to approximate subdivision surface. (a) The input control mesh with enforced C^0 features shown in red; (b) The manifold generated by our approach; (c) The close-up views of three parts of sharp features including crease, dart, and cusp. The first five atom functions that are adaptively selected for representing the local features are shown.



(a) (b) (c)

Figure: (a) The limit subdivision surfaces of input control meshes (in the upper right) where edges in red are enforced with C^0 constraints; (b) manifolds produced by the method of [Ying and Zorin 2004]; (c) manifolds generated by our approach. Upper row: the results from smooth subdivision surface; Lower row: the results from subdivision surface with sharp features.

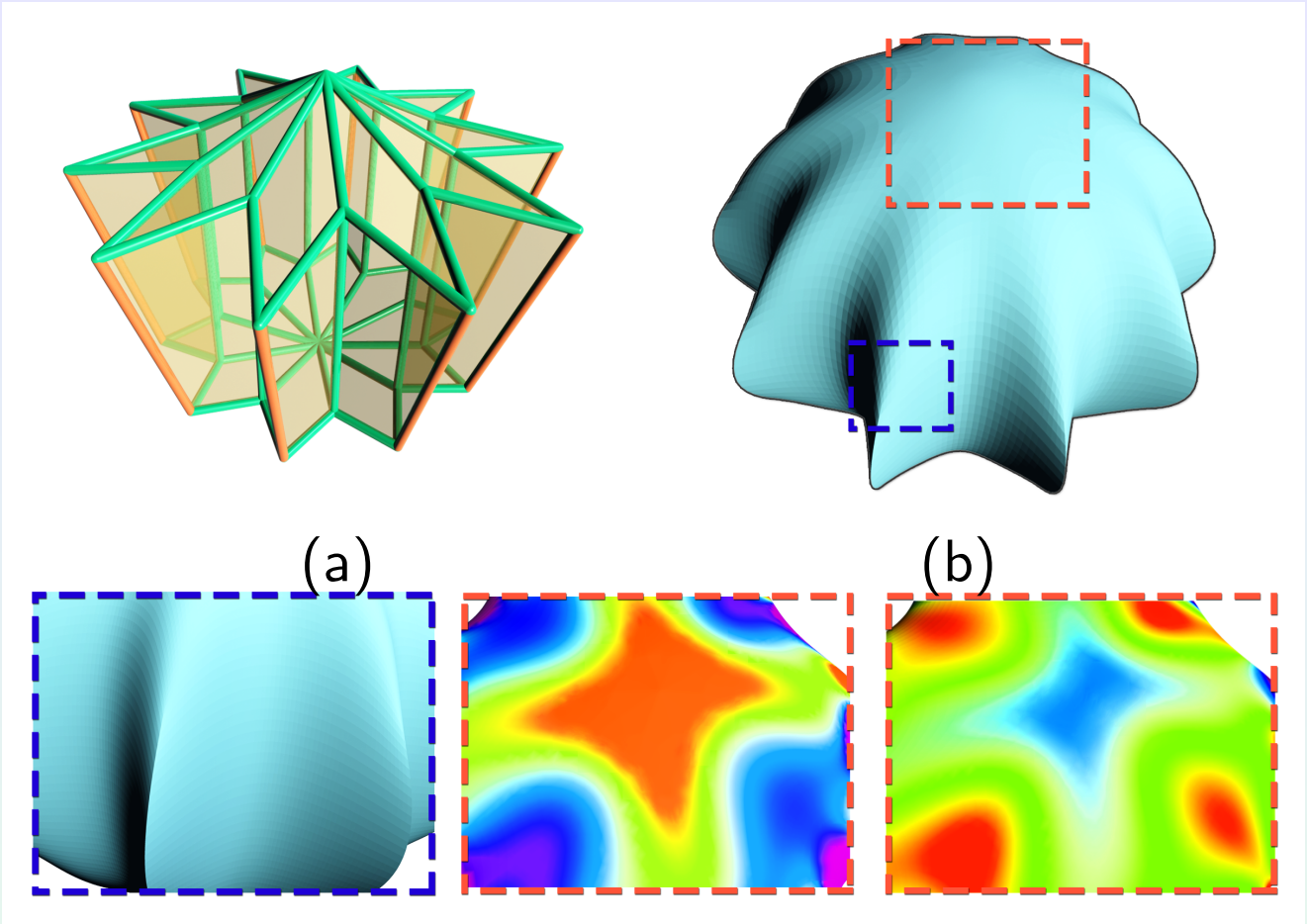
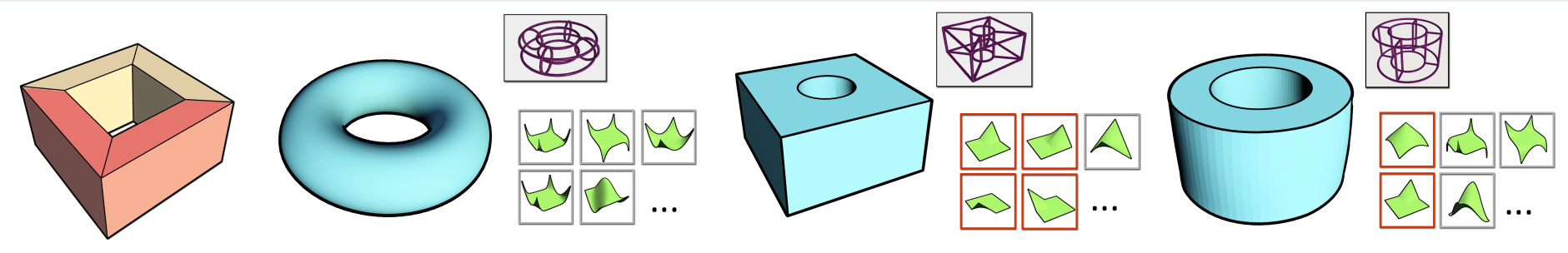


Figure: (a) A control mesh with C^0 constraints shown in red. The top vertex of (a) has a valence of 10. We discard the C^0 atom functions from the dictionary at the chart around this vertex. The manifold surface constructed by our approach is shown in (b). The surface adapts to the sharp features in blue region and is C^∞ in the red region. (c) the close-up view of blue region; (d,e) Gaussian curvature and mean curvature of the red region.

Application to curve network



(a) (b) (c) (d)

Figure: Generating manifold from different geometries defined on the same domain manifold. (a) The domain mesh; (b-d) three manifolds created from given curve networks (in the upper right) and the first five chosen atom functions (in the lower right). C^0 shape functions are shown with red frames.

