Detail-preserving mesh denoising using adaptive patches

Dissertação de Mestrado

Jan Jose Hurtado Jauregui[1]

Advisor: Marcelo Gattass[1]

[1] Departamento de Informática,
Pontifícia Universidade Católica do Rio de Janeiro,
Rio de Janeiro, Brazil

8th March 2018

Outline

Introduction

Noisy meshes acquired from 3D scanning.

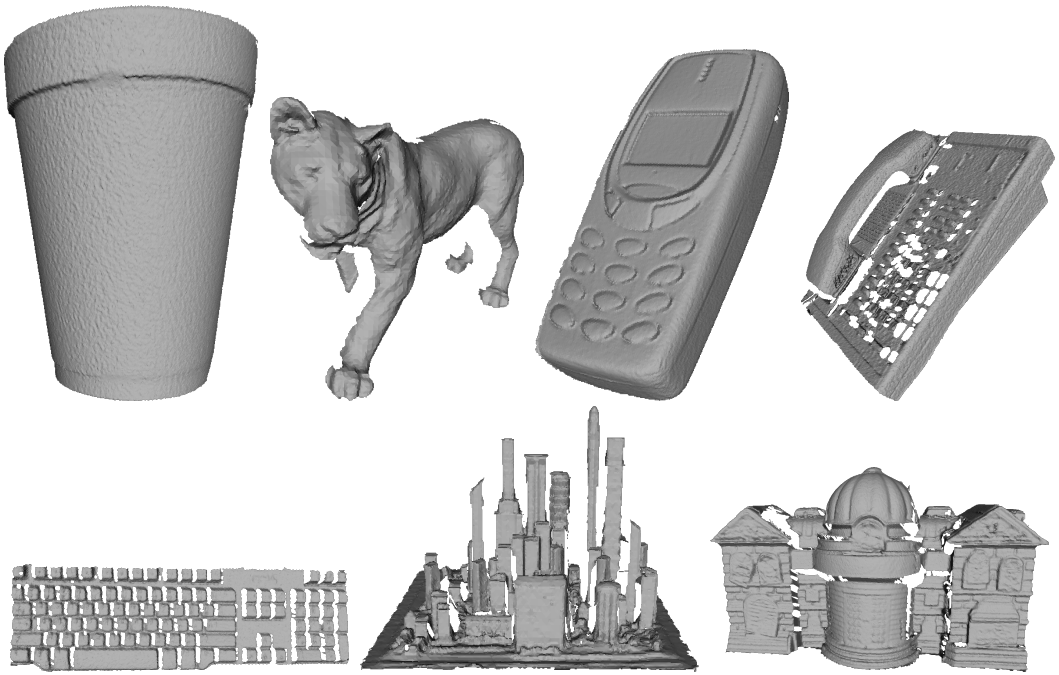


FIGURE – Range scans from SHREC'15.

Introduction

Noisy meshes acquired from medical data reconstruction.

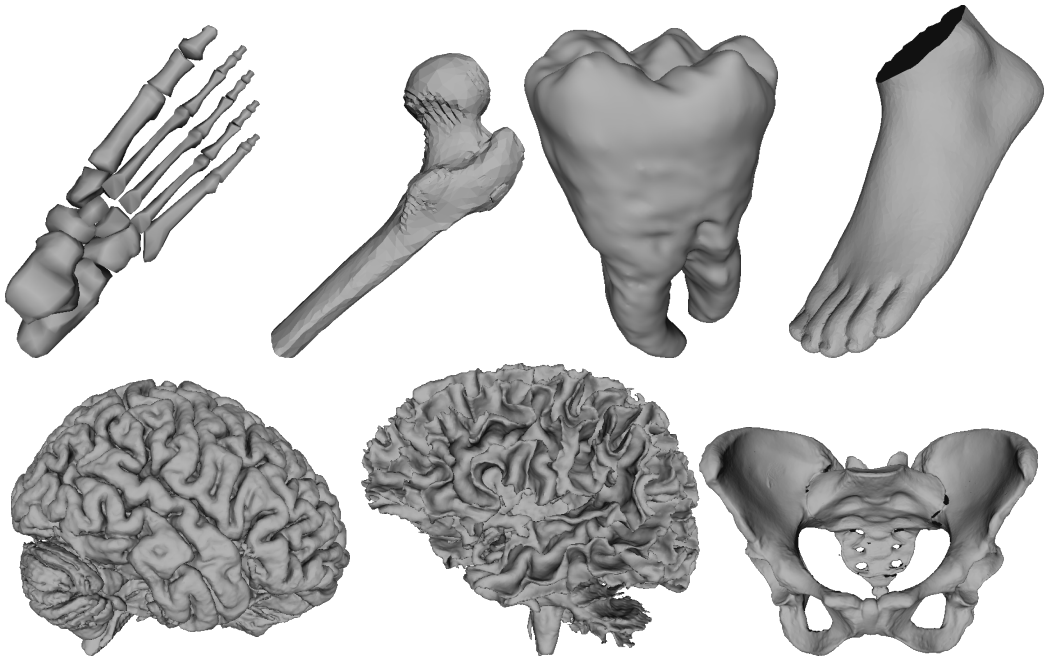


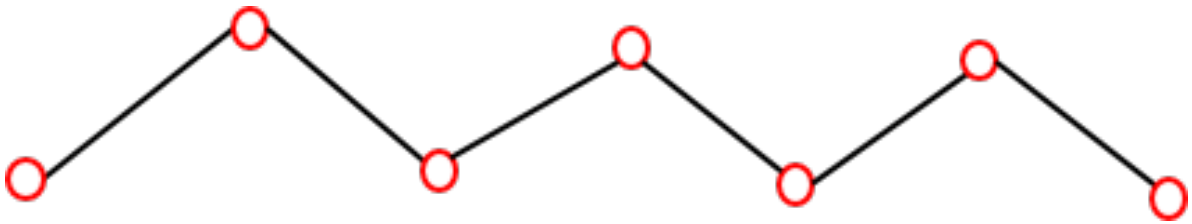
FIGURE – Meshes from the AIM@SHAPE Shape Repository

Introduction

- mesh denoising : noise removal while preserving high-frequency features (details), modifying vertex positions.
- challenging problem : it is difficult to distinguish high-frequency features from noise.
- the denoising step is part of a typical geometry processing pipeline.

Introduction

- mesh denoising : noise removal while preserving high-frequency features (details), modifying vertex positions.
- challenging problem : it is difficult to distinguish high-frequency features from noise.
- the denoising step is part of a typical geometry processing pipeline.



Introduction

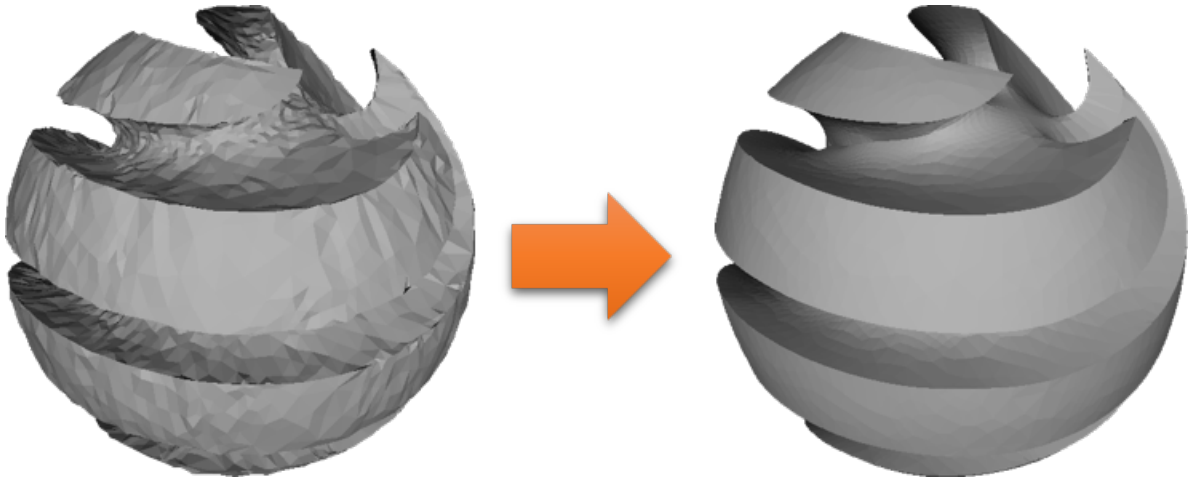
- mesh denoising : noise removal while preserving high-frequency features (details), modifying vertex positions.
- challenging problem : it is difficult to distinguish high-frequency features from noise.
- the denoising step is part of a typical geometry processing pipeline.

Introduction

- mesh denoising : noise removal while preserving high-frequency features (details), modifying vertex positions.
- challenging problem : it is difficult to distinguish high-frequency features from noise.
- the denoising step is part of a typical geometry processing pipeline.

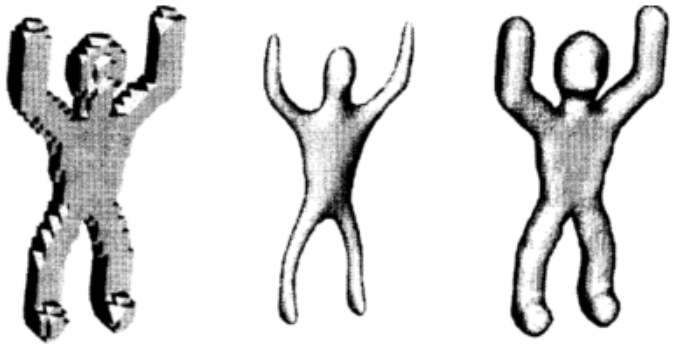
Introduction

- mesh denoising : noise removal while preserving high-frequency features (details), modifying vertex positions.
- challenging problem : it is difficult to distinguish high-frequency features from noise.
- the denoising step is part of a typical geometry processing pipeline.

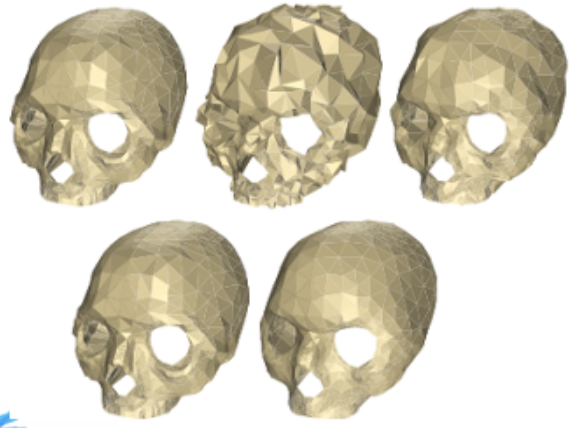


Previous work

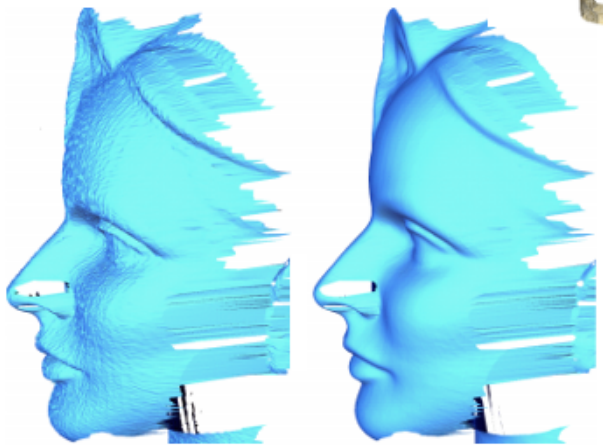
Isotropic denoising



[Taubin 1995]



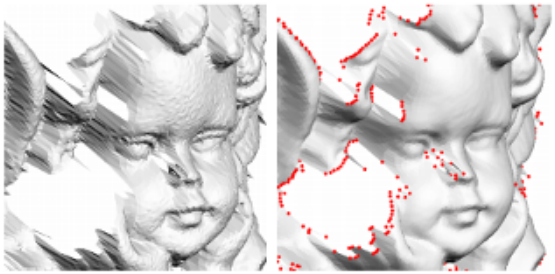
[Vollmer et al. 1999]



[Desbrun et al. 1999]

Previous work

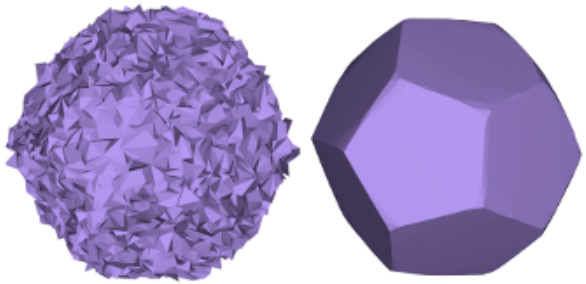
Anisotropic denoising



[Taubin 2001]

[Hildebrandt and Polthier 2004]

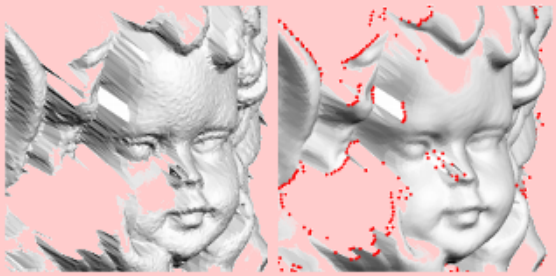
[Sun et al. 2007]



[He and Schaefer 2013]

Previous work

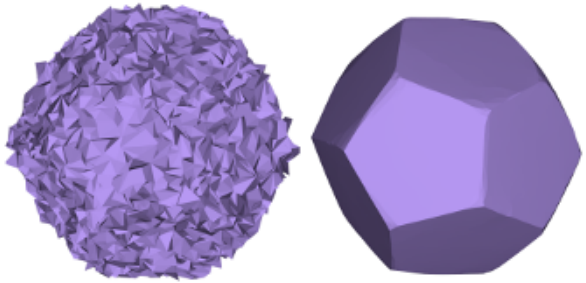
Anisotropic denoising - two-step based methods



[Taubin 2001]

[Hildebrandt and Polthier 2004]

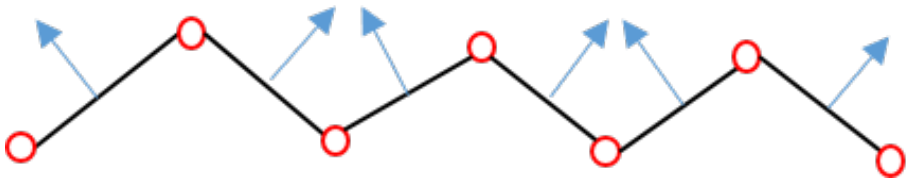
[Sun et al. 2007]



[He and Schaefer 2013]

Previous work

Normal filtering



Previous work

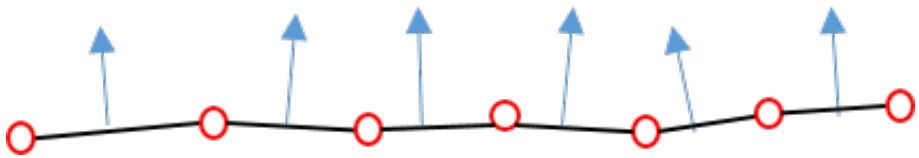
Normal filtering

Previous work

Normal filtering

Previous work

Normal filtering



Previous work

Bilateral filtering for images : We can represent an image as a signal $g(p)$, where p is the minimal structure, i.e. a pixel.

$$g'(p) = \frac{1}{W(p)} \sum_{q \in N(p)} K_c(||p-q||) K_s(||g(q)-g(p)||) g(q) \quad (1)$$

where $g'(p)$ is the new signal value at pixel p, $W(p)$ is the normalization factor, $N(p)$ represents the neighboring pixels of p, $K_c(x)$ and $K_s(x)$ are kernel functions to spatial and signal distance. $W(x) = \sum \omega$. $K_i(x) = e^{-x^2/2\sigma_i^2}$.

FIGURE – [Tomassi and Manduchi 1998]

Previous work

Bilateral normal filtering : Their scheme defines a bilateral filter over face normals, using the normals as signal and the corresponding face centroids as positions.

$$\mathbf{n}'_i = \frac{1}{W(f_i)} \sum_{f_j \in N_f(f_i)} A_j K_c(||\mathbf{c}_i - \mathbf{c}_j||) K_s(||\mathbf{n}_j - \mathbf{n}_i||) \mathbf{n}_j \tag{2}$$

where $\mathbf{n}'_i$ is the new normal, $N_f(f_i)$ represents the set of neighboring faces of face f_i, $\mathbf{c}_k$ is the corresponding centroid of f_k, and A_k is the area of the face f_k.

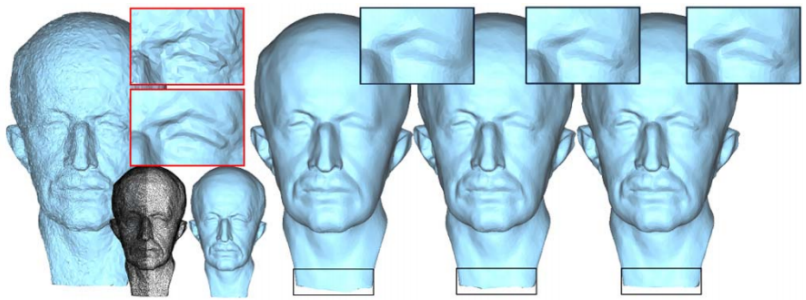


FIGURE – [Zheng et al. 2011]

Previous work

Joint bilateral filtering for images : It is an extension of bilateral filtering which uses a guidance image signal $\tilde{g}(p)$ to denoise the original signal $g(p)$.

$$g'(p) = \frac{1}{W(p)} \sum_{q \in N(p)} K_c(||p-q||) K_s(||\tilde{g}(q) - \tilde{g}(p)||) g(q) \tag{3}$$

FIGURE – [Petschnigg et al. 2004]

Previous work

Image denoising was addressed using shape adaptive patches or anisotropic neighborhoods. This idea is focused on the preservation of features such as edges or corners (detail preservation). In [Foi et al. 2007] the authors proposed an image filtering algorithm using patches generated by consistent 1D signals.

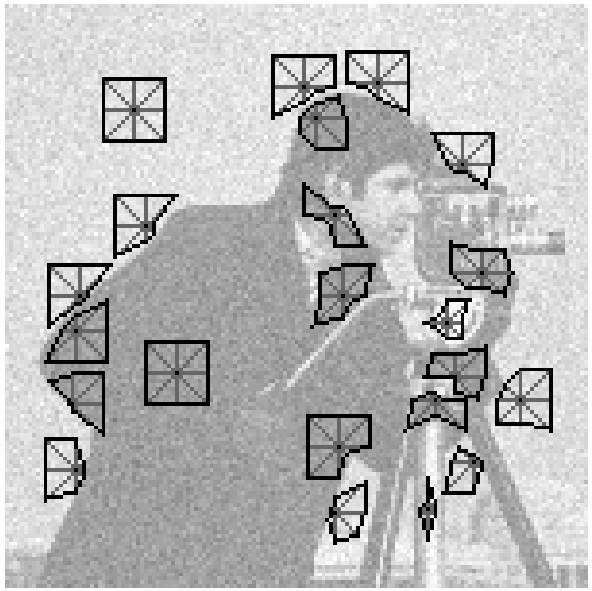


FIGURE – [Foi et al. 2007]

Previous work

Guided bilateral normal filtering : This algorithm is based on the joint bilateral filter and uses estimated normals defined by consistent patches.

$$\mathbf{n}'_i = \frac{1}{W(f_i)} \sum_{f_j \in N_f(f_i)} A_j K_c(||\mathbf{c}_i - \mathbf{c}_j||) K_s(||\tilde{\mathbf{n}}_j - \tilde{\mathbf{n}}_i||) \mathbf{n}_j \tag{4}$$

where $\tilde{\mathbf{n}}_k$ is the corresponding guidance signal value of face f_k. The guidance signal is defined as the average normal of a consistent patch centered at the corresponding face. The consistent patch is selected from a set of regular candidate patches which share the target face. The candidate with the minimum error, regarding a metric that measures normal difference, is the consistent one.

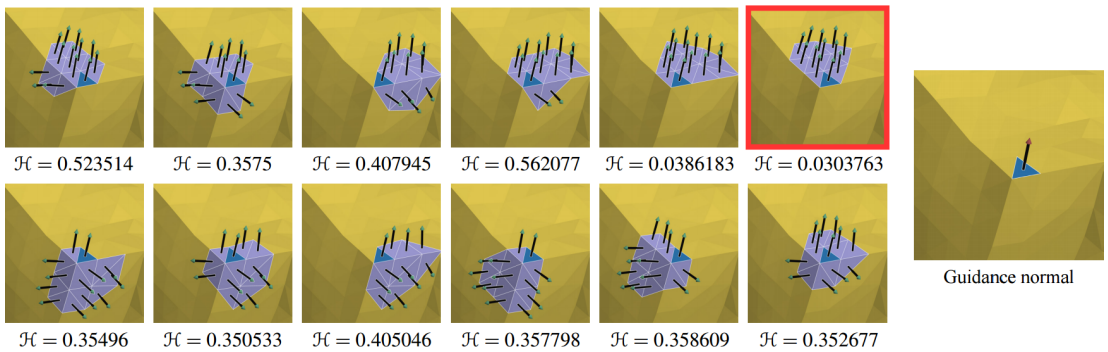
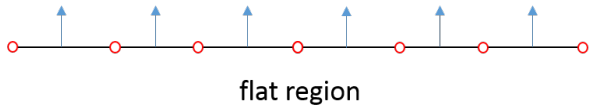


FIGURE – [Zhang et al. 2015]

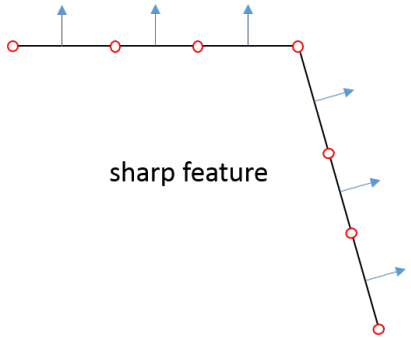
Adaptive patches

Let X be a 2-manifold embedded in $\mathbb{R}^3$, and X' a subset of X representing a patch (neighborhood) for a reference point $x' \in X$. We aim to find an optimal subset X' considering some properties desired in our denoising algorithm.

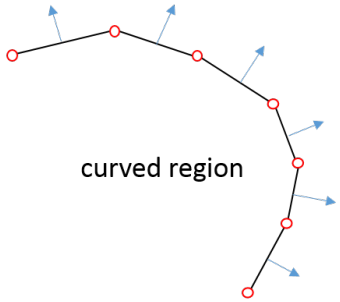
- shape adaptation.
- noise invariance.
- minimize normal variation.
  - low normal variation : flat regions.
  - high normal variation : sharp features, curved regions.



flat region



sharp feature

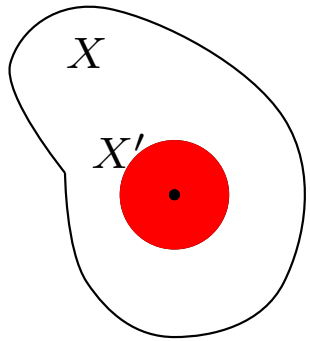


curved region

Adaptive patches (error term)

Finding the solution X' allows us to formulate a quadratic optimization problem penalizing the error between two points $x_i \in X'$ and $x_j \in X'$ as $q_{ij} = ||n_i - n_j||$.

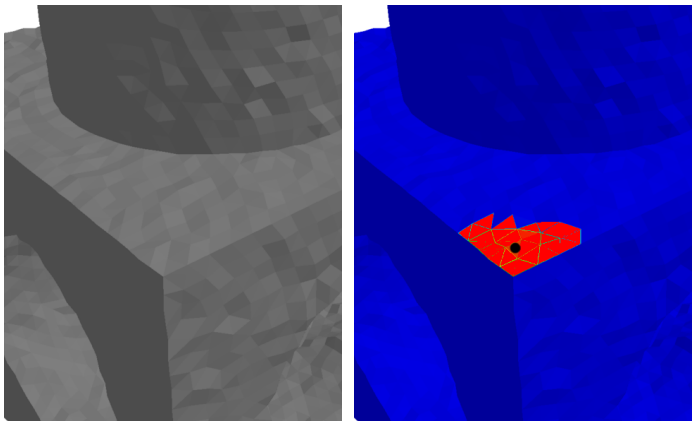
$$\min_{X' \subseteq X} \int_{x_i \in X'} \int_{x_j \in X'} q_{ij} da. \tag{5}$$



Adaptive patches (error term)

Finding the solution X' allows us to formulate a quadratic optimization problem penalizing the error between two points $x_i \in X'$ and $x_j \in X'$ as $q_{ij} = ||n_i - n_j||$.

$$\min_{X' \subseteq X} \int_{x_i \in X'} \int_{x_j \in X'} q_{ij} da. \tag{5}$$



Adaptive patches (error term)

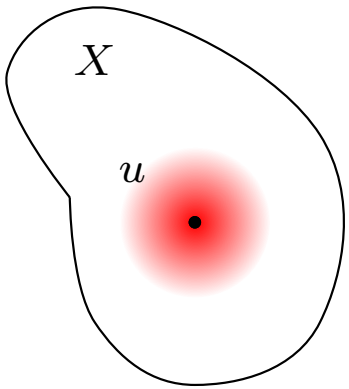
Finding the solution X' allows us to formulate a quadratic optimization problem penalizing the error between two points $x_i \in X'$ and $x_j \in X'$ as $q_{ij} = ||n_i - n_j||$.

$$\min_{X' \subseteq X} \int_{x_i \in X'} \int_{x_j \in X'} q_{ij} da. \tag{5}$$

Finding a crisp subset X' results in a NP-Hard combinatorial complexity problem. We can relax the problem defining a fuzzy membership function $u : X \to [0, 1]$ over all the domain X :

$$\min_{u} \int_{x_i \in X} \int_{x_j \in X} q_{ij} u_i u_j dada \quad s.t. \quad u \in [0, 1], \tag{6}$$

where $u_i = u(x_i)$ and $u_j = u(x_j)$. Now the optimization will find the membership function u and control the upperbound and lowerbound constraints.



Adaptive patches (error term)

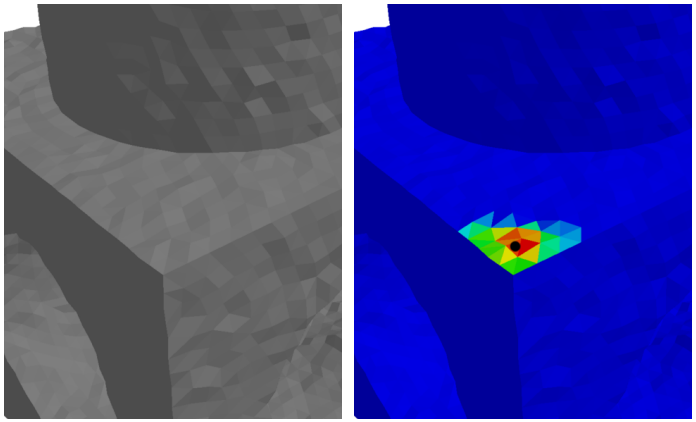
Finding the solution X' allows us to formulate a quadratic optimization problem penalizing the error between two points $x_i \in X'$ and $x_j \in X'$ as $q_{ij} = ||n_i - n_j||$.

$$\min_{X' \subseteq X} \int_{x_i \in X'} \int_{x_j \in X'} q_{ij} da. \tag{5}$$

Finding a crisp subset X' results in a NP-Hard combinatorial complexity problem. We can relax the problem defining a fuzzy membership function $u : X \to [0, 1]$ over all the domain X :

$$\min_{u} \int_{x_i \in X} \int_{x_j \in X} q_{ij} u_i u_j dada \quad s.t. \quad u \in [0, 1], \tag{6}$$

where $u_i = u(x_i)$ and $u_j = u(x_j)$. Now the optimization will find the membership function u and control the upperbound and lowerbound constraints.



Adaptive patches (error term)

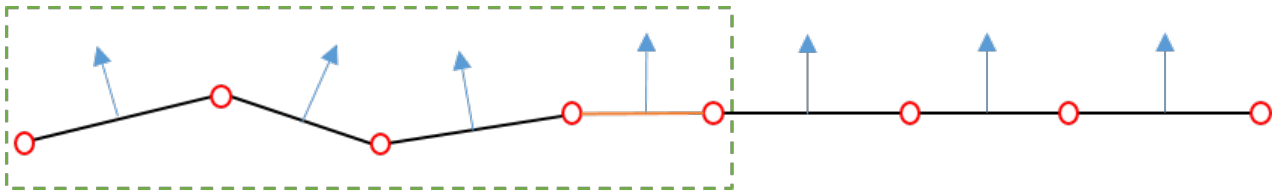
Finding the solution X' allows us to formulate a quadratic optimization problem penalizing the error between two points $x_i \in X'$ and $x_j \in X'$ as $q_{ij} = ||n_i - n_j||$.

$$\min_{X' \subseteq X} \int_{x_i \in X'} \int_{x_j \in X'} q_{ij} da. \tag{5}$$

Finding a crisp subset X' results in a NP-Hard combinatorial complexity problem. We can relax the problem defining a fuzzy membership function $u : X \to [0, 1]$ over all the domain X :

$$\min_{u} \int_{x_i \in X} \int_{x_j \in X} q_{ij} u_i u_j dada \quad s.t. \quad u \in [0, 1], \tag{6}$$

where $u_i = u(x_i)$ and $u_j = u(x_j)$. Now the optimization will find the membership function u and control the upperbound and lowerbound constraints.



high error

Adaptive patches (error term)

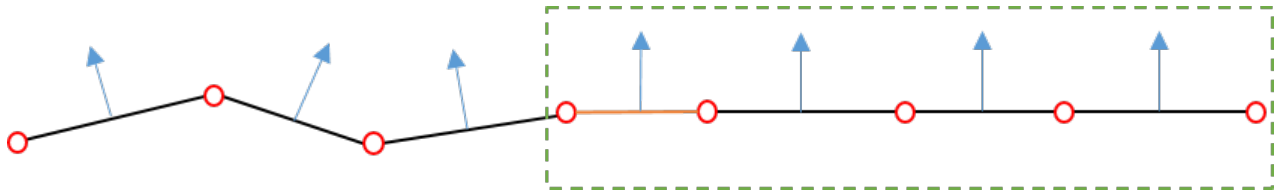
Finding the solution X' allows us to formulate a quadratic optimization problem penalizing the error between two points $x_i \in X'$ and $x_j \in X'$ as $q_{ij} = ||n_i - n_j||$.

$$\min_{X' \subseteq X} \int_{x_i \in X'} \int_{x_j \in X'} q_{ij} da. \tag{5}$$

Finding a crisp subset X' results in a NP-Hard combinatorial complexity problem. We can relax the problem defining a fuzzy membership function $u : X \to [0, 1]$ over all the domain X :

$$\min_{u} \int_{x_i \in X} \int_{x_j \in X} q_{ij} u_i u_j dada \quad s.t. \quad u \in [0, 1], \tag{6}$$

where $u_i = u(x_i)$ and $u_j = u(x_j)$. Now the optimization will find the membership function u and control the upperbound and lowerbound constraints.



low error, $q_{ij} = 0$.

