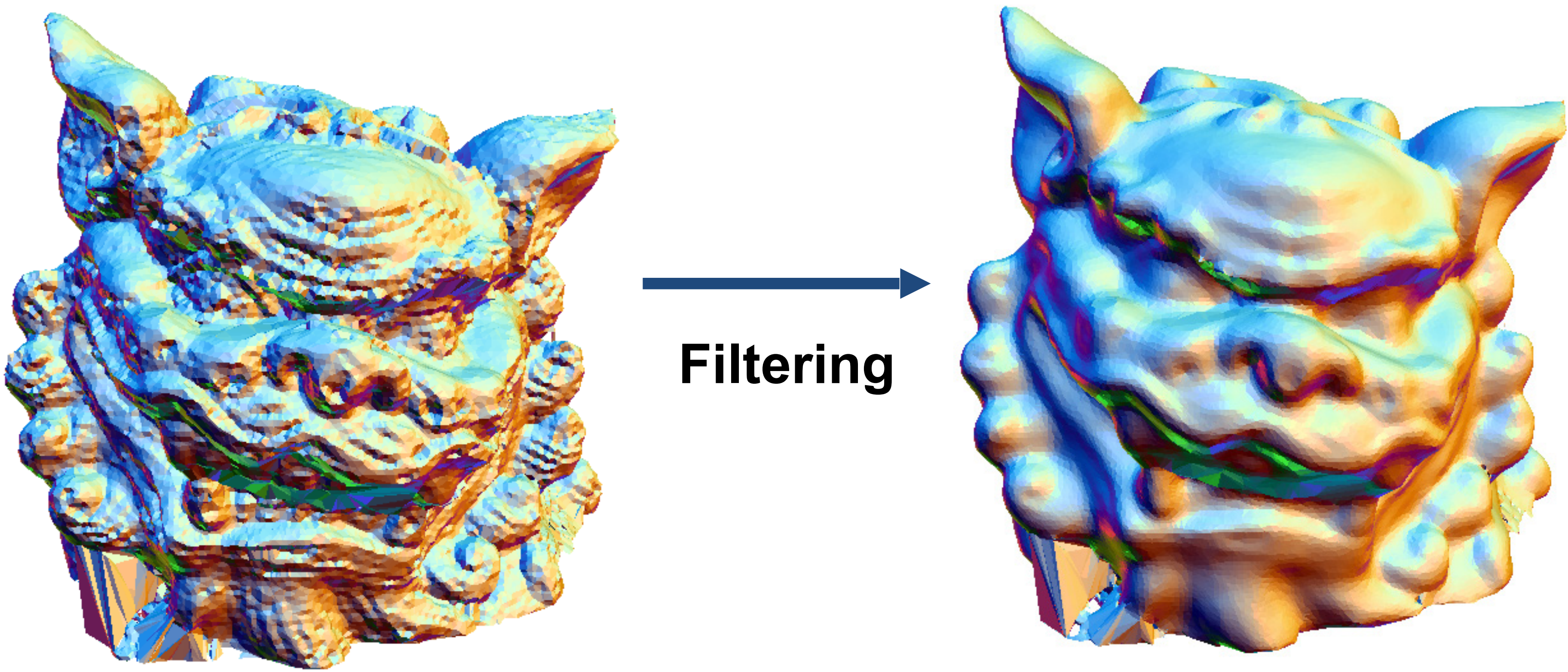
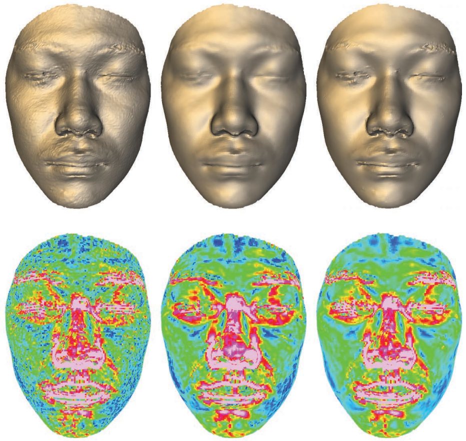
Guided Mesh Normal Filtering

Wangyu Zhang	USTC
Bailin Deng	EPFL, University of Hull
Juyong Zhang	USTC
Sofien Bouaziz	EPFL
Ligang Liu	USTC

Filtering is necessary

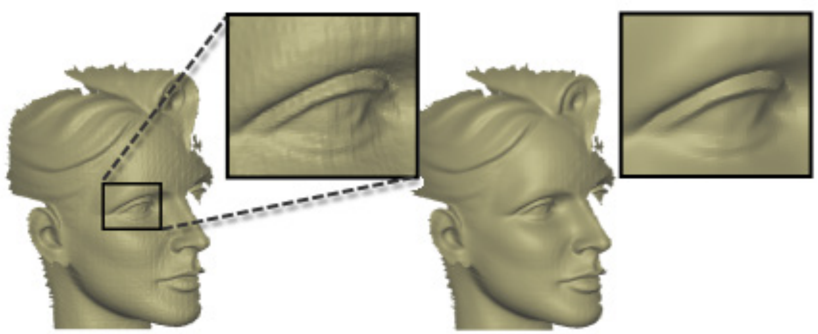


Related Work

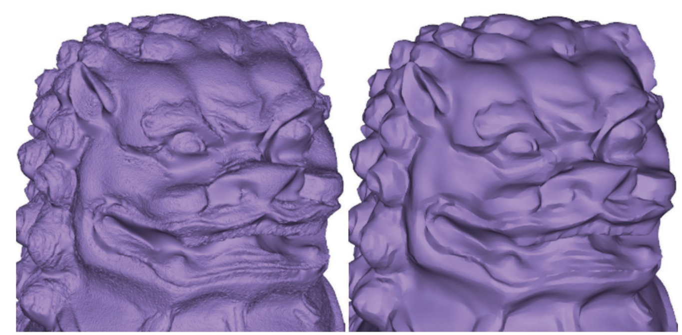


Bilateral mesh denoising
[Fleishman et al. 2003]

Non-iterative, feature-preserving mesh smoothing
[Jones et al. 2003]