Application to curve network

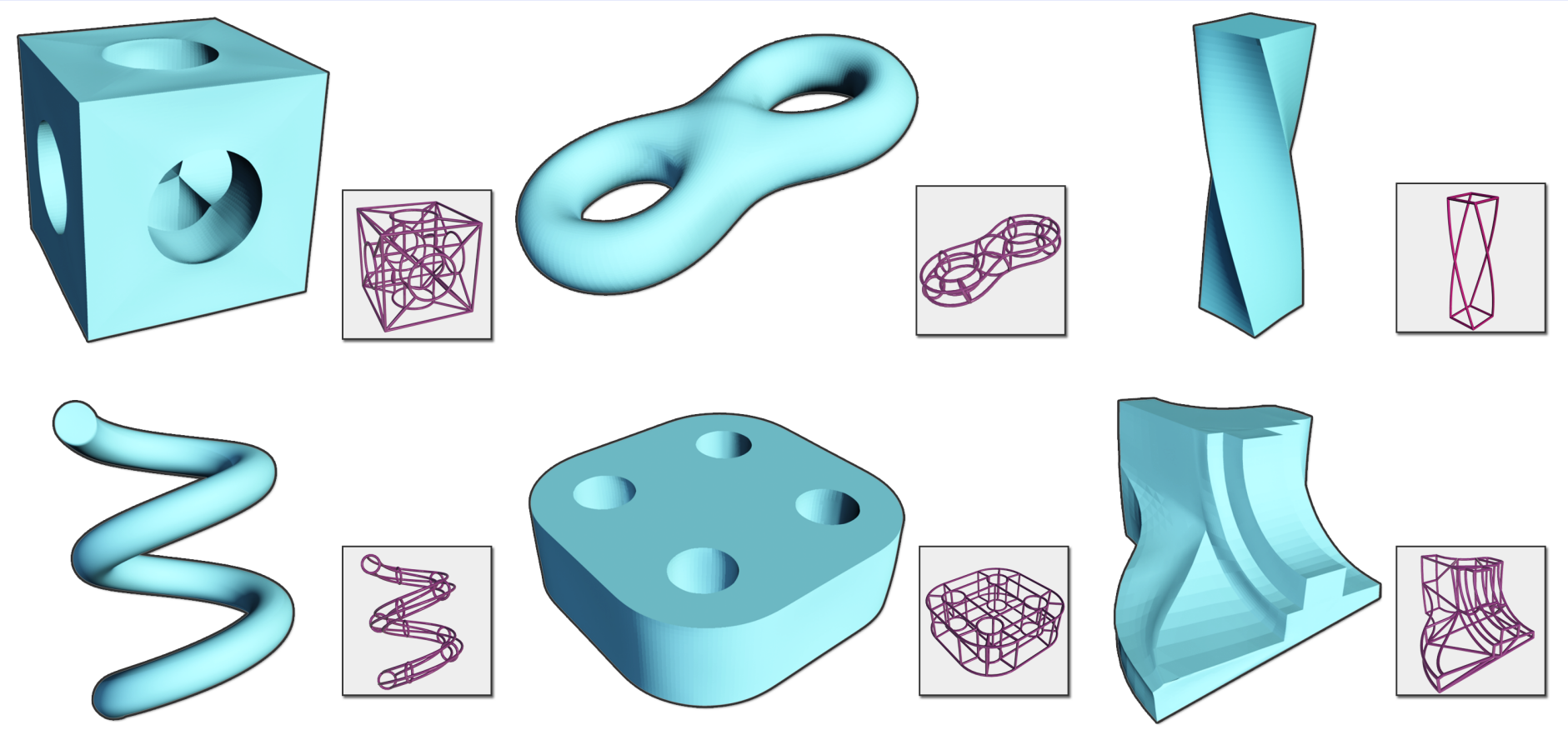


Figure: Examples of manifold surfaces generated from 3D curve networks (in the lower right).

Application to curve network

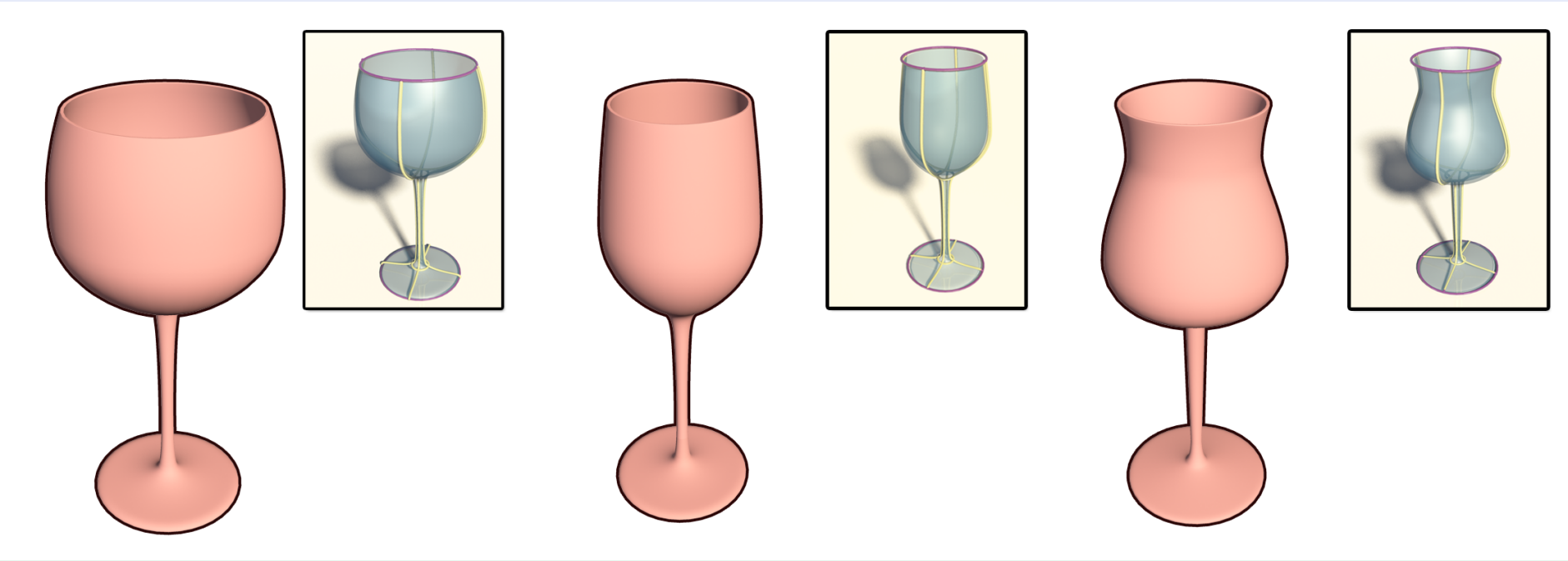


Figure: Three different wine glasses obtained by our method with specifying smoothness across some edges (yellow ones).

