Mumford-Shah Mesh Processing using the Ambrosio-Tortorelli Functional

N. Bonneel*,[1] D. Coeurjolly*,[1] P. Gueth*[2] J.-O. Lachaud*[3]

*Equal contribution

[1]CNRS, Univ. Lyon
[2]Arskan
[3]Université Savoie Mont Blanc

Goal

- A general purpose edge-aware computational geometry tool...

(a) Original (b) Denoising (c) Inpainting (d) Segmentation (e) Embossing

Context

- ... benefitting from the Mumford-Shah functional used in image processing

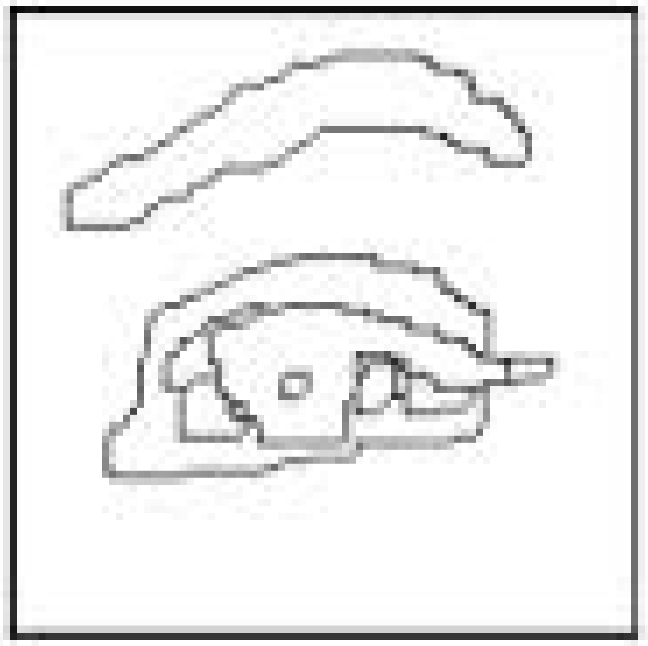
finding a function u and contours C

$$MS[u, C] = \alpha \int_{\Omega} (u - g)^2 \, \mathrm{d}x + \beta \int_{\Omega \setminus C} |\nabla u|^2 \, \mathrm{d}x + \gamma \int_{C} \mathrm{d}s$$

with u matching a function g | **u smooth outside of C** | **and short contours C**

g

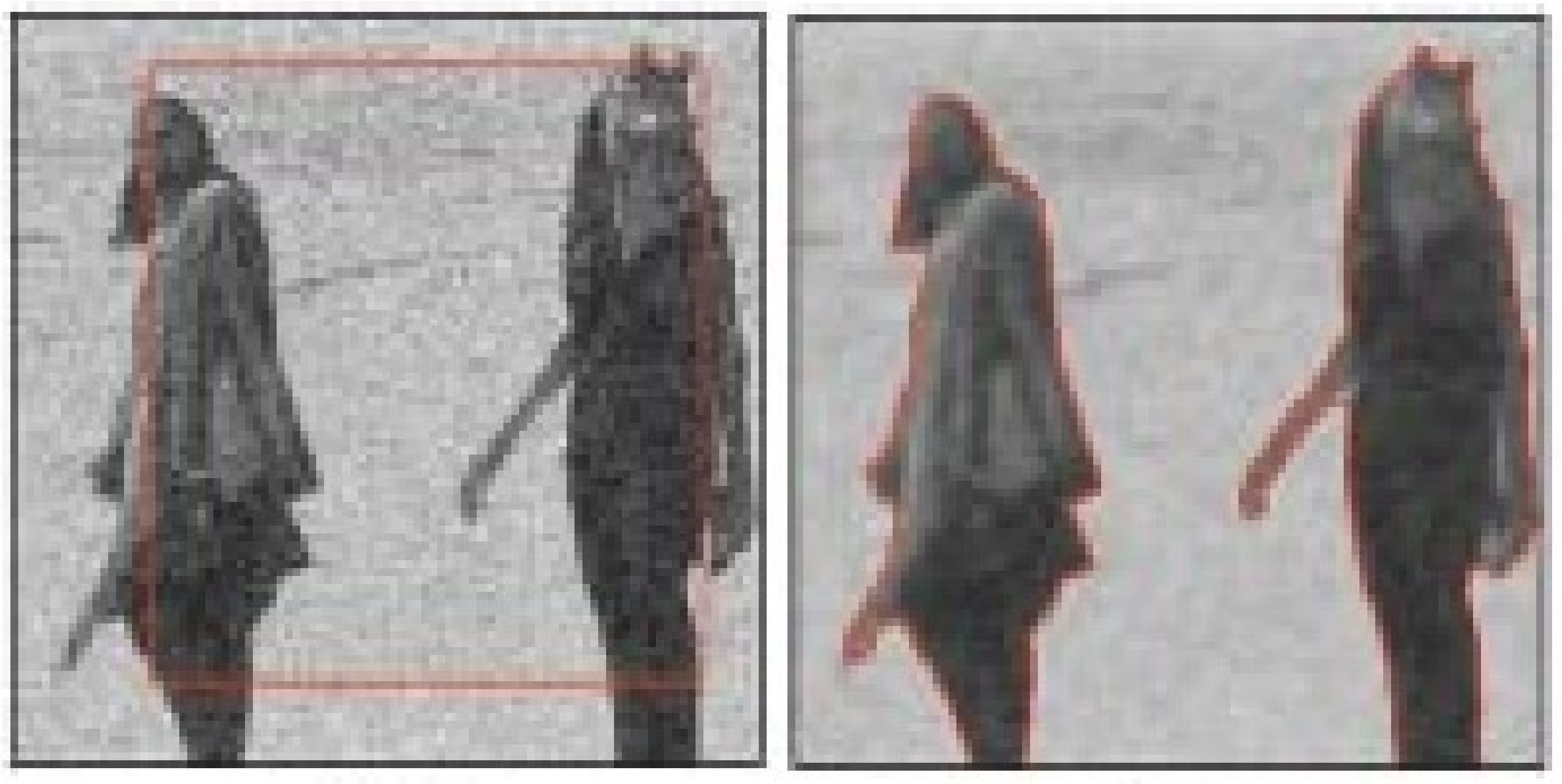
u



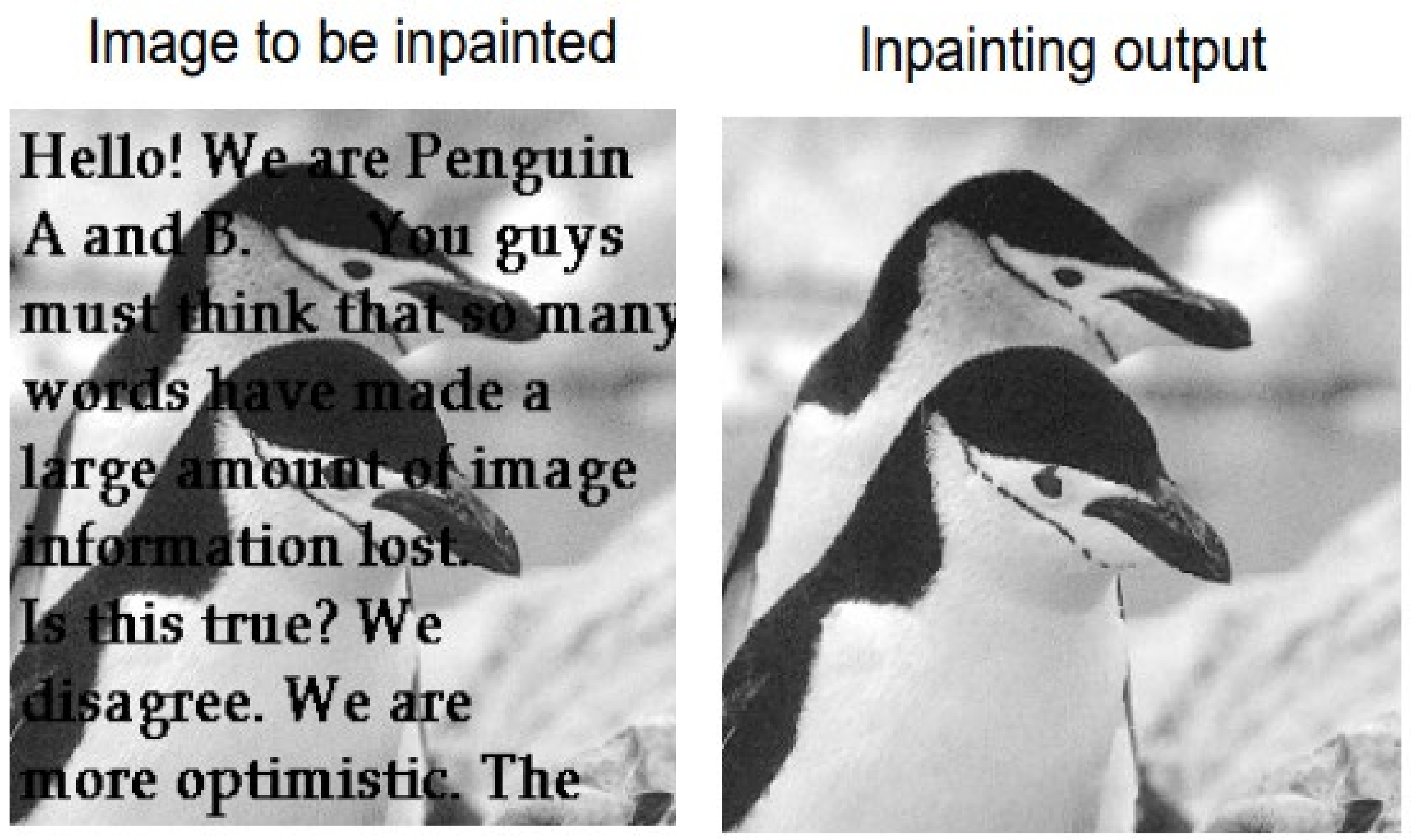
C

Prior work

- Mumford-Shah in image processing



Segmentation+denoising [Tsai et al. 2001]



Inpainting [Esedoglu and Shen 2002]