Adaptive patches (error term)

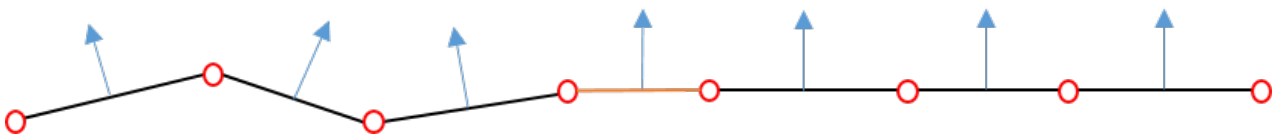
Finding the solution X' allows us to formulate a quadratic optimization problem penalizing the error between two points $x_i \in X'$ and $x_j \in X'$ as $q_{ij} = ||n_i - n_j||$.

$$\min_{X' \subseteq X} \int_{x_i \in X'} \int_{x_j \in X'} q_{ij} da. \tag{5}$$

Finding a crisp subset X' results in a NP-Hard combinatorial complexity problem. We can relax the problem defining a fuzzy membership function $u : X \to [0, 1]$ over all the domain X :

$$\min_{u} \int_{x_i \in X} \int_{x_j \in X} q_{ij} u_i u_j dada \quad s.t. \quad u \in [0, 1], \tag{6}$$

where $u_i = u(x_i)$ and $u_j = u(x_j)$. Now the optimization will find the membership function u and control the upperbound and lowerbound constraints.



low error, $q_{ij} = 0.\ area = 0$

Adaptive patches (error term)

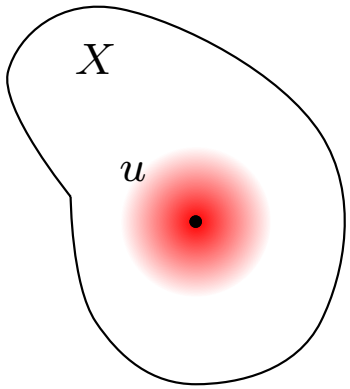
To obtain a solution with area we added a constraint for the solution u, such that the sum of the area of X weighted by the membership function u should be equal to a fixed value a_0.

$$\min_u \int_{x_i \in X} \int_{x_j \in X} q_{ij} u_i u_j dada \quad s.t. \quad u \in [0,1] \quad \wedge \quad \int_{x_i \in X} uda = a_0. \tag{7}$$

Adaptive patches (error term)

To obtain a solution with area we added a constraint for the solution u, such that the sum of the area of X weighted by the membership function u should be equal to a fixed value a_0.

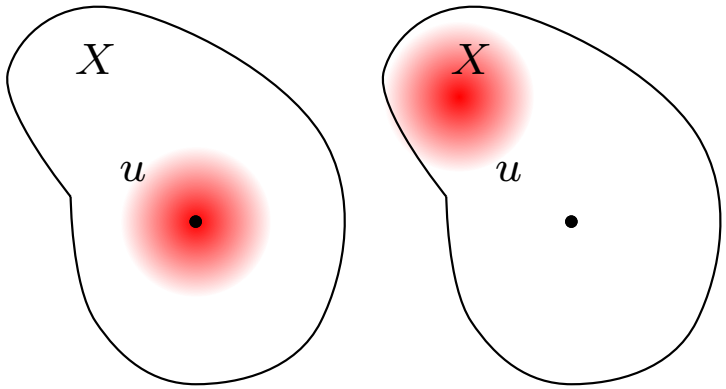
$$\min_u \int_{x_i \in X} \int_{x_j \in X} q_{ij} u_i u_j dada \quad s.t. \quad u \in [0,1] \quad \wedge \quad \int_{x_i \in X} uda = a_0. \tag{7}$$



Adaptive patches (error term)

To obtain a solution with area we added a constraint for the solution u, such that the sum of the area of X weighted by the membership function u should be equal to a fixed value a_0.

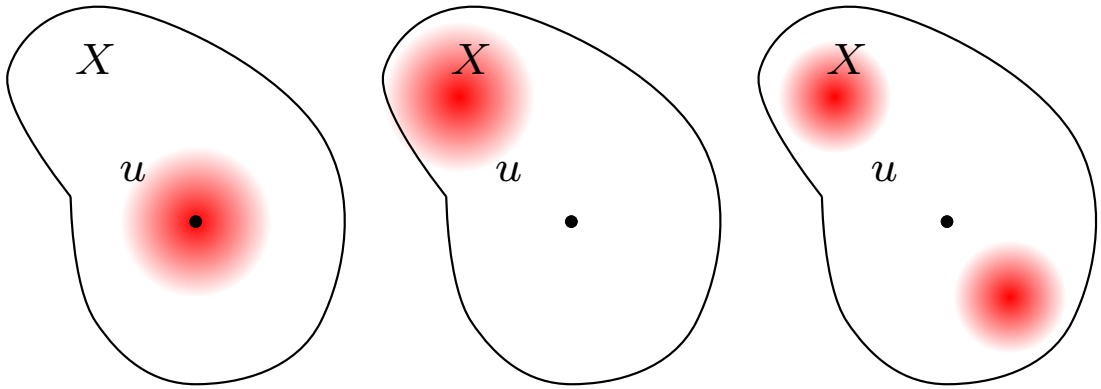
$$\min_u \int_{x_i \in X} \int_{x_j \in X} q_{ij} u_i u_j dada \quad s.t. \quad u \in [0,1] \quad \wedge \quad \int_{x_i \in X} uda = a_0. \tag{7}$$



Adaptive patches (error term)

To obtain a solution with area we added a constraint for the solution u, such that the sum of the area of X weighted by the membership function u should be equal to a fixed value a_0.

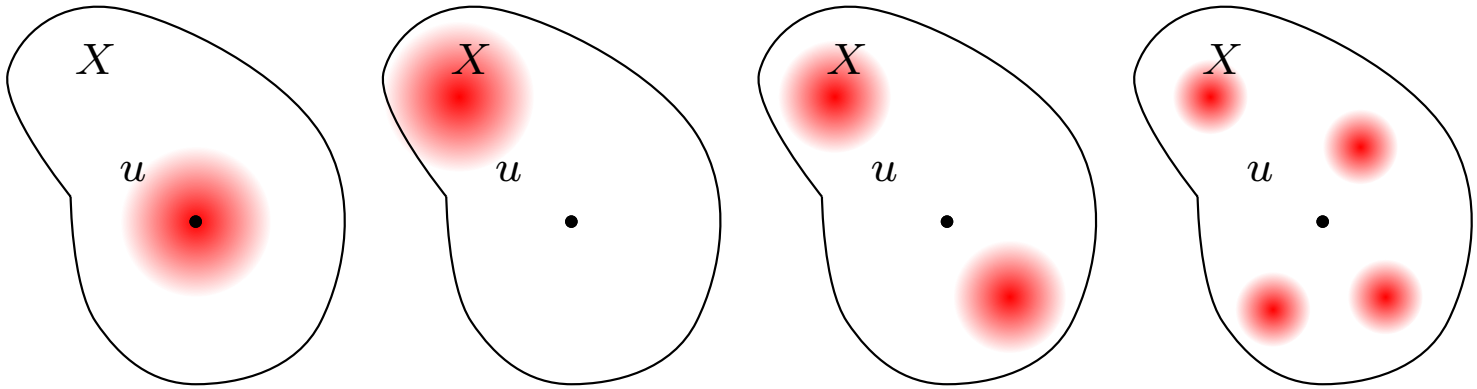
$$\min_u \int_{x_i \in X} \int_{x_j \in X} q_{ij} u_i u_j dada \quad s.t. \quad u \in [0,1] \quad \wedge \quad \int_{x_i \in X} uda = a_0. \tag{7}$$



Adaptive patches (error term)

To obtain a solution with area we added a constraint for the solution u, such that the sum of the area of X weighted by the membership function u should be equal to a fixed value a_0.

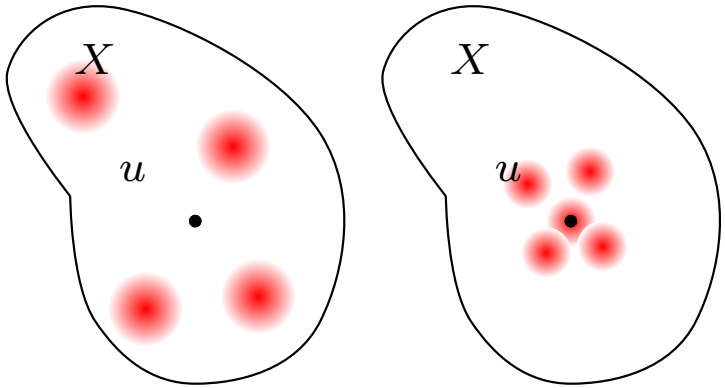
$$\min_u \int_{x_i \in X} \int_{x_j \in X} q_{ij} u_i u_j dada \quad s.t. \quad u \in [0,1] \quad \wedge \quad \int_{x_i \in X} uda = a_0. \tag{7}$$



Adaptive patches (distance to the reference point term)

The patch must be “centered” on the reference point or close to it. To address this problem we added a term to penalize the distance between any point of X to the reference point x' considering the weights of function u :

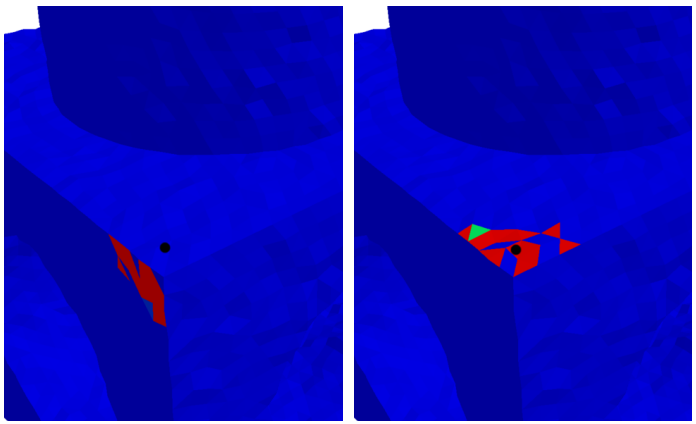
$$\min_u \int_{x_i \in X} \int_{x_j \in X} q_{ij} u_i u_j da da + \int_{x_i \in X} ||x' - x_i|| u_i da \qquad (8)$$
$$s.t. \quad u \in [0,1] \quad \wedge \quad \int_{x_i \in X} u da = a_0,$$



Adaptive patches (distance to the reference point term)

The patch must be “centered” on the reference point or close to it. To address this problem we added a term to penalize the distance between any point of X to the reference point x' considering the weights of function u :

$$\min_u \int_{x_i \in X} \int_{x_j \in X} q_{ij} u_i u_j da da + \int_{x_i \in X} ||x' - x_i|| u_i da \qquad (8)$$
$$s.t. \quad u \in [0,1] \quad \wedge \quad \int_{x_i \in X} u da = a_0,$$



Adaptive patches (regularization term)

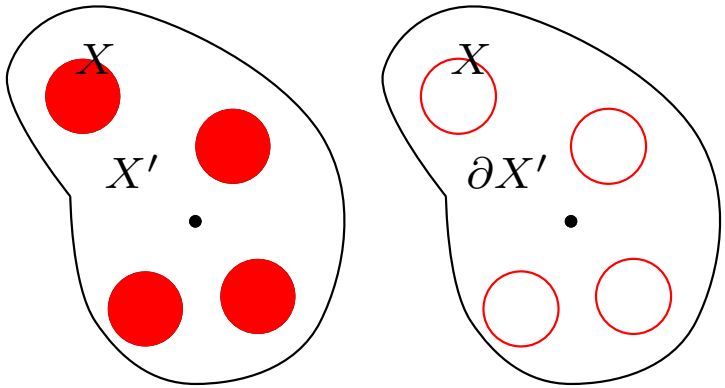
To avoid irregularity, we follow the idea of the Mumford-Shah functional trying to minimize the boundary length of the solution. Because we have a continuous function u instead of a crisp set X', we opted to use the squared gradient norm of u as a penalization term.

$$\begin{aligned}\min_u \int_{x_i \in X} \int_{x_j \in X} q_{ij} u_i u_j da da + \int_{x_i \in X} ||x' - x_i|| u_i da \\ + \int_X ||\nabla u||^2 da \quad s.t. \quad u \in [0,1] \quad \wedge \quad \int_{x_i \in X} u da = a_0,\end{aligned} \tag{9}$$

Adaptive patches (regularization term)

To avoid irregularity, we follow the idea of the Mumford-Shah functional trying to minimize the boundary length of the solution. Because we have a continuous function u instead of a crisp set X', we opted to use the squared gradient norm of u as a penalization term.

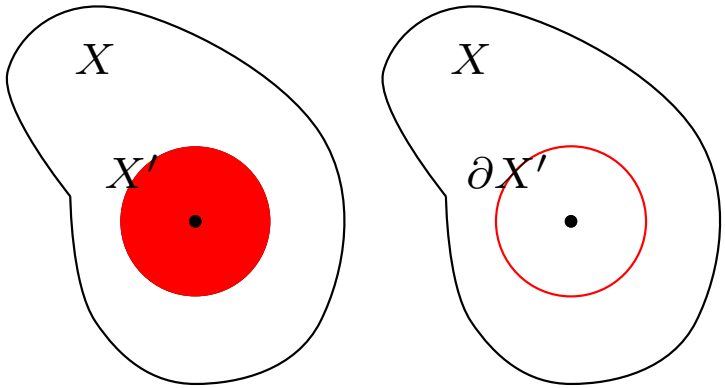
$$\min_u \int_{x_i \in X} \int_{x_j \in X} q_{ij} u_i u_j da da + \int_{x_i \in X} ||x' - x_i|| u_i da + \int_X ||\nabla u||^2 da \quad s.t. \quad u \in [0,1] \ \wedge \ \int_{x_i \in X} u da = a_0, \tag{9}$$



Adaptive patches (regularization term)

To avoid irregularity, we follow the idea of the Mumford-Shah functional trying to minimize the boundary length of the solution. Because we have a continuous function u instead of a crisp set X', we opted to use the squared gradient norm of u as a penalization term.

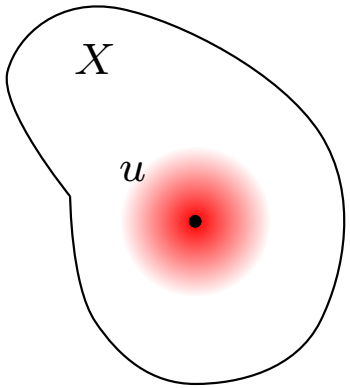
$$\min_u \int_{x_i \in X} \int_{x_j \in X} q_{ij} u_i u_j da da + \int_{x_i \in X} ||x' - x_i|| u_i da + \int_X ||\nabla u||^2 da \quad s.t. \quad u \in [0,1] \ \wedge \ \int_{x_i \in X} u da = a_0, \tag{9}$$



Adaptive patches (regularization term)

To avoid irregularity, we follow the idea of the Mumford-Shah functional trying to minimize the boundary length of the solution. Because we have a continuous function u instead of a crisp set X', we opted to use the squared gradient norm of u as a penalization term.

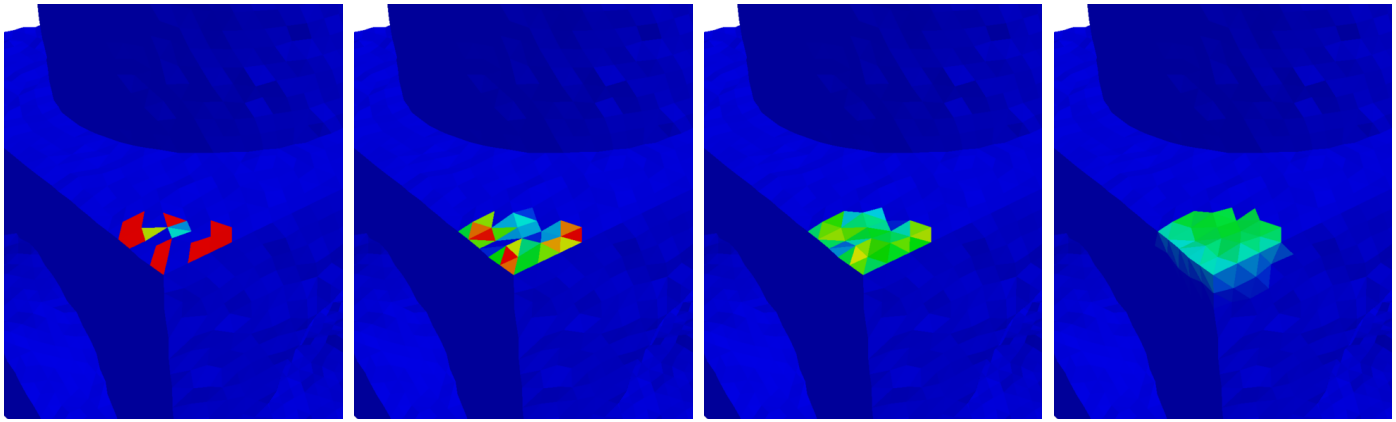
$$\min_u \int_{x_i \in X} \int_{x_j \in X} q_{ij} u_i u_j dada + \int_{x_i \in X} ||x' - x_i|| u_i da + \int_X ||\nabla u||^2 da \quad s.t. \quad u \in [0,1] \quad \wedge \quad \int_{x_i \in X} u da = a_0, \tag{9}$$



Adaptive patches (regularization term)

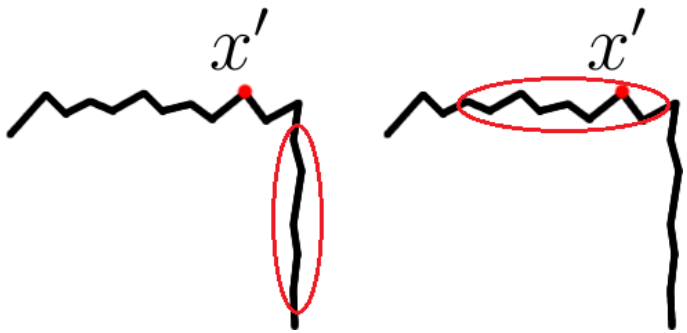
To avoid irregularity, we follow the idea of the Mumford-Shah functional trying to minimize the boundary length of the solution. Because we have a continuous function u instead of a crisp set X', we opted to use the squared gradient norm of u as a penalization term.

$$\min_u \int_{x_i \in X} \int_{x_j \in X} q_{ij} u_i u_j da da + \int_{x_i \in X} ||x' - x_i|| u_i da + \int_X ||\nabla u||^2 da \quad s.t. \quad u \in [0,1] \ \wedge \ \int_{x_i \in X} u da = a_0, \tag{9}$$



Adaptive patches (normal coherence term)

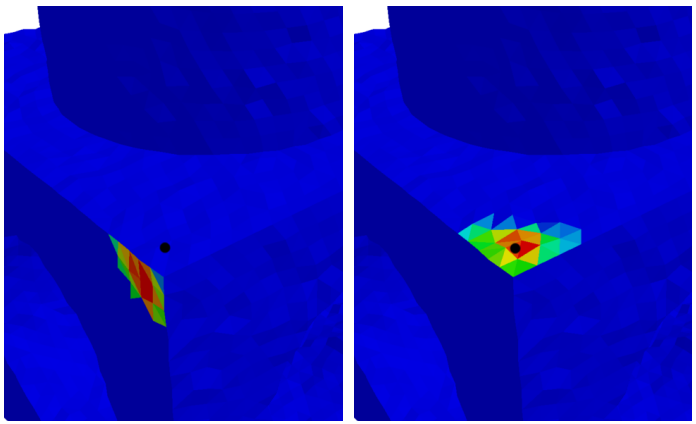
Depending on the influence of each of the previous terms we can obtain a solution including the reference point or not. If not it is possible to obtain a solution which has no coherence with the desired reference point normal.



Adaptive patches (normal coherence term)

$$\min_u \int_{x_i \in X} \int_{x_j \in X} q_{ij} u_i u_j da da + \int_{x_i \in X} ||x' - x_i|| u_i da$$
$$+ \int_X ||\nabla u||^2 da + \int_{x_i \in X} ||n' - n_i|| u_i da \quad (10)$$
$$s.t. \quad u \in [0,1] \quad \wedge \quad \int_{x_i \in X} u da = a_0,$$

where n' is the normal of the reference point



Adaptive patches (parameters)

$$\min_u \alpha \int_{x_i \in X} \int_{x_j \in X} q_{ij} u_i u_j da da + \beta \int_{x_i \in X} ||x' - x_i|| u_i da$$
$$+\gamma \int_X ||\nabla u||^2 da + \delta \int_{x_i \in X} ||n' - n_i|| u_i da \quad (11)$$
$$s.t. \quad u \in [0,1] \quad \wedge \quad \int_{x_i \in X} u da = a_0,$$

Adaptive patches (discretization)

Discretization using face centroids as sampled points.

- A triangular mesh M can be represented as a set of m vertices $V = \{v_1, \ldots, v_m\}$ and a set of n faces $F = \{f_1, \ldots, f_n\}$
- Vertex positions : $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$
- Face centroids : $C = \{\mathbf{c}_1, \ldots, \mathbf{c}_n\}$
- Face normals : $N = \{\mathbf{n}_1, \ldots, \mathbf{n}_n\}$
- Membership function : $\mathbf{u} = \{u_1, \ldots, u_n\}^T$
- Face areas : $\mathbf{a} = \{a_1, \ldots, a_n\}^T$
- Face areas diagonal matrix : $\mathbf{A}$ such that $(a_{ii}) = a_i$

Adaptive patches (discretization)

$$\min_u \alpha \int_{x_i \in X} \int_{x_j \in X} q_{ij} u_i u_j da da + \beta \int_{x_i \in X} ||x' - x_i|| u_i da$$
$$+\gamma \int_X ||\nabla u||^2 da + \delta \int_{x_i \in X} ||n' - n_i|| u_i da \quad (12)$$
$$s.t. \quad u \in [0,1] \quad \wedge \quad \int_{x_i \in X} u da = a_0,$$

$$\sum_{i=1}^{n} \sum_{j=1}^{n} q_{ij} u_i a_i u_j a_j,$$

using a matrix form we have :

$$\mathbf{u}^T \mathbf{A}^T \mathbf{Q} \mathbf{A} \mathbf{u},$$

where $\mathbf{A}$ is the diagonal matrix containing face areas and each entry of $\mathbf{Q}$ is defined by $q_{ij} = ||\mathbf{n}_i - \mathbf{n}_j||$.

Adaptive patches (discretization)

$$\min_u \alpha \int_{x_i \in X} \int_{x_j \in X} q_{ij} u_i u_j dada + \beta \int_{x_i \in X} ||x' - x_i|| u_i da$$
$$+\gamma \int_X ||\nabla u||^2 da + \delta \int_{x_i \in X} ||n' - n_i|| u_i da \quad (13)$$
$$s.t. \quad u \in [0,1] \ \wedge \ \int_{x_i \in X} u da = a_0,$$

The lowerbound and upperbound constraints can be represented by the vectors $\mathbf{0}$ and $\mathbf{1}$, which are n-dimensional vectors containing in all their entries 0s and 1s respectively. The area constraint results in a single linear constraint $\sum_i^n a_i u_i = a_0$, whose matrix representation is : $\mathbf{a}^T \mathbf{u} = a_0$.

Adaptive patches (discretization)

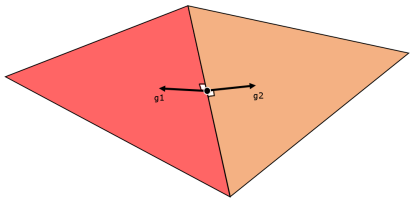
$$\min_u \alpha \int_{x_i \in X} \int_{x_j \in X} q_{ij} u_i u_j da da + \beta \int_{x_i \in X} ||x' - x_i|| u_i da$$
$$+\gamma \int_X ||\nabla u||^2 da + \delta \int_{x_i \in X} ||n' - n_i|| u_i da \quad (14)$$
$$s.t. \quad u \in [0,1] \quad \wedge \quad \int_{x_i \in X} u da = a_0,$$

$\mathbf{d}$ represents the distance between each centroid and the reference centroid. This term can be described by $a' \sum_i^n d_i a_i u_i$, and in a matrix form by $\mathbf{d}^T a' \mathbf{A} \mathbf{u}$

Adaptive patches (discretization)

$$\min_u \alpha \int_{x_i \in X} \int_{x_j \in X} q_{ij} u_i u_j dada + \beta \int_{x_i \in X} ||x' - x_i|| u_i da$$
$$+ \gamma \int_X ||\nabla u||^2 da + \delta \int_{x_i \in X} ||n' - n_i|| u_i da \quad (15)$$
$$s.t. \quad u \in [0,1] \quad \wedge \quad \int_{x_i \in X} u da = a_0,$$

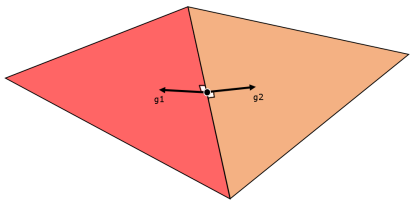
- u is constant in the face, gradient norm is 0 within it.
- integrate the gradient norm over the edges
- the gradient should be orthogonal to the edge
- two possible directions depending on u values
- gradient norm over an edge point is equal to $|u_i - u_j|$
- integrating the gradient norm over an edge is equal to $l|u_i - u_j|$



Adaptive patches (discretization)

$$\min_u \alpha \int_{x_i \in X} \int_{x_j \in X} q_{ij} u_i u_j dada + \beta \int_{x_i \in X} ||x' - x_i|| u_i da$$
$$+ \gamma \int_X ||\nabla u||^2 da + \delta \int_{x_i \in X} ||n' - n_i|| u_i da \quad (15)$$
$$s.t. \quad u \in [0,1] \quad \wedge \quad \int_{x_i \in X} u da = a_0,$$

$$\mathbf{G} = (g_{ij}) = \begin{cases} \sum_{f_k \in N_f(f_i)} l_{ik} & i = j \\ -l_{ij} & e_{ij} \in E \\ 0 & \text{otherwise} \end{cases} \quad (16)$$



$$||\nabla u||^2 \approx (\mathbf{Gu})^2 = \mathbf{u}^T \mathbf{G}^T \mathbf{Gu}. \quad (17)$$

Adaptive patches (discretization)

$$\min_u \alpha \int_{x_i \in X} \int_{x_j \in X} q_{ij} u_i u_j dada + \beta \int_{x_i \in X} ||x' - x_i|| u_i da$$
$$+\gamma \int_X ||\nabla u||^2 da + \delta \int_{x_i \in X} ||n' - n_i|| u_i da \quad (18)$$
$$s.t. \quad u \in [0,1] \quad \wedge \quad \int_{x_i \in X} uda = a_0,$$

We can write this term as $a' \sum_i^n ||\mathbf{n}_i - \mathbf{n}'|| a_i u_i$, and in a matrix form as $\mathbf{f}^T a' \mathbf{A}\mathbf{u}$, where $\mathbf{f}$ is a n dimensional vector containing in ith position the normal difference between the reference face and the face f_i.