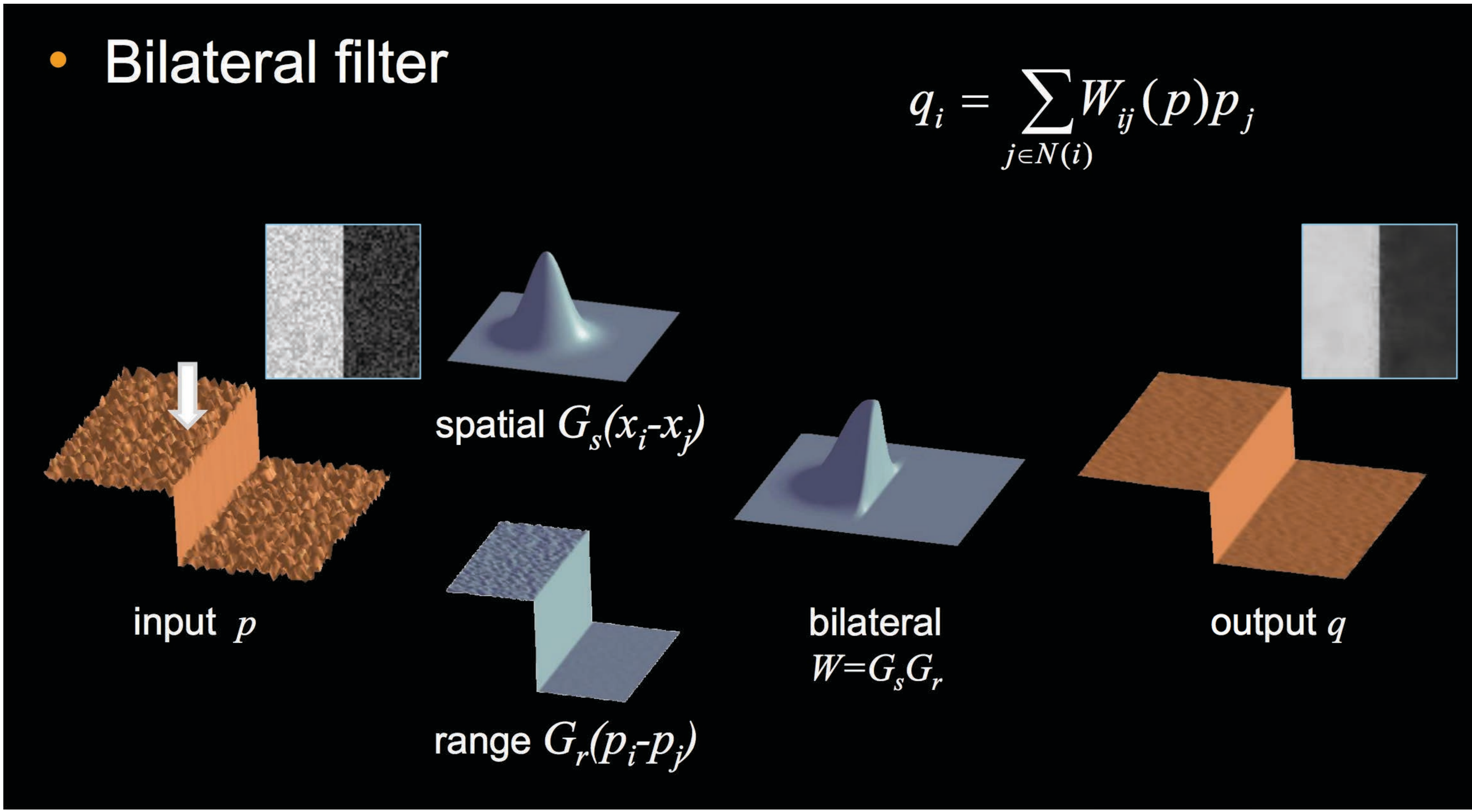


Bilateral Normal Filtering for Mesh Denoising
[Zheng et al. 2011]