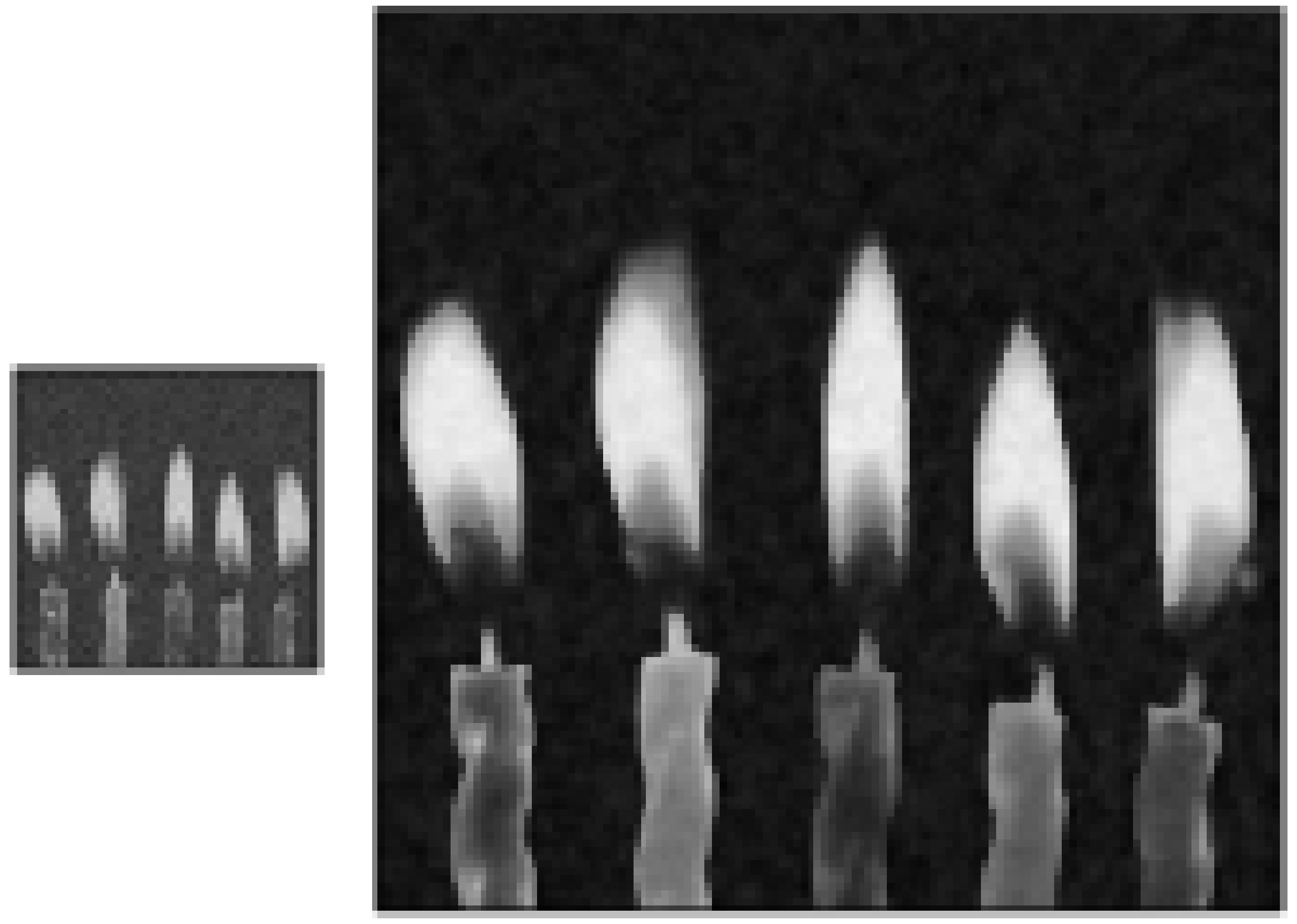


Image magnification [Tsai et al. 2001]

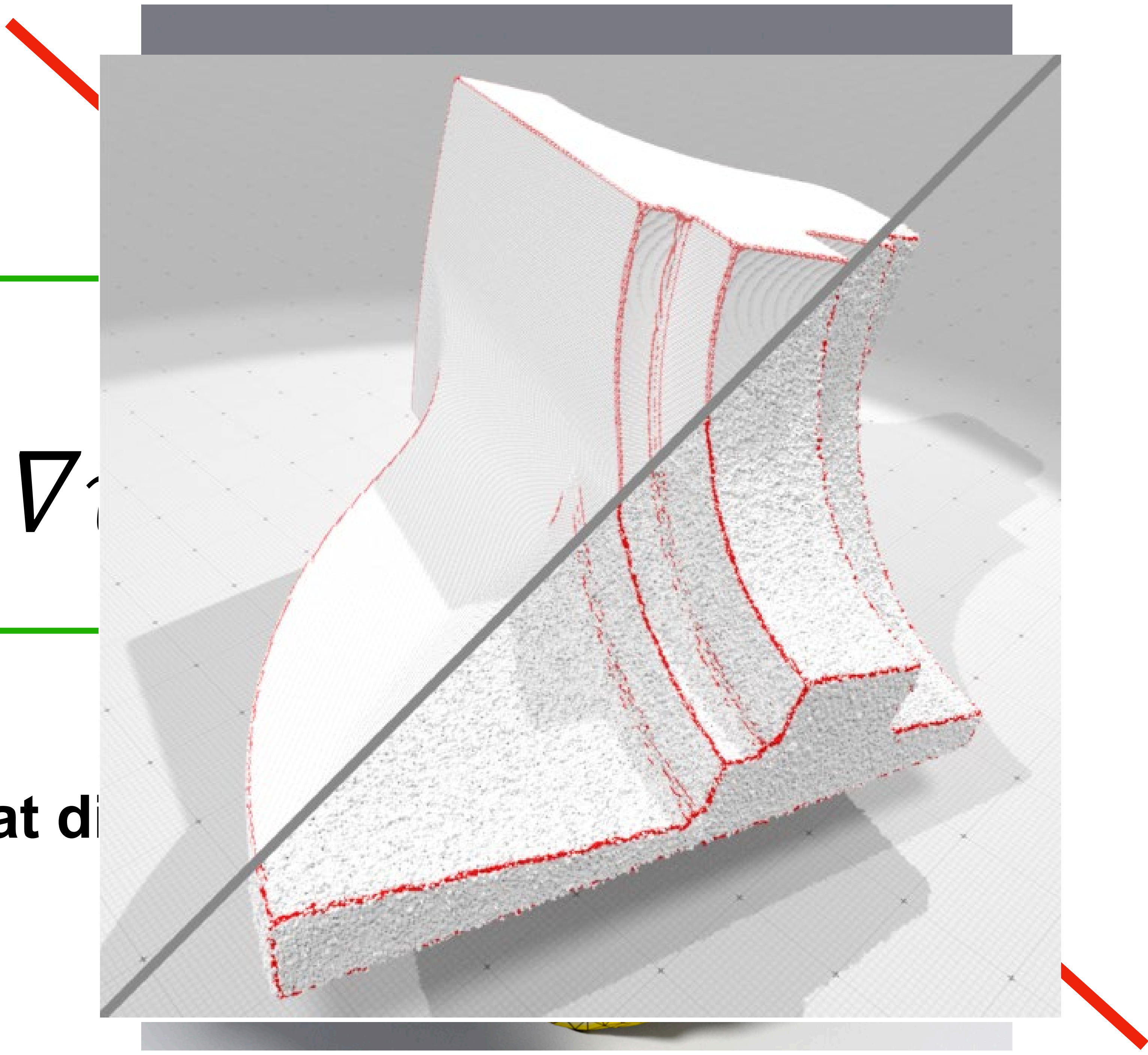
other edge-aware applications include deconvolution and stereo matching.

Prior work

- For solving Mumford-Shah
  - Replacing contours by closed contours, and optimizing indicator function instead [Chan et a. 2006]. Unclear how to adapt to meshes.
  - Ambrosio and Tortorelli [1990] approximation (AT)
    - Used on voxelized shapes [Coeurjolly et al. 2016]

$$AT_\epsilon[u,v] = \int_\Omega \alpha(u-g)^2 + |v\nabla u|^2 + \lambda\epsilon|\nabla$$

smooth feature field v : 0 at di



Our method

Pipeline

- Regularize normal field using AT, treating x, y, z separately
  (AT step)

- Optimize vertices position so mesh triangles match regularized normal
  (Projection step)

Ambrosio-Tortorelli DEC discretization

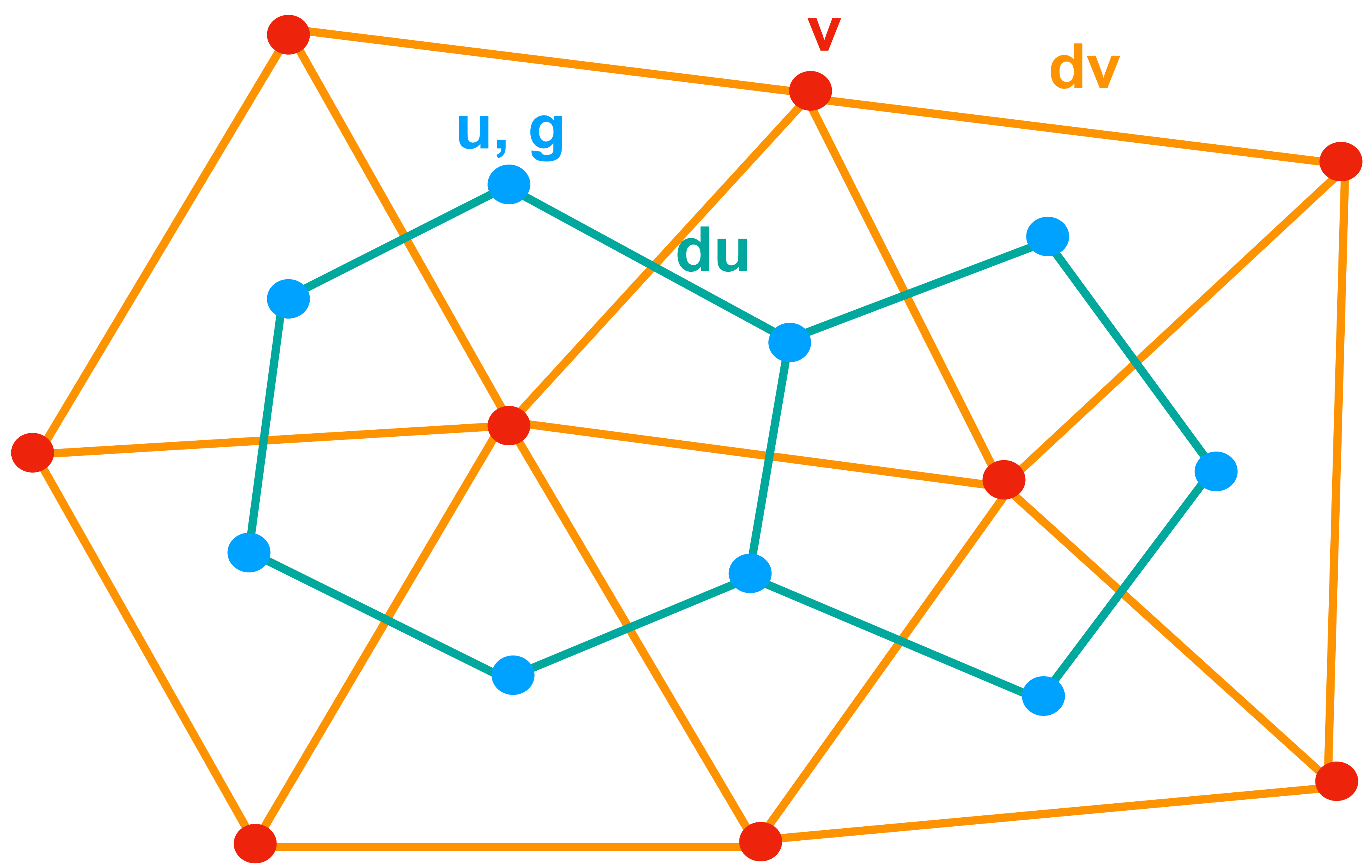
becomes du, a 1-form on dual edges

$$AT_\epsilon[u,v] = \int_\Omega \alpha(u-g)^2 + |v\nabla u|^2 + \lambda\epsilon|\nabla v|^2 + \frac{\lambda}{\epsilon}\frac{(1-v)^2}{4}\,\mathrm{d}x$$