Outline I

Outline II

Motivation & Purpose

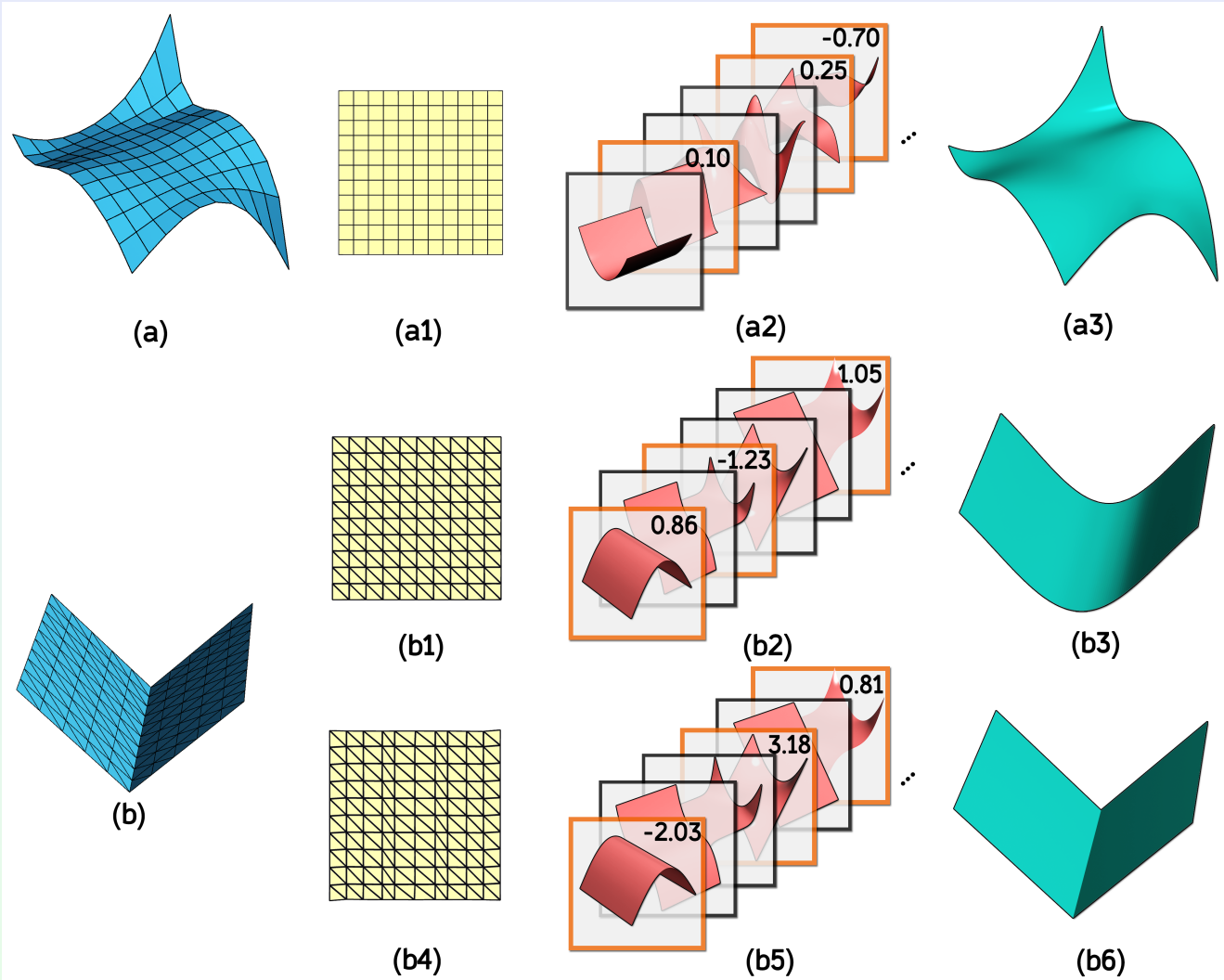
- Fitting geometry data based on a sparse optimization can efficiently overcome the overfitting artifacts.
- The parameterization of input data is simultaneously optimized to respect sharp features of target geometry.

Composite function

$$f(\mathrm{u}) = h \circ \Phi(\mathrm{u}), \quad \mathrm{u} \in M \tag{33}$$

$$\begin{cases} \Phi & : \quad M \to M, \\ h & : \quad M \to \mathbb{R}. \end{cases}$$

Composite function



Fitting problem

Denote $\mathrm{b} = (b_1, \cdots, b_m)^T$ as a set of scalar observation from some underlying Euclidean space $\mathbb{R}^k$.

Suppose $D = \{d_j(\mathrm{u})\}_{j=1}^n$ is a family of n basis functions defined on $\mathbb{R}^k$.

The fitting problem is to find the coefficients $\mathrm{c} = (c_1, \cdots, c_m)^T$ by minimizing the error function

$$F(\mathrm{c}, U) = \sum_{i=1}^{m} \left[b_i - \sum_{j=1}^{n} c_j d_j(\mathrm{u}_i) \right]^2,$$