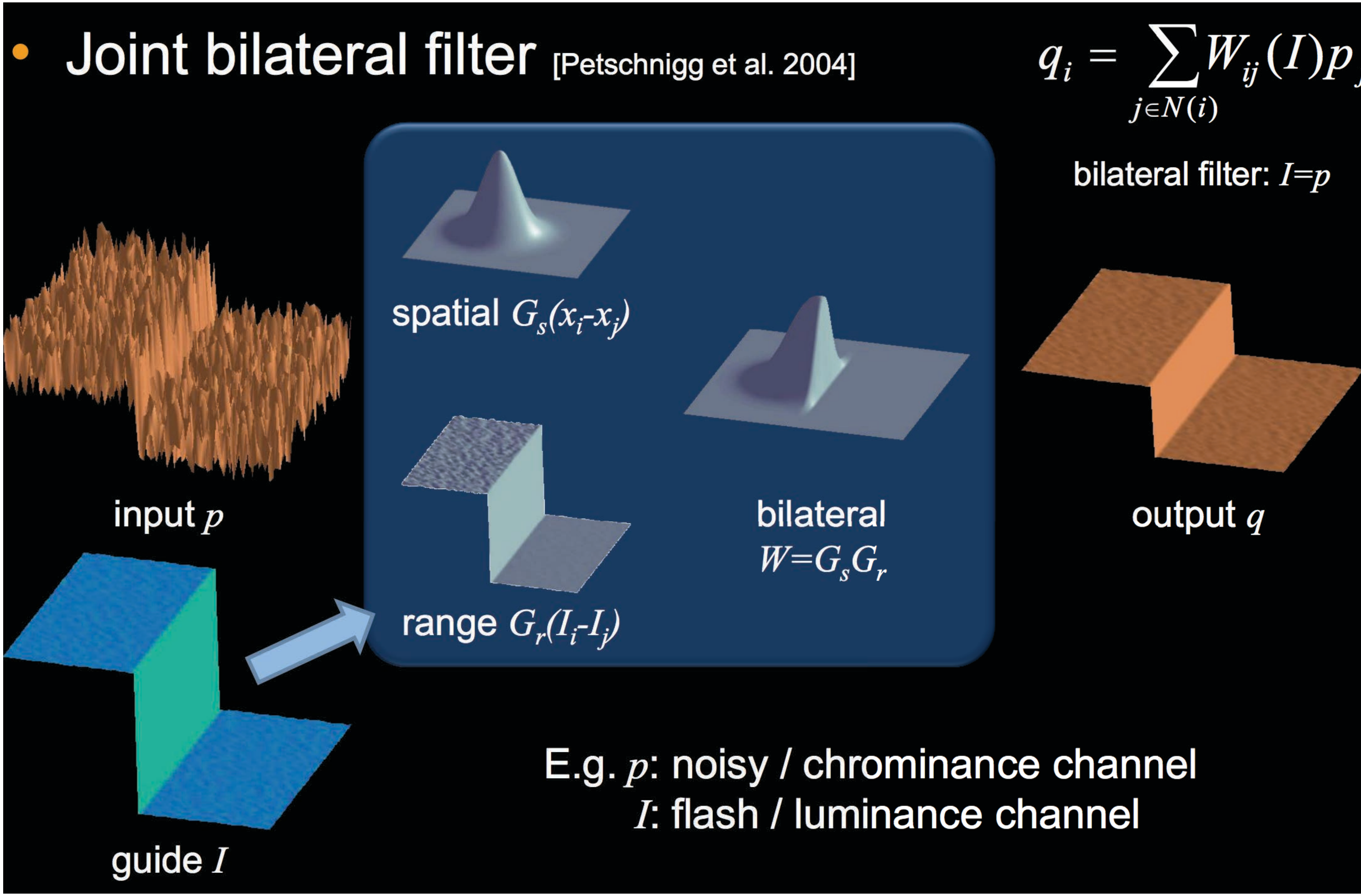


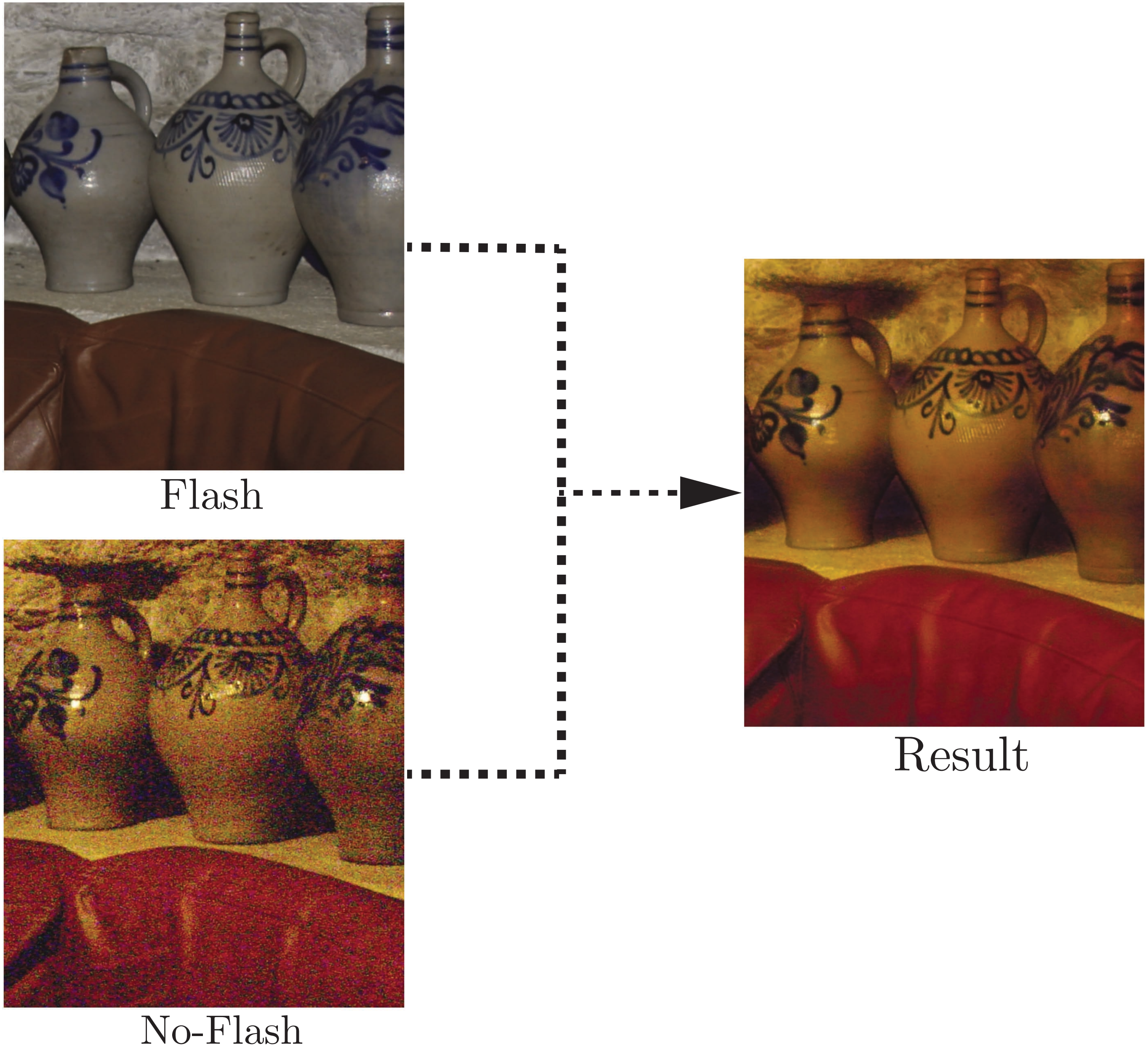
Mesh Denoising via L0 Minimization
[He & Schaefer 2013]