becomes dv, a 1-form on primal edges

v stored as 0-form on primal vertices

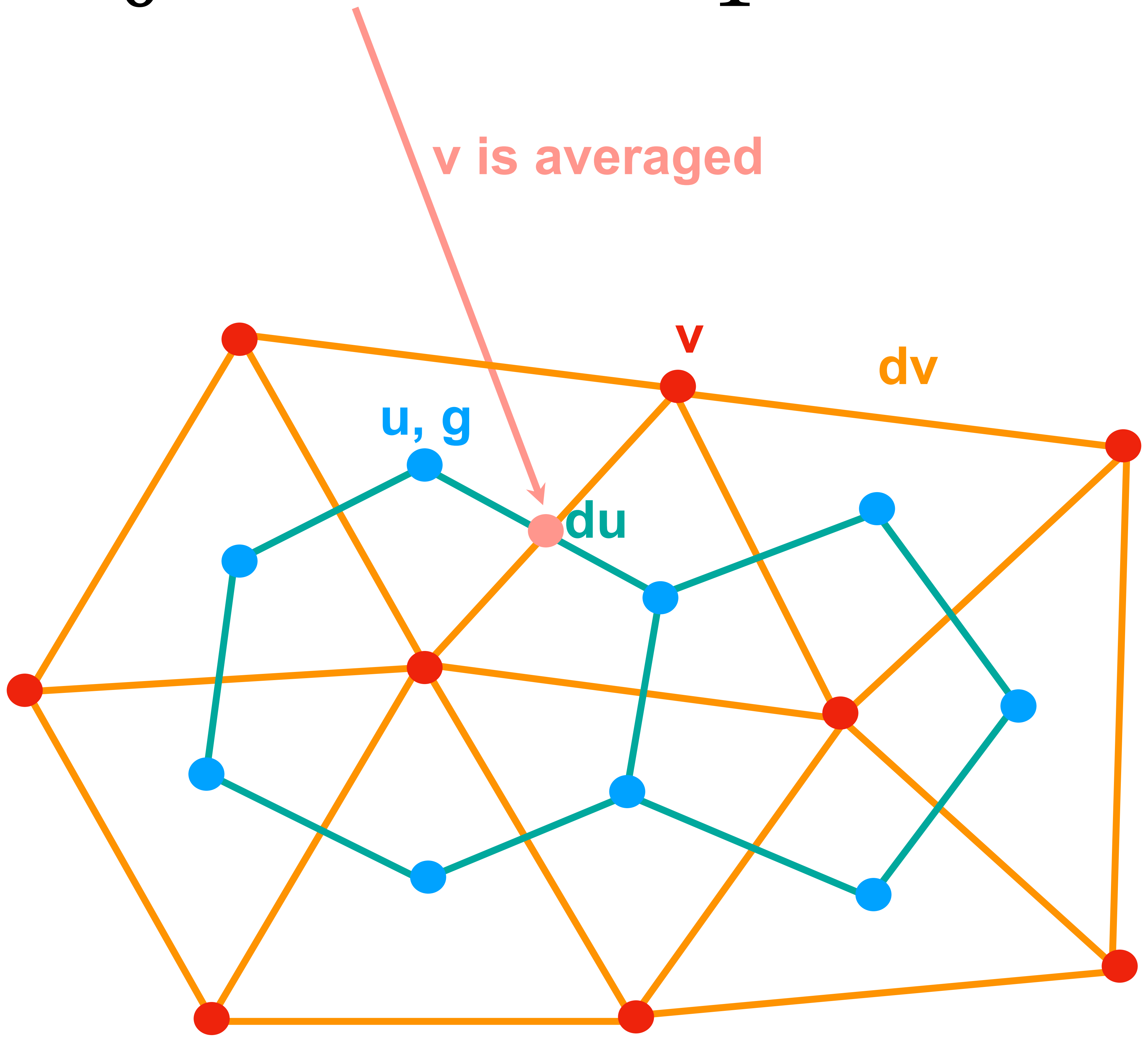
u and g stored as 0-forms on dual vertices



Ambrosio-Tortorelli DEC discretization

- Discretization reads

$$AT_\epsilon[u,v] = \alpha\langle u-g, u-g\rangle_{\bar{0}} + \langle v\bar{\mathrm{d}}u, v\bar{\mathrm{d}}u\rangle_{\bar{1}} + \lambda\epsilon\langle \mathrm{d}v, \mathrm{d}v\rangle_1 + \frac{\lambda}{4\epsilon}\langle 1-v, 1-v\rangle_0$$



Ambrosio-Tortorelli DEC discretization

- In matrix form

averaging matrix

$$AT_\epsilon[u,v] = \alpha(u-g)^T S_{\bar{0}}(u-g) + u^T B^T \mathrm{Diag}(Mv) S_{\bar{1}} \mathrm{Diag}(Mv) Bu + \lambda\epsilon v^T A^T S_1 A v + \frac{\lambda}{4\epsilon}(1-v)^T S_0 (1-v)$$

Hodge star　Hodge star　Hodge star　Hodge star

Cancelling gradients results in two linear systems in u and v, solved alternatively :

$$\left[\alpha S_{\bar{0}} - B^T \mathrm{Diag}(Mv) S_{\bar{1}} \mathrm{Diag}(Mv) B\right] u = \alpha S_0 g$$

$$\left[\frac{\lambda}{4\epsilon} S_0 + \lambda\epsilon A^T S_1 A + M^T \mathrm{Diag}(Bu) S_{\bar{1}} \mathrm{Diag}(Bu) M\right] v = \frac{\lambda}{4\epsilon} S_0$$

Projection

- Optimize vertices to minimize

$$E = E_m + w_1 E_f + w_2 E_d$$

Matching normals

$$E_m = \sum_{f_i \in F} ((p_{f_i^1} - p_{f_i^0}) \cdot u)^2 + ((p_{f_i^2} - p_{f_i^1}) \cdot u)^2 + ((p_{f_i^0} - p_{f_i^2}) \cdot u)^2$$

=> Each face normal is near the regularized normal

fairness

$$E_f(e) = \left(\frac{v(p_{i_1}) + v(p_{i_2})}{2}\right)^2 \left\| p_{i_1} + p_{i_2} - p_{i_3} - p_{i_4} \right\|^2$$

=> Favors Delaunay-like meshes

data attachment

$$E_d = \sum_i \| p_i - q_i \|^2$$

=> Prevents large departures from original mesh

Projection

- Optimize vertices to minimize

$$E = E_m + w_1 E_f + w_2 E_d$$

Matching normals

$$\nabla_{p_{f_i^j}} E_m(f_i) = 2\left(\left(2p_{f_i^j} - \sum_{k \neq j} p_{f_i^k}\right) \cdot u\right)u$$

data attachment

$$\nabla_p E_d = 2(p - q)$$

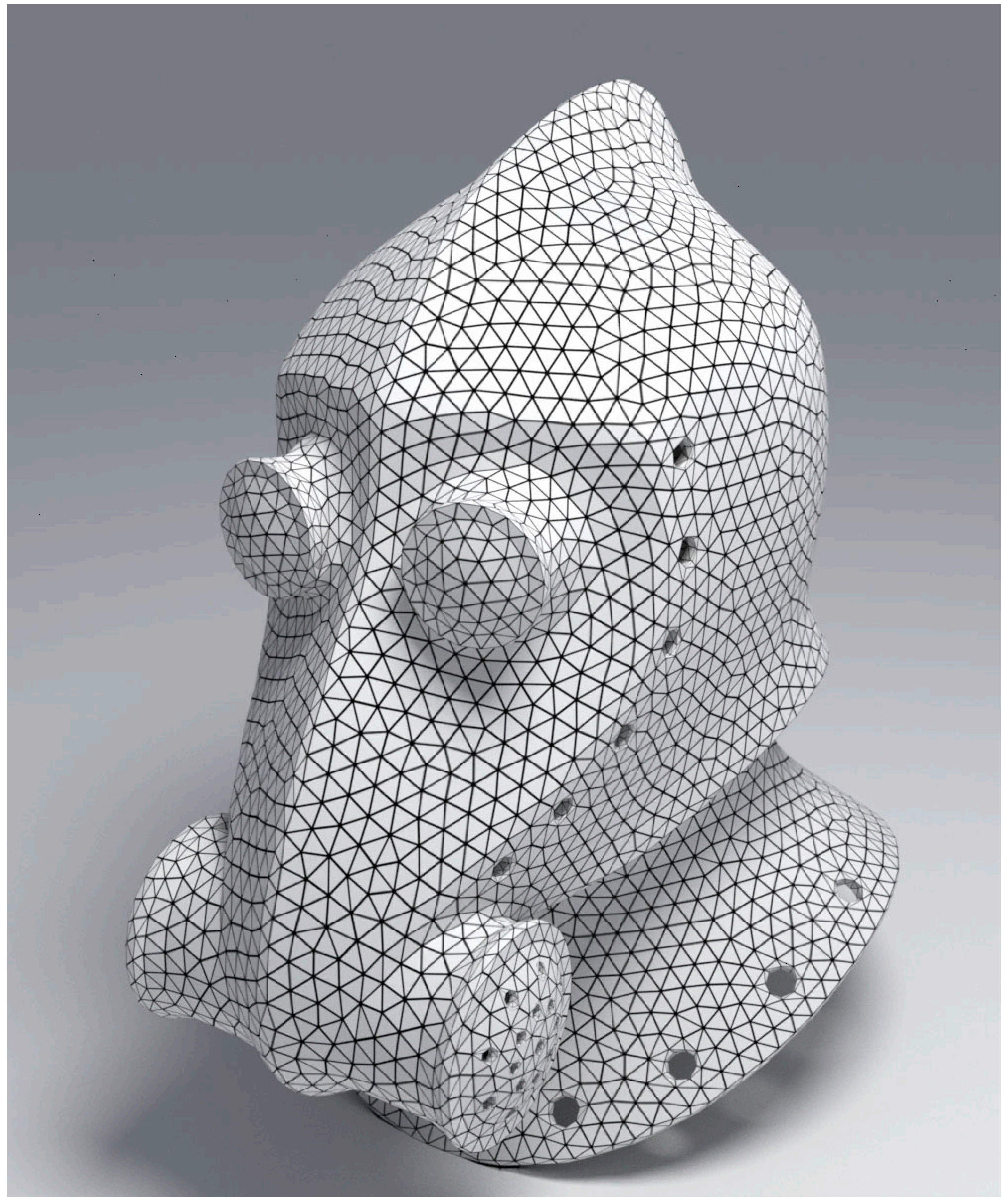
fairness

$$\nabla_{p_{i_1}} E_f(e) = 2\left(\frac{v(p_{i_1}) + v(p_{i_2})}{2}\right)^2 (p_{i_1} + p_{i_2} - p_{i_3} - p_{i_4})$$

=> Single linear system to solve

Results

Robustness to noise

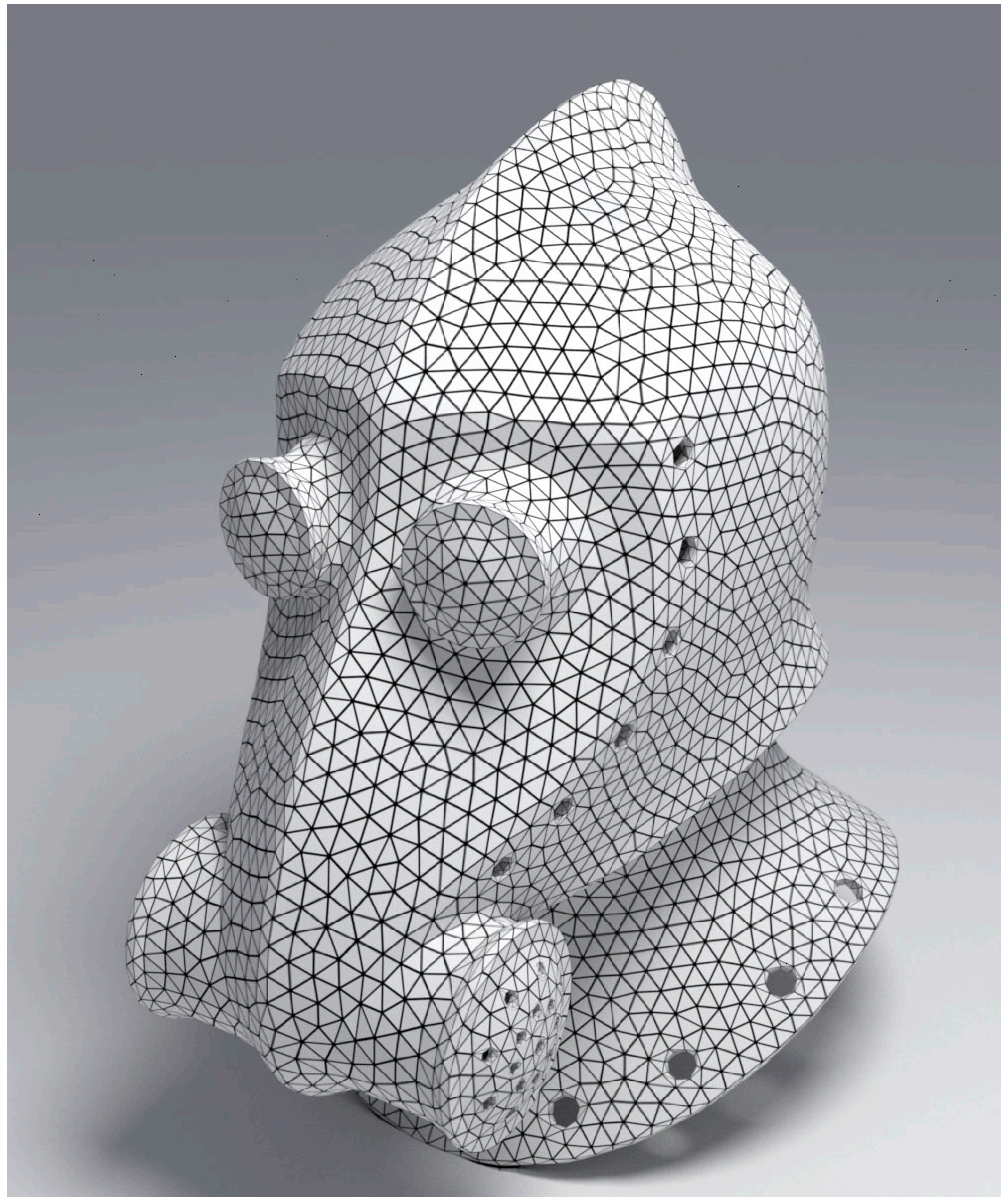


Original

Low noise
[Tong et al. 2016]