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

Sparse fitting

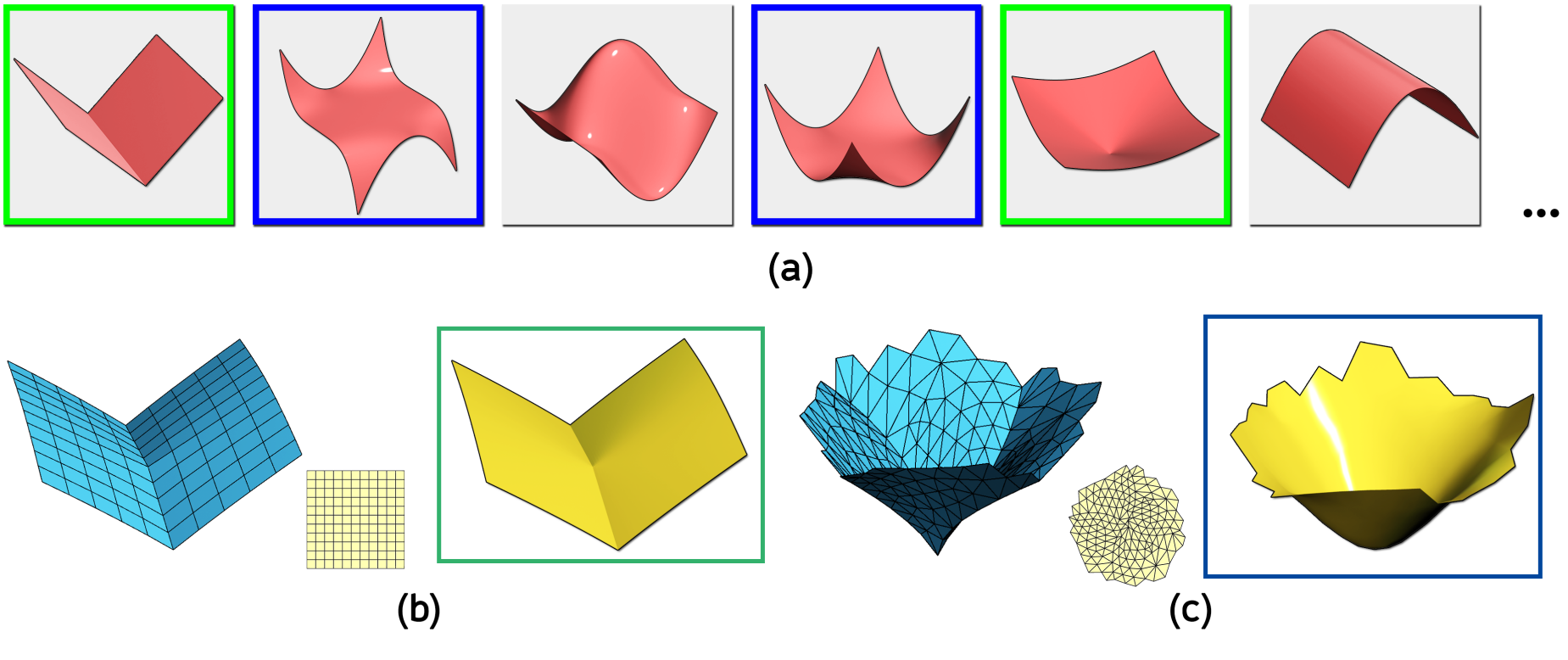
We usually adopt a set of redundant basis functions $\{d_j(\mathrm{u})\}_{j=1}^{n}$ and fit the input data by a sparse combination of these functions, i.e.,

$$\begin{aligned} \min_{\mathrm{c}} \quad & F(\mathrm{c}, U) \\ \text{s.t.} \quad & \|\mathrm{c}\|_0 \leq s \end{aligned} \tag{34}$$

where s is the sparsity parameter.

With sparse fitting, we can use redundant basis functions and then rely on the sparsity regularization to choose the appropriate basis functions to represent the input geometry data.

Sparse fitting



Sparse fitting with parameterization optimization

We optimize the coefficients c and the parameterization U of the input data simultaneously as follows:

$$\begin{aligned} \min_{\mathsf{c},U} \quad & F(\mathsf{c}, U) \\ \text{s.t.} \quad & \|\mathsf{c}\|_0 \leq s. \end{aligned} \tag{35}$$

Parametric transformation

The parametric transformation is defined by a mapping $\Phi : \mathbb{R}^2 \to \mathbb{R}^2$.

Assume that $\{u_i\}_{i=1}^m$ is a set of vertices in domain M and $\{v_i\}_{i=1}^m$ is a set of vertices in domain N with $\Phi(u_i) = v_i$. Let T_M be the triangulation of $\{u_i\}_{i=1}^m$ and T_N the triangulation of $\{v_i\}_{i=1}^m$.

Thus we express the mapping Φ as a piecewise linear transformation of two-dimensional triangulated structure, i.e.,

$$\Phi(u) = w_1\Phi(u_{i_1}) + w_2\Phi(u_{i_2}) + w_3\Phi(u_{i_3})$$

for any $u = w_1u_{i_1} + w_2u_{i_2} + w_3u_{i_3}$ lying in the triangle $\triangle u_{i_1}u_{i_2}u_{i_3}$ with its barycenter coordinates (w_1, w_2, w_3).