Adaptive patches (discretization)

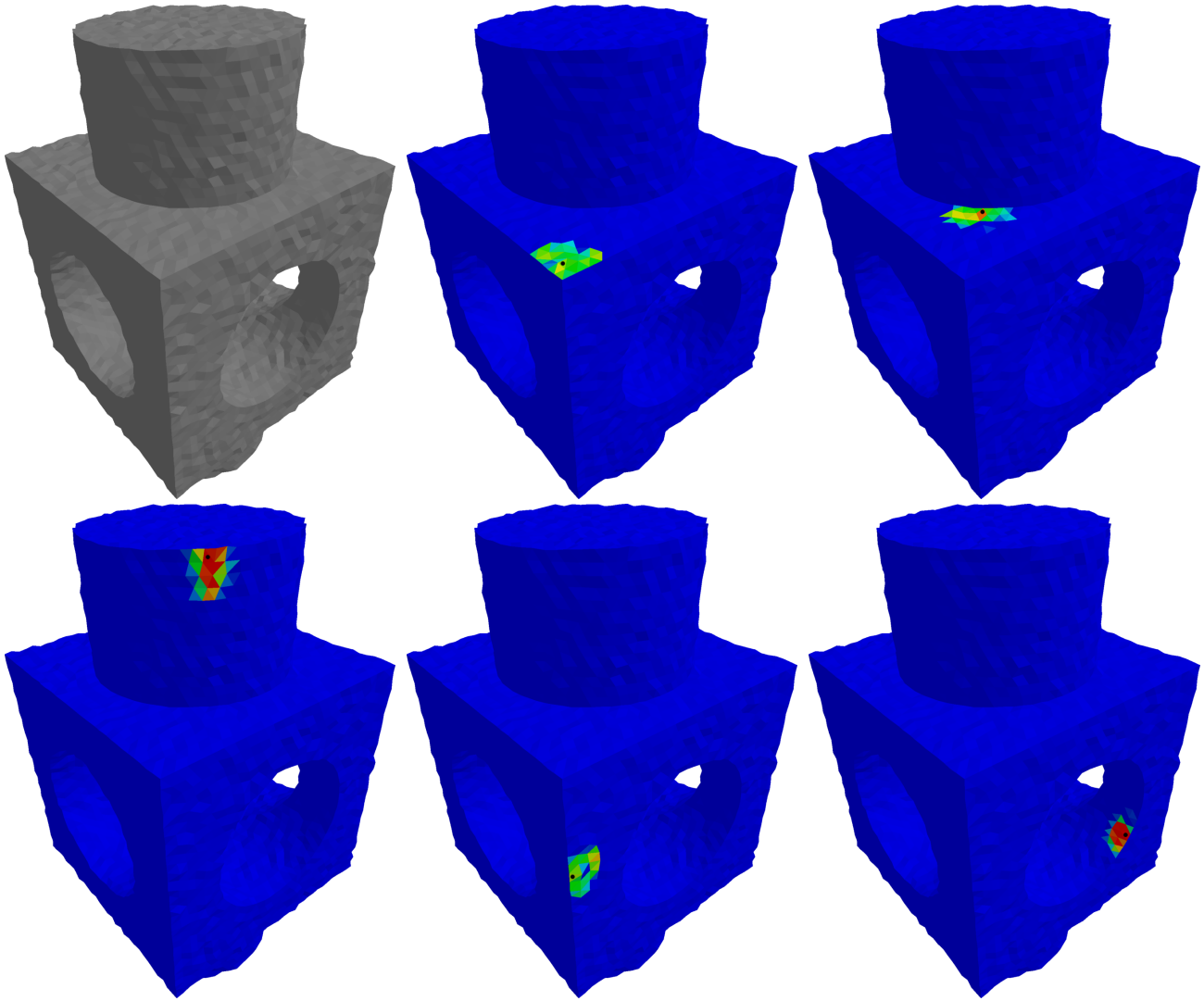
$$\min_{u} \alpha \int_{x_i \in X} \int_{x_j \in X} q_{ij} u_i u_j dada + \beta \int_{x_i \in X} ||x' - x_i|| u_i da$$
$$+\gamma \int_{X} ||\nabla u||^2 da + \delta \int_{x_i \in X} ||n' - n_i|| u_i da \qquad (19)$$
$$s.t. \quad u \in [0,1] \quad \wedge \quad \int_{x_i \in X} u da = a_0,$$

Considering the parameters that control the optimization behavior we have the following quadratic optimization problem for each face centroid of the mesh :

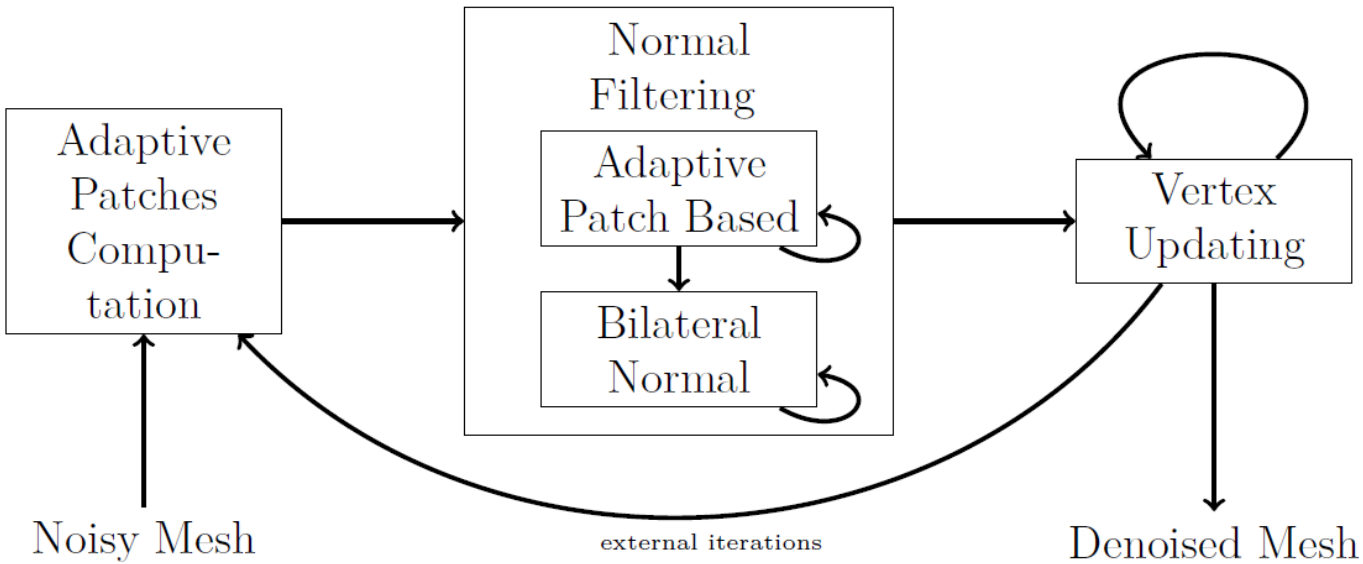
$$\min_{\mathbf{u}} \alpha \mathbf{u}^T \mathbf{A}^T \mathbf{Q} \mathbf{A} \mathbf{u} + \beta \mathbf{d}^T a' \mathbf{A} \mathbf{u} + \gamma \mathbf{u}^T \mathbf{G}^T \mathbf{G} \mathbf{u} + \delta \mathbf{f}^T a' \mathbf{A} \mathbf{u}$$
$$s.t. \quad \mathbf{0} \leq \mathbf{u} \leq \mathbf{1} \wedge \mathbf{a}^T \mathbf{u} = a_0. \qquad (20)$$

Factorizing, it leads to a typical quadratic optimization problem with a quadratic term, a linear term, upper bound constraints, lower bound constraints, and a single linear equality constraint.

Adaptive patches (examples)



Denoising algorithm



Denoising algorithm

Adaptive patch based filtering :

$$\mathbf{n}' = \sum_{f_i \in F} \mathbf{n}_i u_i a_i, \tag{21}$$

Bilateral normal filtering :

$$\mathbf{n}'_i = \frac{1}{W(f_i)} \sum_{f_j \in F} A_j K_c(distanceMatrix(i,j)) K_s(||\mathbf{n}_j - \mathbf{n}_i||)\mathbf{n}_j. \tag{22}$$

Denoising algorithm

Vertex updating :

$$E(M) = \begin{cases} \mathbf{n}_f \cdot (\mathbf{x}_j - \mathbf{x}_i) \\ \mathbf{n}_f \cdot (\mathbf{x}_k - \mathbf{x}_j) \\ \mathbf{n}_f \cdot (\mathbf{x}_i - \mathbf{x}_k) \end{cases} \quad , \quad \forall f(i,j,k) \tag{23}$$

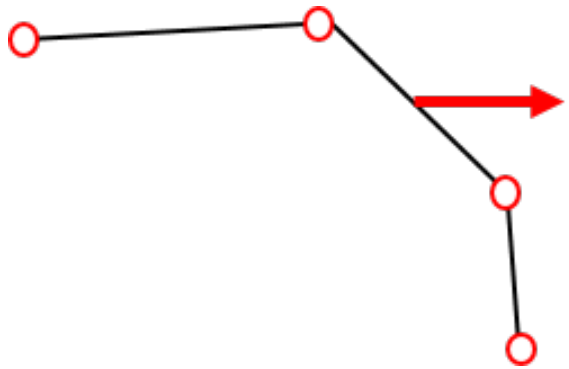
$$\mathbf{x}'_i = \mathbf{x}_i + \frac{1}{|F_v(v_i)|} \sum_{f_k \in F_v(v_i)} \mathbf{n}'_k(\mathbf{n}'_k \cdot (\mathbf{c}_k - \mathbf{x}_i)) \tag{24}$$

Denoising algorithm

Vertex updating :

$$E(M) = \begin{cases} \mathbf{n}_f \cdot (\mathbf{x}_j - \mathbf{x}_i) \\ \mathbf{n}_f \cdot (\mathbf{x}_k - \mathbf{x}_j) \\ \mathbf{n}_f \cdot (\mathbf{x}_i - \mathbf{x}_k) \end{cases} , \quad \forall f(i,j,k) \tag{23}$$

$$\mathbf{x}'_i = \mathbf{x}_i + \frac{1}{|F_v(v_i)|} \sum_{f_k \in F_v(v_i)} \mathbf{n}'_k (\mathbf{n}'_k \cdot (\mathbf{c}_k - \mathbf{x}_i)) \tag{24}$$

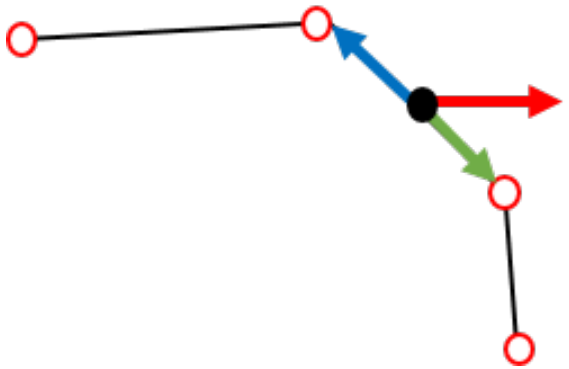


Denoising algorithm

Vertex updating :

$$E(M) = \begin{cases} \mathbf{n}_f \cdot (\mathbf{x}_j - \mathbf{x}_i) \\ \mathbf{n}_f \cdot (\mathbf{x}_k - \mathbf{x}_j) \\ \mathbf{n}_f \cdot (\mathbf{x}_i - \mathbf{x}_k) \end{cases} , \quad \forall f(i,j,k) \tag{23}$$

$$\mathbf{x}'_i = \mathbf{x}_i + \frac{1}{|F_v(v_i)|} \sum_{f_k \in F_v(v_i)} \mathbf{n}'_k(\mathbf{n}'_k \cdot (\mathbf{c}_k - \mathbf{x}_i)) \tag{24}$$

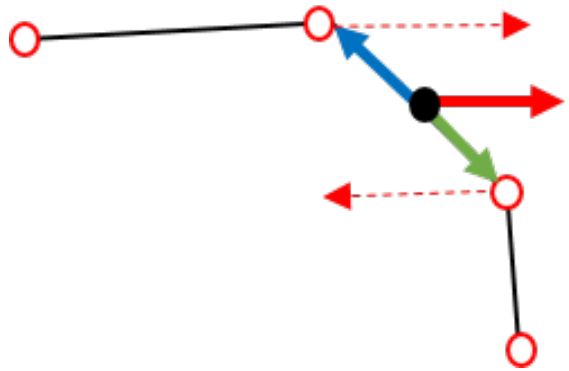


Denoising algorithm

Vertex updating :

$$E(M) = \begin{cases} \mathbf{n}_f \cdot (\mathbf{x}_j - \mathbf{x}_i) \\ \mathbf{n}_f \cdot (\mathbf{x}_k - \mathbf{x}_j) \\ \mathbf{n}_f \cdot (\mathbf{x}_i - \mathbf{x}_k) \end{cases} , \quad \forall f(i,j,k) \tag{23}$$

$$\mathbf{x}'_i = \mathbf{x}_i + \frac{1}{|F_v(v_i)|} \sum_{f_k \in F_v(v_i)} \mathbf{n}'_k(\mathbf{n}'_k \cdot (\mathbf{c}_k - \mathbf{x}_i)) \tag{24}$$

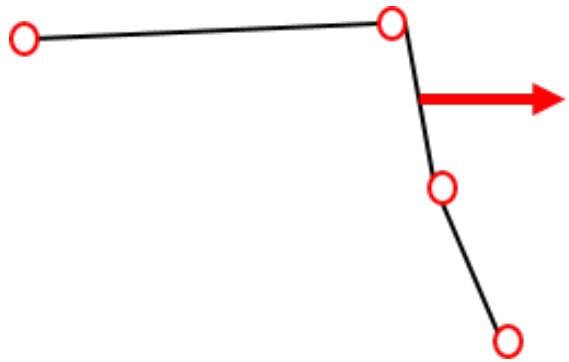


Denoising algorithm

Vertex updating :

$$E(M) = \begin{cases} \mathbf{n}_f \cdot (\mathbf{x}_j - \mathbf{x}_i) \\ \mathbf{n}_f \cdot (\mathbf{x}_k - \mathbf{x}_j) \\ \mathbf{n}_f \cdot (\mathbf{x}_i - \mathbf{x}_k) \end{cases} , \quad \forall f(i,j,k) \tag{23}$$

$$\mathbf{x}'_i = \mathbf{x}_i + \frac{1}{|F_v(v_i)|} \sum_{f_k \in F_v(v_i)} \mathbf{n}'_k(\mathbf{n}'_k \cdot (\mathbf{c}_k - \mathbf{x}_i)) \tag{24}$$



Denoising algorithm evaluation

Visual comparison

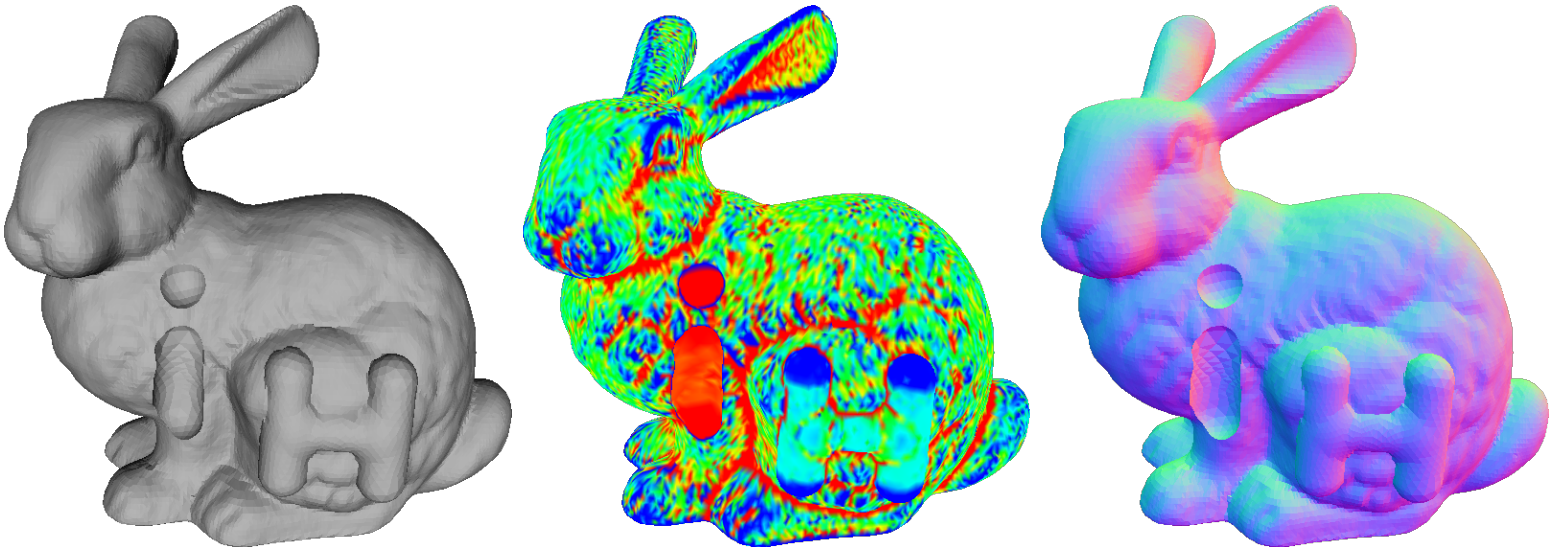


FIGURE – Visualization methods. From left to right : Flat shading, curvature color mapping and normal mapping.

Denoising algorithm evaluation

Numerical comparison - metrics

Type	**Metrics**
Distance based	Mean distance error (CRS96) Quadric error metric (GH97) L^2 vertex-based mesh-to-mesh error metric (YOB02, BO03)
Normal based	Tangential Error Metric (KKK02) L^2 normal-based mesh-to-mesh error metric (YOB02, BO03) Mean Square Angular Error (NH00, SB04)
Curvature based	Discrete curvature error metric (KKK02)

(*) volume/area preservation

Initial Comparison

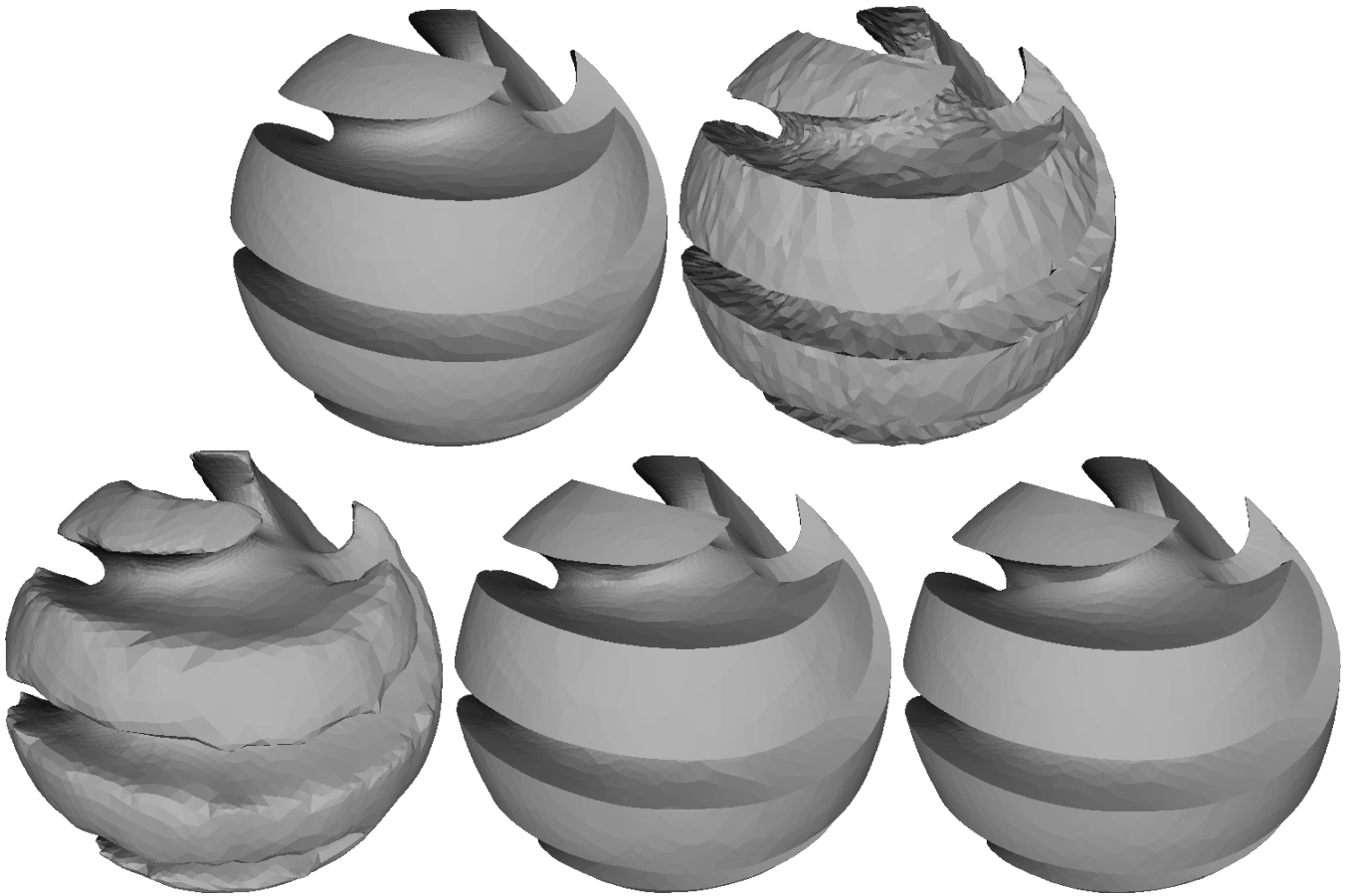


FIGURE – An example using our denoising algorithm. First row from left to right : original mesh and noisy mesh. Second row from left to right : resulting mesh using uniform Laplacian smoothing, resulting mesh of our proposal wihout using the bilateral normal filtering step, and resulting mesh of our proposal using it.

Initial Comparison

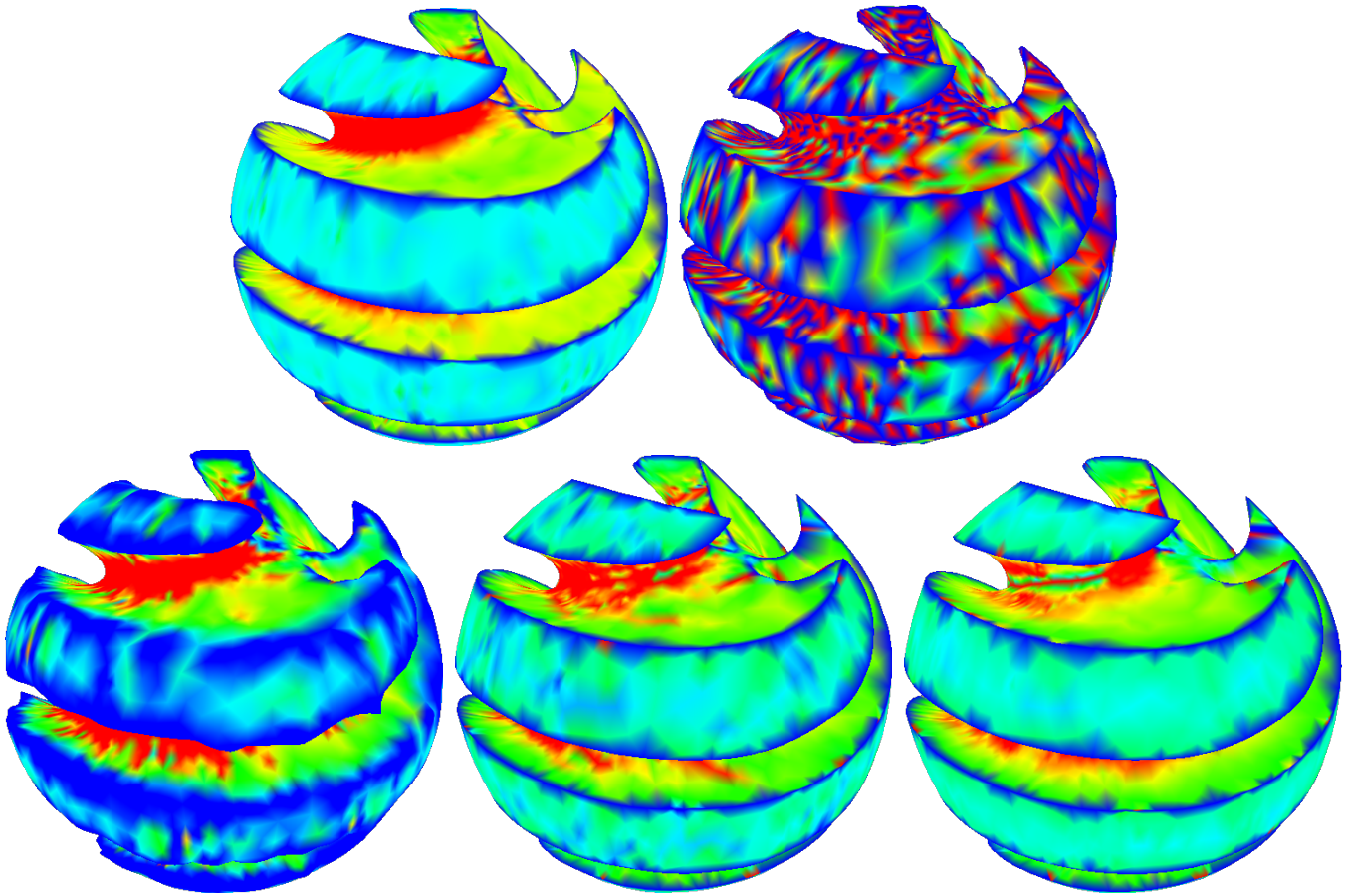


FIGURE – An example using our denoising algorithm (showing curvature mapping). First row from left to right : original mesh and noisy mesh. Second row from left to right : resulting mesh using uniform Laplacian smoothing, resulting mesh of our proposal wihout using the bilateral normal filtering step, and resulting mesh of our proposal using it.

Initial comparison

	Mean Vertex Distance	L2 Vertex Based	Mean Quadric	MSAE	L2 Normal Based	Tangential	Mean Discrete Curvature	Area Error	Volume Error
Uniform	0.010836	0.022663	0.019531	8.975630	0.059854	0.086991	0.192965	0.120947	0.029940
Ours	0.000962	0.001673	0.001288	6.572030	0.012719	0.011032	0.076498	0.005788	0.000364
Ours (*)	0.001548	0.003932	0.001643	9.372330	0.036556	0.016079	0.085570	0.009088	0.001526

Guided vs Ours

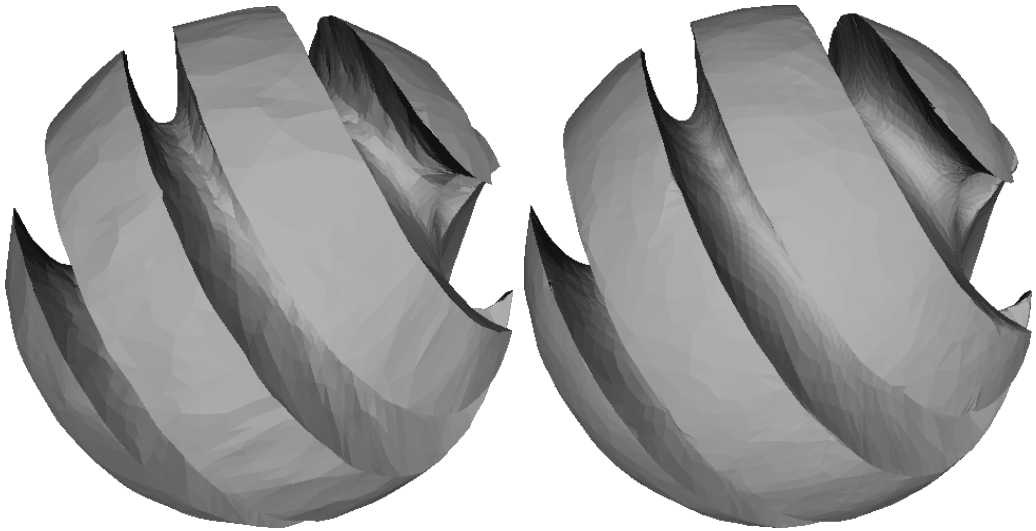


FIGURE – Results after 20 iterations of vertex updating using estimated normals. Left : Guided normals using (ZDZBL15). Right : Average normal weighted by patch membership function.