Low noise
Our method

Robustness to noise

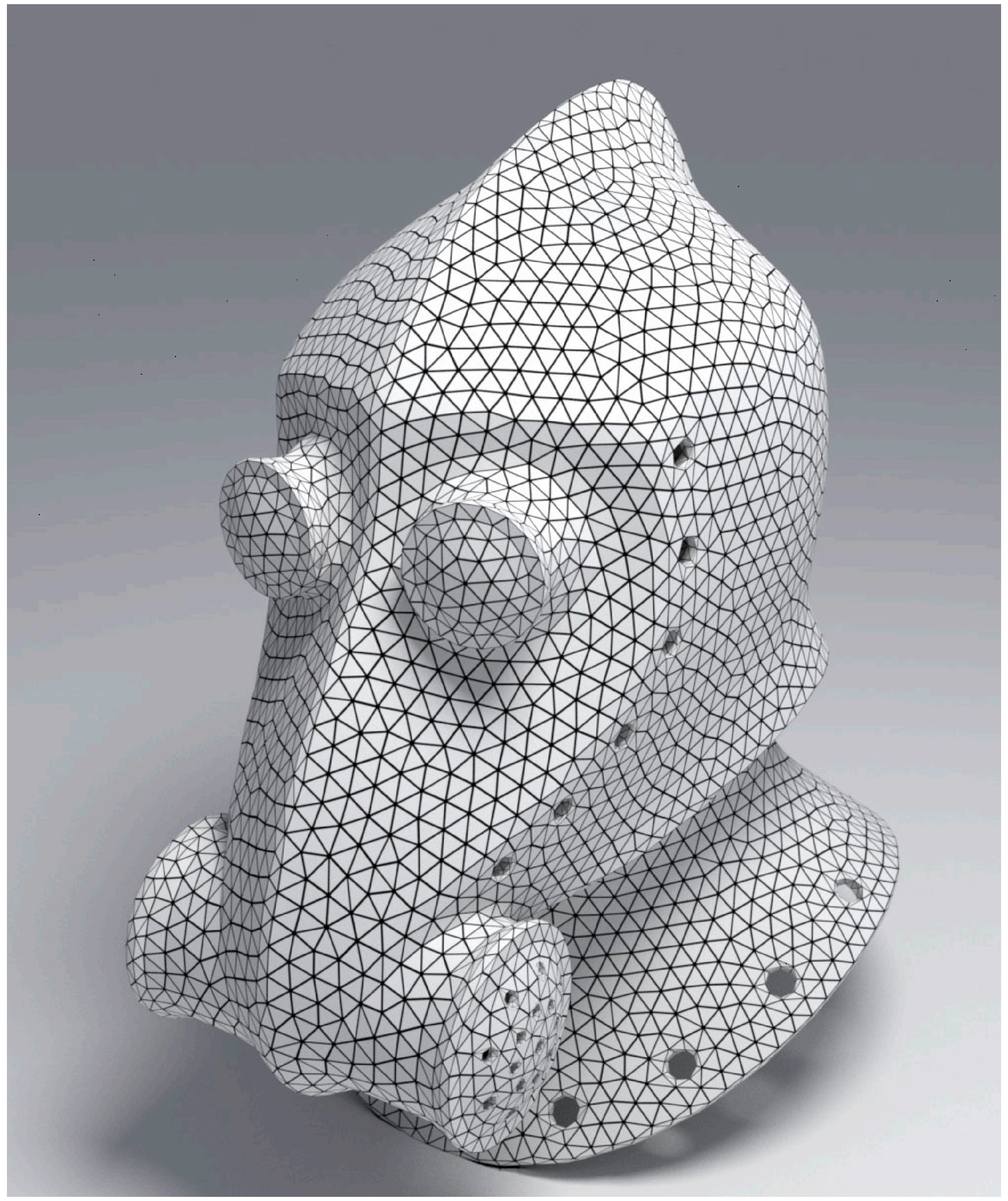


Original

Moderate noise
[Tong et al. 2016]

Moderate noise
Our method

Robustness to noise



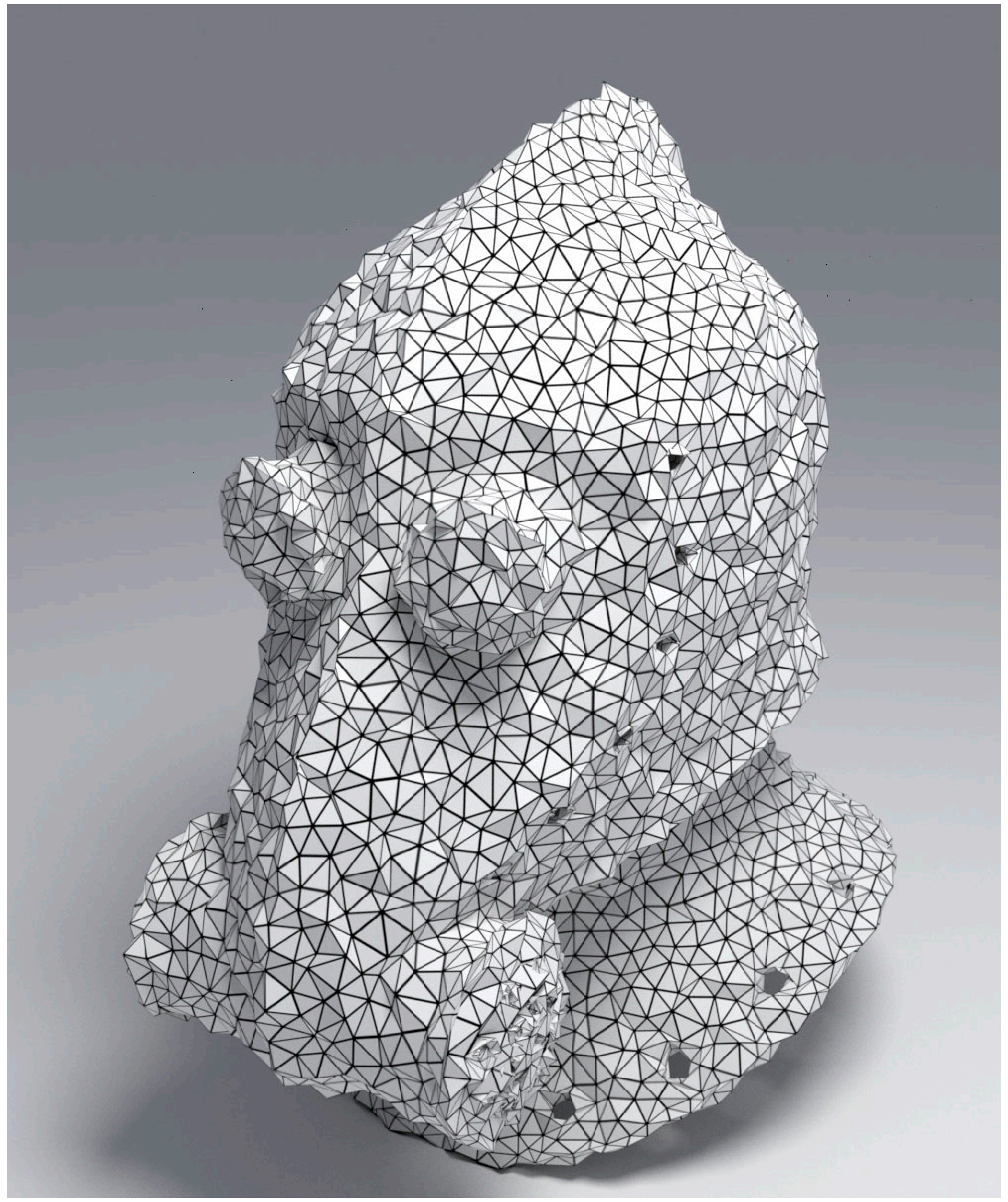
Original

Large noise
[Tong et al. 2016]