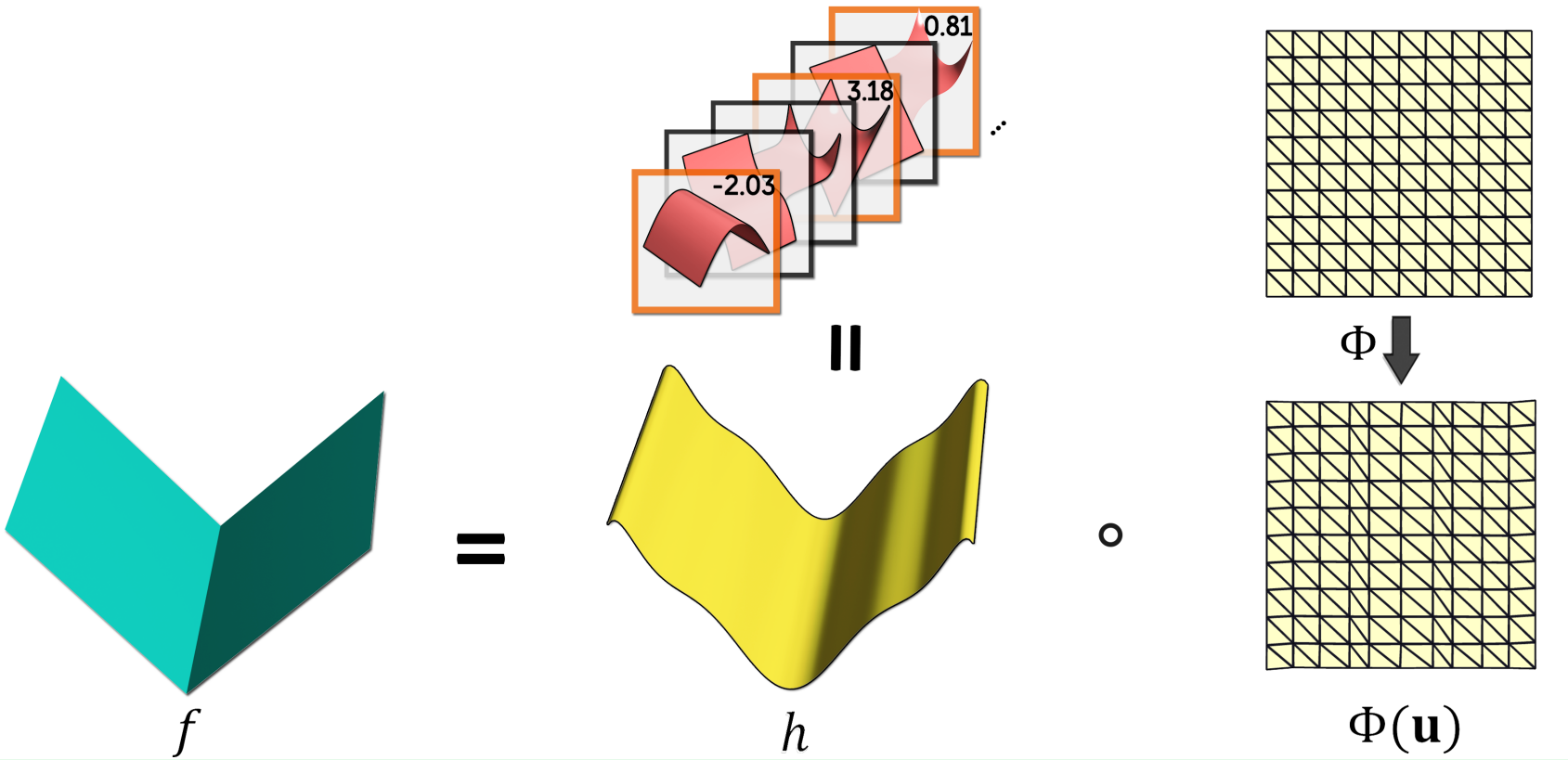
Composite representation

A function f defined on domain $M \subset \mathbb{R}^2$ composites of some function $h : \mathbb{R}^2 \to \mathbb{R}$ and a mapping $\Phi : \mathbb{R}^2 \to \mathbb{R}^2$.

Here h is a linear combination of a certain set of basis functions with $h(\mathrm{u}) = \sum_{j=1}^{n} c_j d_j(\mathrm{u})$. The composite function is written as:

$$f(\mathrm{u}) = h \circ \Phi(\mathrm{u}) = \sum_{j=1}^{n} c_j d_j(\Phi(\mathrm{u})). \tag{36}$$

Composite representation



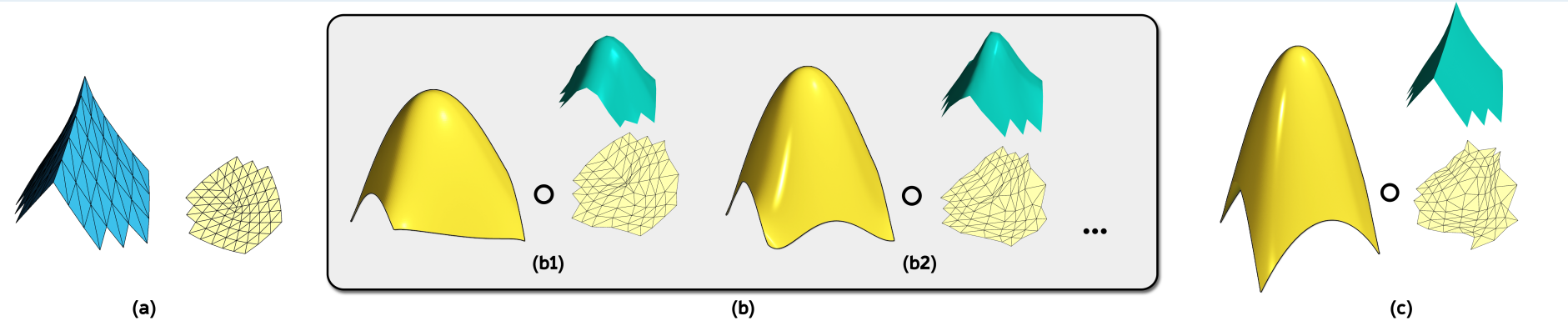
Optimization framework

We formulate the sparse representation of geometry data as the following optimization:

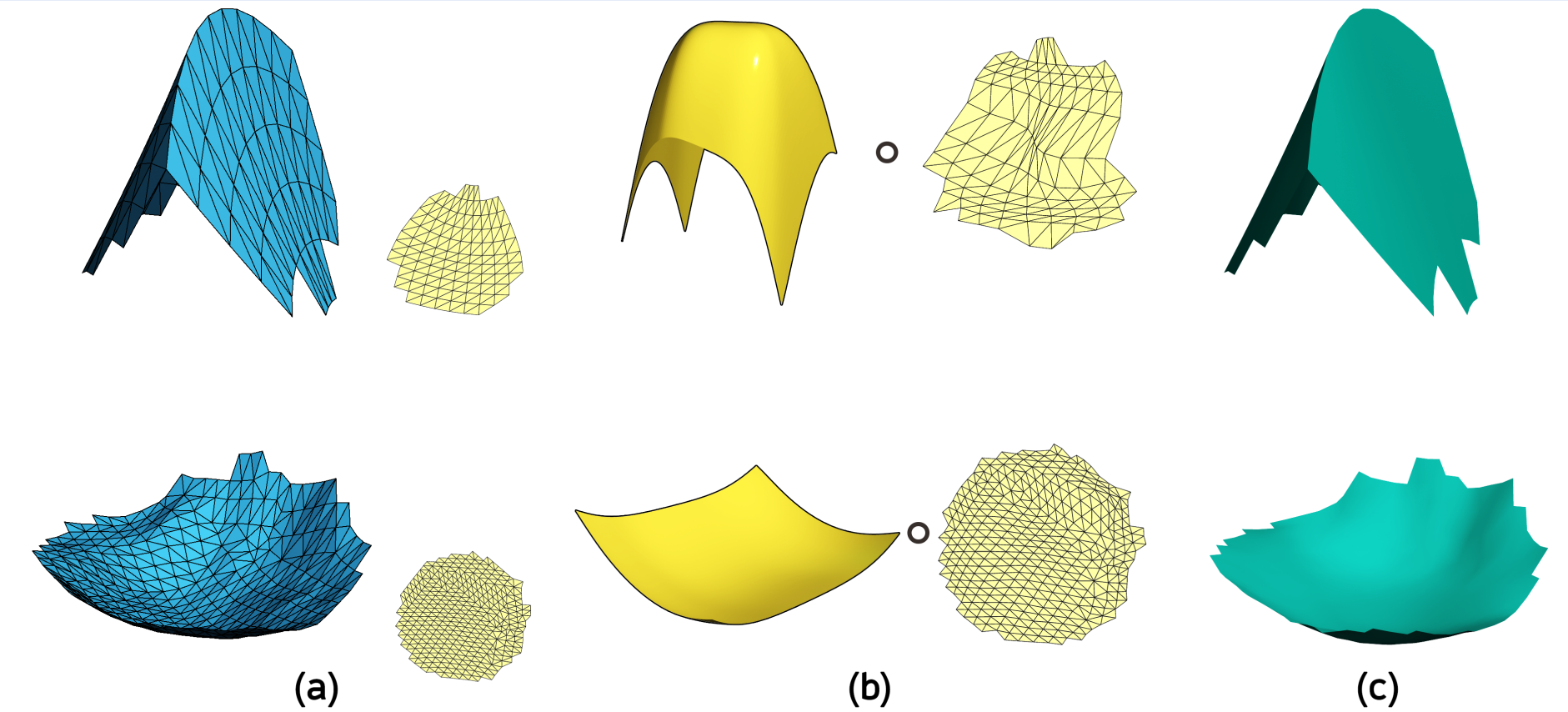
$$\begin{aligned} \min_{\mathrm{c},U} \quad & F(\mathrm{c},U)+\alpha E(\mathrm{c},U)+Q(U) \\ \text{s.t.} \quad & \|\mathrm{c}\|_0 \le s \end{aligned} \tag{37}$$