Bilateral filter



Joint bilateral filter



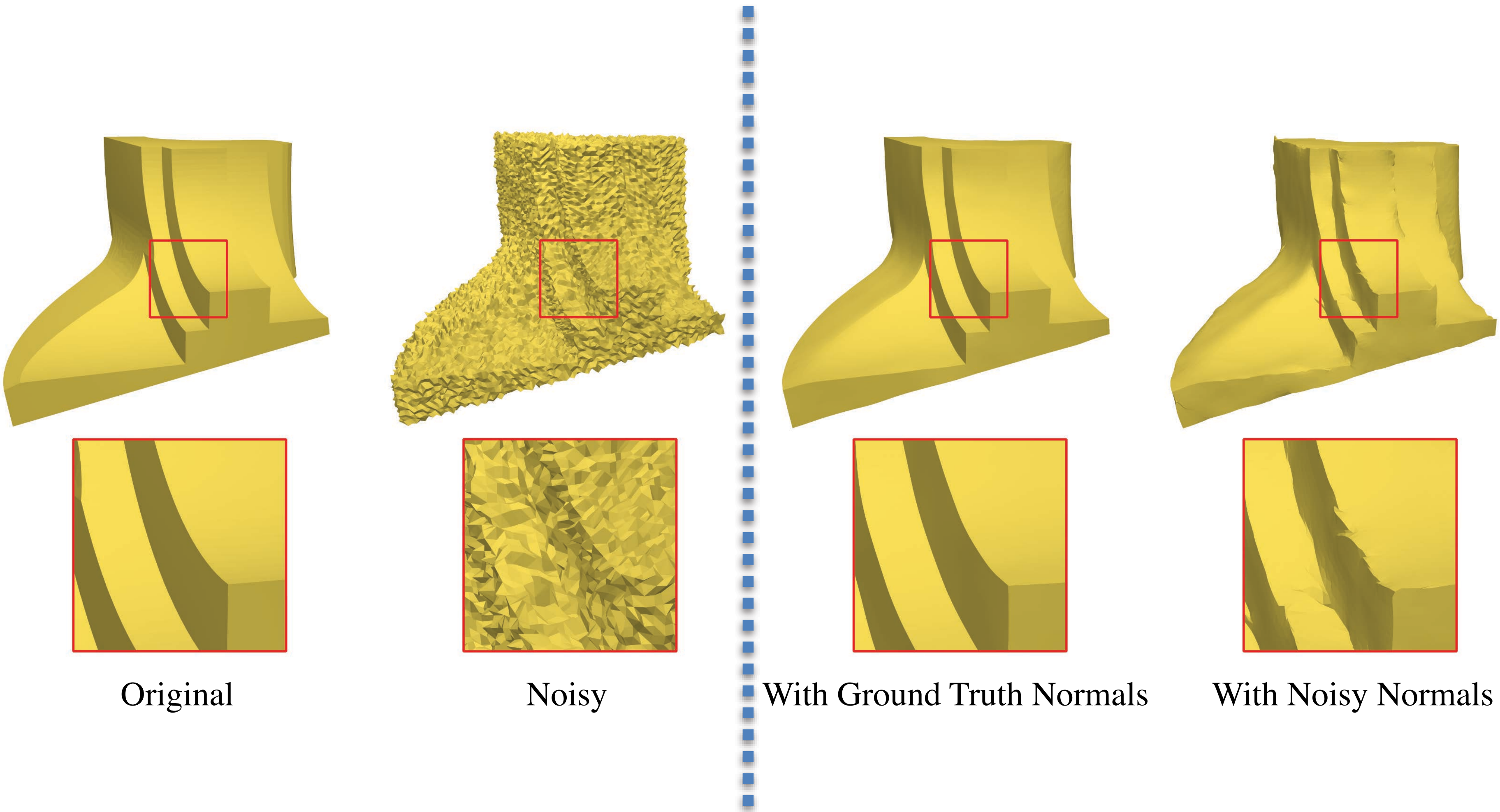

Flash
No-Flash
Result

The role of guidance

- The success of joint bilateral filtering is heavily dependent on the guidance signal.

- The guidance signal should provide a robust estimation about the features of the output signal

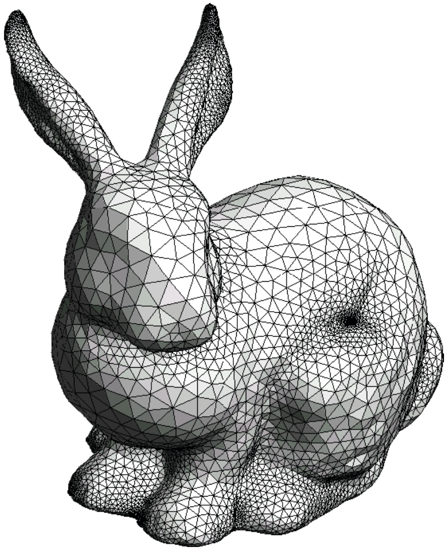
The importance of guidance



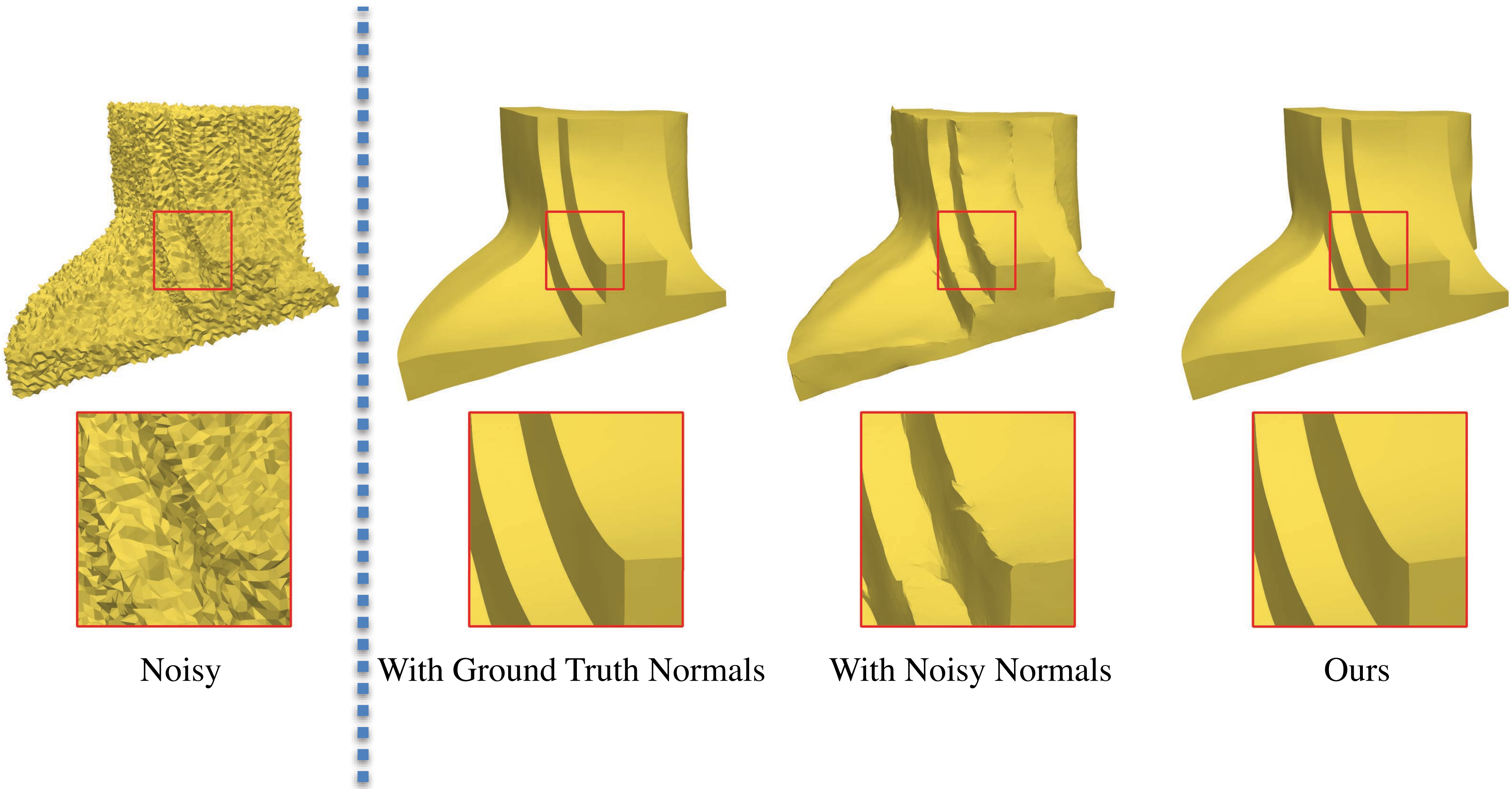
Guidance geometry

- Contrary to the case of images, such guidance geometry is not easily available from measure devices.