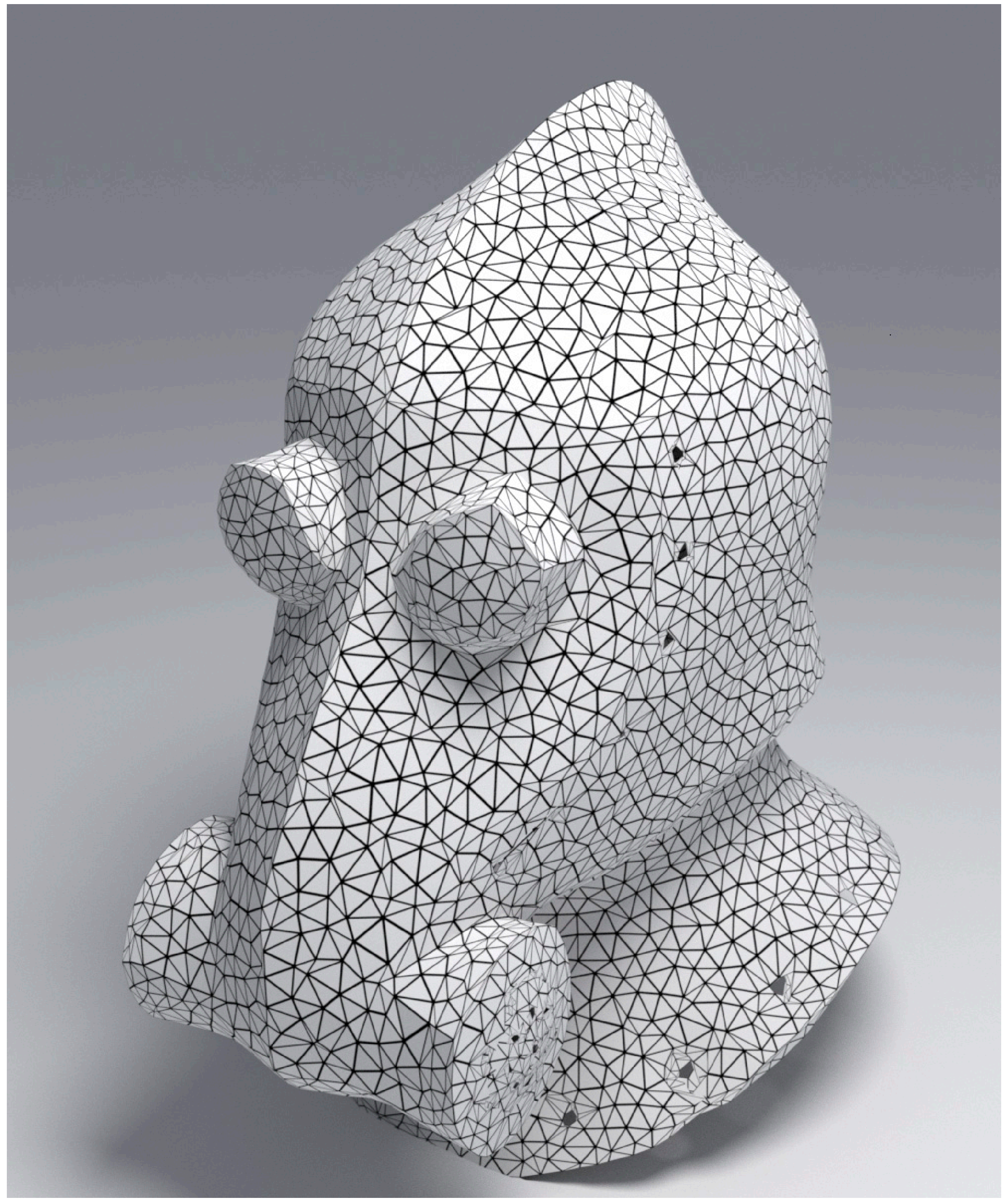
Large noise
Our method

=> More robust than state-of-the-art Mumford-Shah solvers

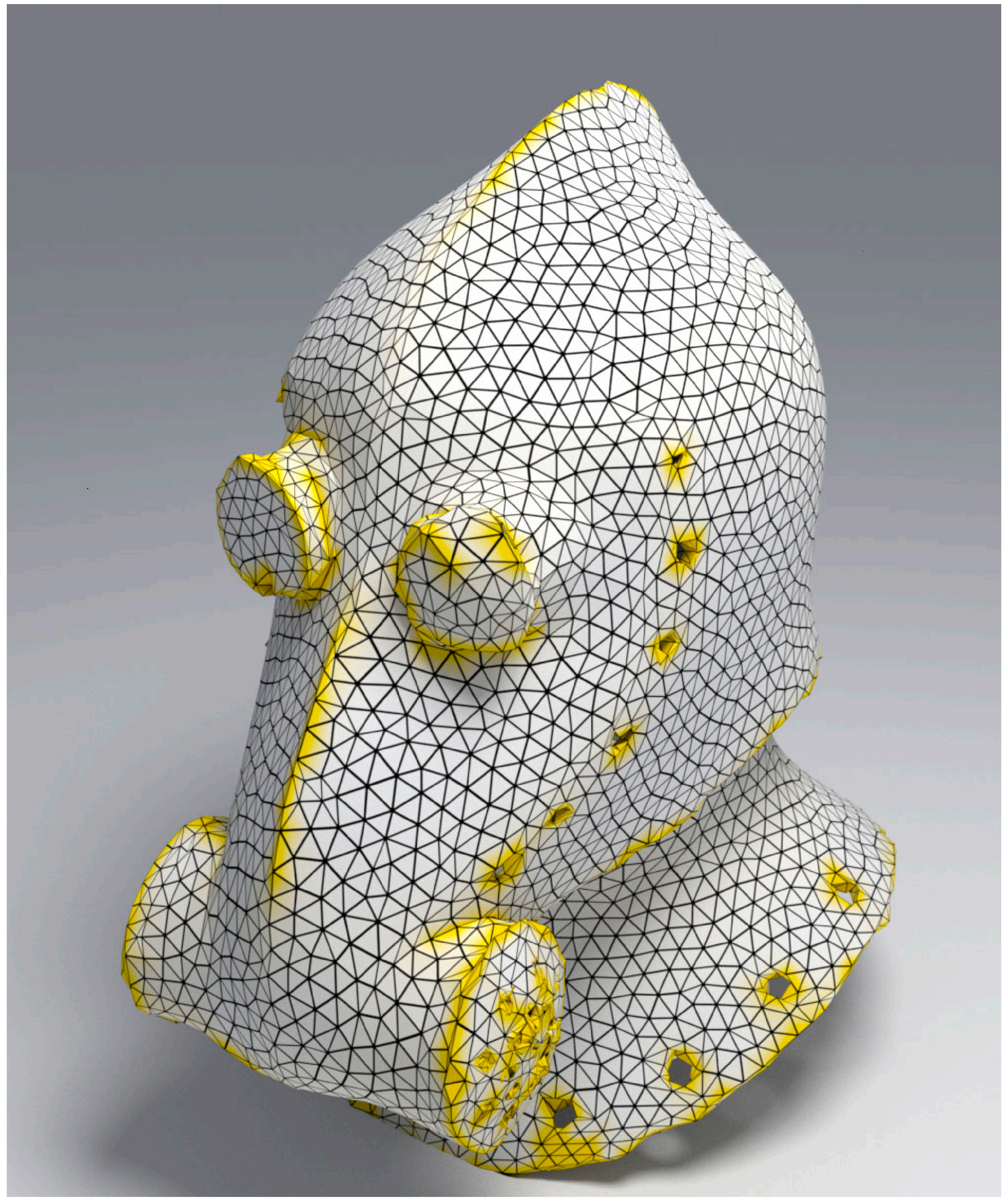
Denoising



Synthetic noise added



Guided Filtering [Zhang et al. 2015]



Our denoising

=> State-of-the-art denoising results compared with 5 methods (similar Hausdorff, sligthly better perceptual metric)

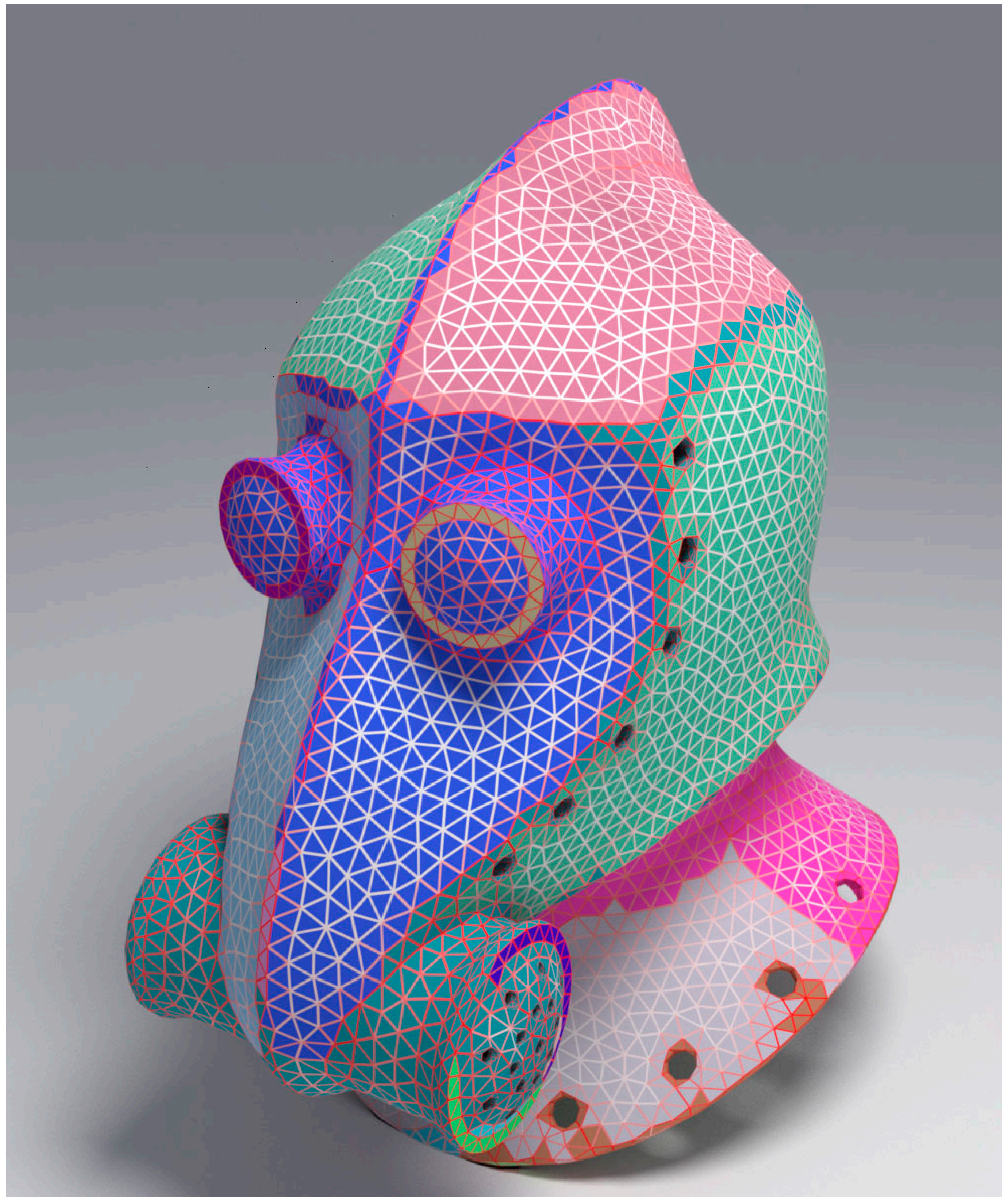
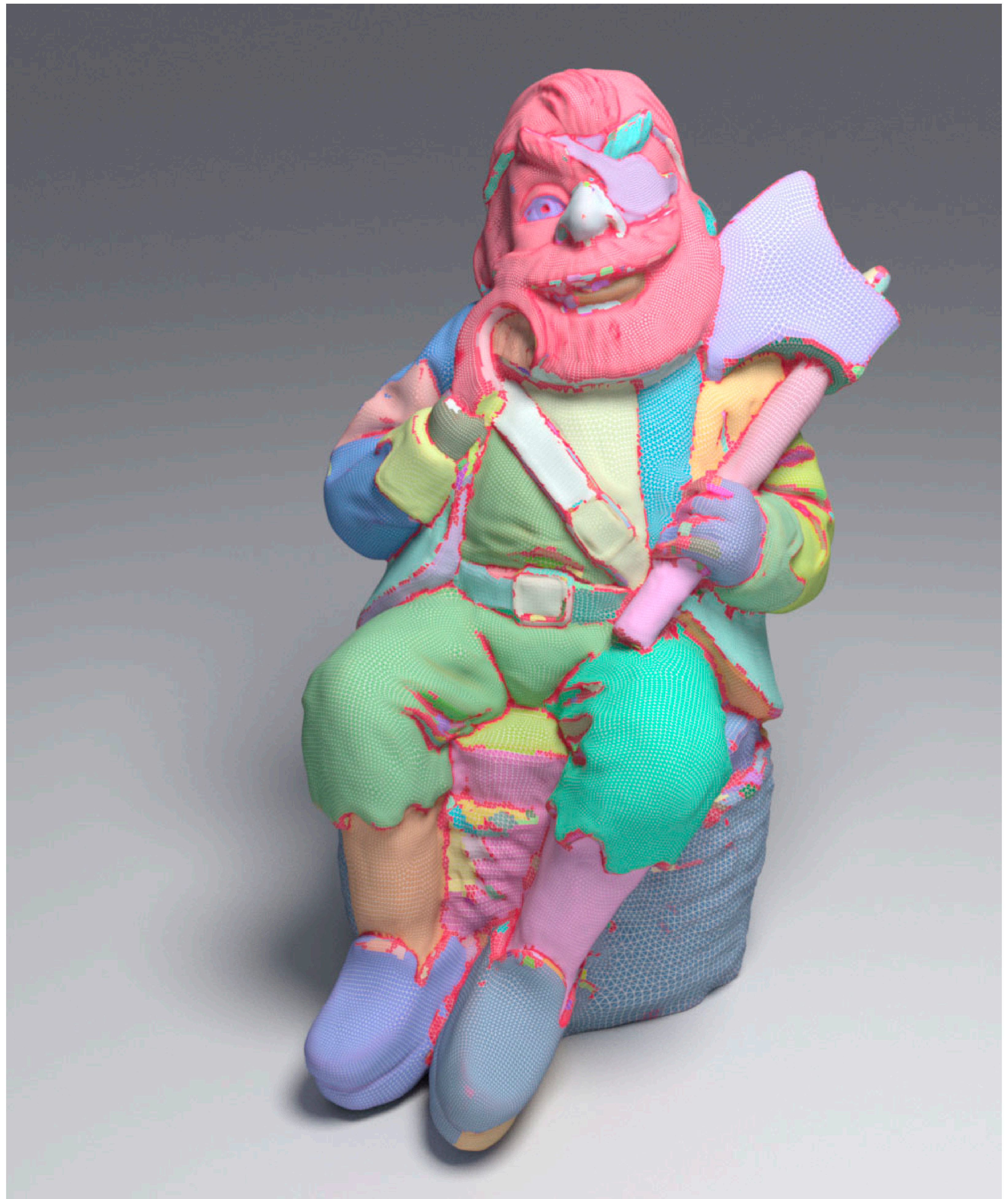
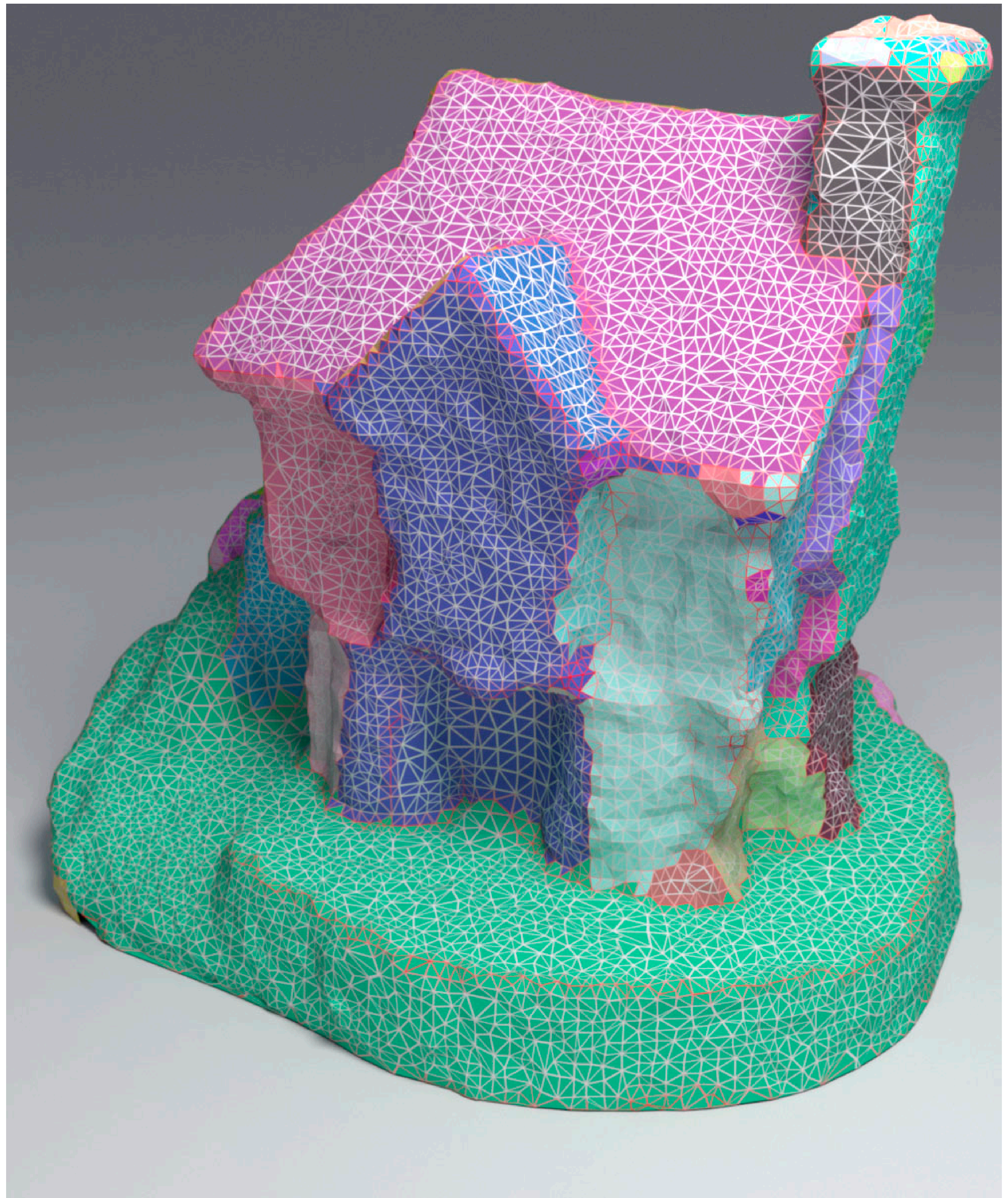
Denoising (LiDaR noise)