Guided vs Ours

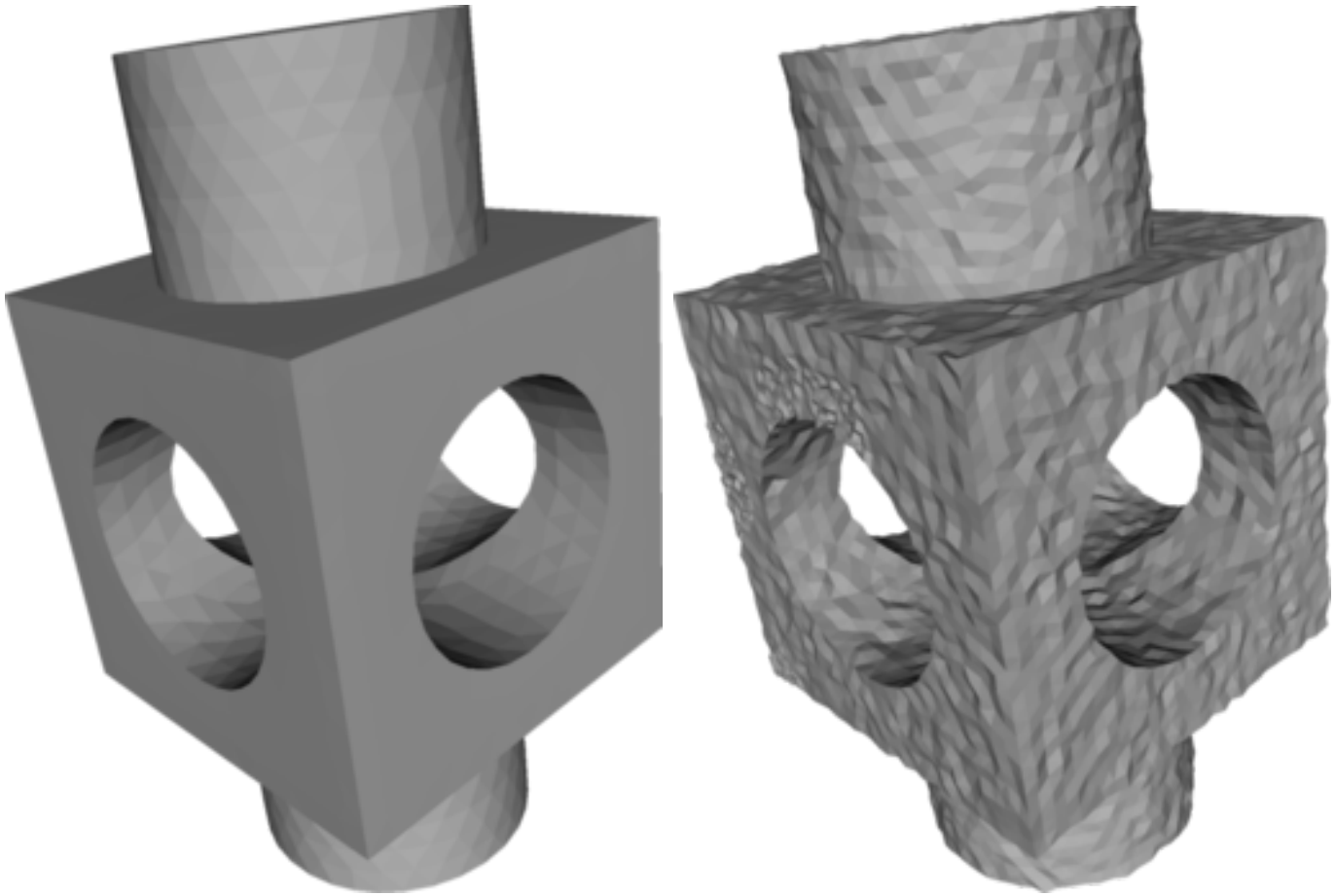
	MSAE	L2 Normal Based
Guidance normals (ZDZBL15)	8.34329	0.01603
Our normals	5.90240	0.00816

Execution time

Step	time	percentage
Distance computation	1.379s	7%
Adaptive patches	15.676s	84%
Normal filtering	1.524s	8%
Vertex updating	0.023s	1%