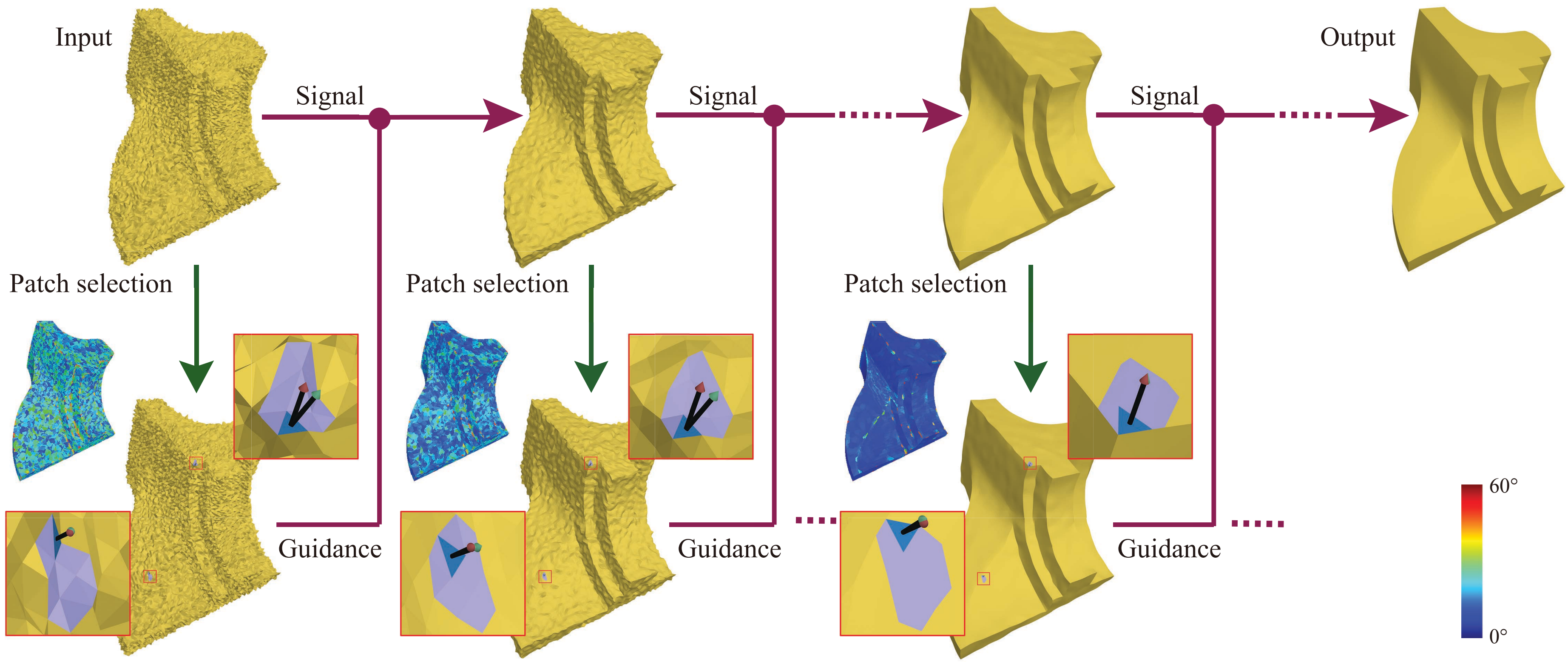
- It often has to be constructed computationally.