LiDaR scan with real noise

Guided Filtering [Zhang et al. 2015]

Our denoising

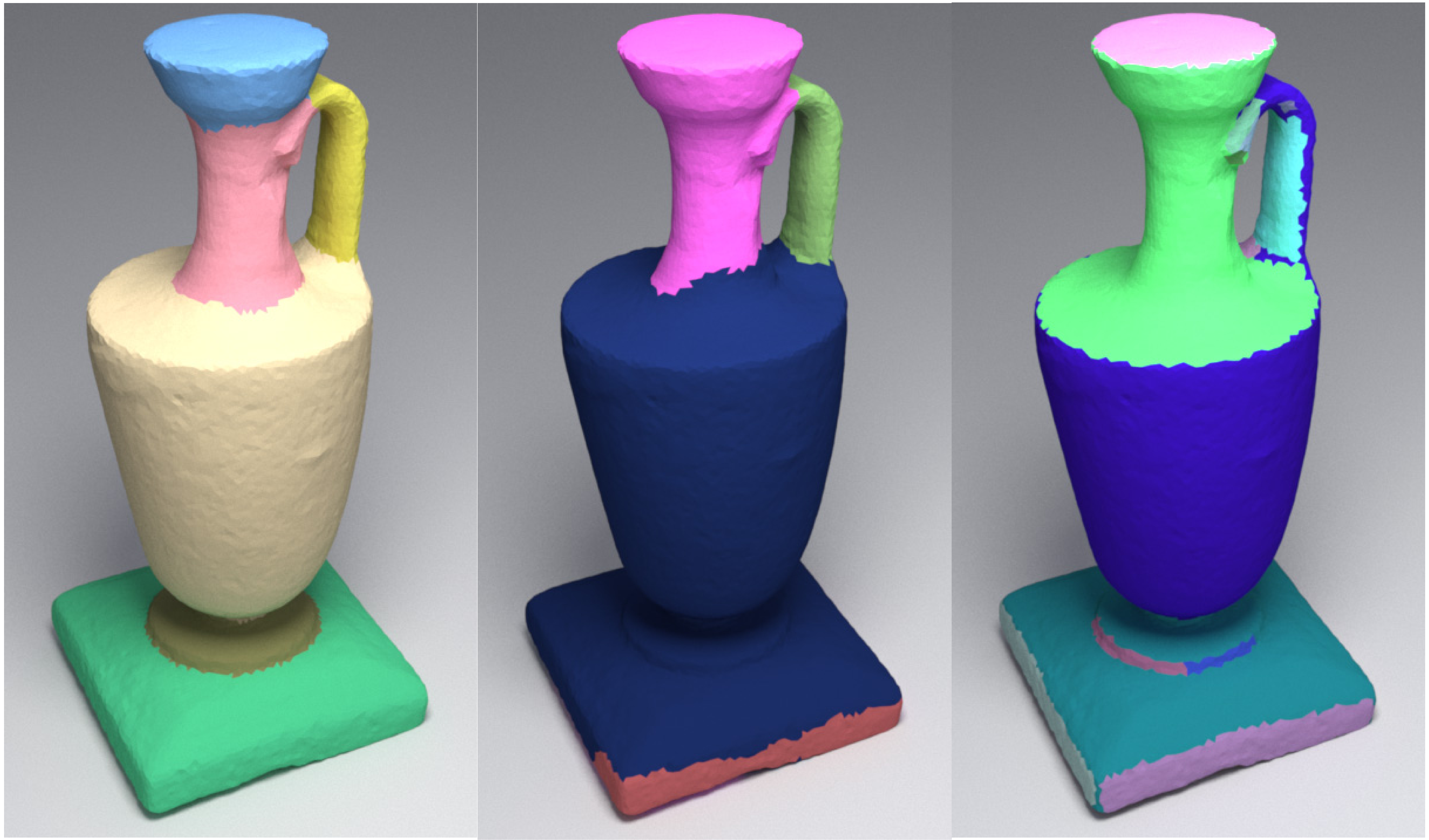
Segmentation

- Features v combined with Lifted Multicuts [Keuper et al. 2015]

Segmentation

- Provides geometric (not semantic) segmentation

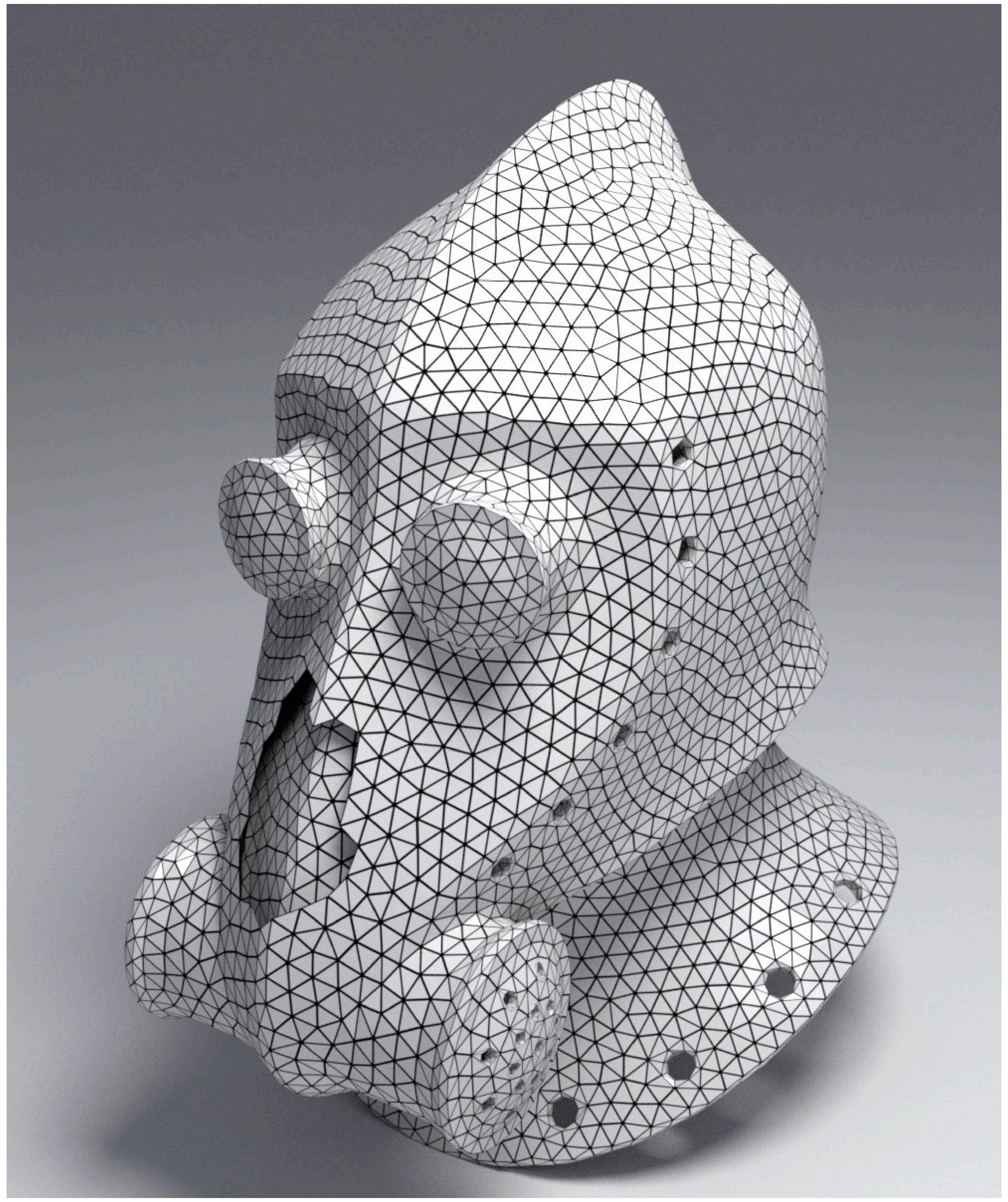


Randomized Cuts [Golovinskiy et al. 2008]