where $F(\mathrm{c},U)$ is the data term, $E(\mathrm{c},U)$ is a geometric smoothness term decided by both coordinate c and U, and $Q(U)$ is the regularization term of the triangle structure of $\{\mathrm{u}_i\}$.

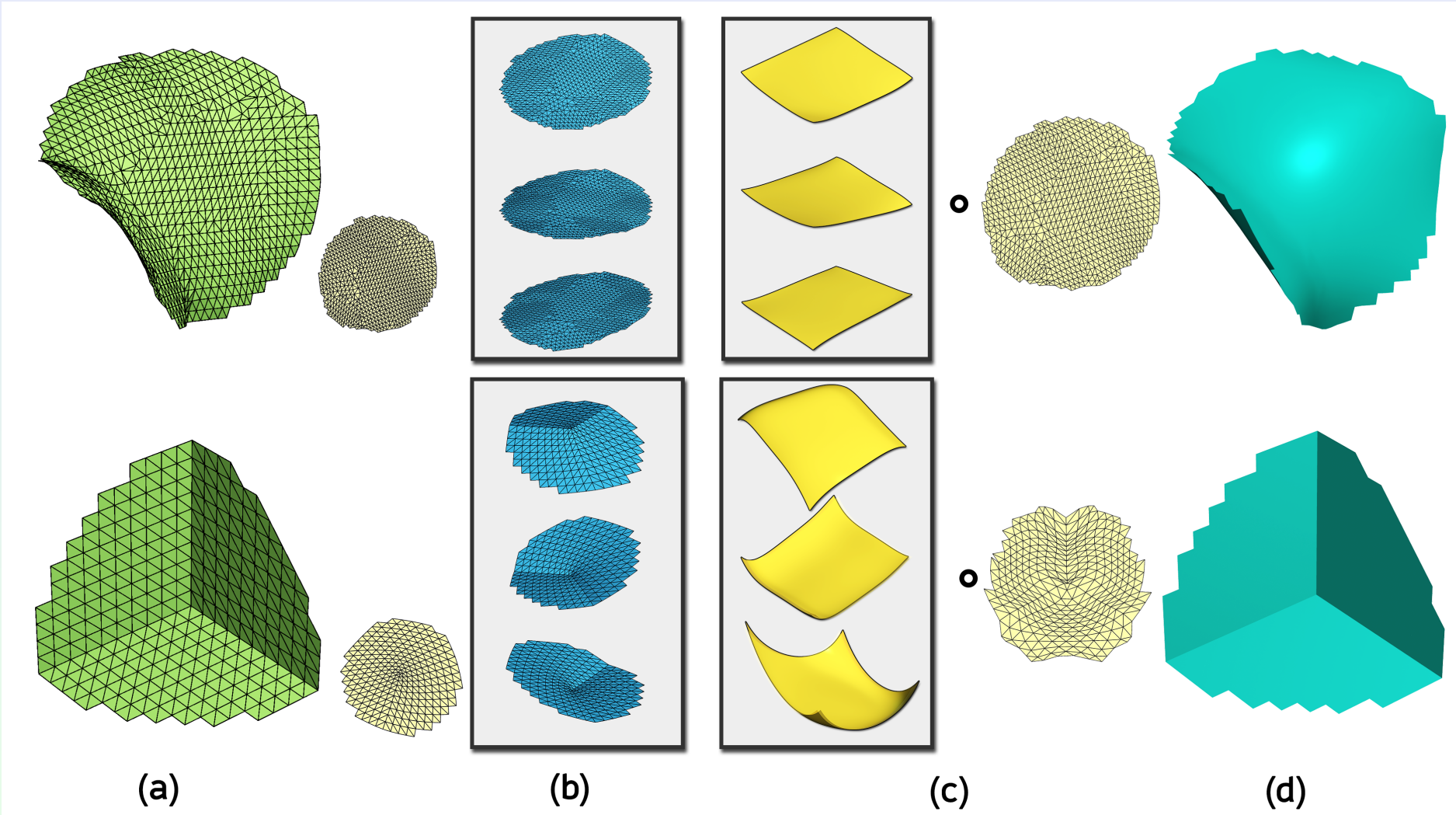
Overview of our algorithm