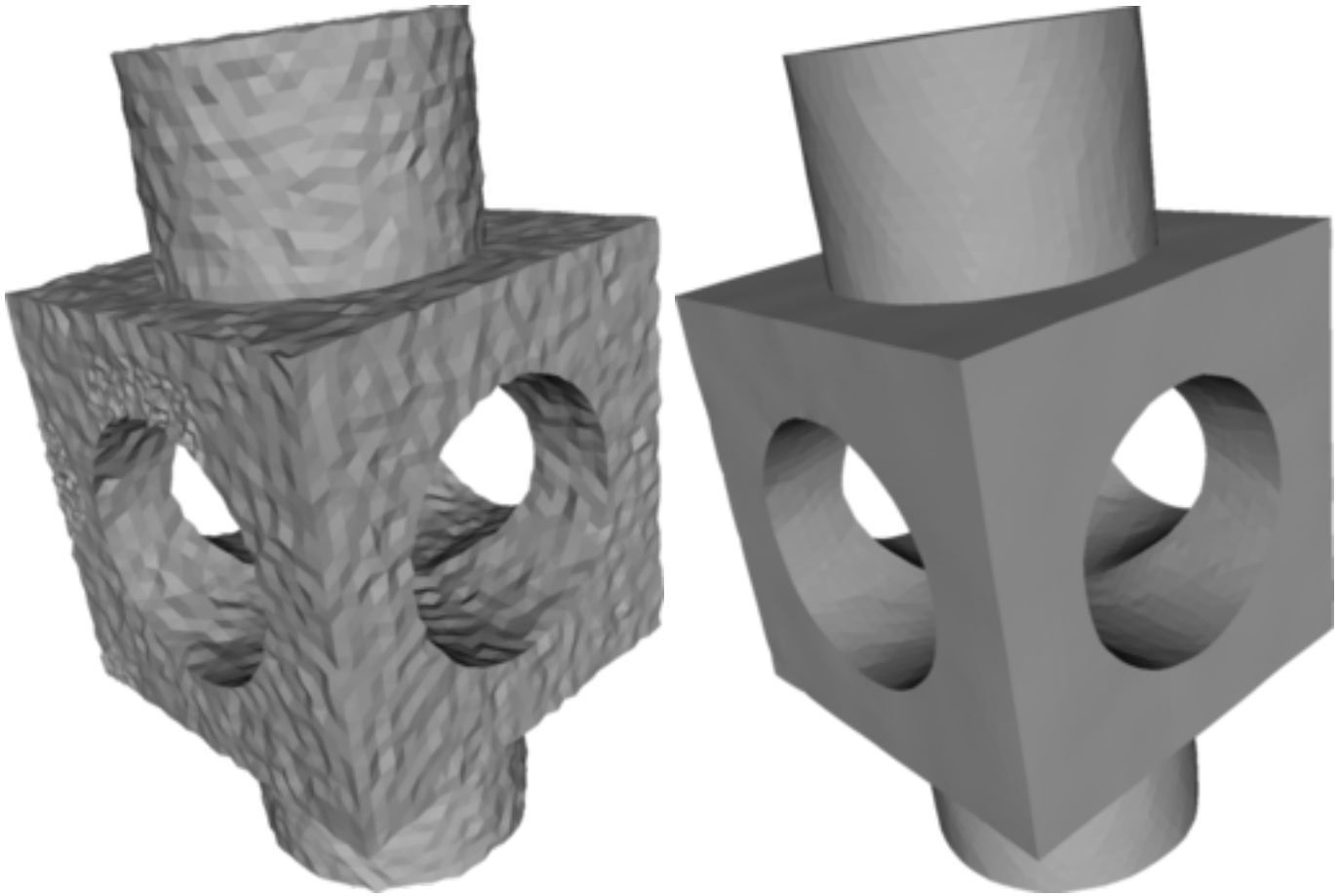
Block mesh

original - noisy



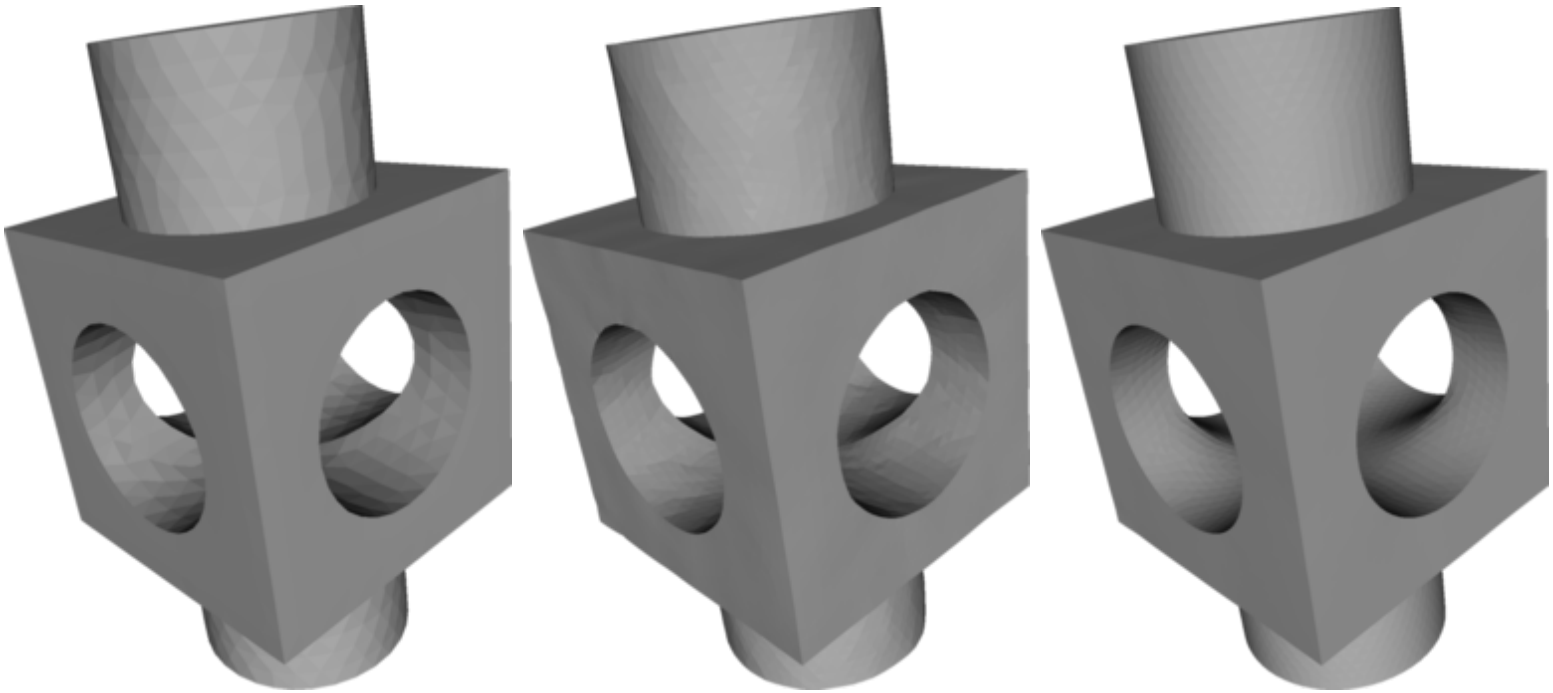
Block mesh

noisy - ours



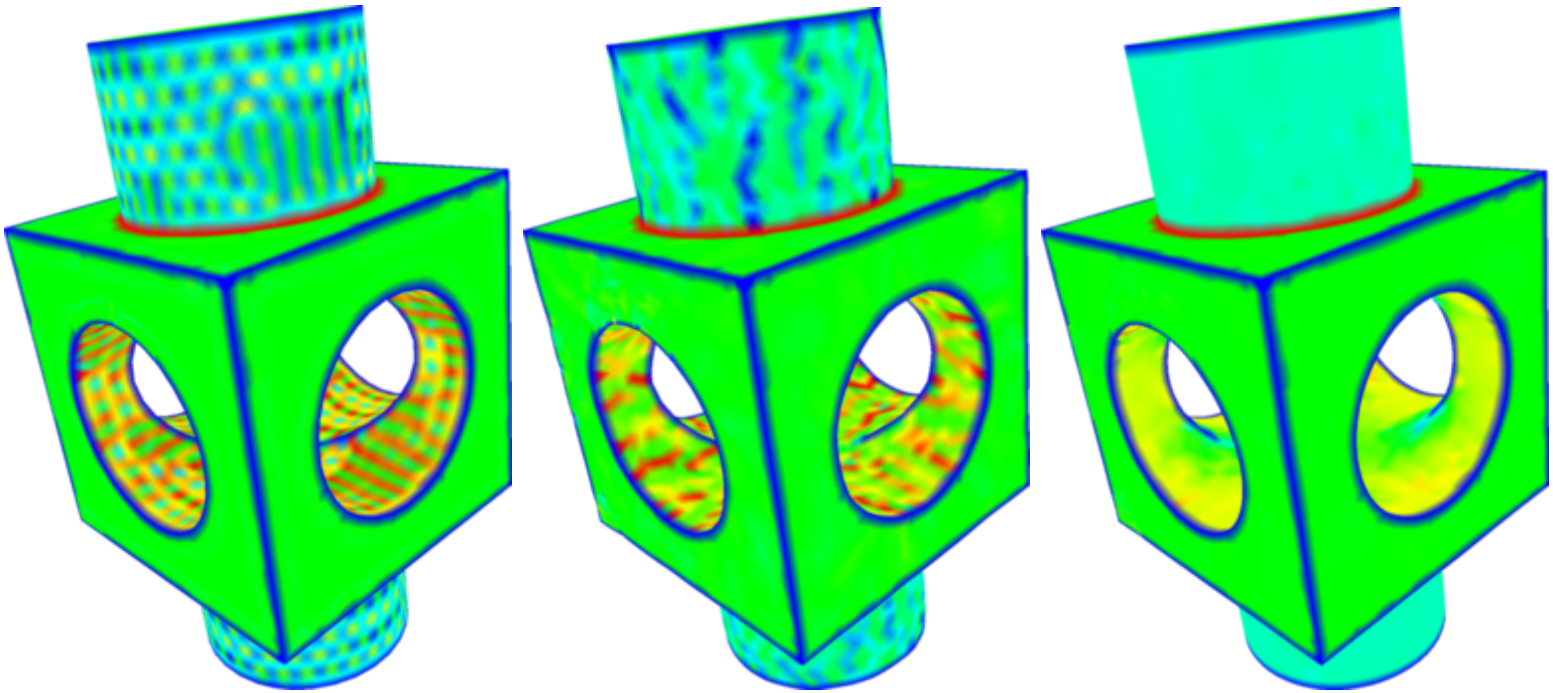
Block mesh

original - ours - bilateralNormal



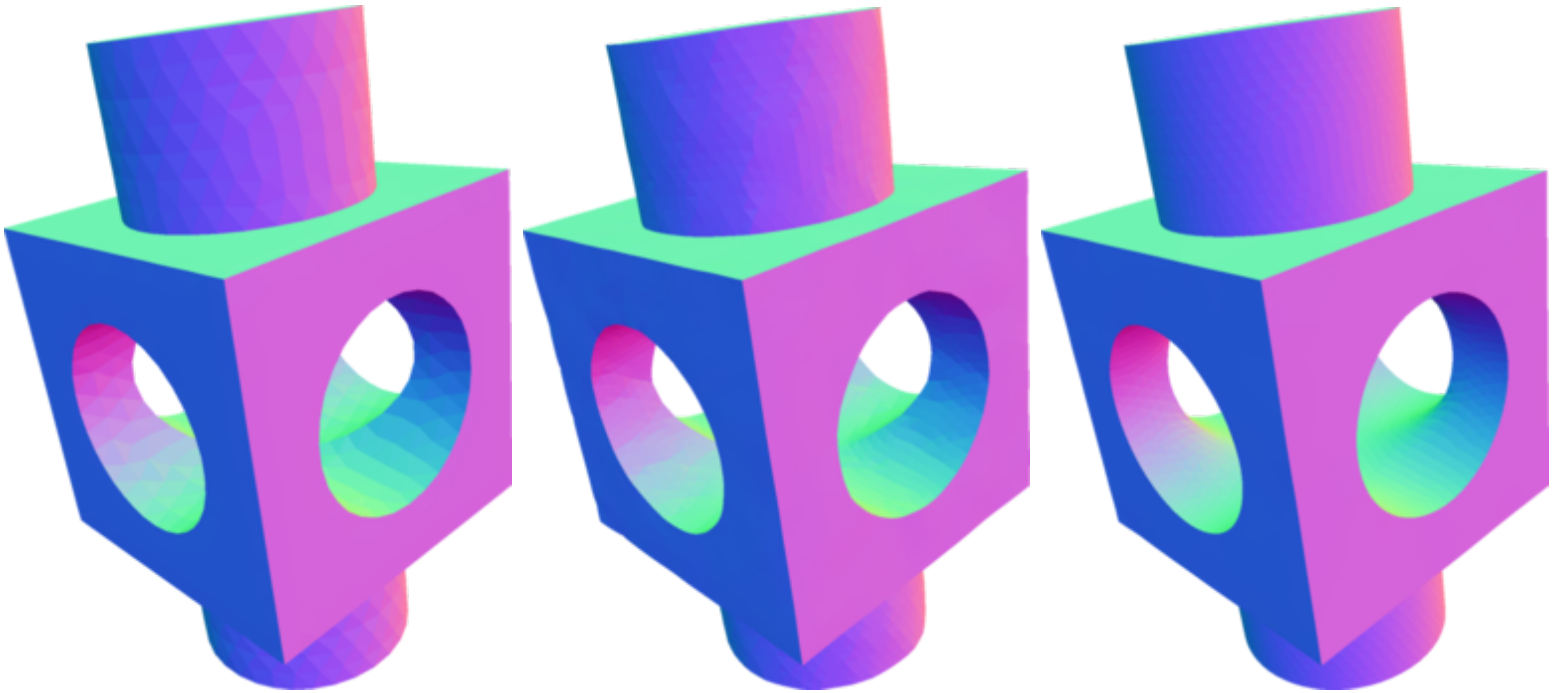
Block mesh

original - ours - bilateralNormal



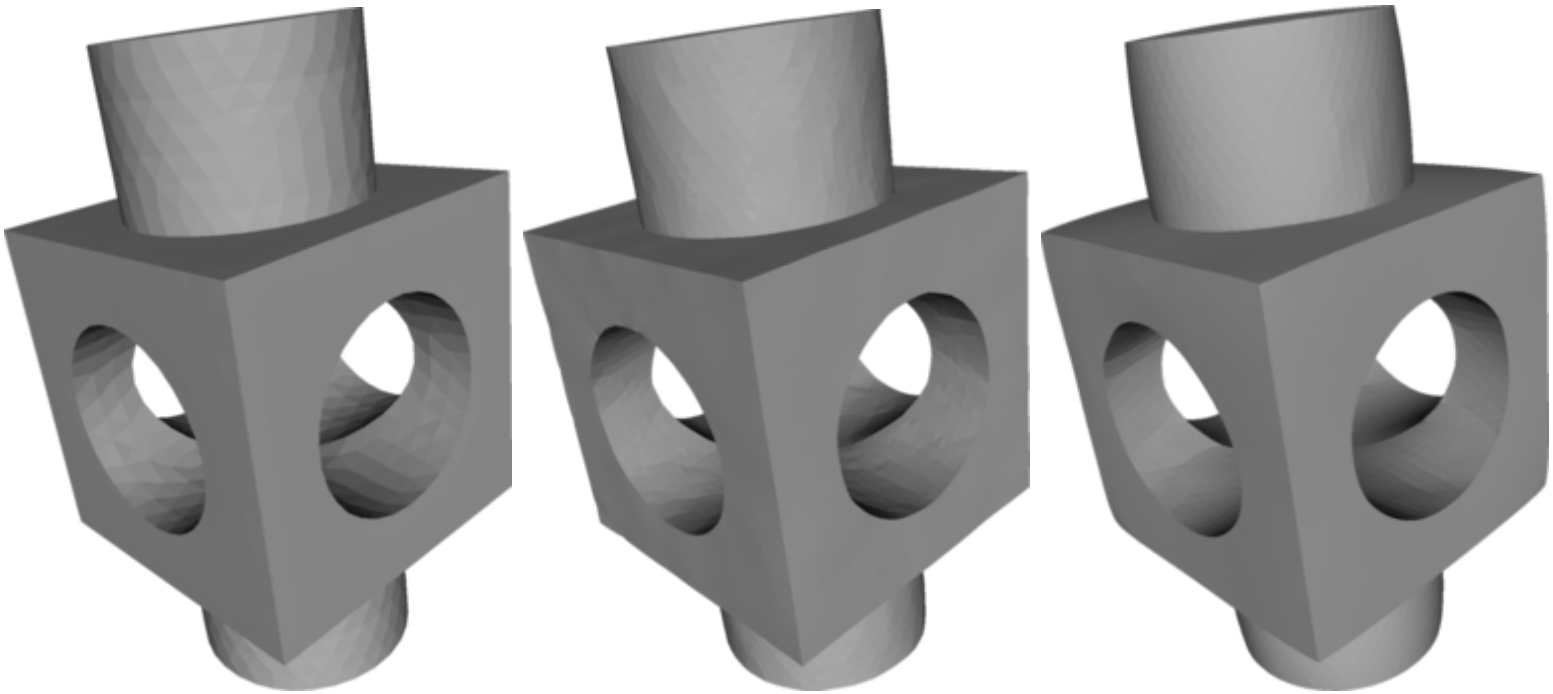
Block mesh

original - ours - bilateralNormal



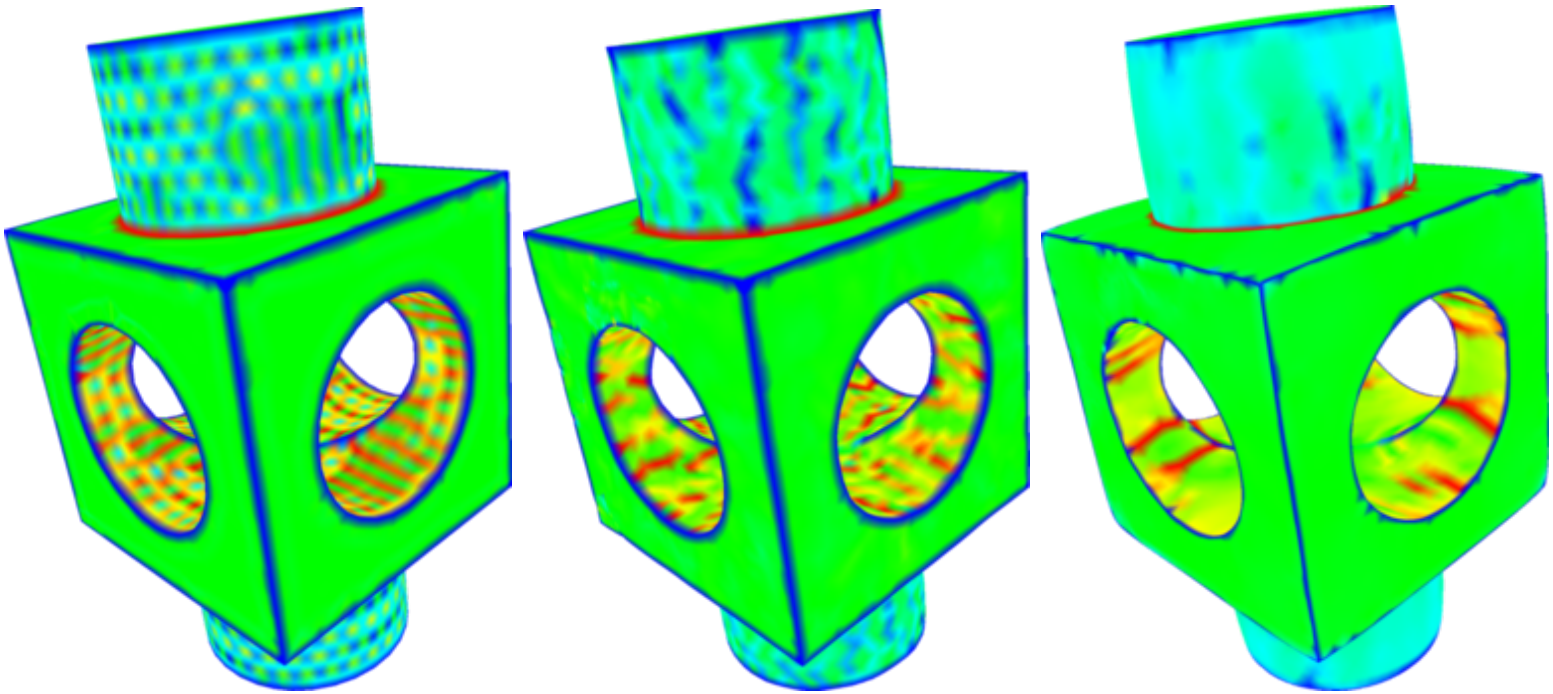
Block mesh

original - ours - L0



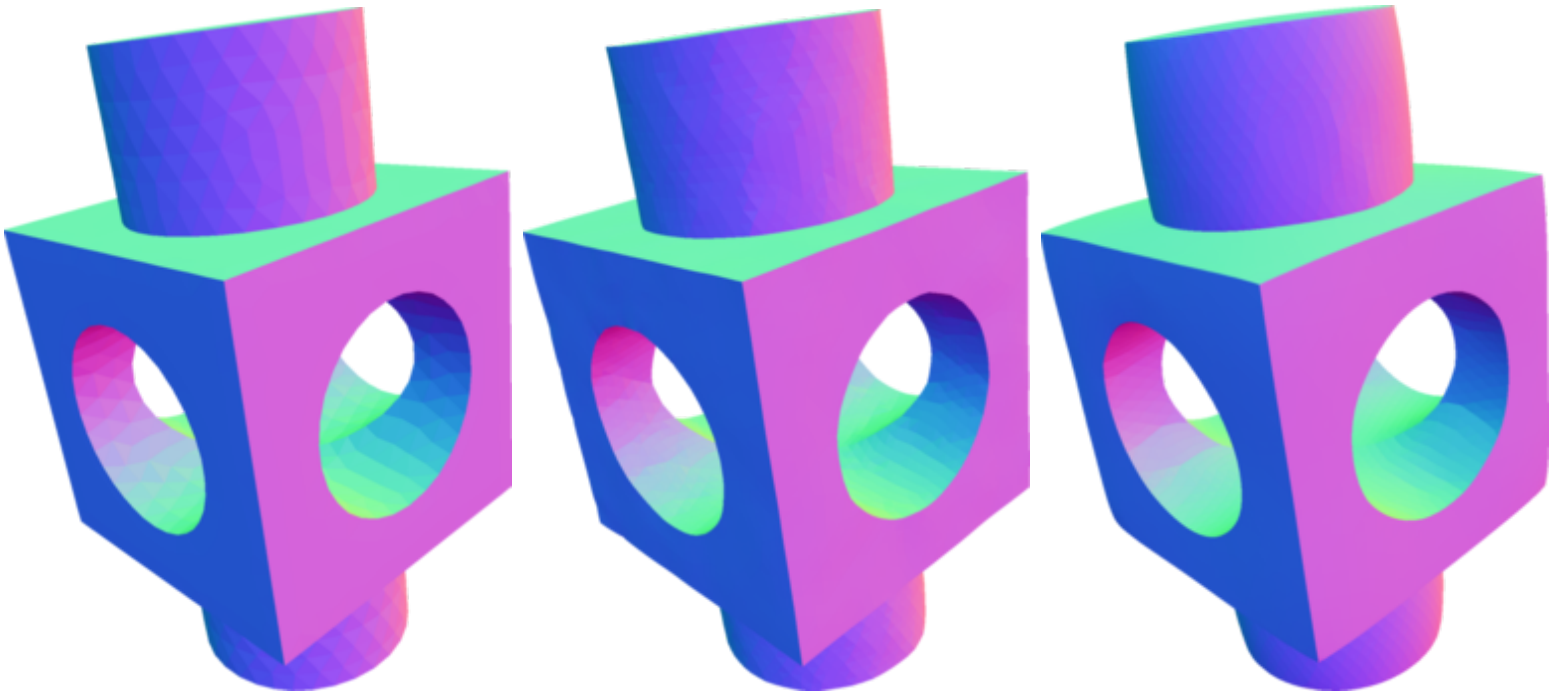
Block mesh

original - ours - L0



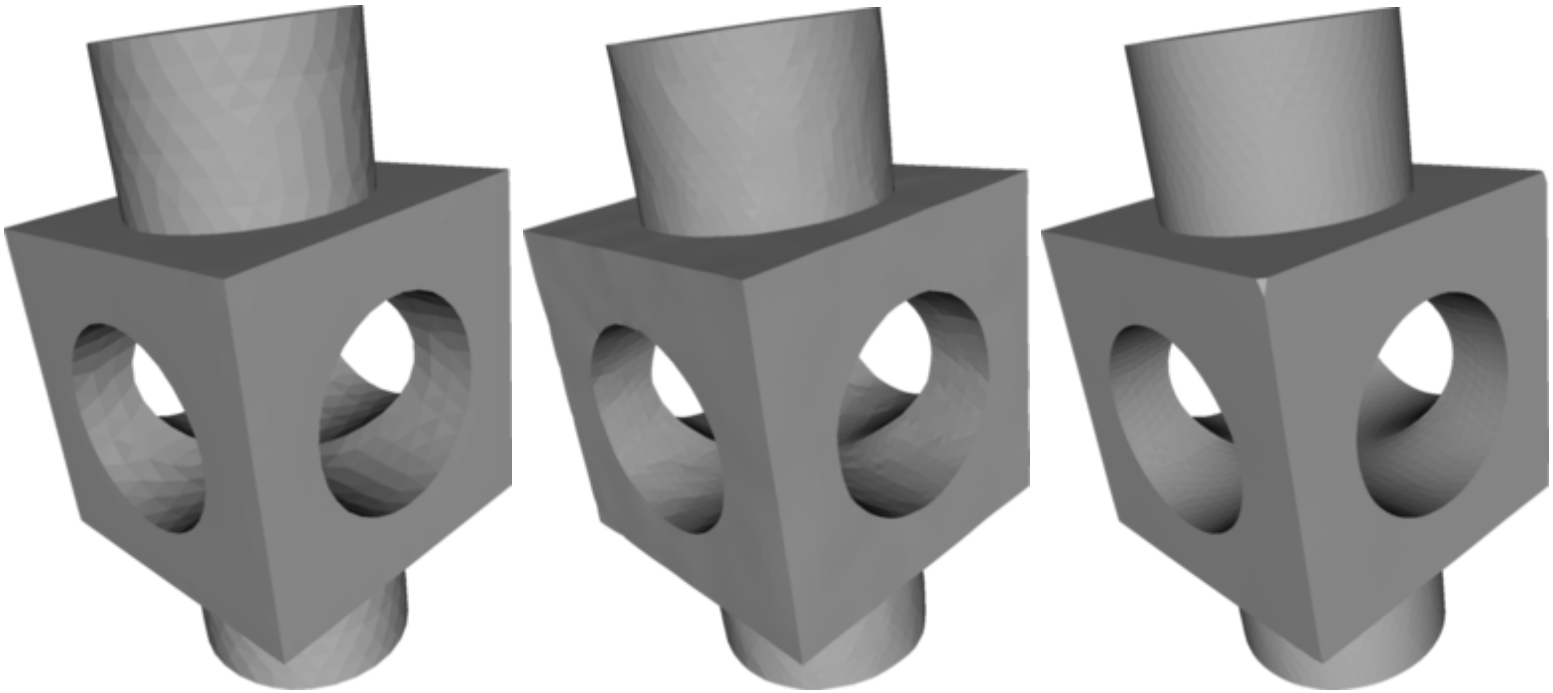
Block mesh

original - ours - L0



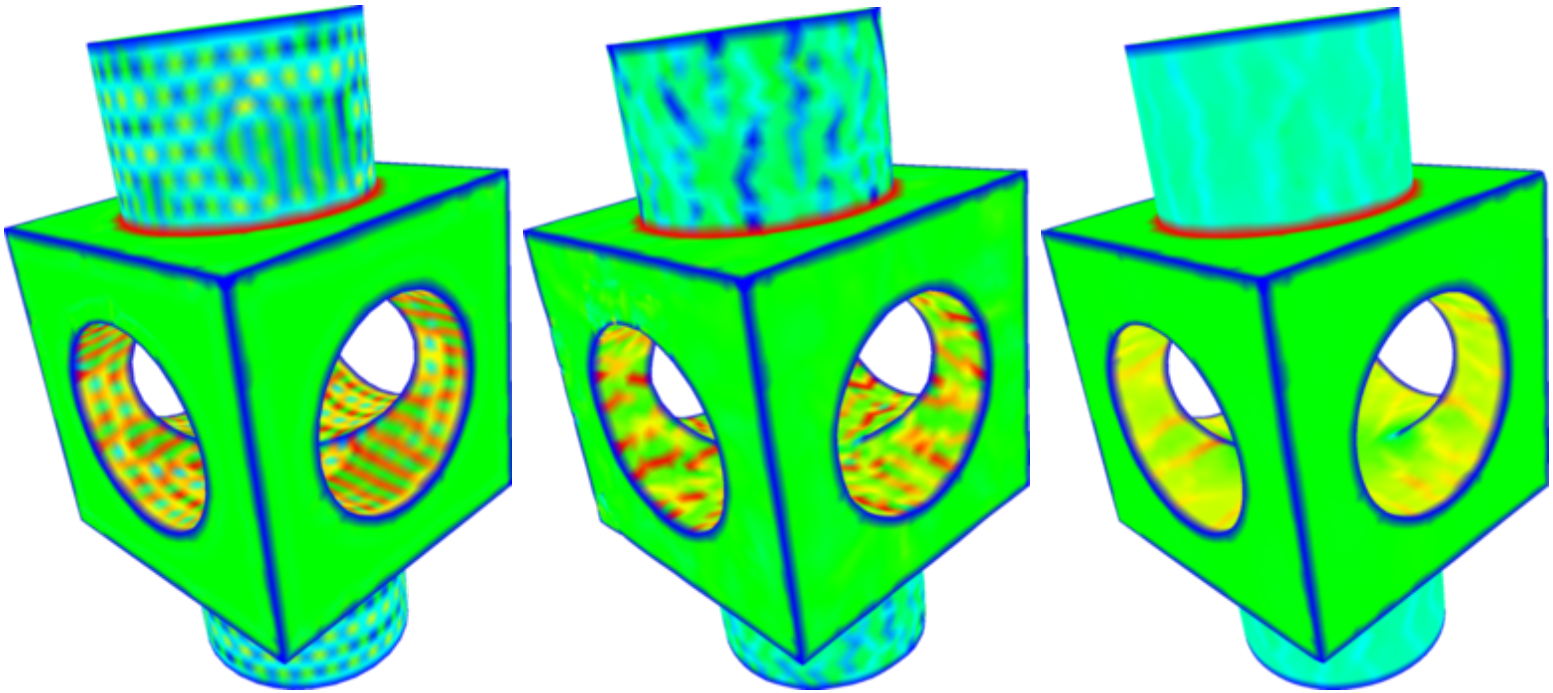
Block mesh

original - ours - guided



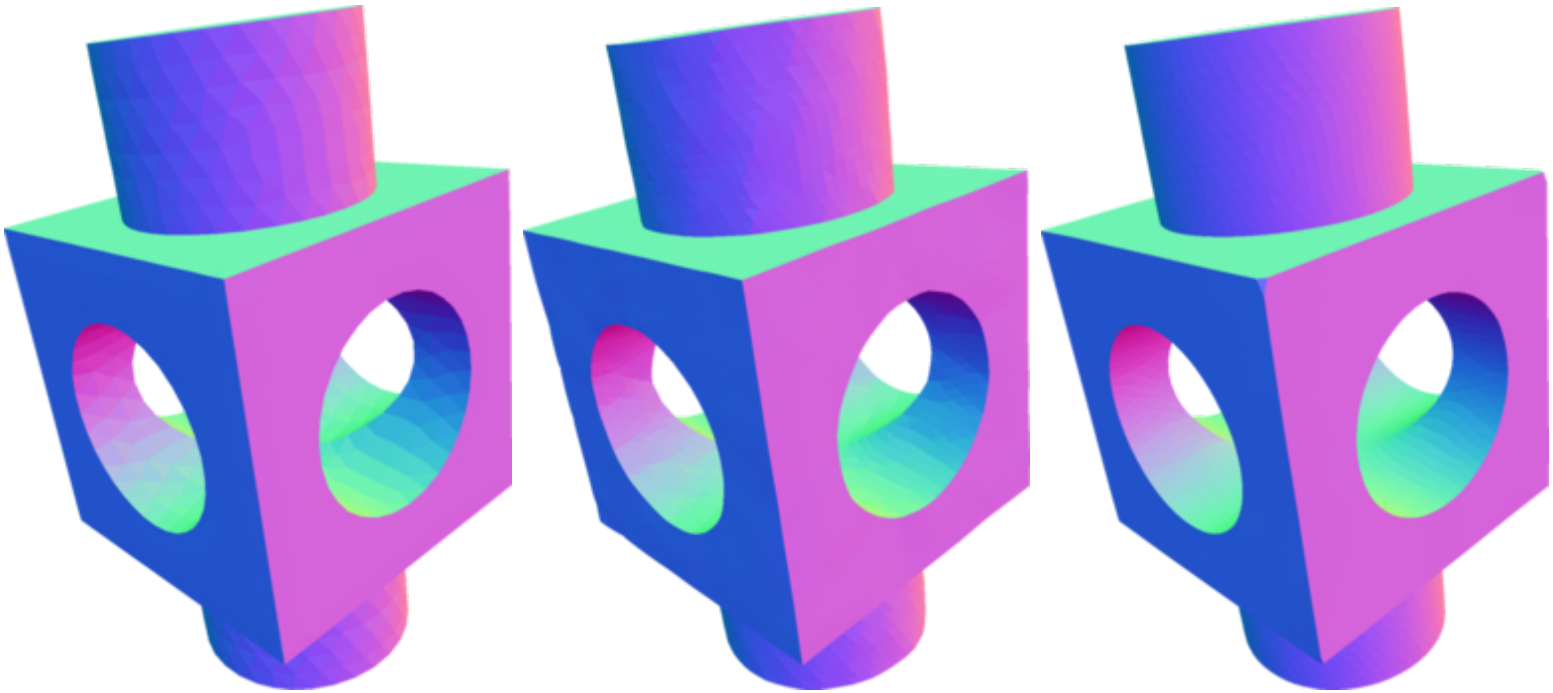
Block mesh

original - ours - guided



Block mesh

original - ours - guided



Block mesh

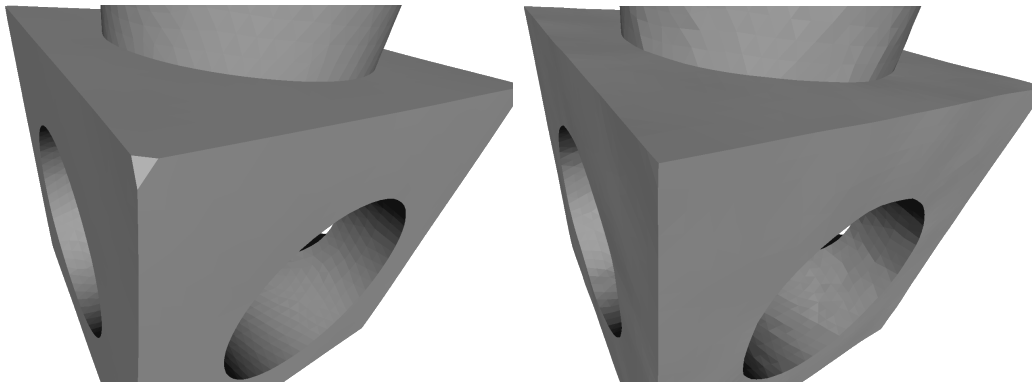


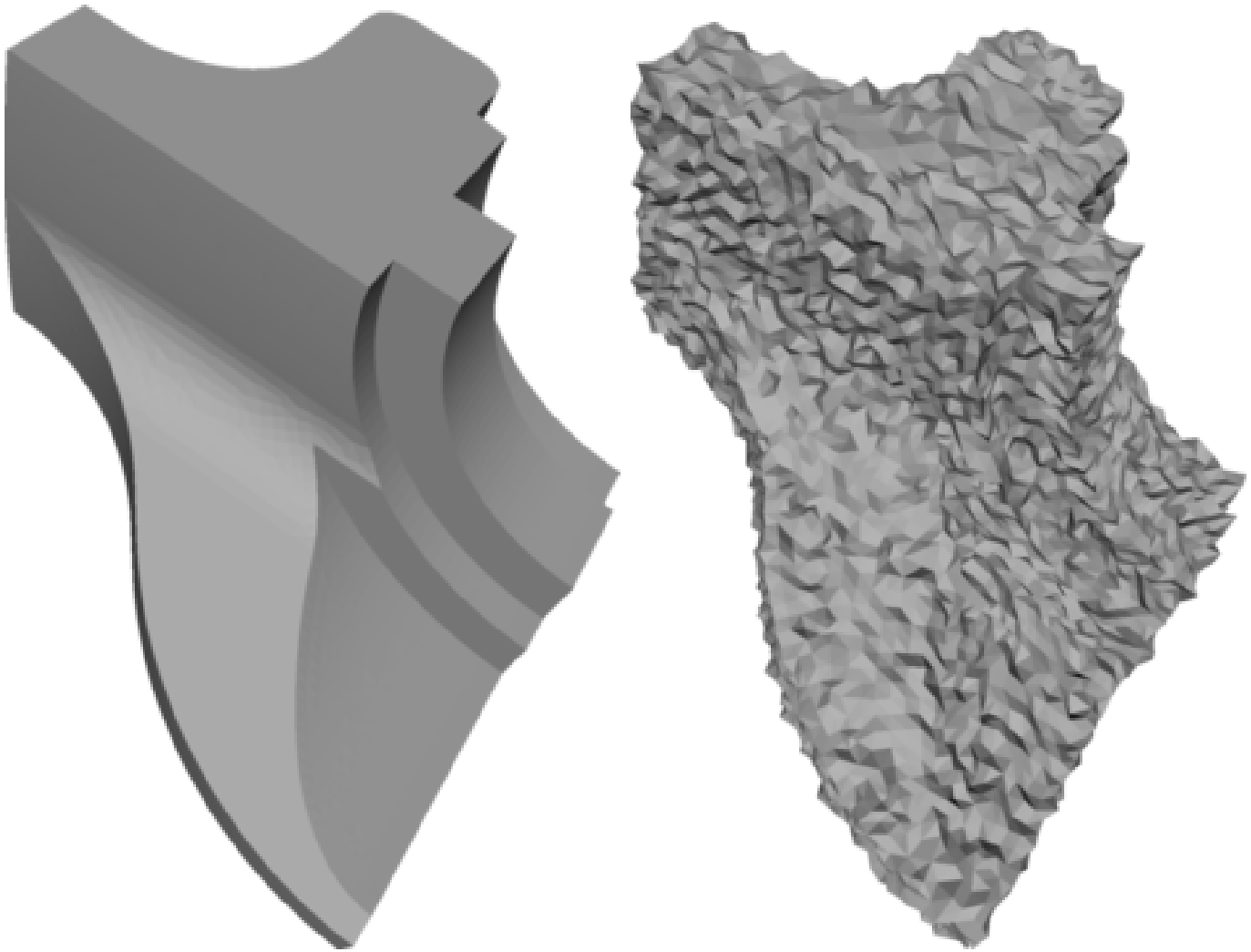
FIGURE – Wrong normal direction in the corner of the resulting mesh using the Guided mesh normal filtering. Left : resulting mesh using guided filter. Right : our result.

Block mesh

	Mean Vertex Distance	L2 Vertex Based	Mean Quadric	MSAE	L2 Normal Based	Tangential	Mean Discrete Curvature	Area Error	Volume Error
(FDC03)	0.030743	0.052414	0.042734	9.477970	0.035246	0.043968	0.109589	0.103718	0.045399
(JDD03)	0.002910	0.004648	0.004019	5.537650	0.011929	0.004498	0.074113	0.011371	0.009549
(SRML07)	0.002159	0.011088	0.002844	2.172140	0.004683	0.002574	0.029590	0.003375	0.000300
(ZFAT11)	0.001960	0.010192	0.002583	2.088220	0.004298	0.002258	0.029038	0.003223	0.000210
(HS13)	0.012640	0.020213	0.019013	6.127930	0.024389	0.021592	0.109556	0.040437	0.024038
(ZDZBL15)	0.001249	0.004180	0.001698	1.845830	0.002676	0.001282	0.028262	0.002161	0.000849
Ours	0.000565	0.001280	0.000743	1.906880	0.002001	0.000741	0.029958	0.000186	0.000433