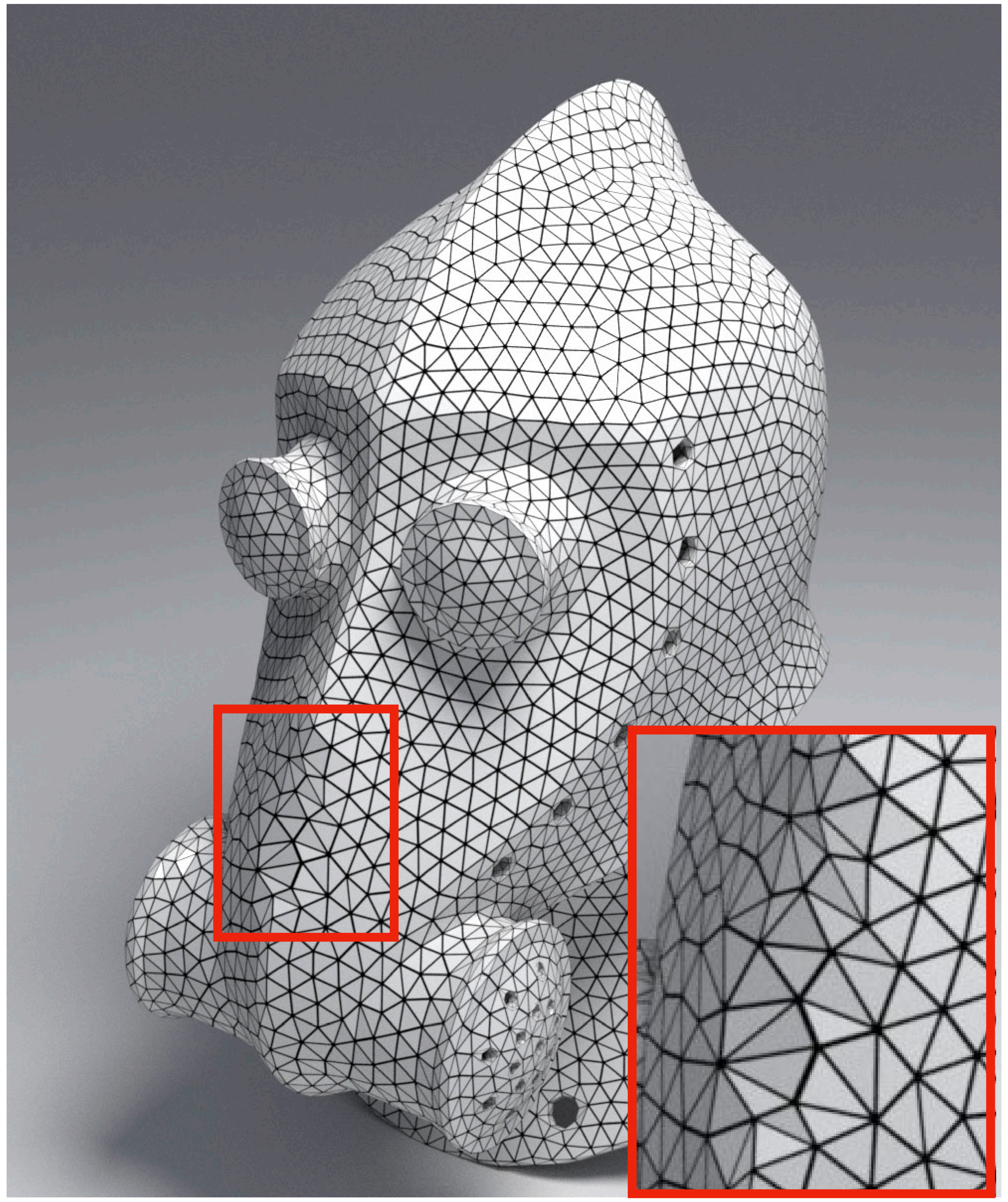
Projective ConvNet [Kalogerakis et al. 2017]