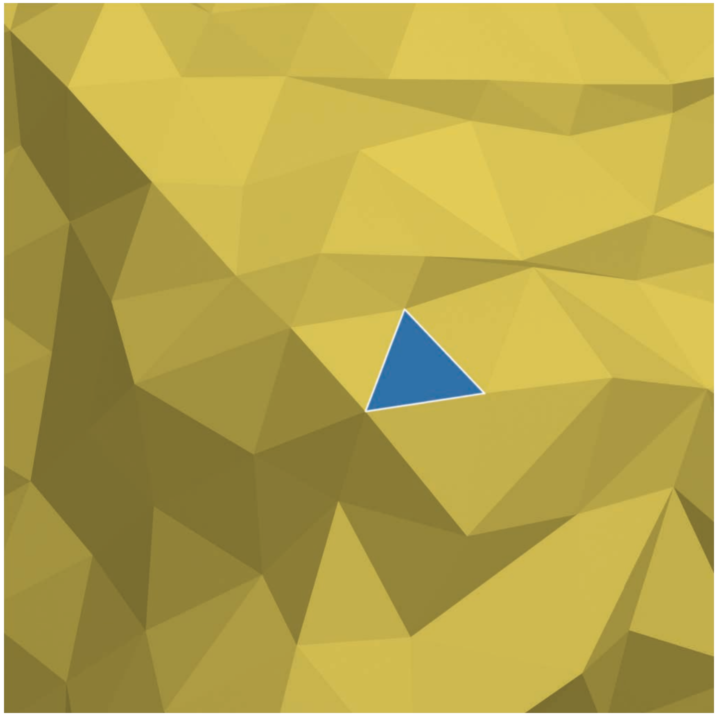
Example

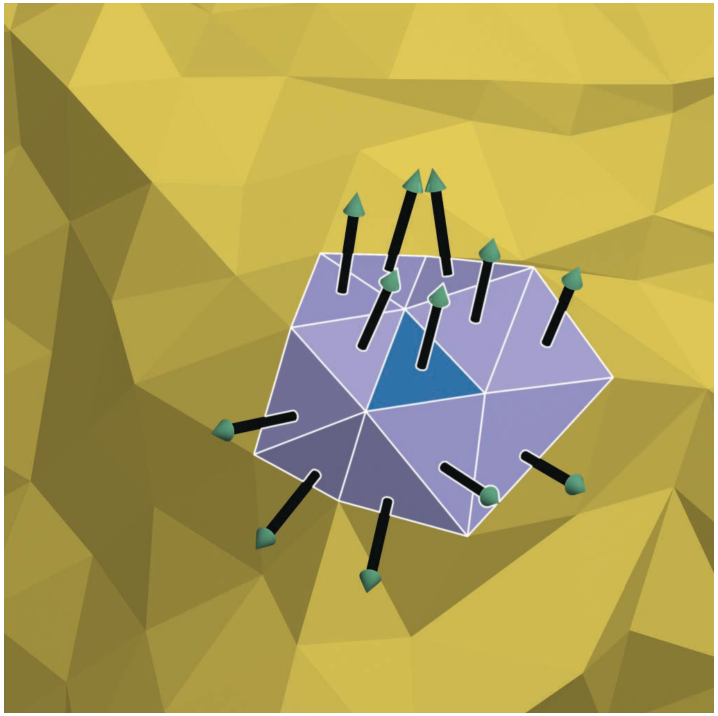


Denoising pipeline



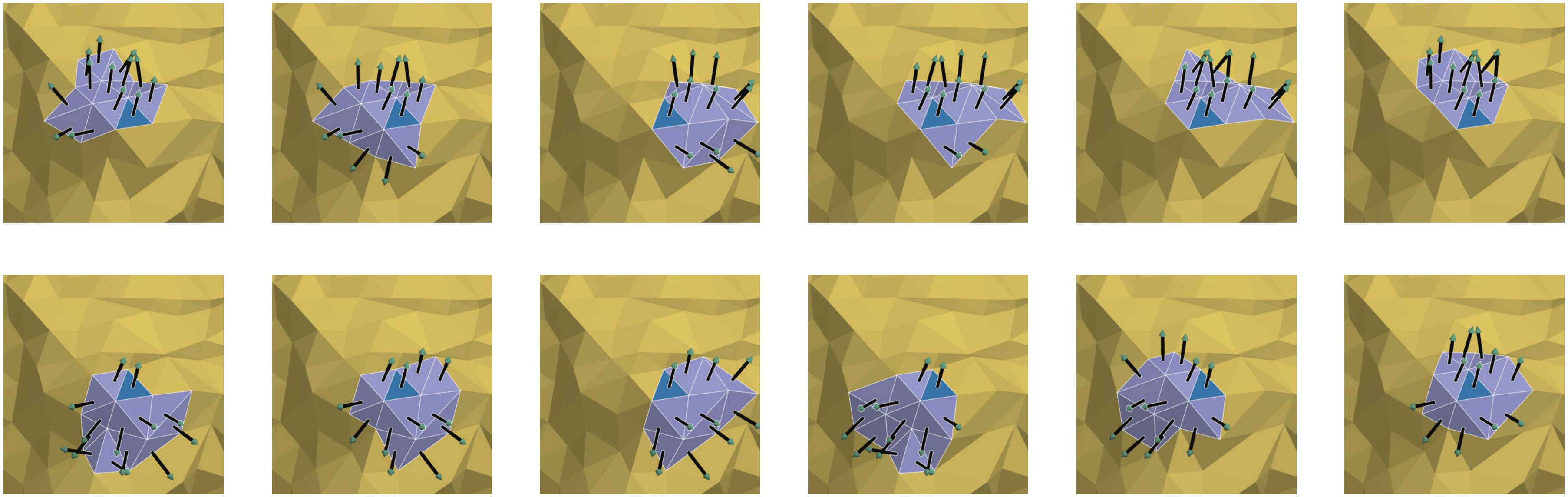
Guidance normal computation





Normal consistency

For each candidate patch $\mathcal{P} \in \mathcal{C}(f_i)$



we measure the consistency of its normals using

$$\mathcal{H}(\mathcal{P}) = \Phi(\mathcal{P}) \cdot \mathcal{R}(\mathcal{P})$$

Maximum normal difference

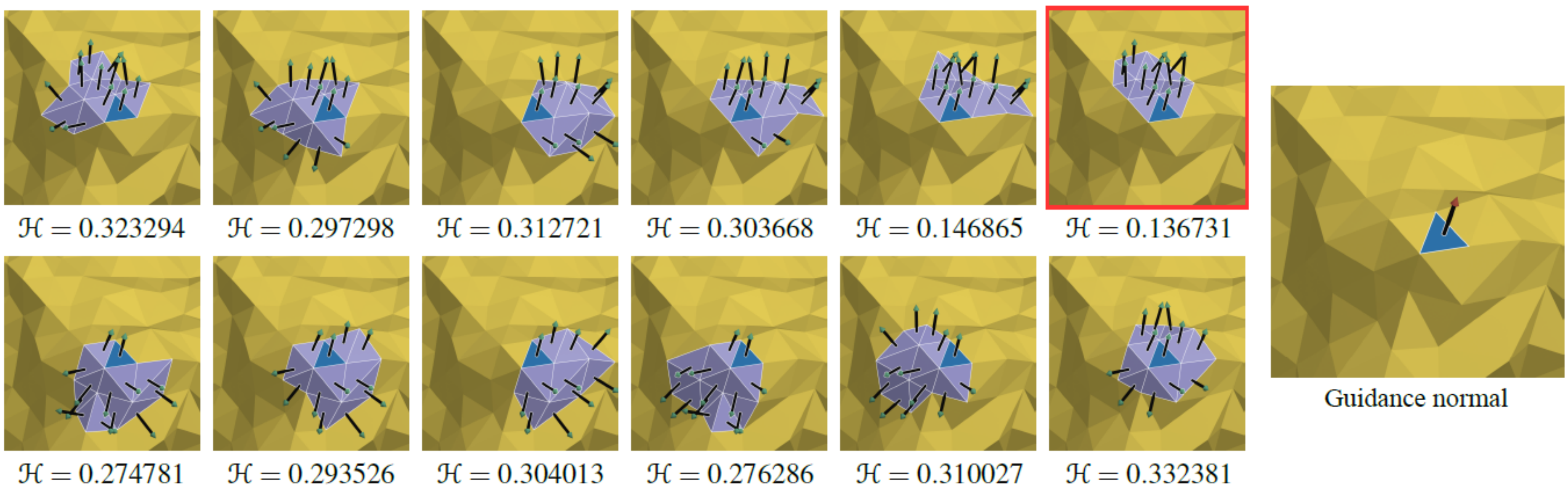
$$\Phi(\mathcal{P}) = \max_{f_j, f_k \in \mathcal{P}} \|\mathbf{n}_j - \mathbf{n}_k\|$$