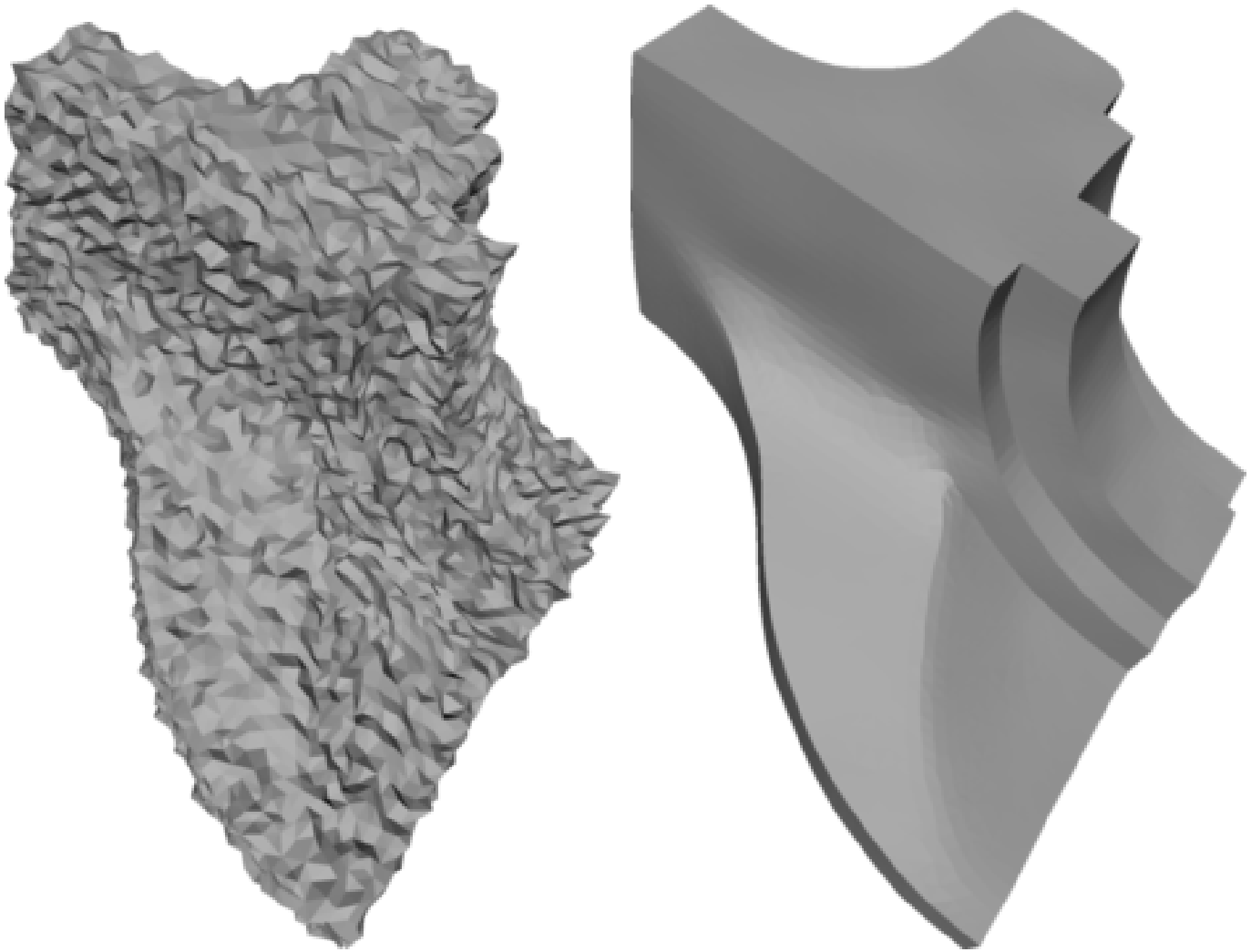
Fandisk mesh

original - noisy



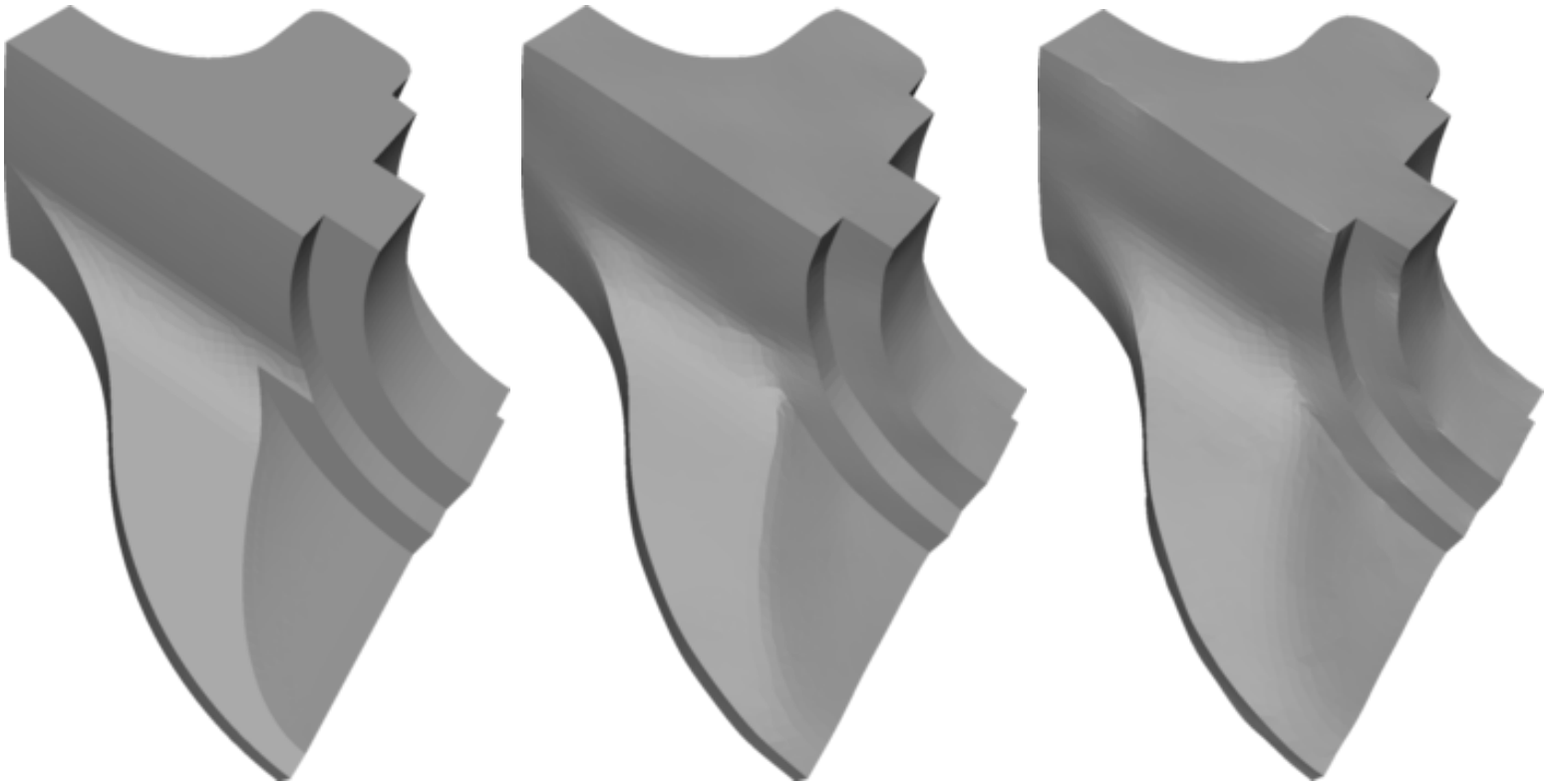
Fandisk mesh

noisy - ours



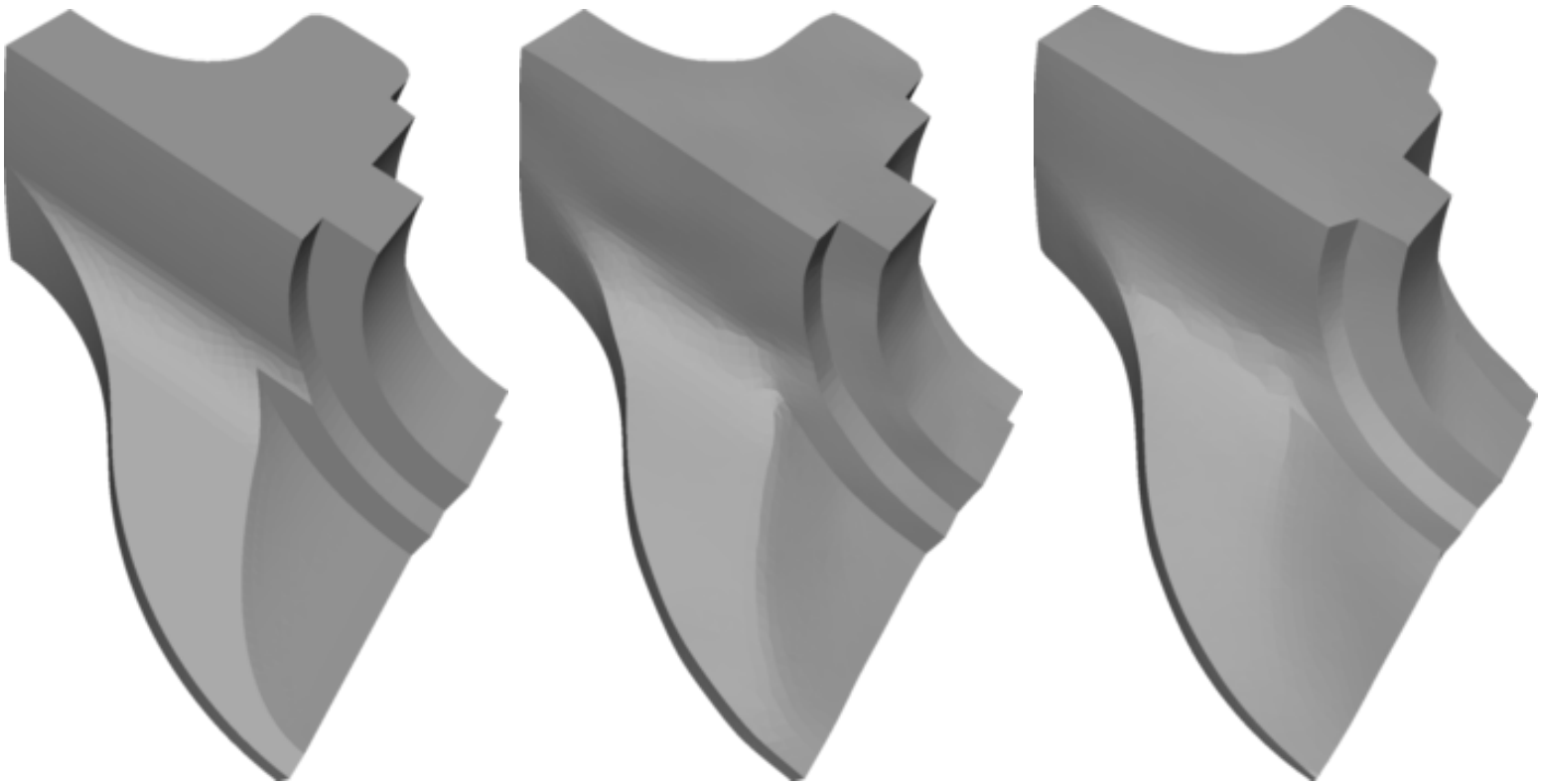
Fandisk mesh

original - ours - bilateralNormal



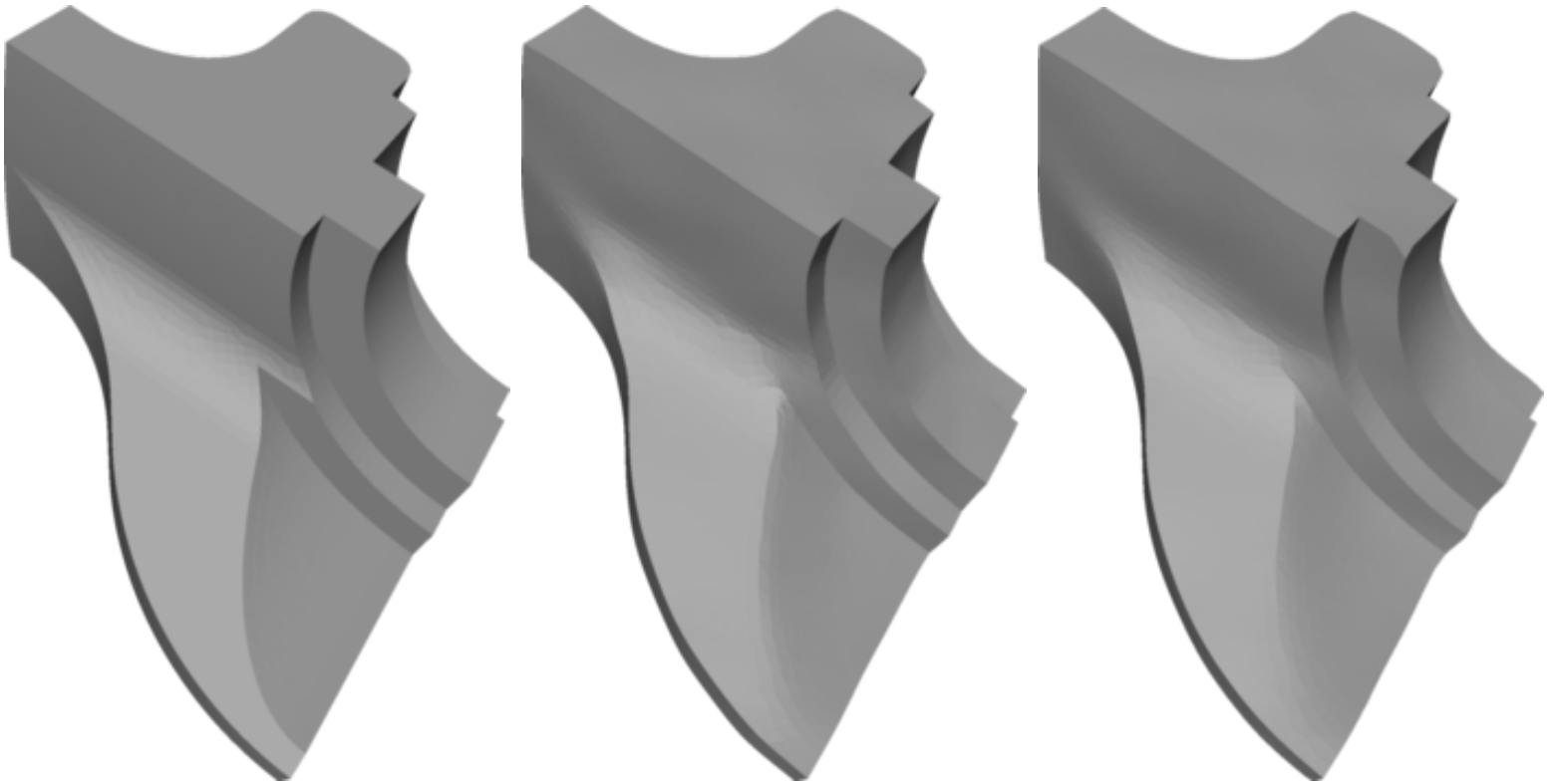
Fandisk mesh

original - ours - L0



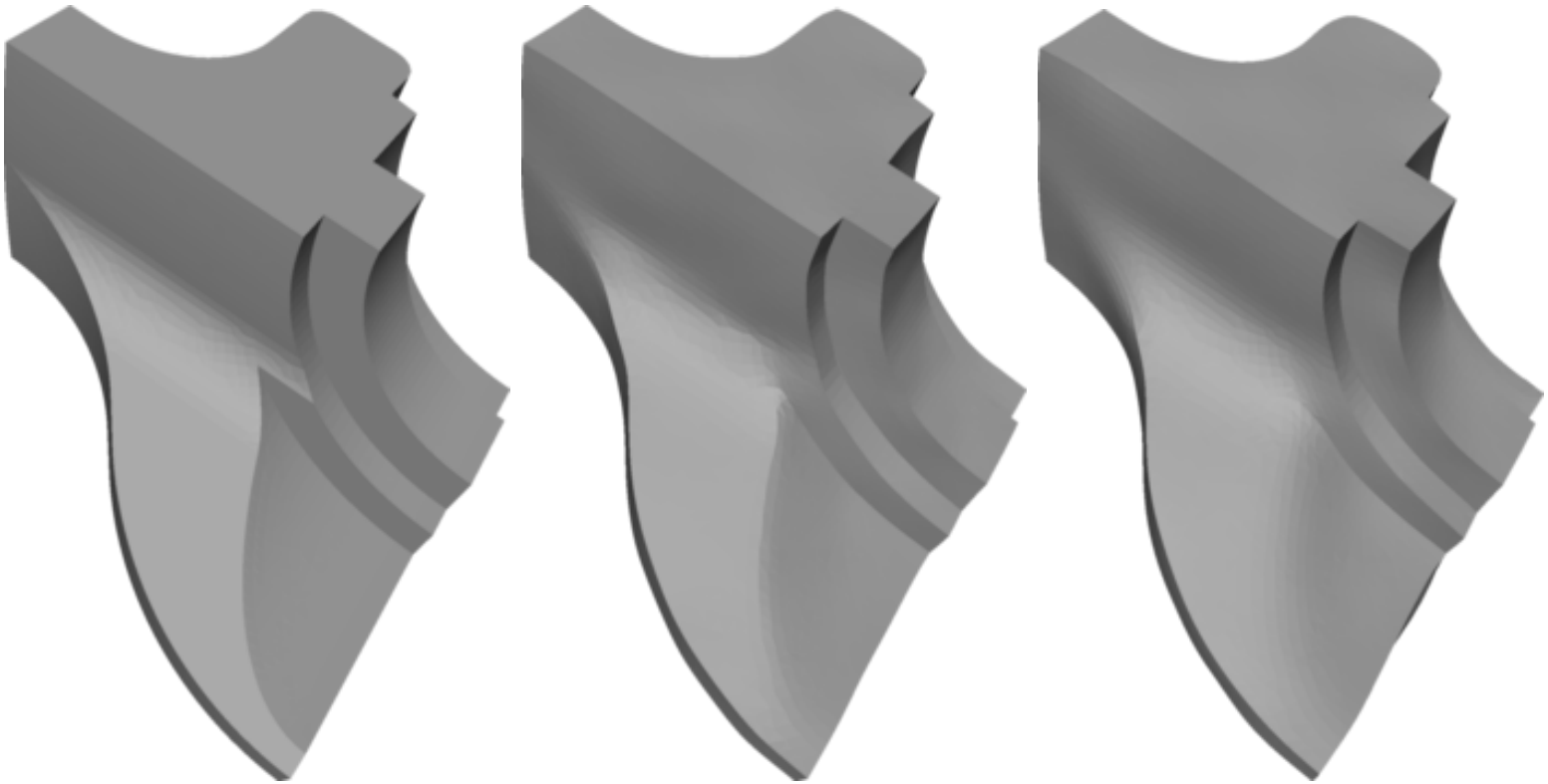
Fandisk mesh

original - ours - guided



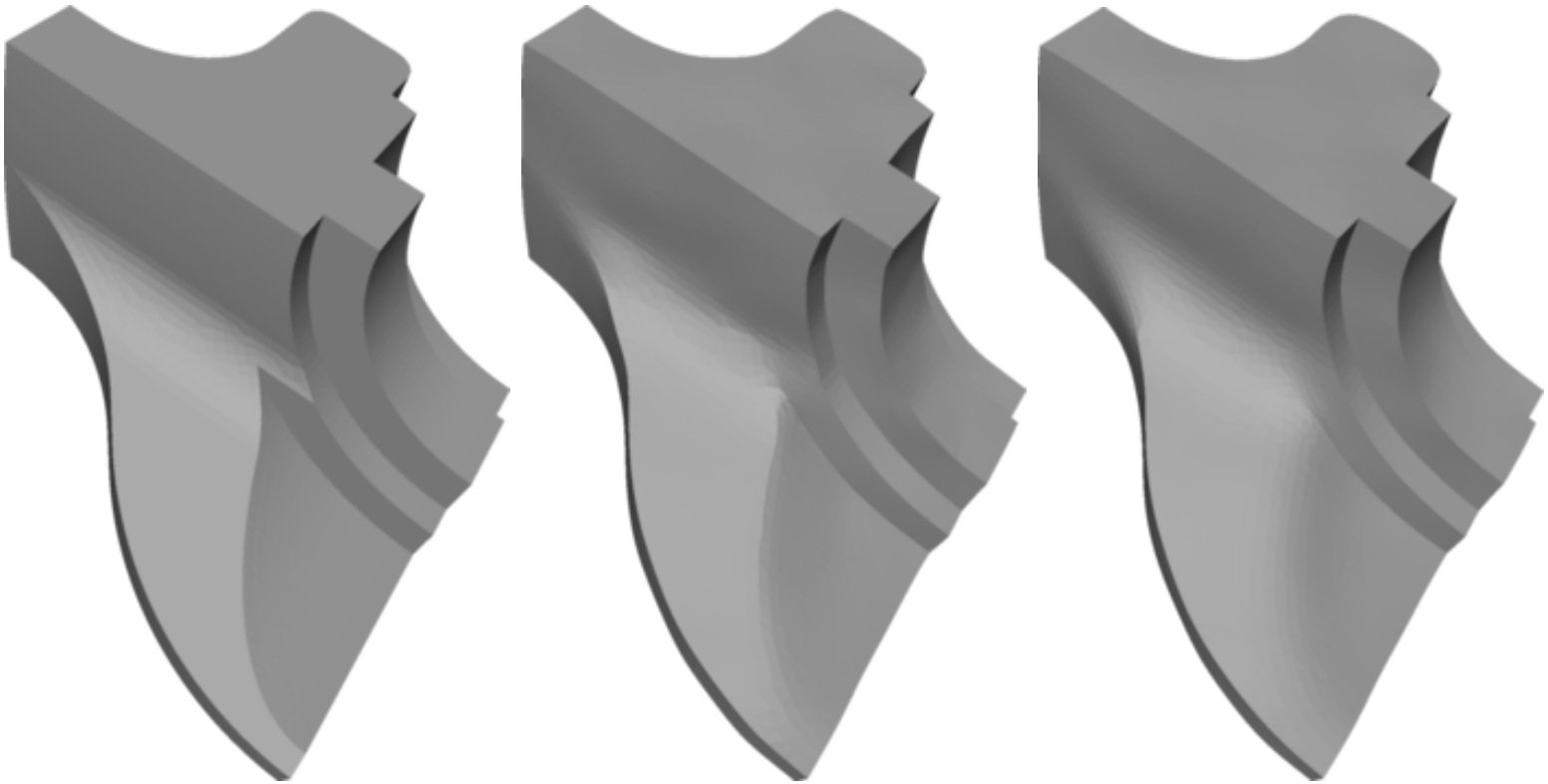
Fandisk mesh

original - ours - nvt



Fandisk mesh

original - ours - robust

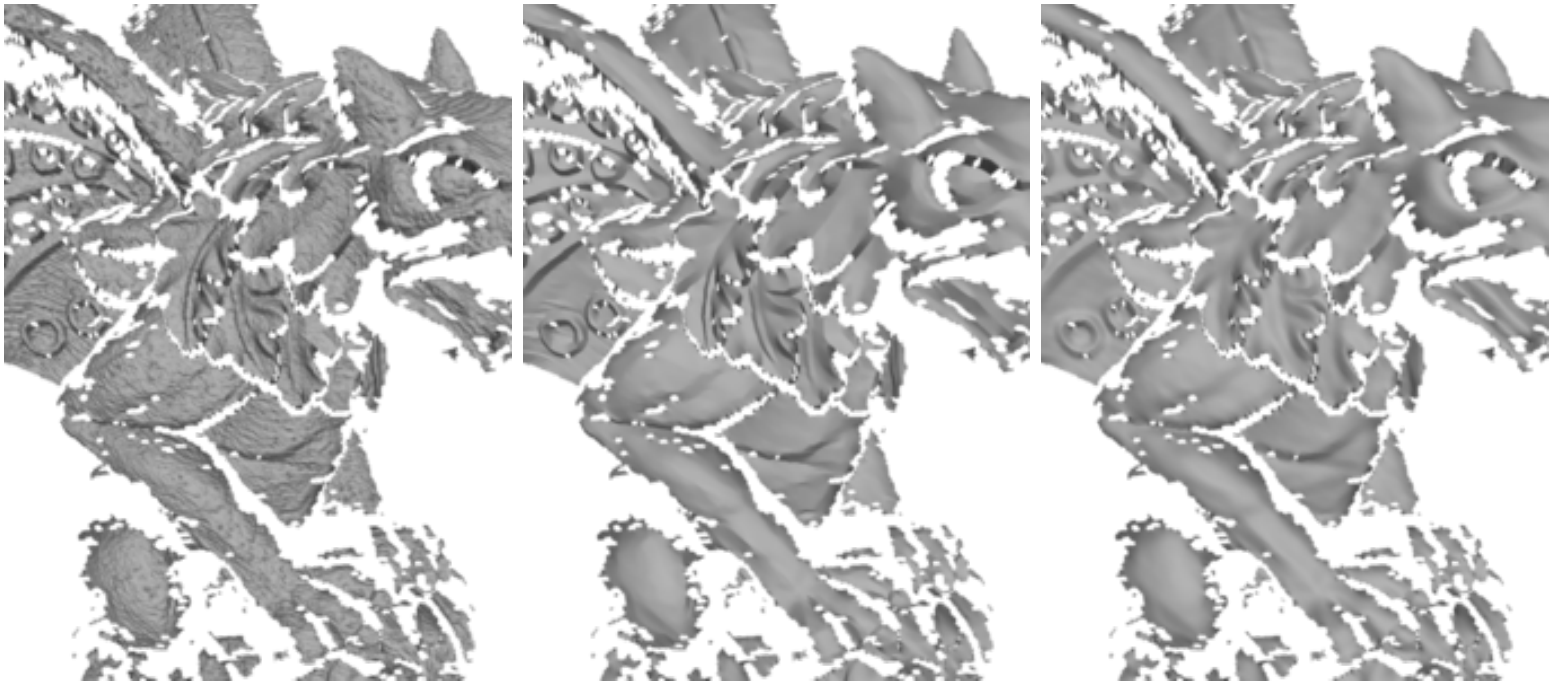


Fandisk mesh

	Mean Vertex Distance	L2 Vertex Based	Mean Quadric	MSAE	L2 Normal Based	Tangential	Mean Discrete Curvature	Area Error	Volume Error
(FDC03)	0.000330	0.000671	0.000010	9.821170	0.041399	0.038493	0.598858	0.073705	0.016398
(JDD03)	0.000238	0.000433	0.000007	9.984250	0.043886	0.016946	0.780315	0.007715	0.008510
(SRML07)	0.000128	0.000364	0.000004	4.188150	0.014546	0.017077	0.138360	0.016164	0.000733
(ZFAT11)	0.000081	0.000213	0.000003	3.377610	0.008664	0.011561	0.131381	0.012807	0.001007
(HS13)	0.000355	0.000630	0.000011	6.979130	0.028042	0.029579	0.197488	0.056925	0.017092
(ZDZBL15)	0.000046	0.000117	0.000002	2.627640	0.007719	0.006241	0.113402	0.007576	0.000049
(YRP16)	0.000072	0.000191	0.000002	3.289700	0.009118	0.009494	0.122689	0.008392	0.001656
(YRP17)	0.000066	0.000185	0.000002	2.493830	0.006311	0.004888	0.097299	0.010766	0.000840
Ours	0.000048	0.000156	0.000001	2.351650	0.005791	0.004378	0.116578	0.004723	0.000813

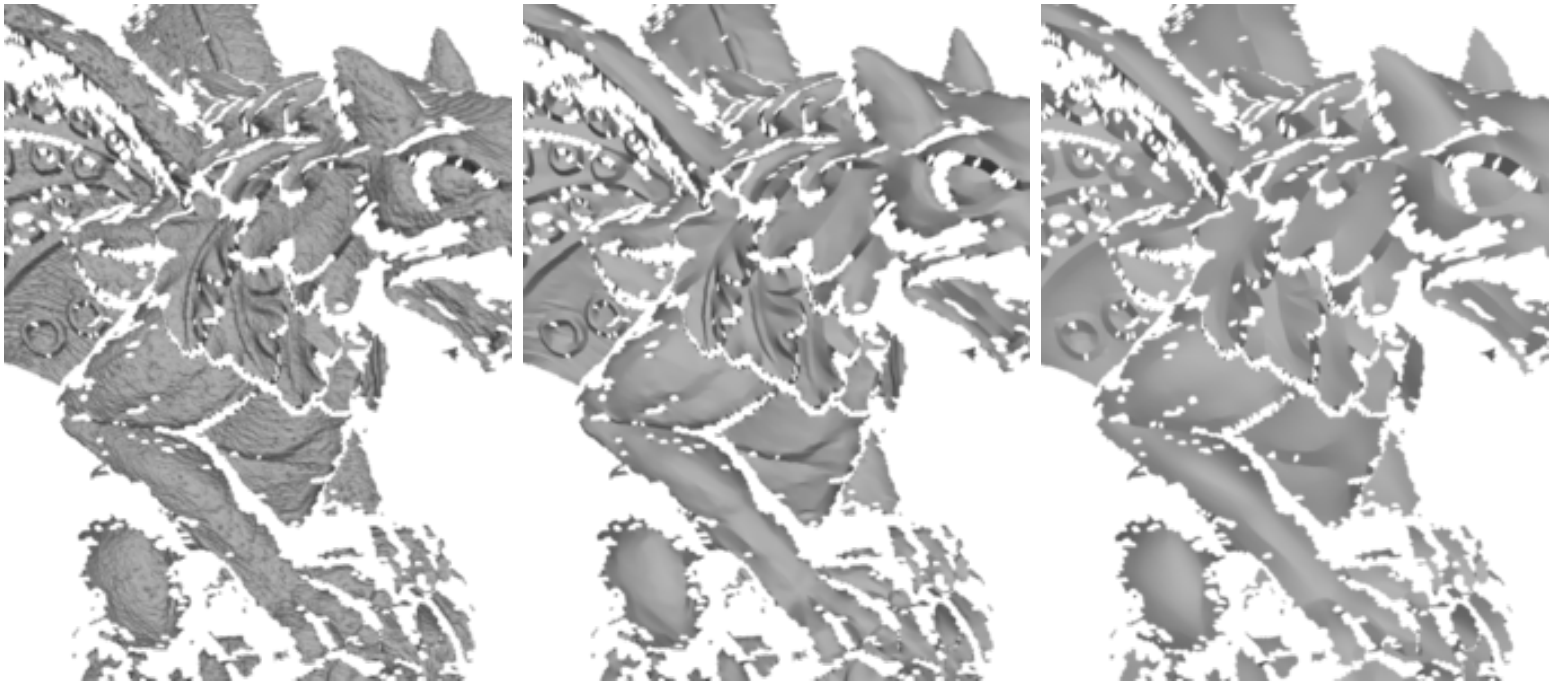
Gargoyle mesh

noisy - ours - bilateralNormal



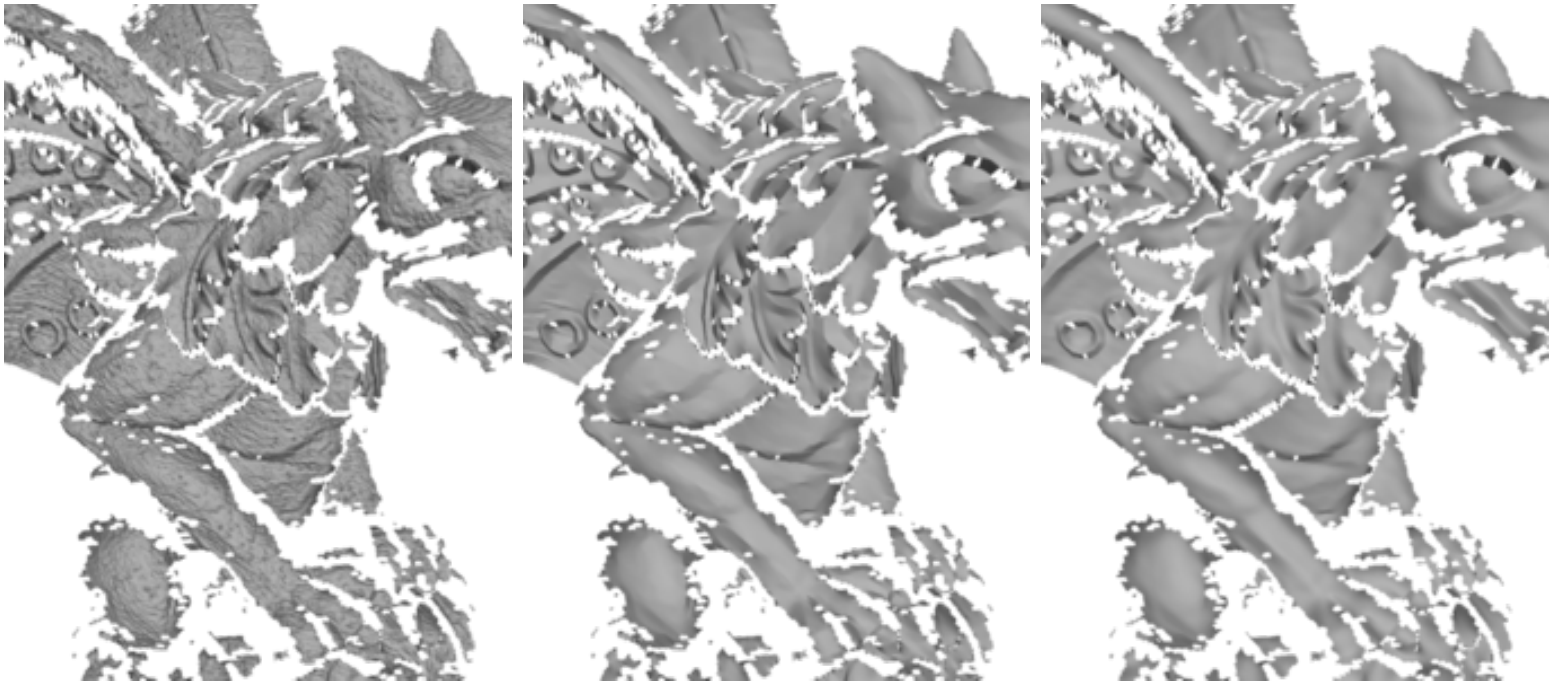
Gargoyle mesh

noisy - ours - L0



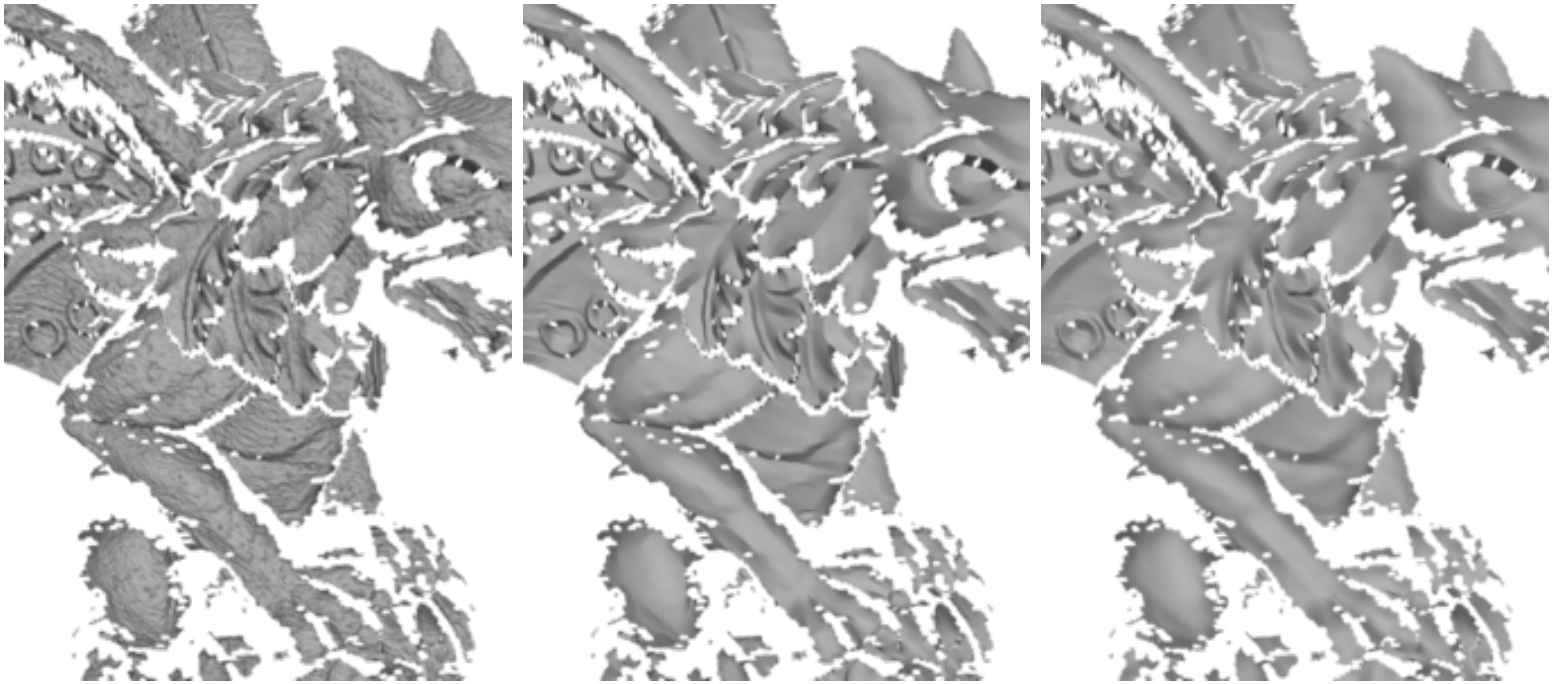
Gargoyle mesh

noisy - ours - guided



Gargoyle mesh

noisy - ours - nvt



Gargoyle mesh

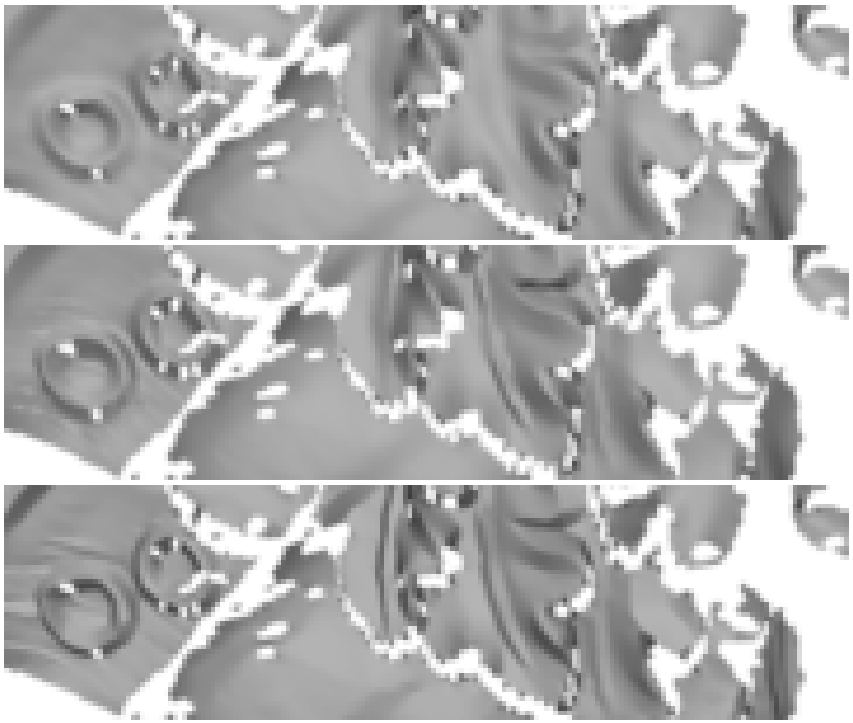
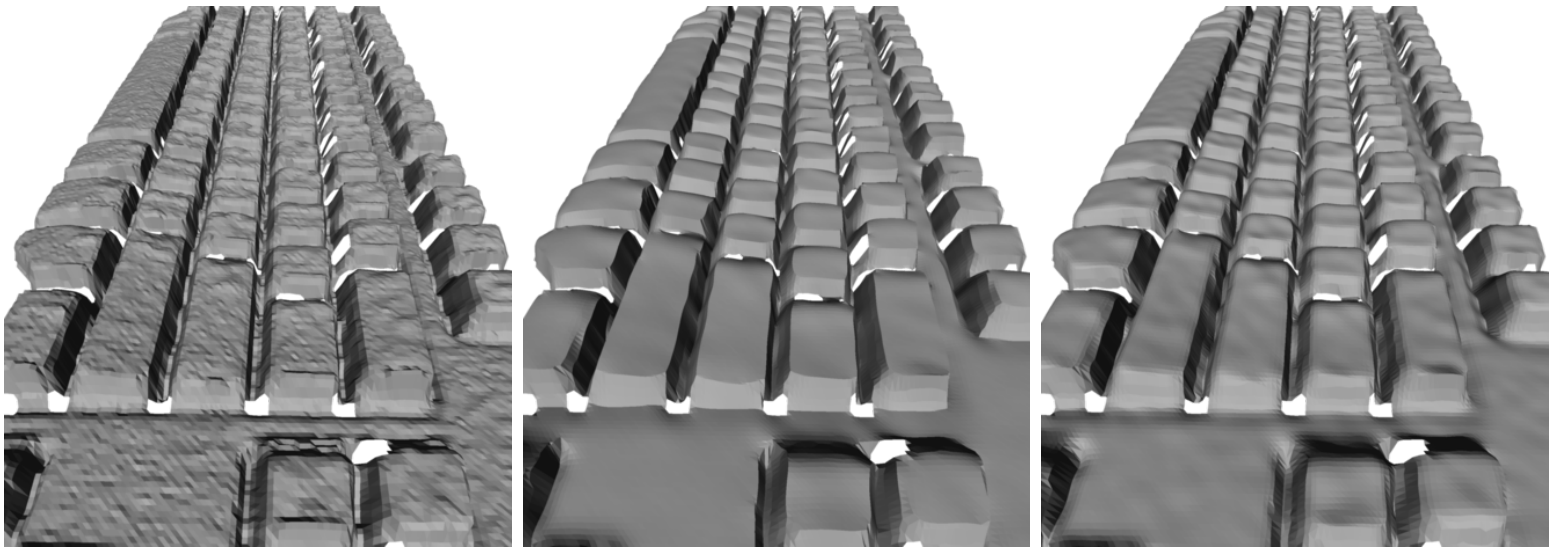


FIGURE – Gargoyle mesh details preserved with our denoising algorithm. First row : bilateralNormal. Second row : NVT. Third row : ours.