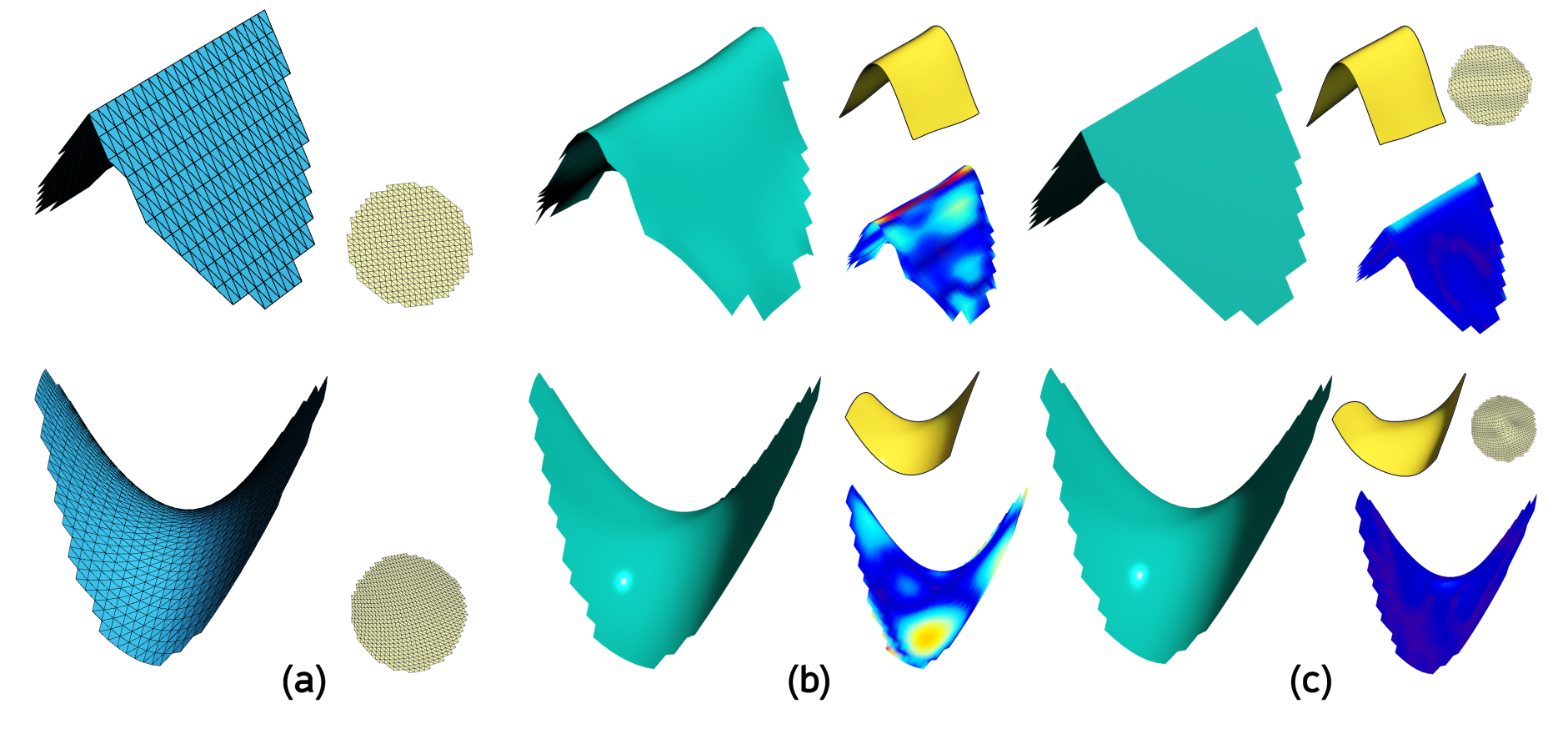
Experimental results



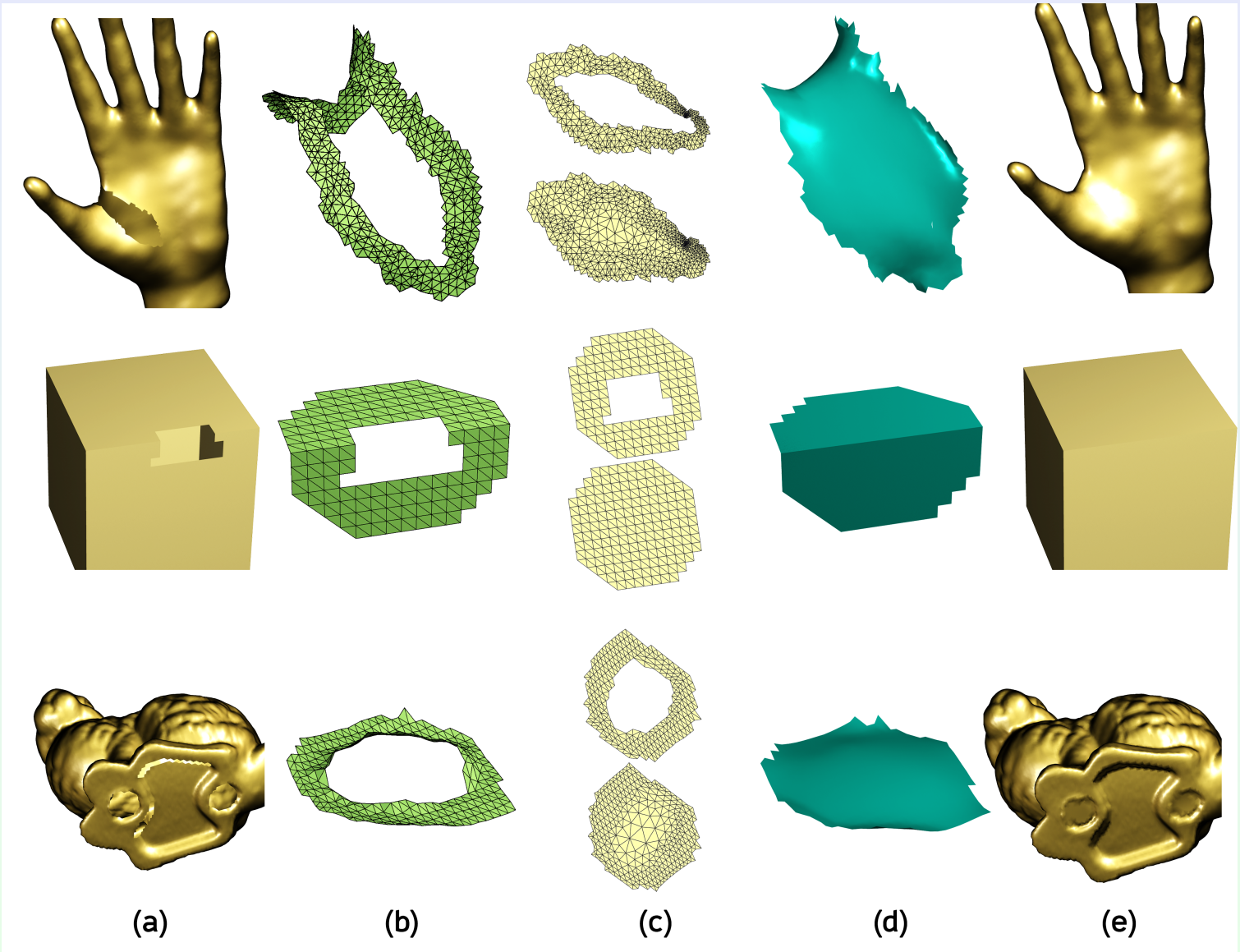
Experimental results



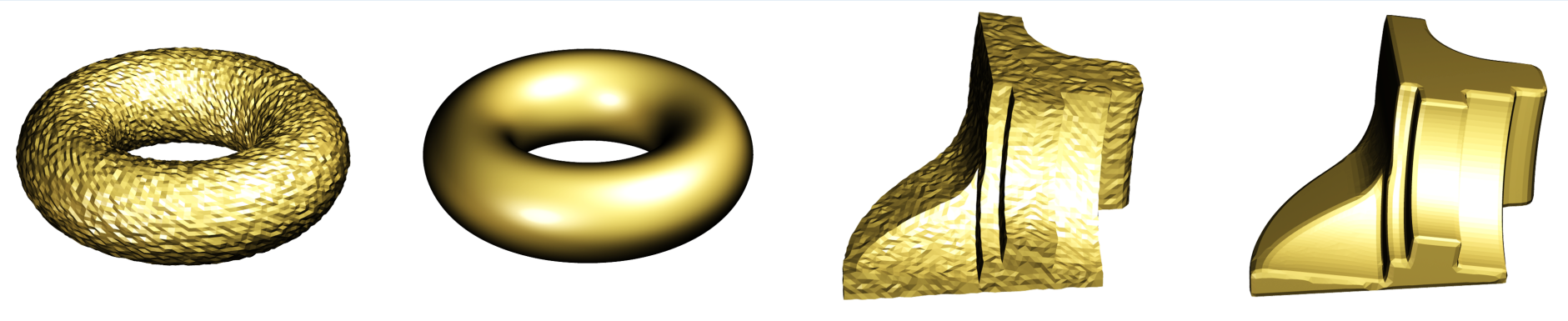
Comparison with naive sparse fitting



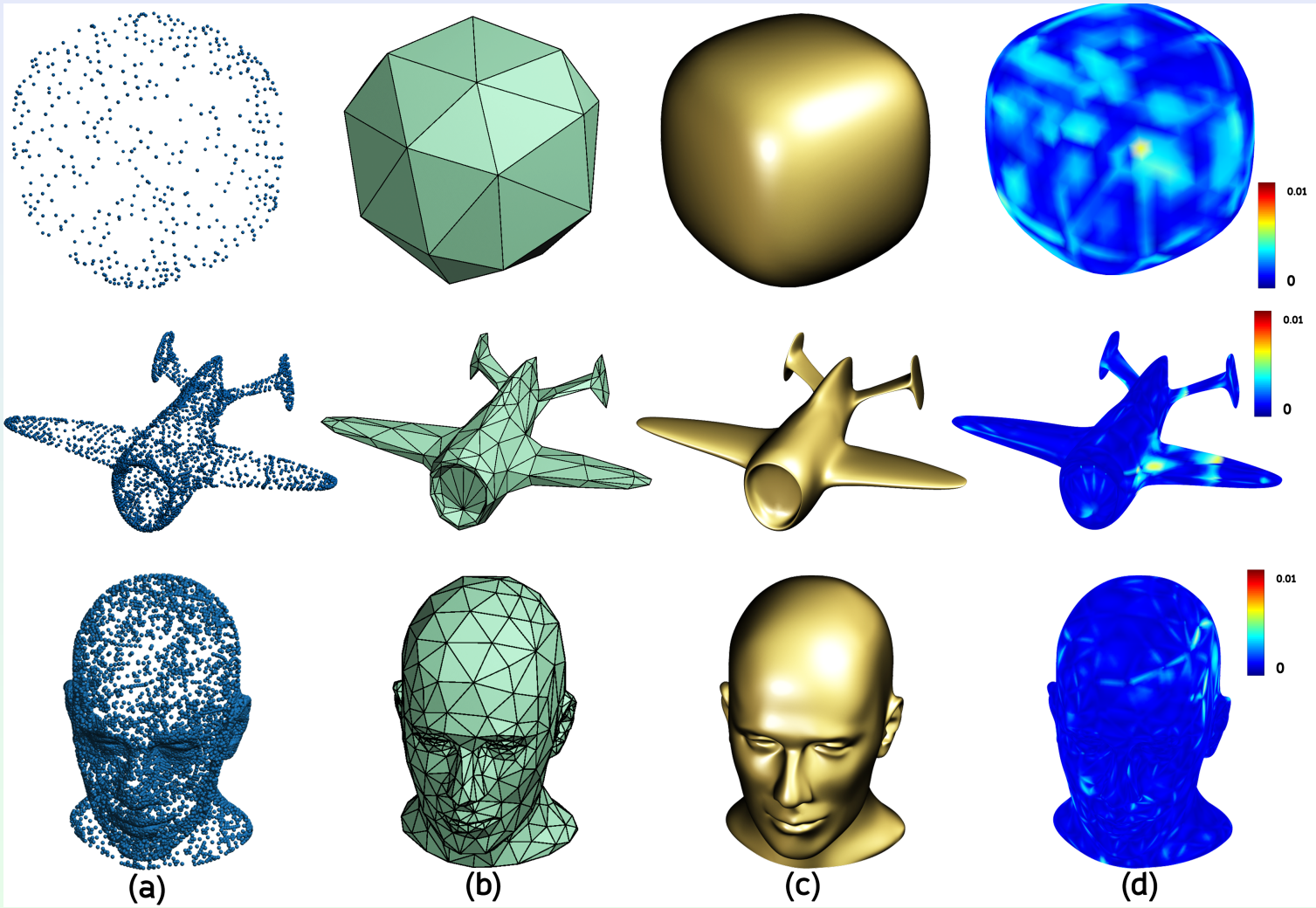
Application to mesh inpainting



Application to mesh smoothing



Application to surface reconstruction of point cloud



Outline I

Outline II

Future plans . . .

We would like to explore the emerging framework of sparse modeling and sparse optimization to interdisciplinary applications.

Thanks for your attention!