Edge saliency measurement

$$\mathcal{R}(\mathcal{P}) = \frac{\max_{e_j \in E_{\mathcal{P}}} \varphi(e_j)}{\varepsilon + \sum_{e_j \in E_{\mathcal{P}}} \varphi(e_j)}$$

$$\varphi(e_j) = \|\mathbf{n}_{j_1} - \mathbf{n}_{j_2}\|$$

Patch selection



Normal filtering

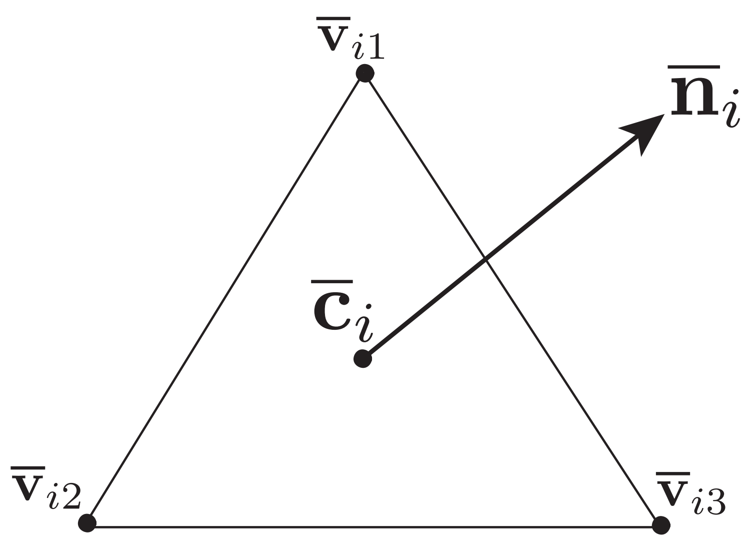
- Our normal filtering computes a new unit normal for each face via joint bilateral filter:

$$\overline{\mathbf{n}}_i = \frac{1}{W_i} \sum_{f_j \in \mathcal{N}_i} A_j \; K_s(\mathbf{c}_i, \mathbf{c}_j) \; K_r(\mathbf{g}_i, \mathbf{g}_j) \; \mathbf{n}_j$$

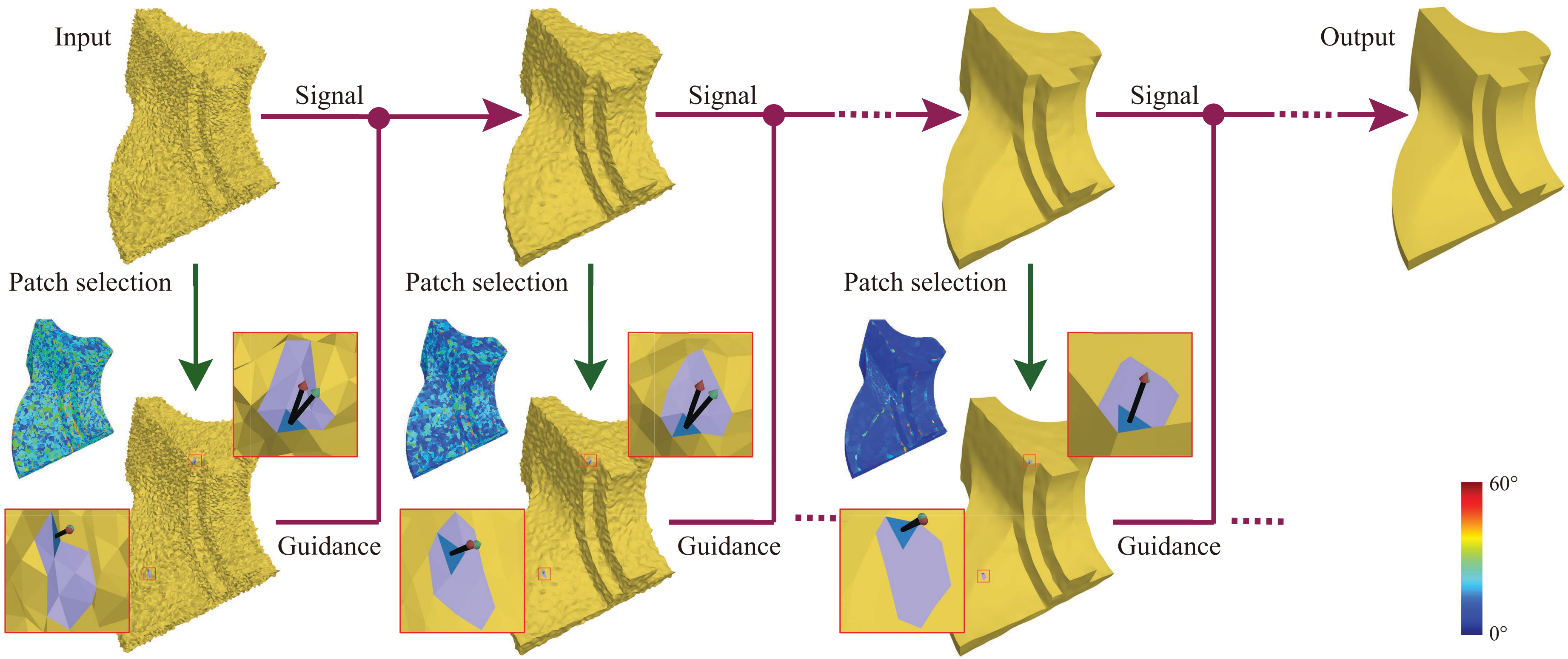
Updating vertices

- Based on the filtering face normals, the vertex positions are updated by minimizing the ℓ_2 error of the compatibility conditions:

$$\overline{\mathbf{n}}_i \cdot (\overline{\mathbf{v}}_{i_k} - \overline{\mathbf{c}}_i) = 0 \ \ (k = 1, 2, 3)$$