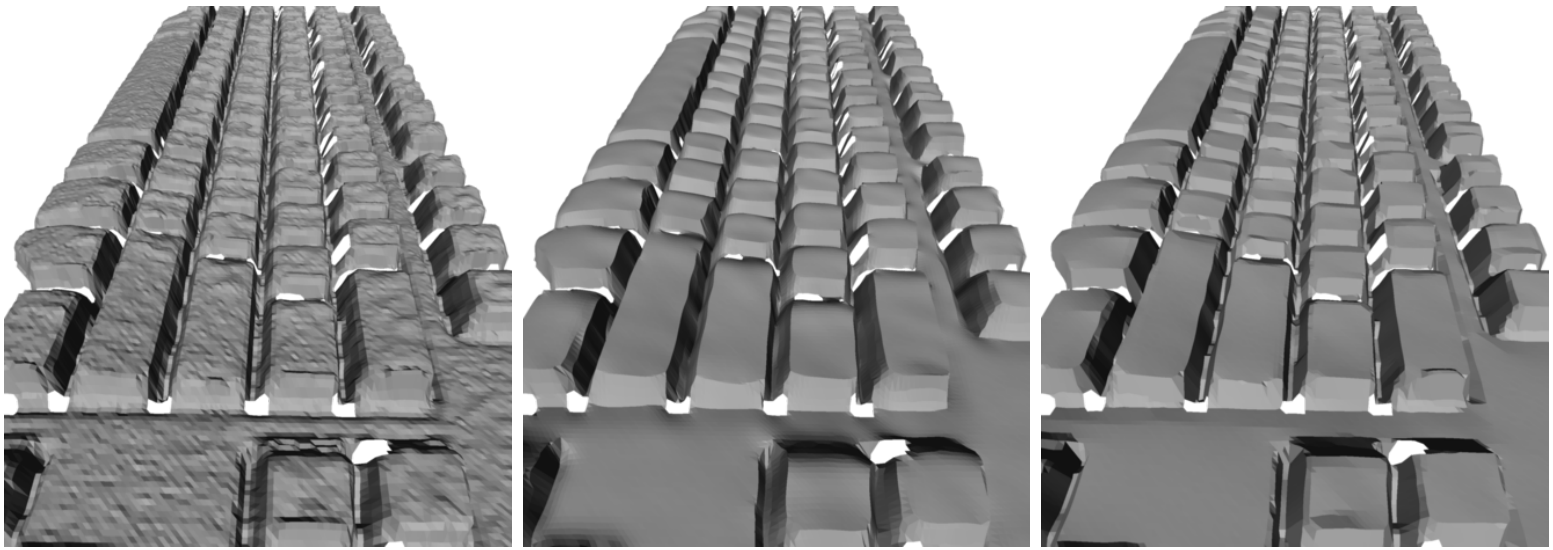
Keyboard mesh

noisy - ours - bilateralNormal



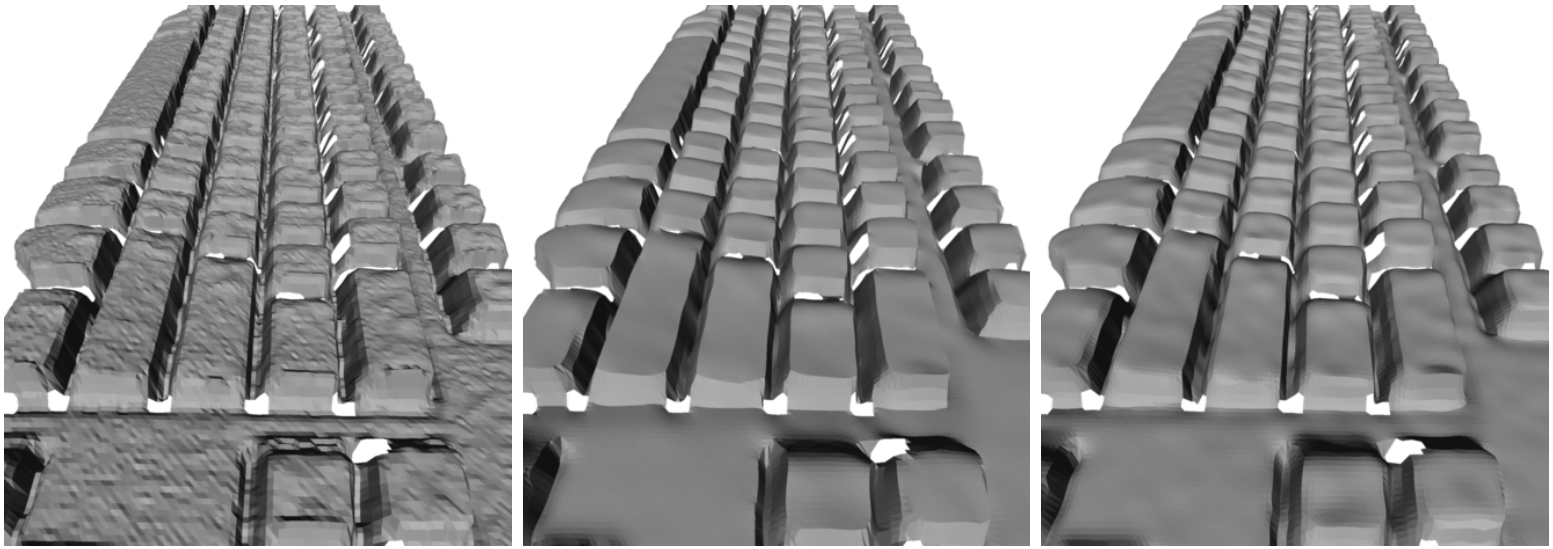
Keyboard mesh

noisy - ours - L0

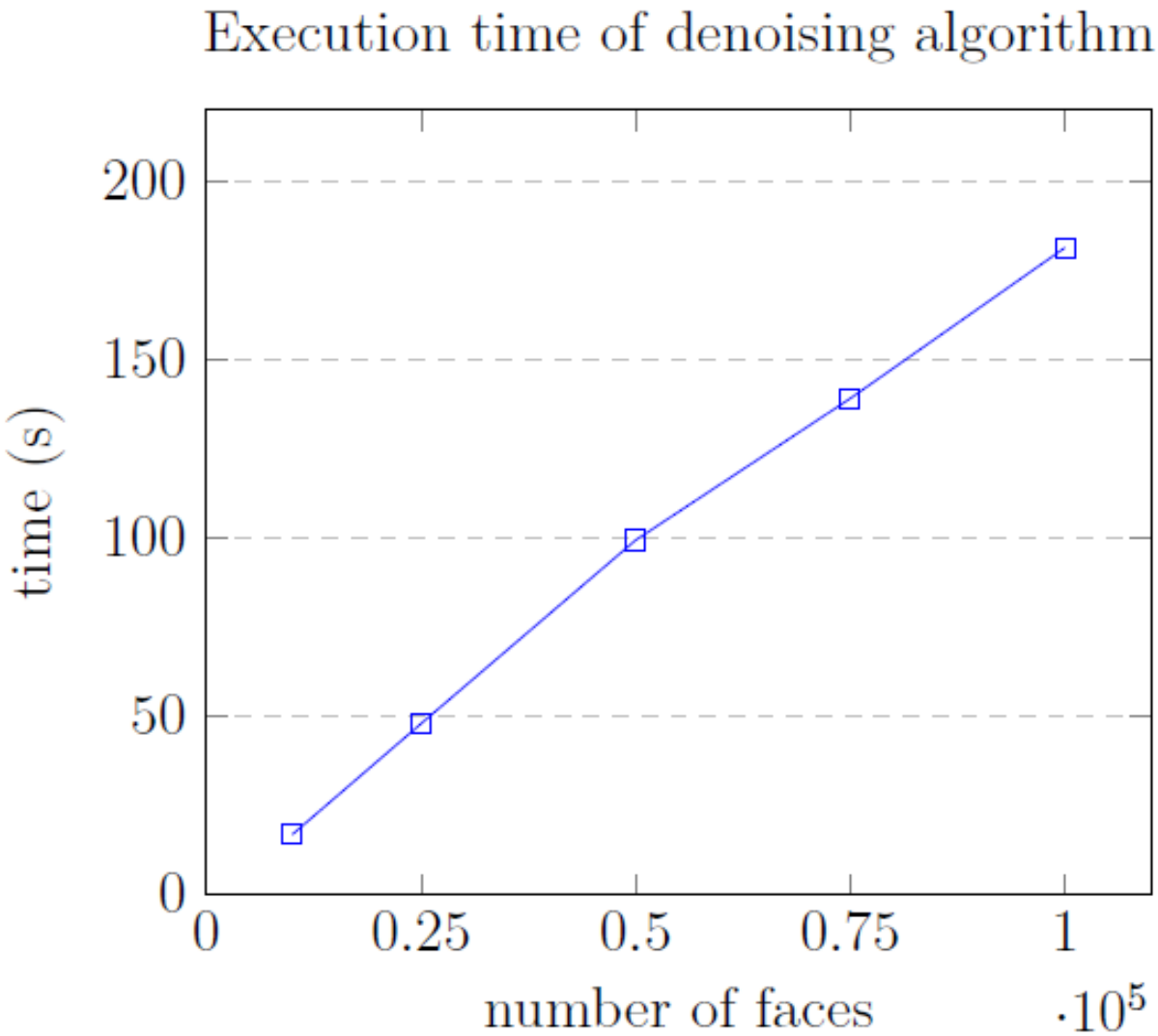


Keyboard mesh

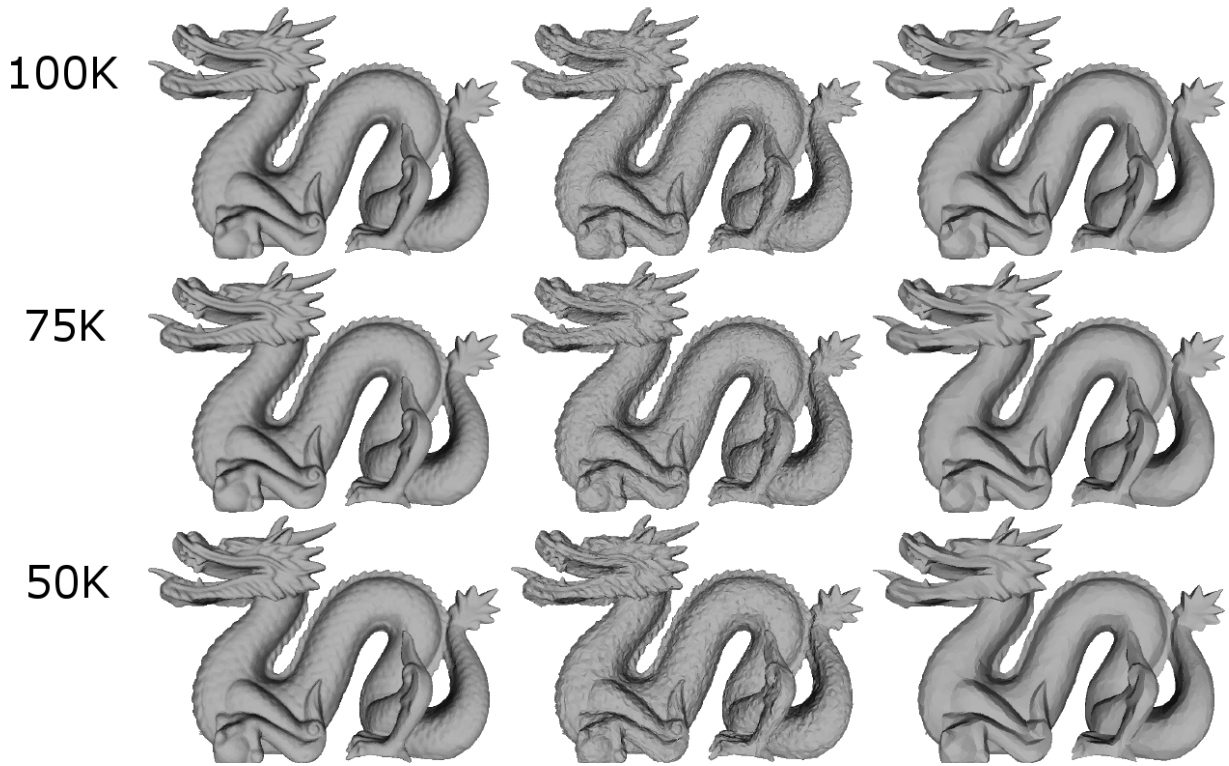
noisy - ours - guided



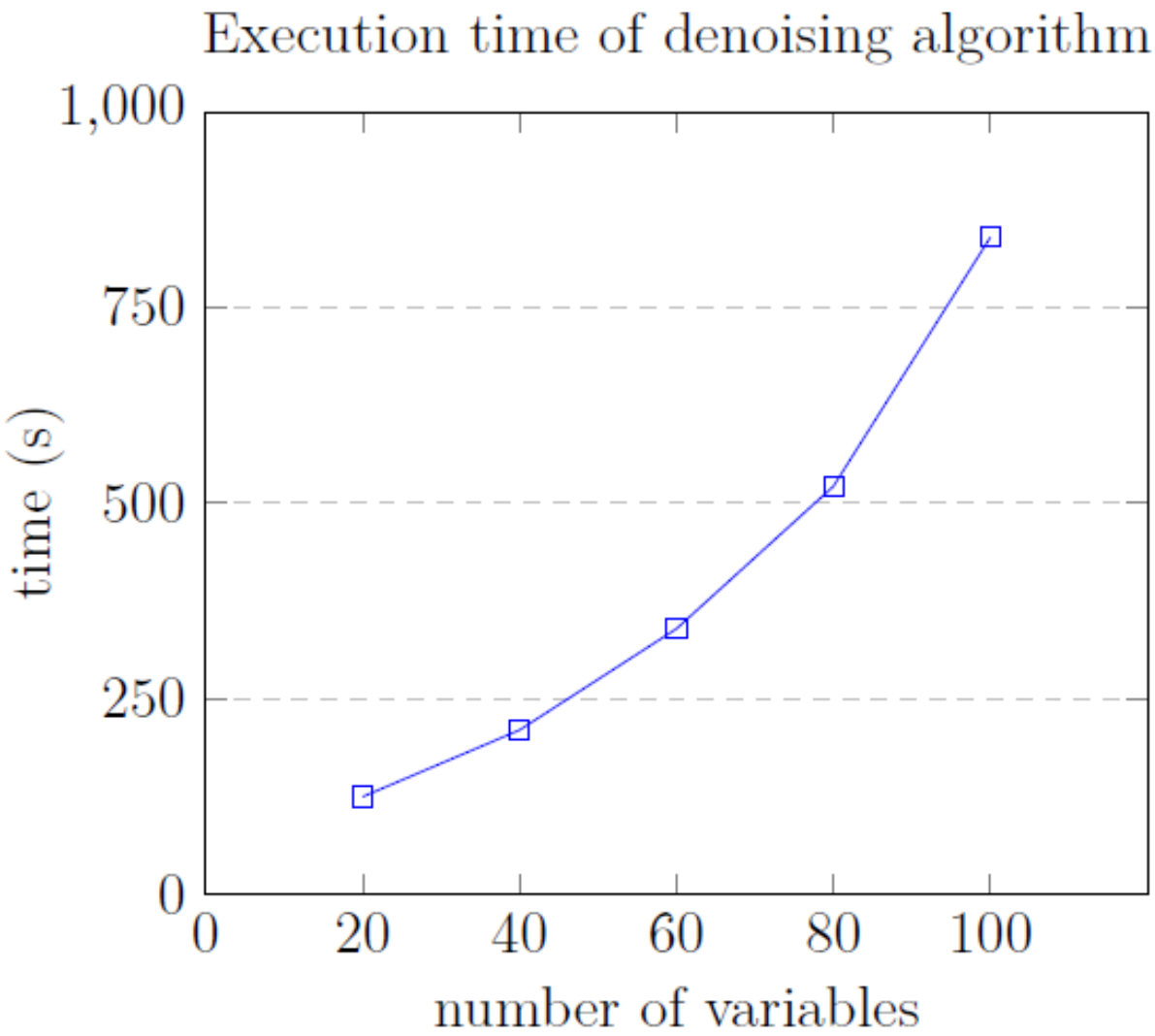
Performance



Performance



Performance



Conclusion and Future work

Conclusions :

- The proposed method removes the noise while preserving details.
- Using synthetic noise we obtained low error and in most cases the lowest one compared to other state-of-the-art methods.
- Using real data, most of our results have a better definition of details compared to toher state-of-the-art methods.
- The pipeline of our denoising algorithm is simple and similar to other state-of-the-art methods.
- The result of our method is dependent on the choice of the optimization parameters, that presents some complexity but also allows great flexibility.
- We explain the behavior of the method when changing the values of the parameters, allowing the reader to have a more intuitive notion of how the solution is influenced.
- The denoising algorithm can have multiple behaviors when tuning these parameters.

Conclusion and Future work

Conclusions :

- It is important to normalize the mesh due that the proportion between a parameter of a quadratic term and a parameter of a linear term strongly depends on the scale of the mesh.
- Non-convex quadratic optimization problems can be treated using local minimas or tunning the parameters.
- The computational time of our algorithm has a near-linear cost but with high constant time for the number of triangles.

Future work :

- face normal field + vertex normal field + (approximated tangent planes normal field).
- Vertex positions as sampled points of the 2-manifold. Point cloud denoising.
- Use our adaptive patches in other applications (segmentation, remeshing or feature detection). Use different descriptors.

The end

Thank you

Detail-preserving mesh denoising using adaptive patches

Dissertação de Mestrado

Jan Jose Hurtado Jauregui[1]

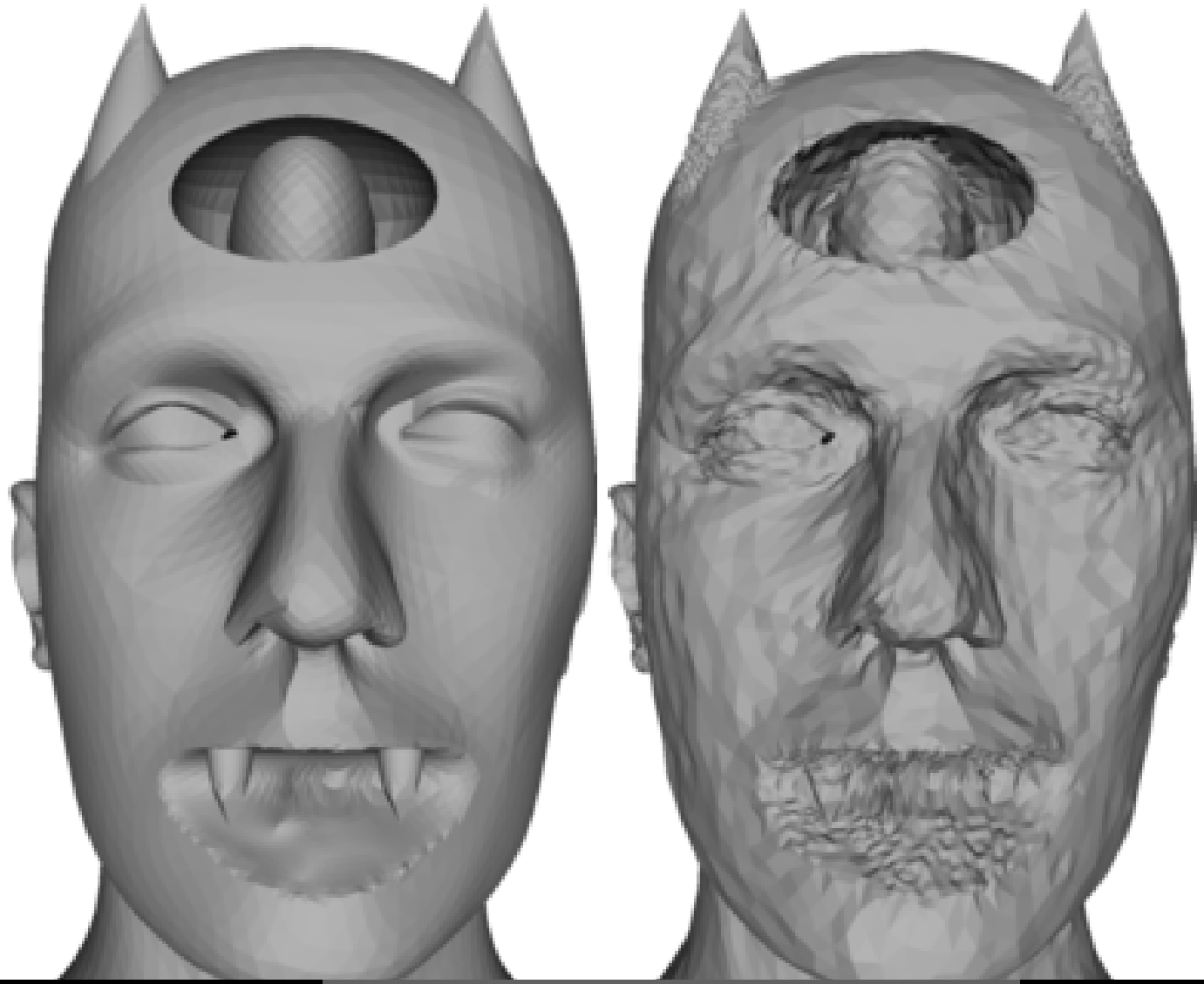
Advisor: Marcelo Gattass[1]

[1] Departamento de Informática,
Pontifícia Universidade Católica do Rio de Janeiro,
Rio de Janeiro, Brazil

8th March 2018

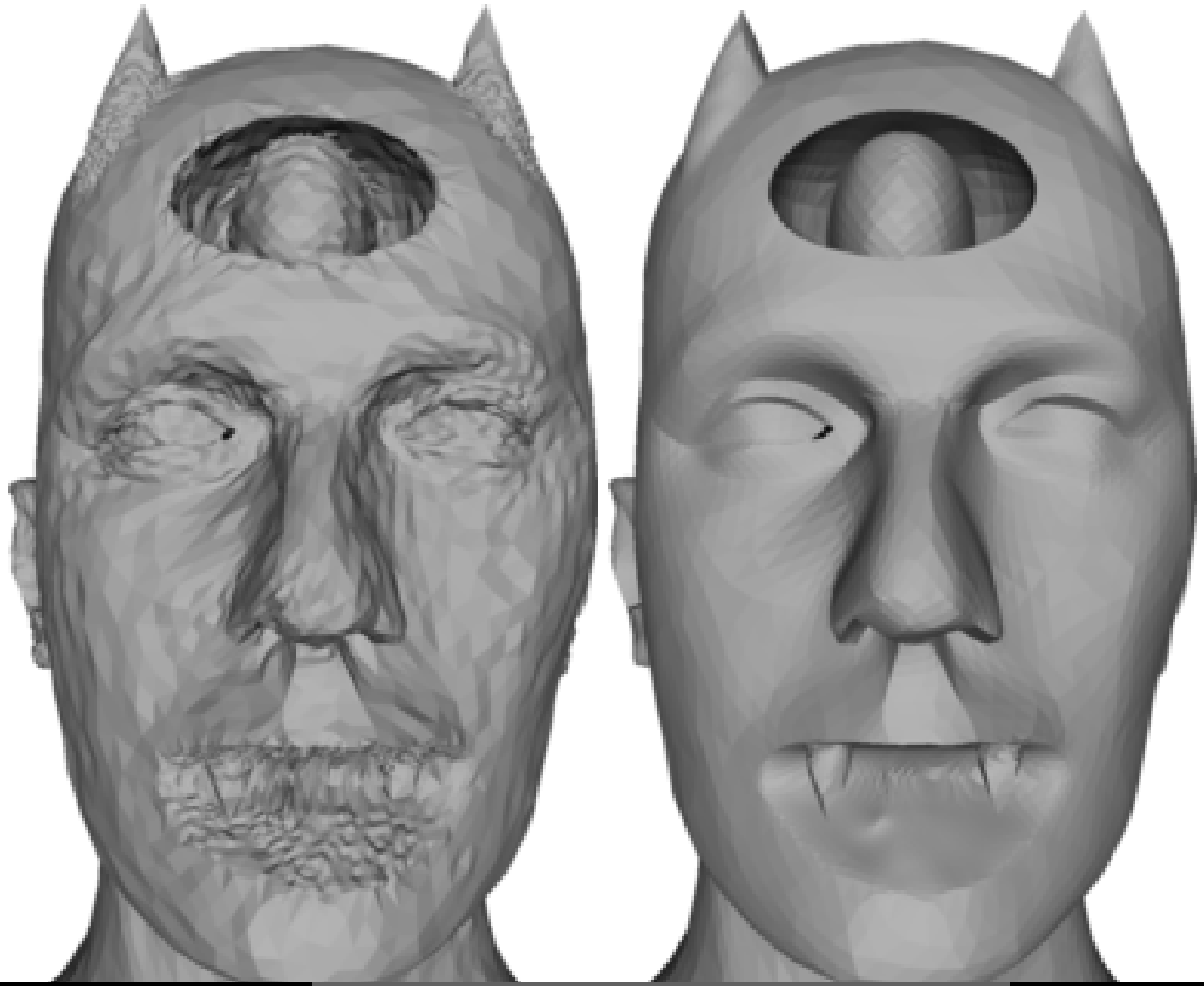
Devil mesh

original - noisy



Devil mesh

noisy - ours



Devil mesh

original - ours - bilateralNormal

Devil mesh

original - ours - L0

Devil mesh

original - ours - guided

Devil mesh

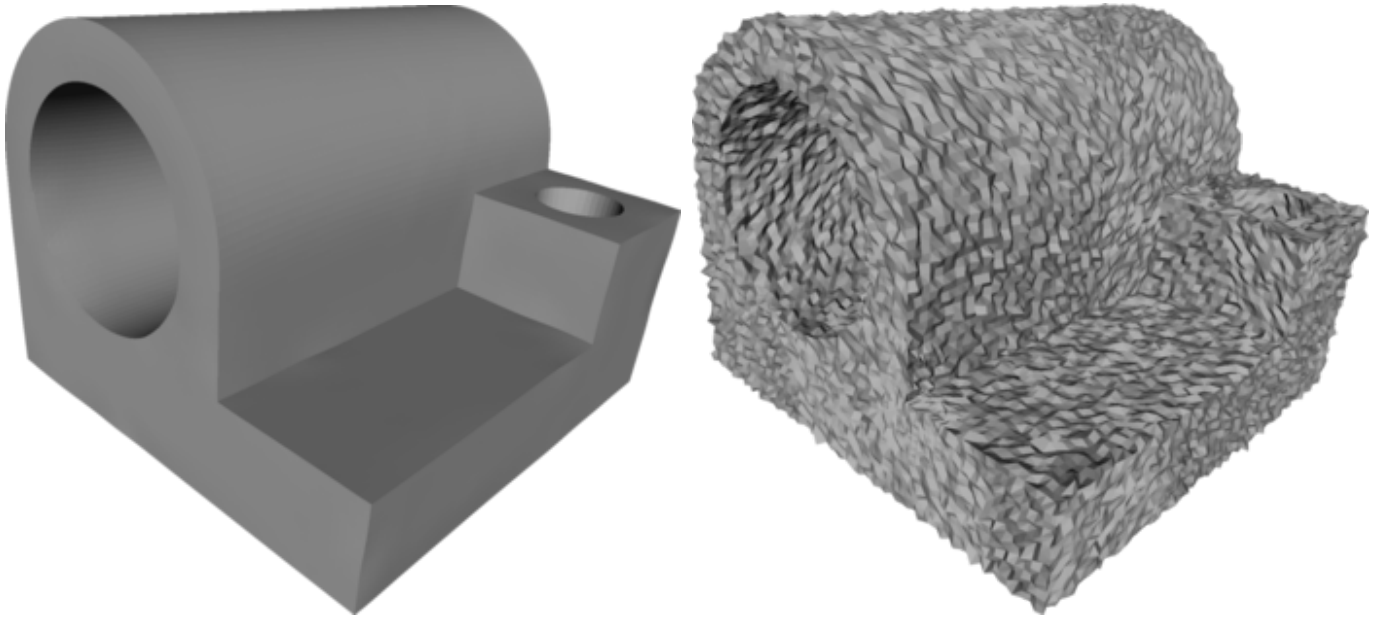
original - ours - nvt

Devil mesh

	Mean Vertex Distance	L2 Vertex Based	Mean Quadric	MSAE	L2 Normal Based	Tangential	Mean Discrete Curvature	Area Error	Volume Error
(FDC03)	0.061277	0.110973	0.236219	19.697900	0.055170	0.047678	0.090284	0.051443	0.045645
(JDD03)	0.001293	0.002800	0.002289	21.237300	0.021589	0.013026	0.087991	0.000364	0.002621
(SRML07)	0.001439	0.002880	0.003540	14.043200	0.012654	0.008911	0.055849	0.007806	0.000582
(ZFAT11)	**0.000713**	**0.001537**	0.001824	12.171400	**0.009654**	**0.005781**	**0.054567**	0.005617	**0.000425**
(HS13)	0.002531	0.004560	0.007108	13.830100	0.017459	0.010314	0.114528	0.001686	0.001786
(ZDZBL15)	0.001623	0.003079	0.005048	**10.454200**	0.015233	0.008054	0.094668	0.002629	0.001326
(YRP16)	0.000737	0.001548	**0.001493**	16.880800	0.014129	0.006974	0.079952	**0.000209**	0.002375
Ours	0.000987	0.001902	0.002686	11.574200	0.010632	0.006796	0.075106	0.003970	0.000722