Our segmentation

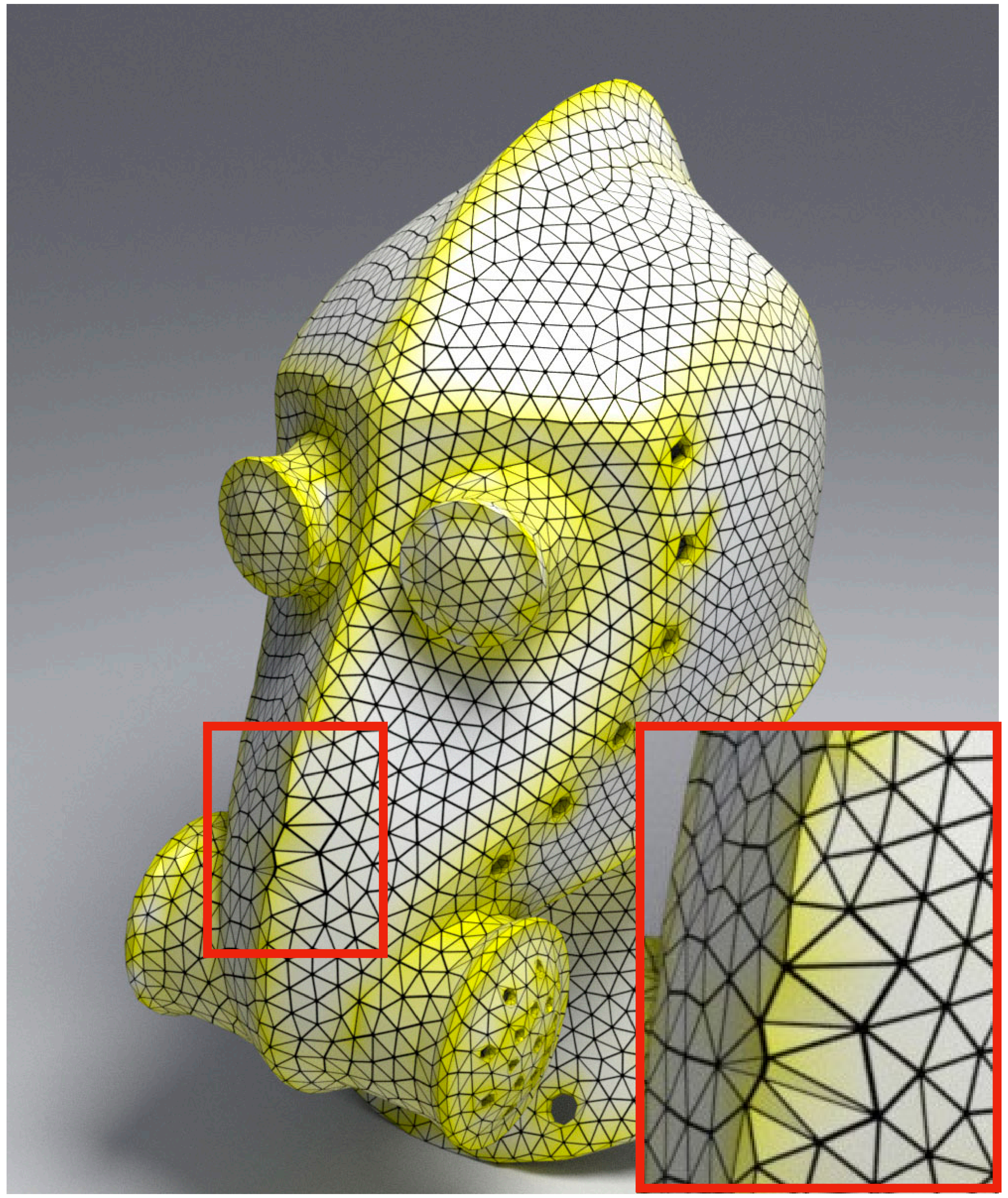
Inpainting



Hole

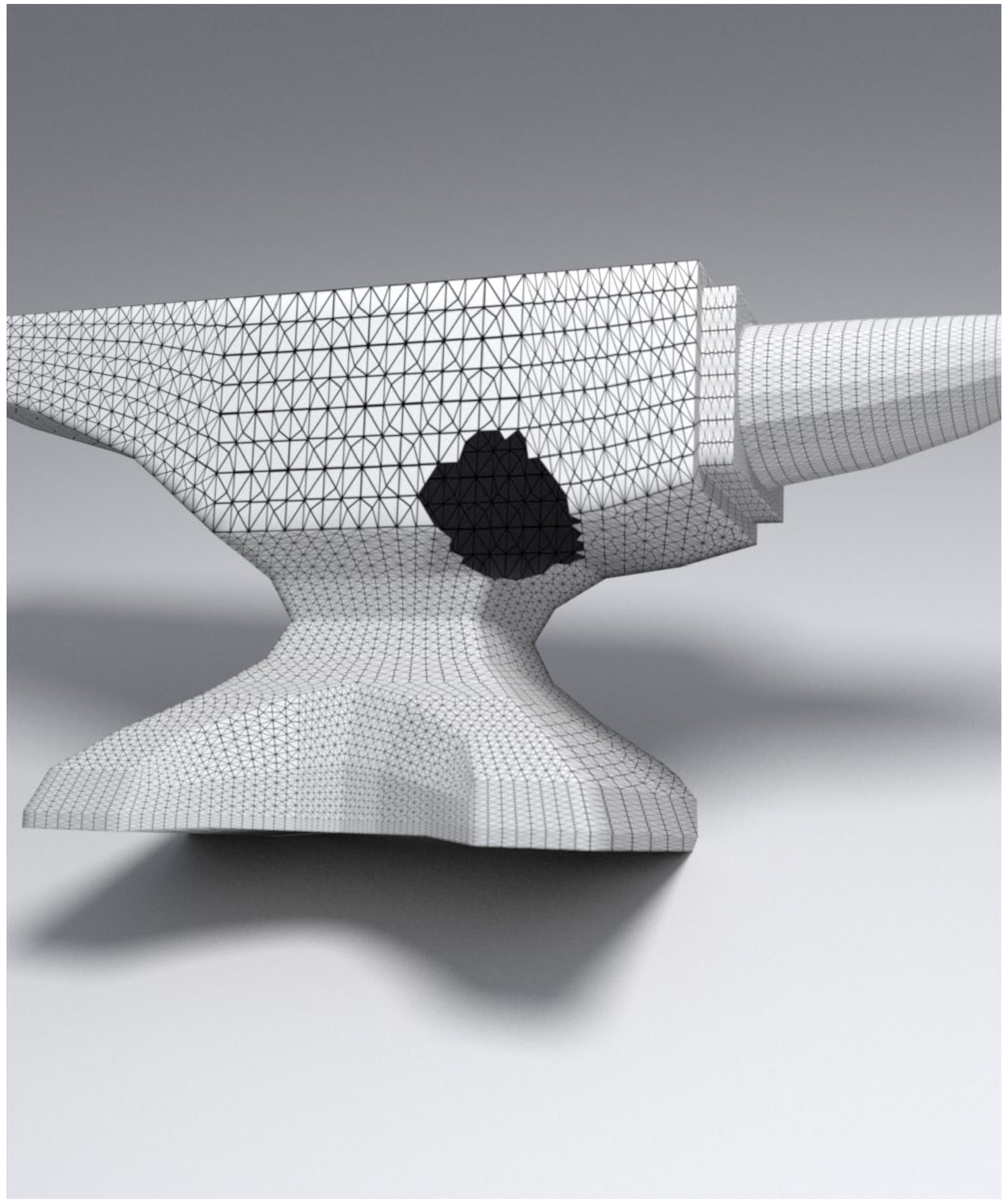


CGAL hole filling

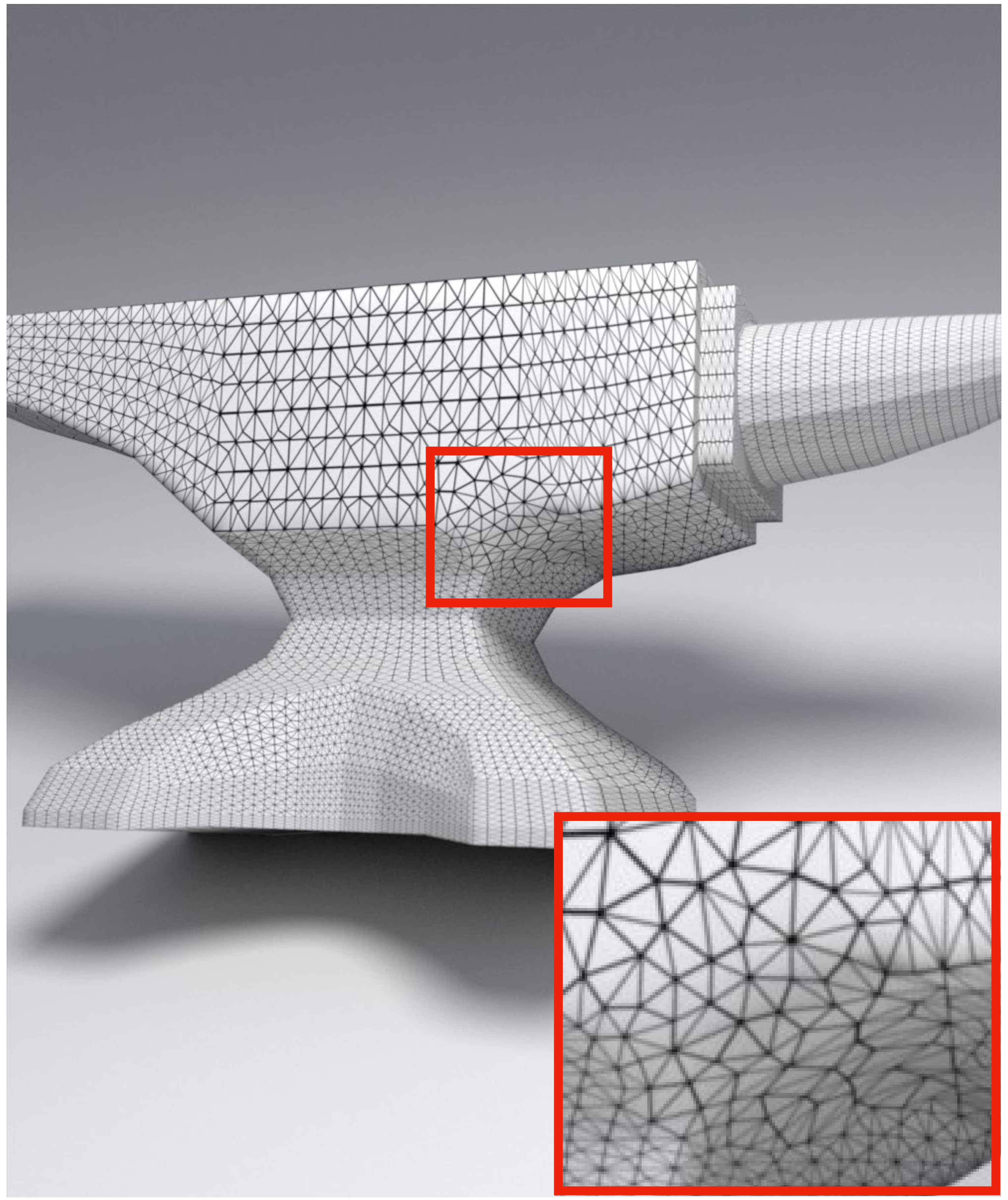


Our inpainting (CGAL + AT)

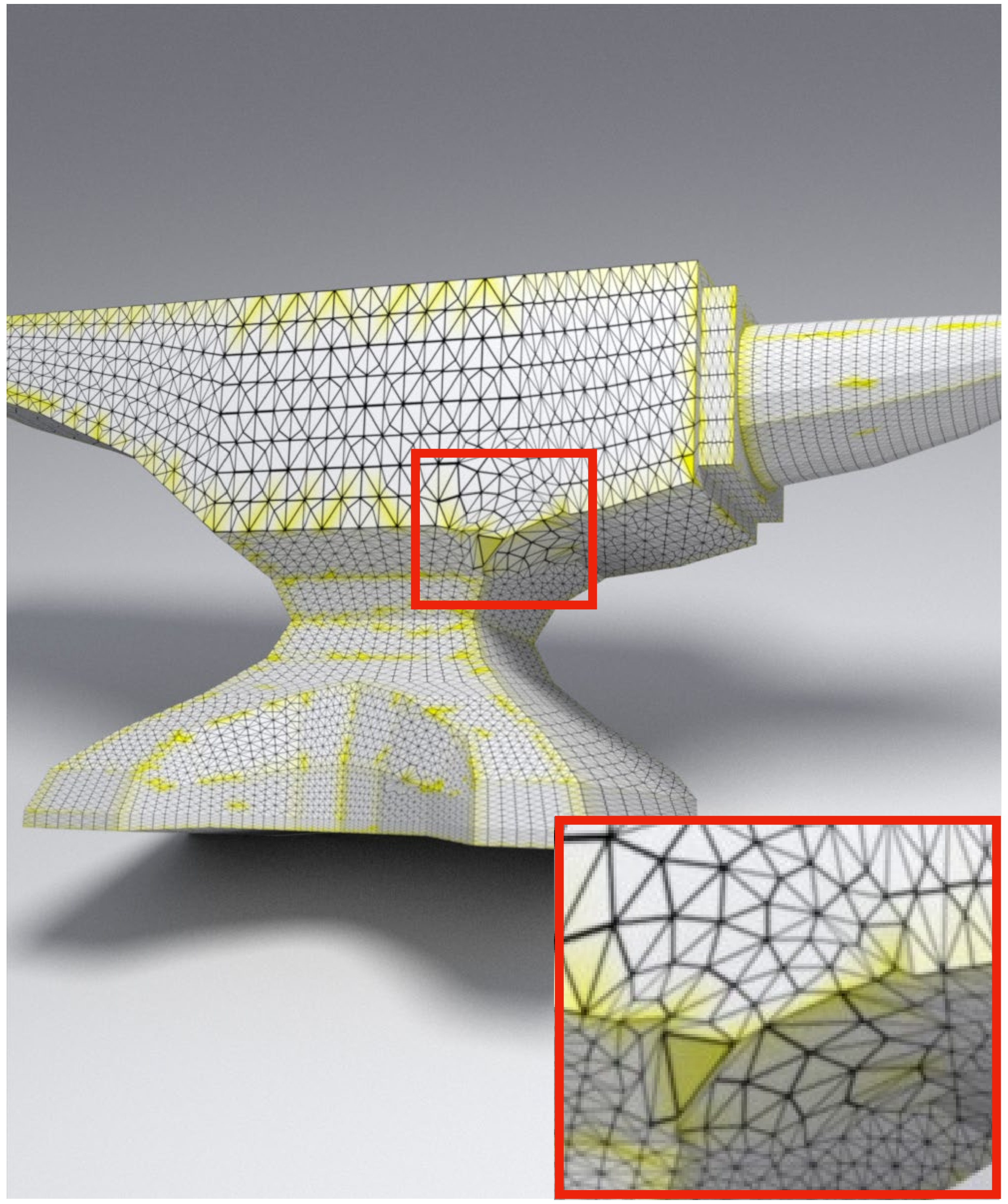
Inpainting



Hole



CGAL hole filling

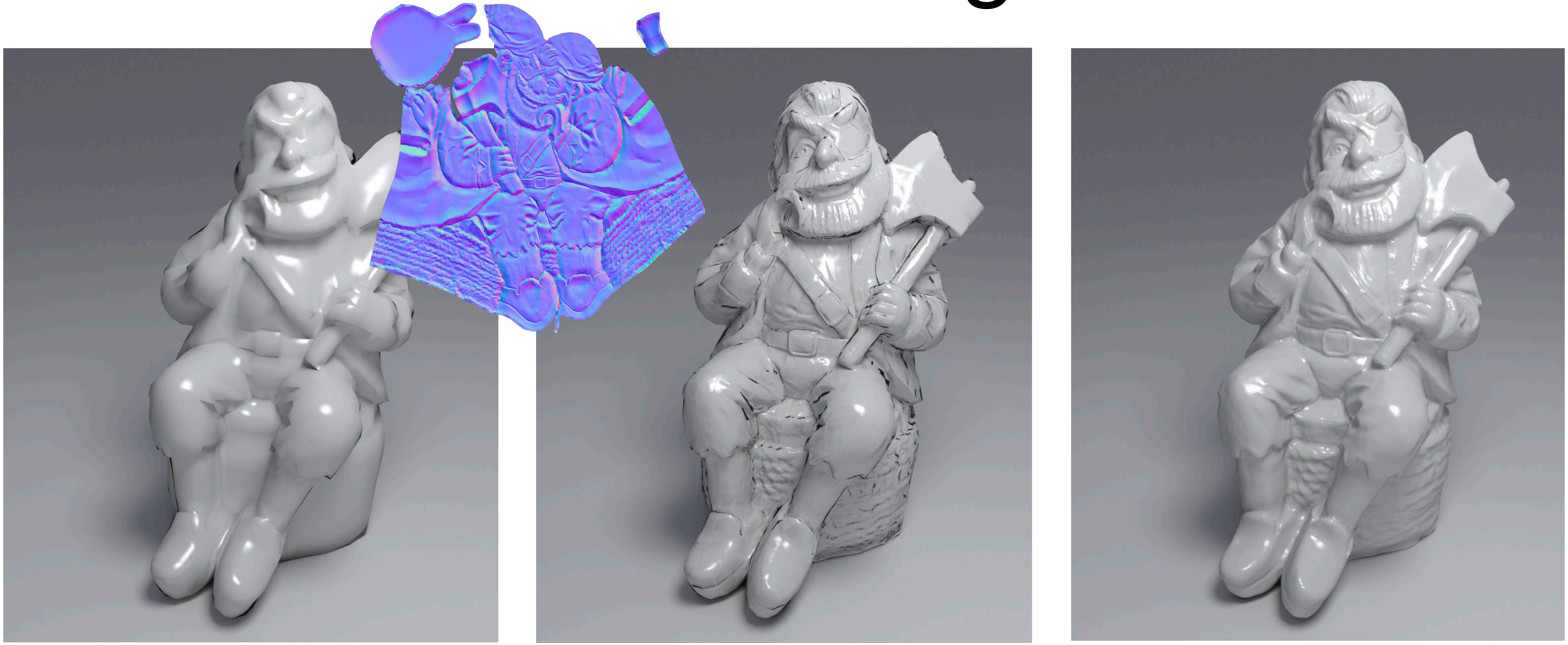


Our inpainting (CGAL + AT)

Embossing

Original mesh | **Normal mapping** | **Our embossing (no AT needed)**

Embossing



Original mesh

Normal mapping

Our embossing (no AT needed)

Conclusions

- Generic edge-aware framework
- AT runs within tens to hundreds of seconds for 10--300k triangle meshes
- Projection runs within tens of seconds for 10--800k triangle meshes
- State-of-the-art for denoising, produces geometric segmentations, improves inpainting. Embossing does not require AT.
- Source code at: https://github.com/dcoeurjo/MS-AT-MeshProcessing