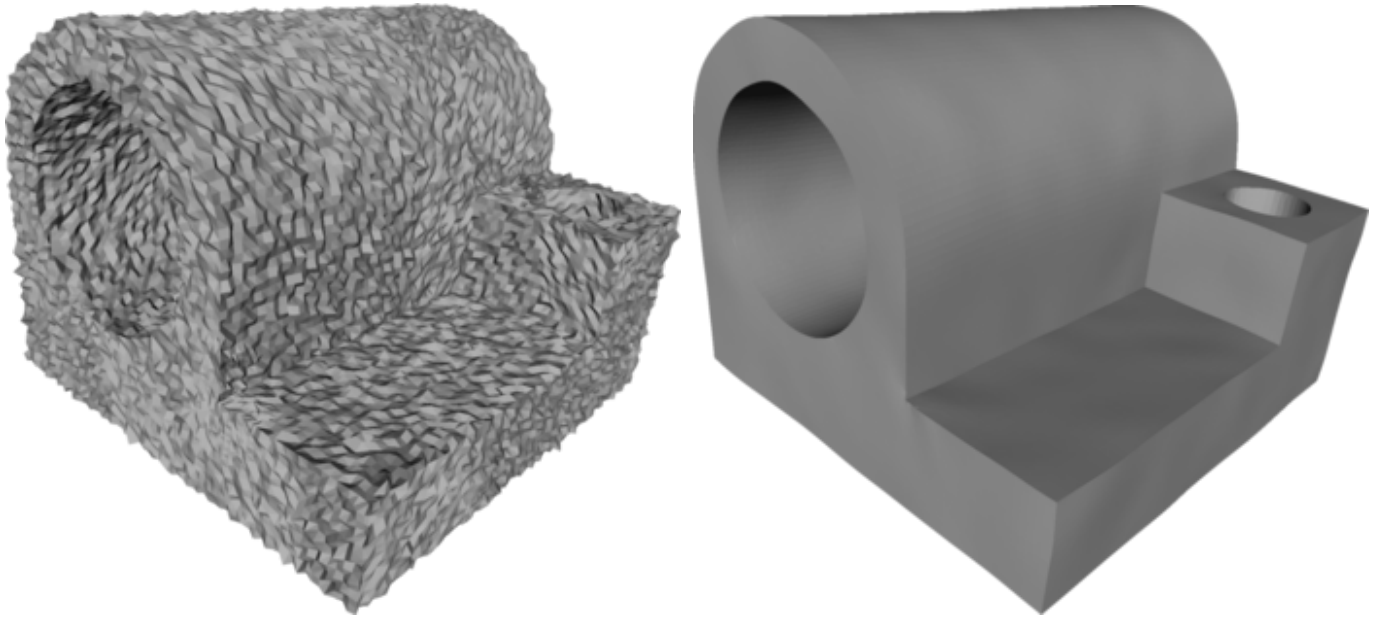
Joint mesh

original - noisy



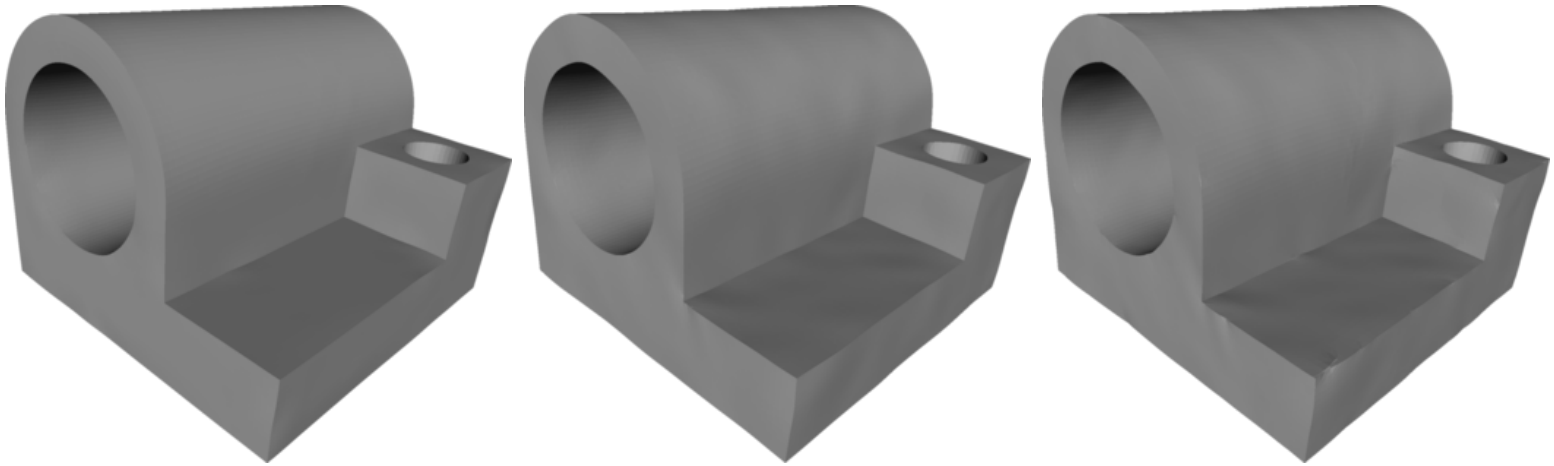
Joint mesh

noisy - ours



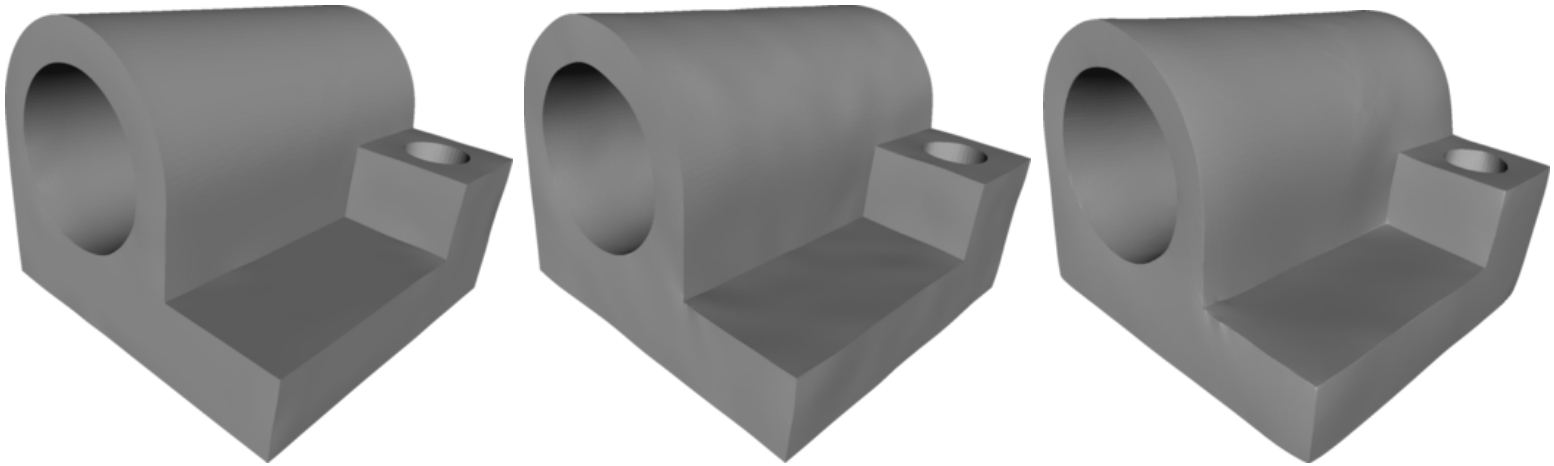
Joint mesh

original - ours - bilateralNormal



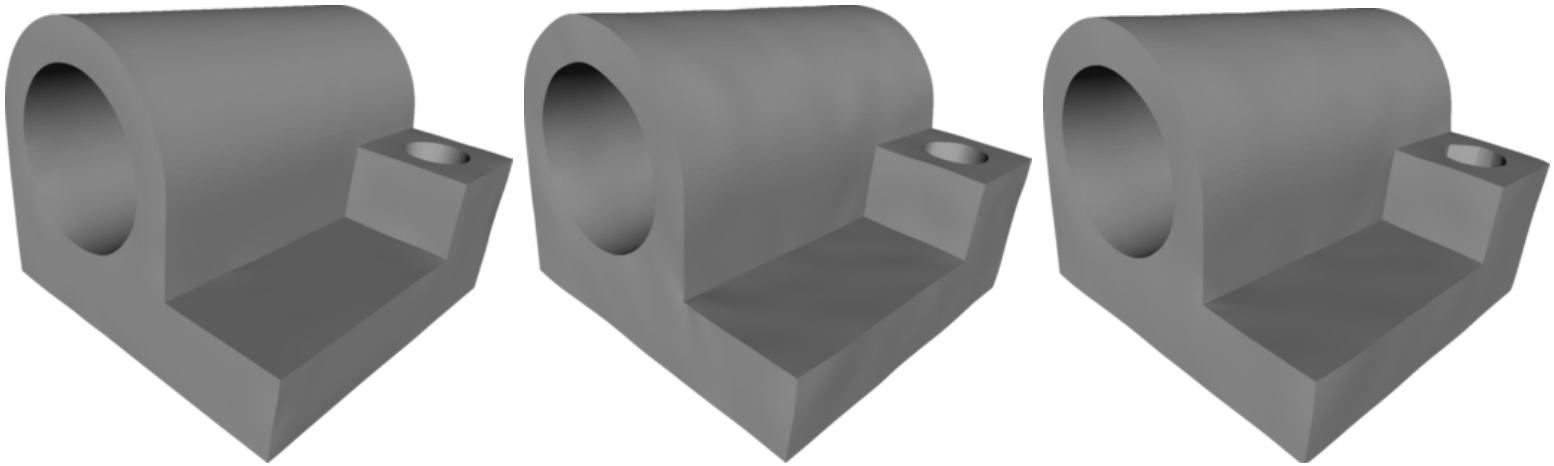
Joint mesh

original - ours - L0



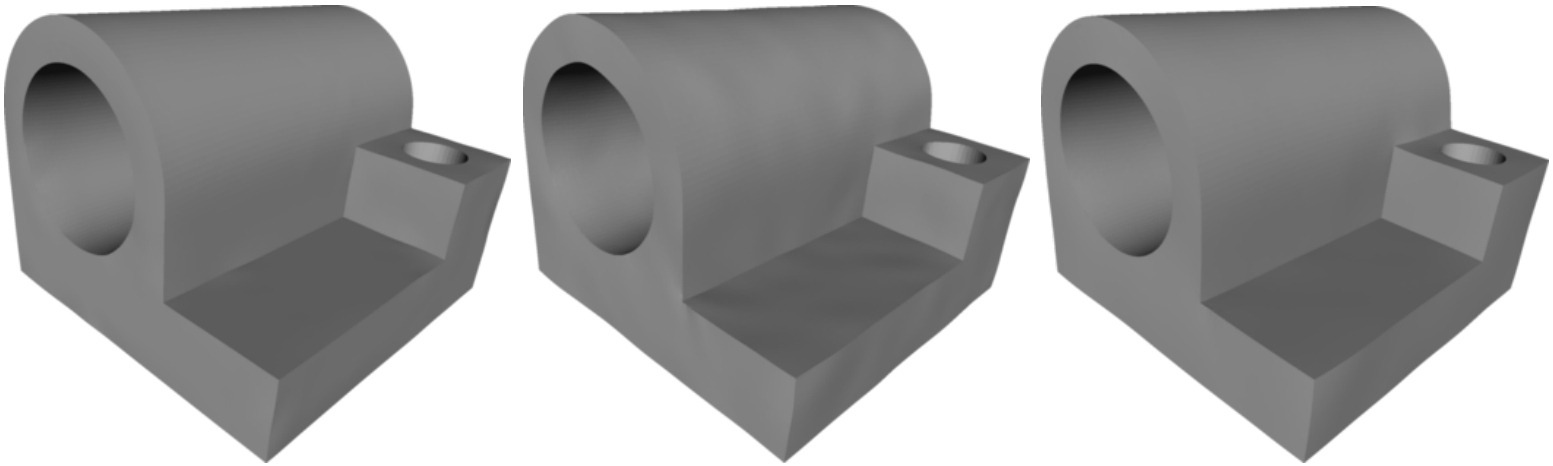
Joint mesh

original - ours - guided



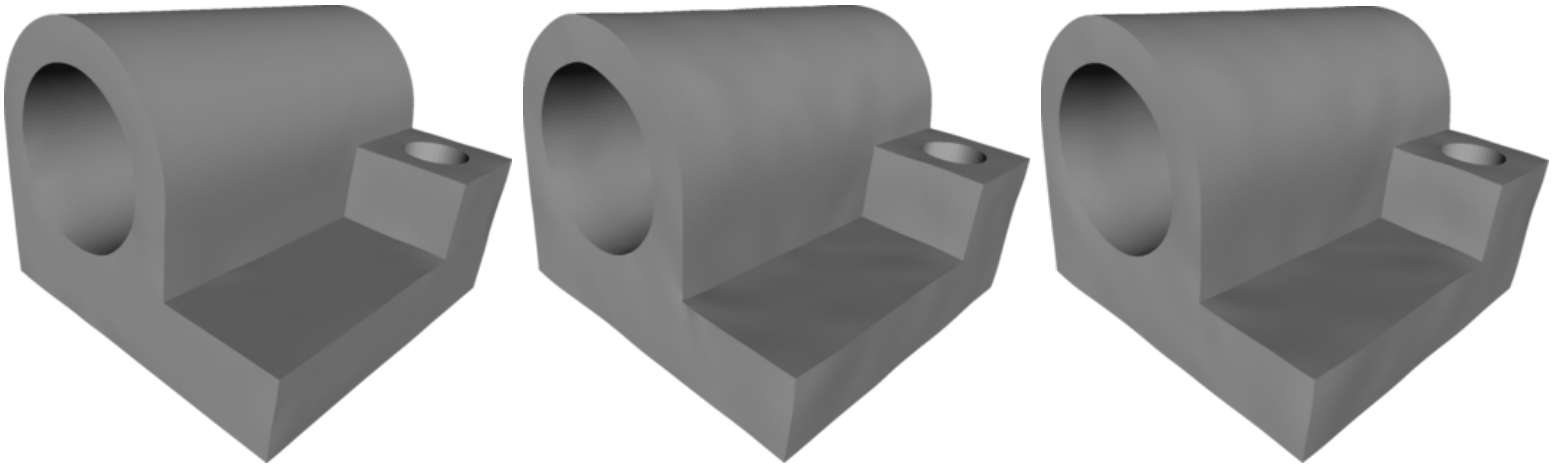
Joint mesh

original - ours - nvt



Joint mesh

original - ours - robust

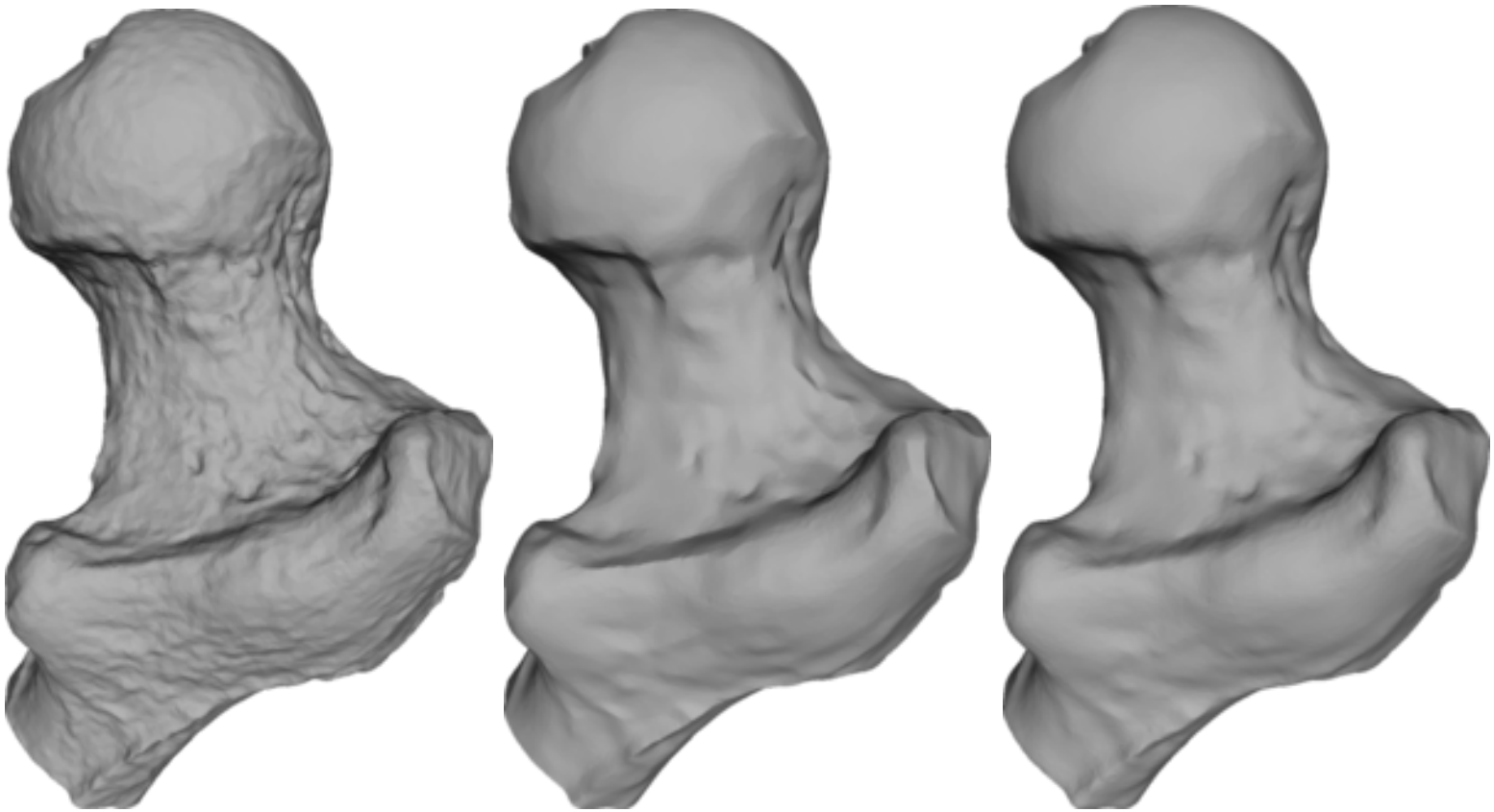


Joint mesh

	Mean Vertex Distance	L2 Vertex Based	Mean Quadric	MSAE	L2 Normal Based	Tangential	Mean Discrete Curvature	Area Error	Volume Error
(FDC03)	5.28e-6	1.23e-5	4.28e-9	5.339180	0.016951	0.020971	0.973744	0.026317	0.003270
(JDD03)	4.44e-6	8.12e-6	3.62e-9	6.708570	0.020659	0.011200	1.242240	0.005358	0.001581
(SRML07)	7.06e-7	2.43e-6	6.09e-10	1.470930	0.002508	0.003977	0.227227	0.002814	0.000115
(ZFAT11)	5.5e-7	1.11e-6	4.86e-10	1.250500	0.001493	0.003002	0.201129	0.000855	0.000152
(HS13)	4.9e-6	1.34e-5	4.14e-9	3.187840	0.005773	0.010923	0.218453	0.019501	0.002809
(ZDZBL15)	5.82e-7	1.48e-6	4.78e-10	1.304530	0.001645	0.003050	0.154981	0.000719	0.000215
(YRP16)	**4.84e-7**	1.29e-6	**4.2e-10**	**0.930990**	0.000756	0.003247	**0.113735**	0.000917	0.000223
(YRP17)	5.69e-7	1.06e-6	4.89e-10	0.986475	0.000882	**0.000977**	0.160960	0.003271	0.000583
Ours	5.67e-7	**1.04e-6**	4.67e-10	1.051820	**0.000601**	0.001352	0.180188	**0.000203**	**0.000066**

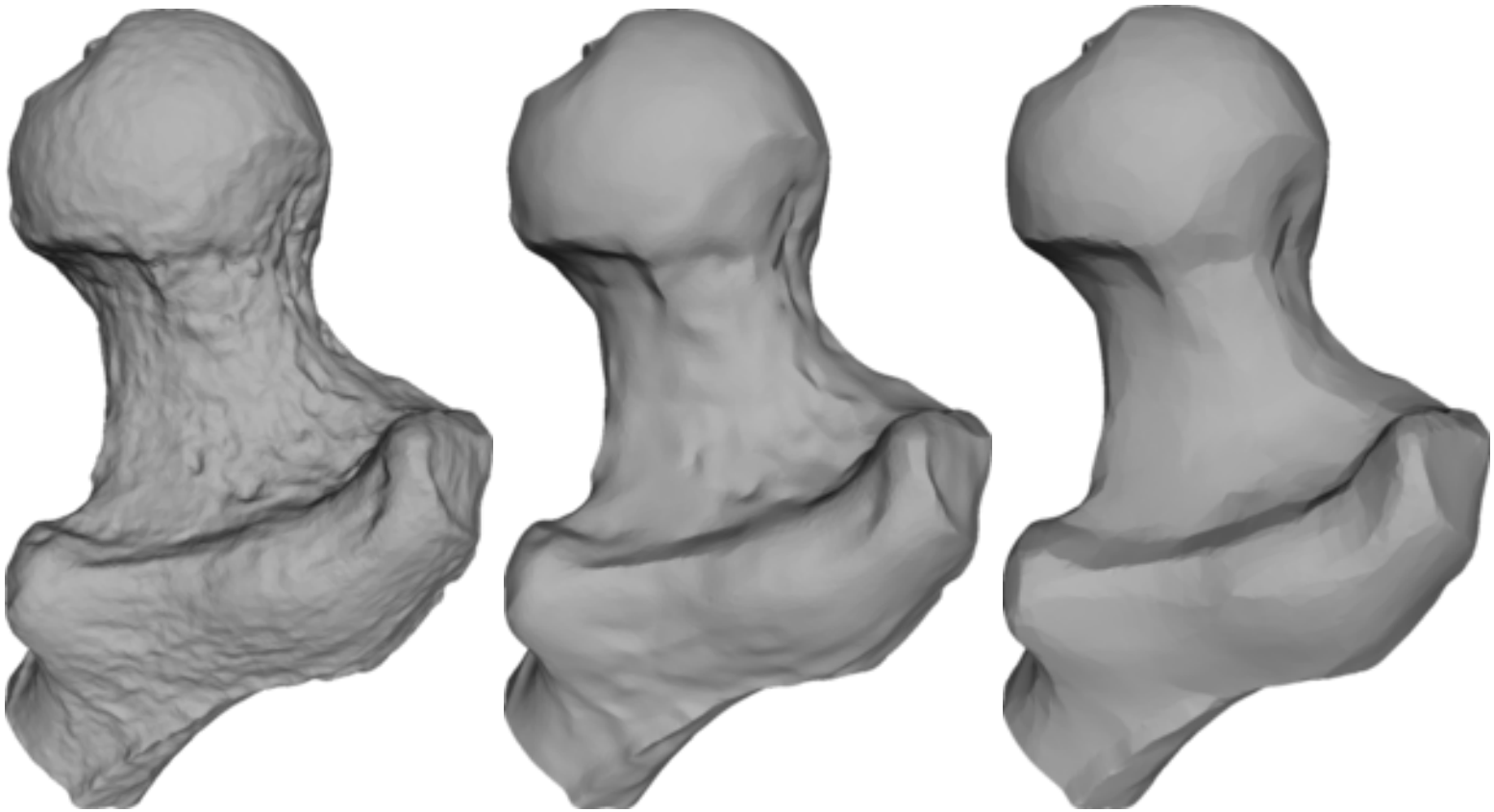
BallJoint mesh

noisy - ours - bilateralNormal



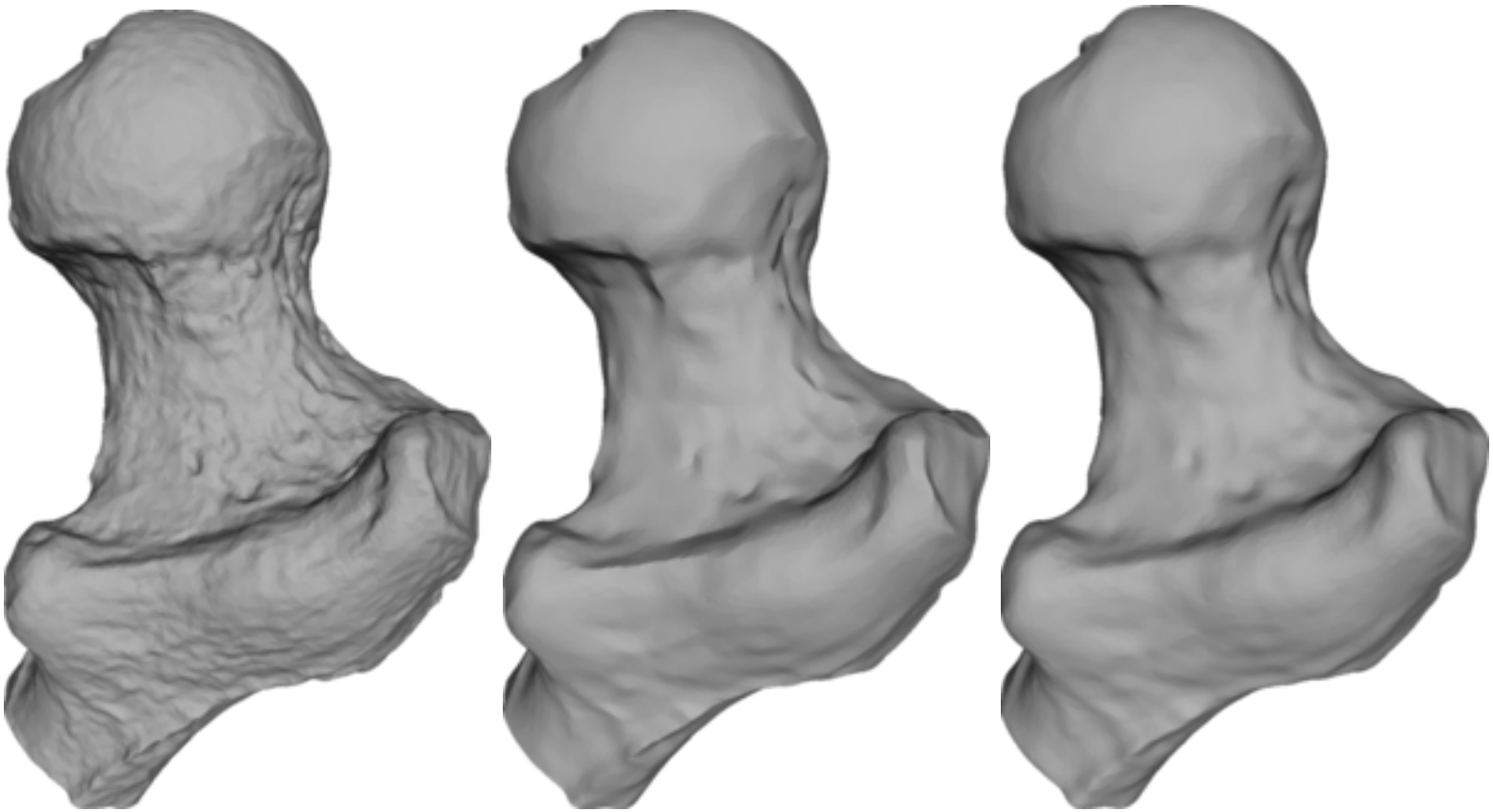
BallJoint mesh

noisy - ours - L0



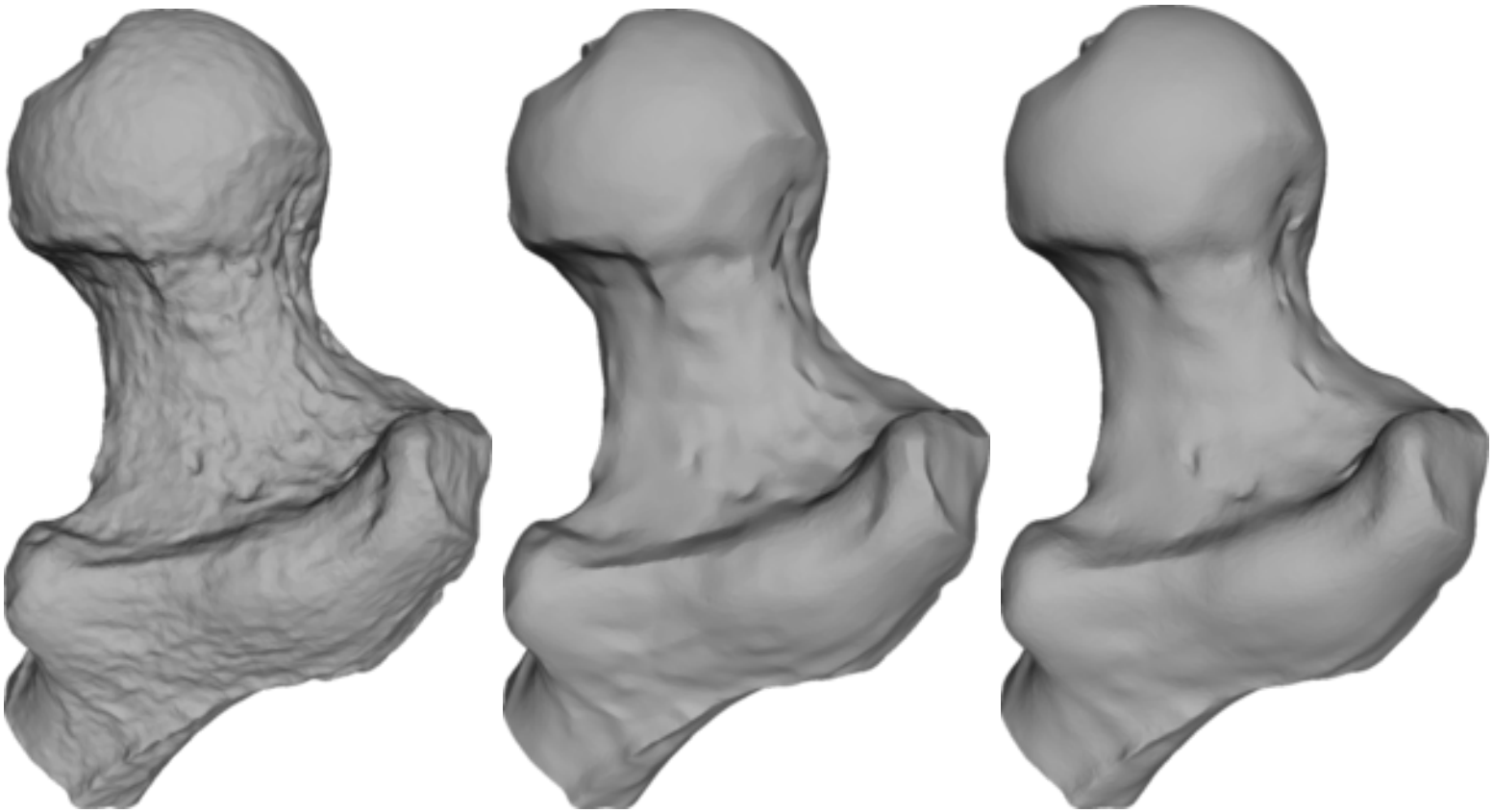
BallJoint mesh

noisy - ours - guided



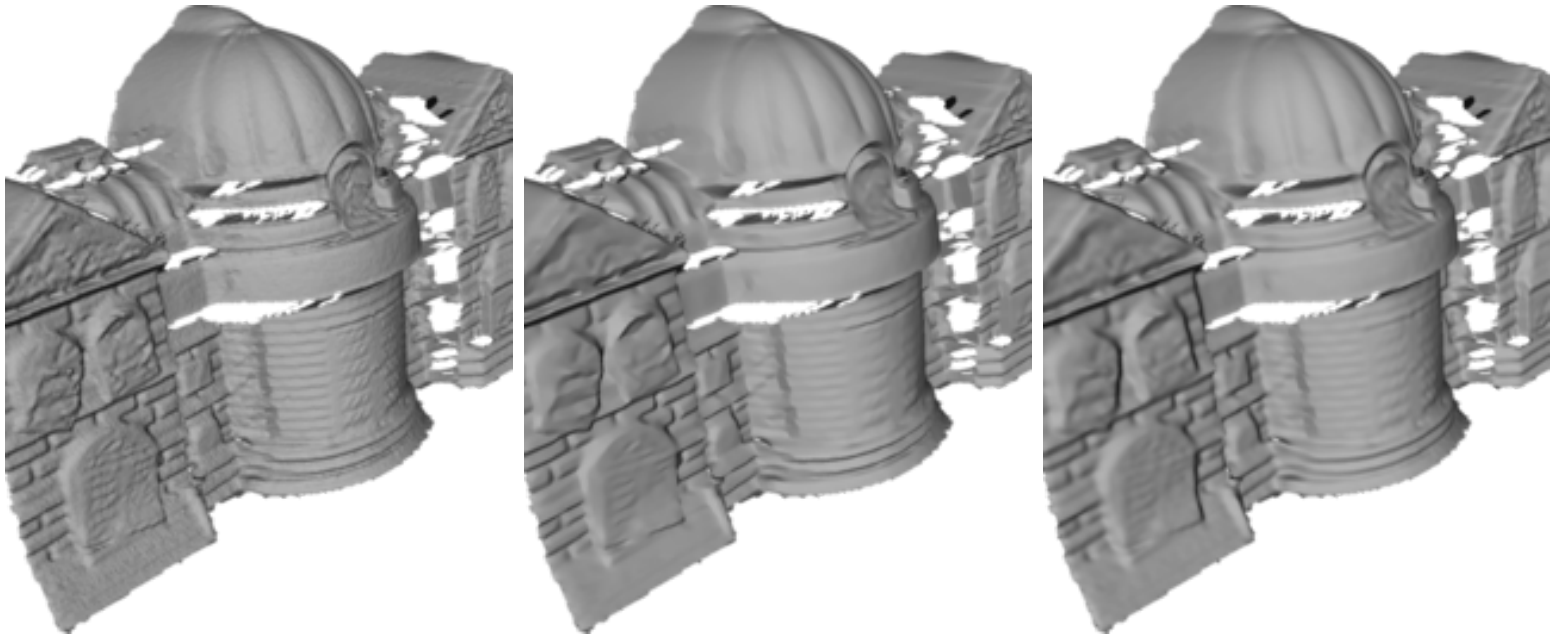
BallJoint mesh

noisy - ours - nvt



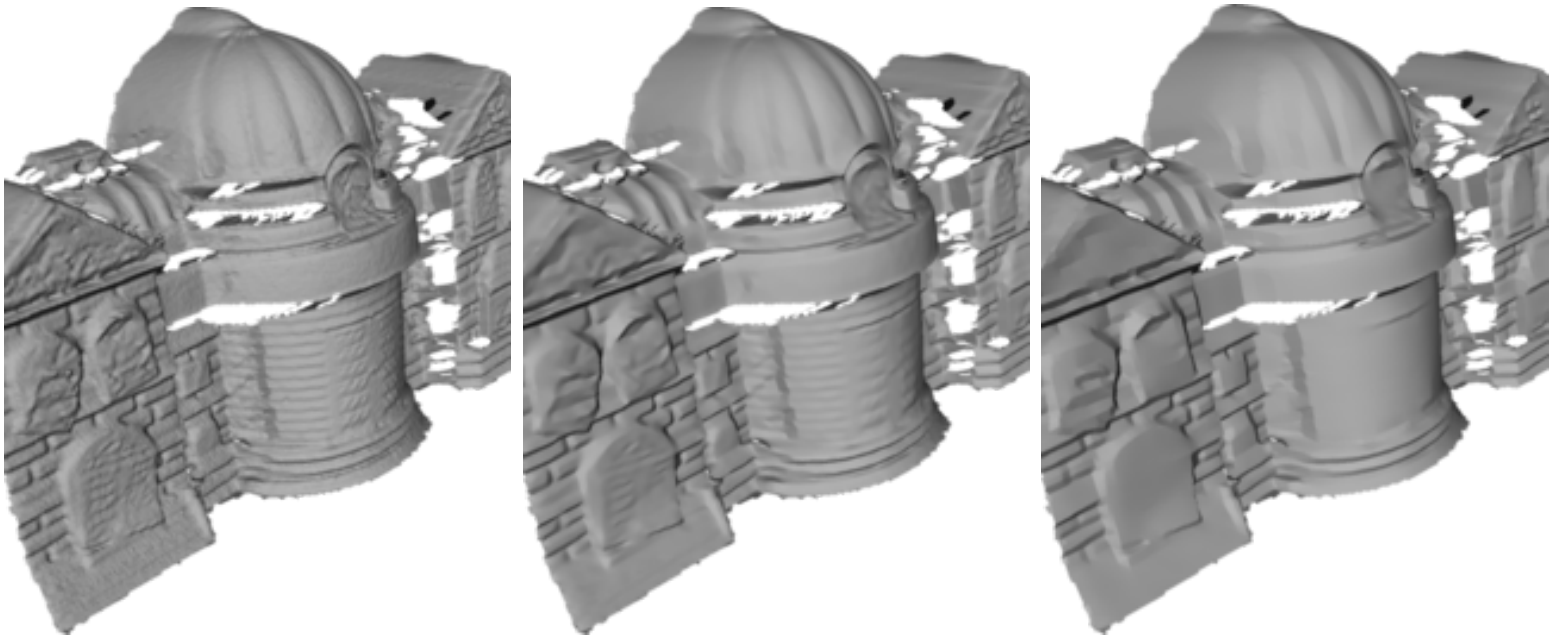
Building mesh

noisy - ours - bilateralNormal



Building mesh

noisy - ours - L0



Building mesh

noisy - ours - guided