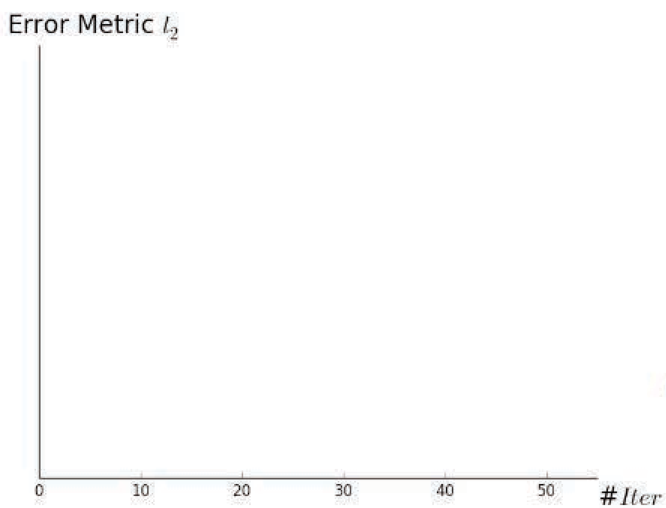
Recap: pipeline

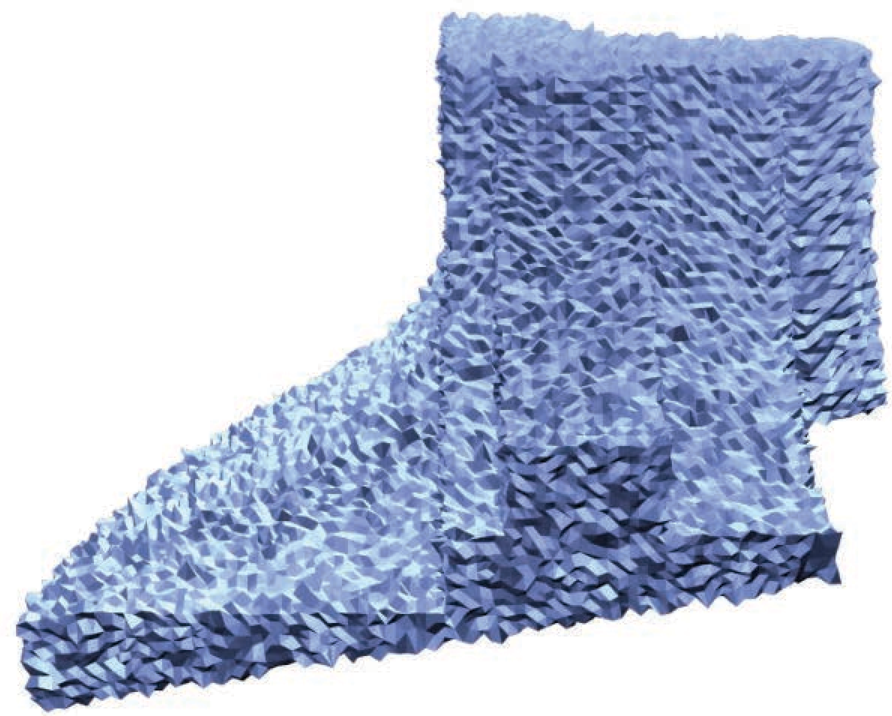


Denoising process

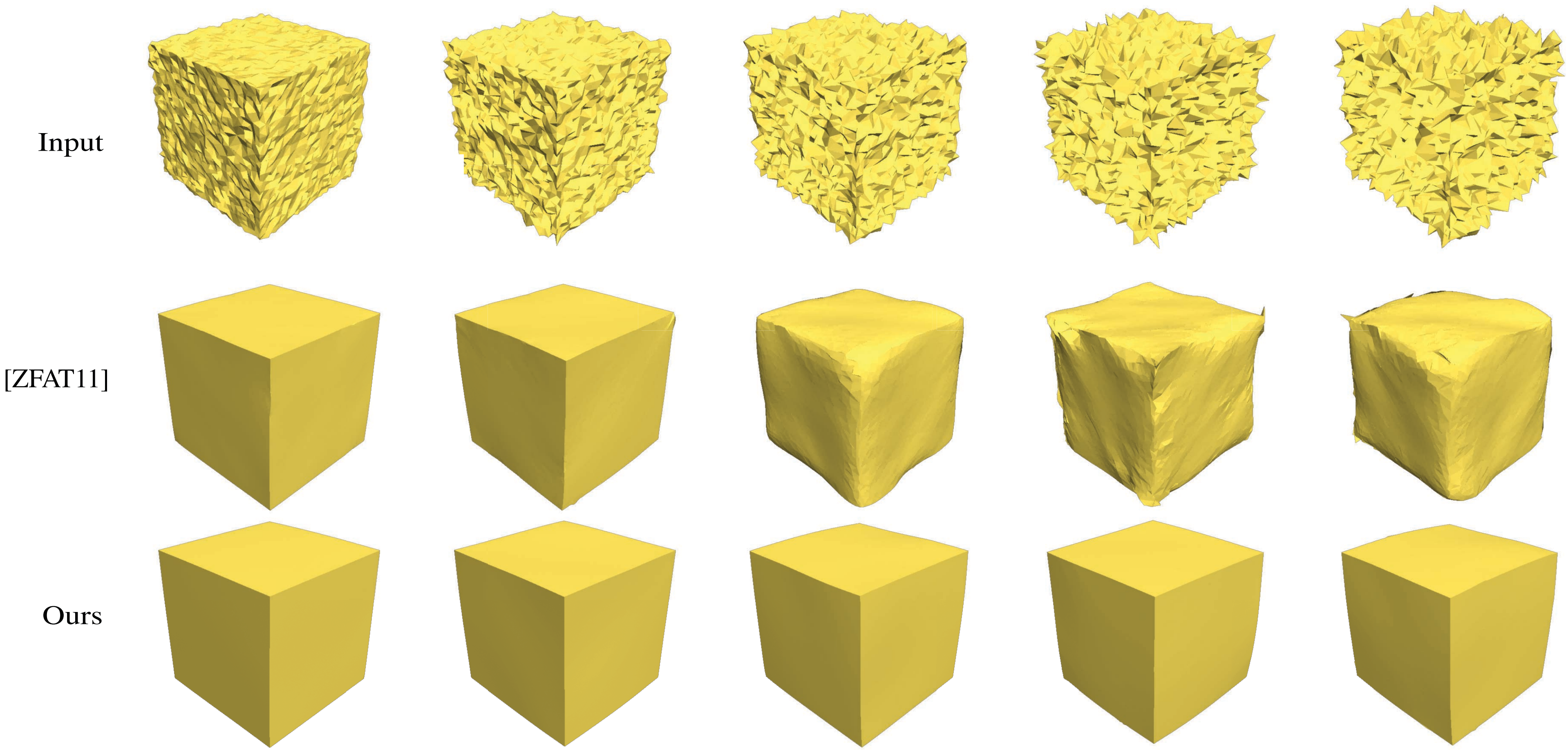
Denoising Process

Input

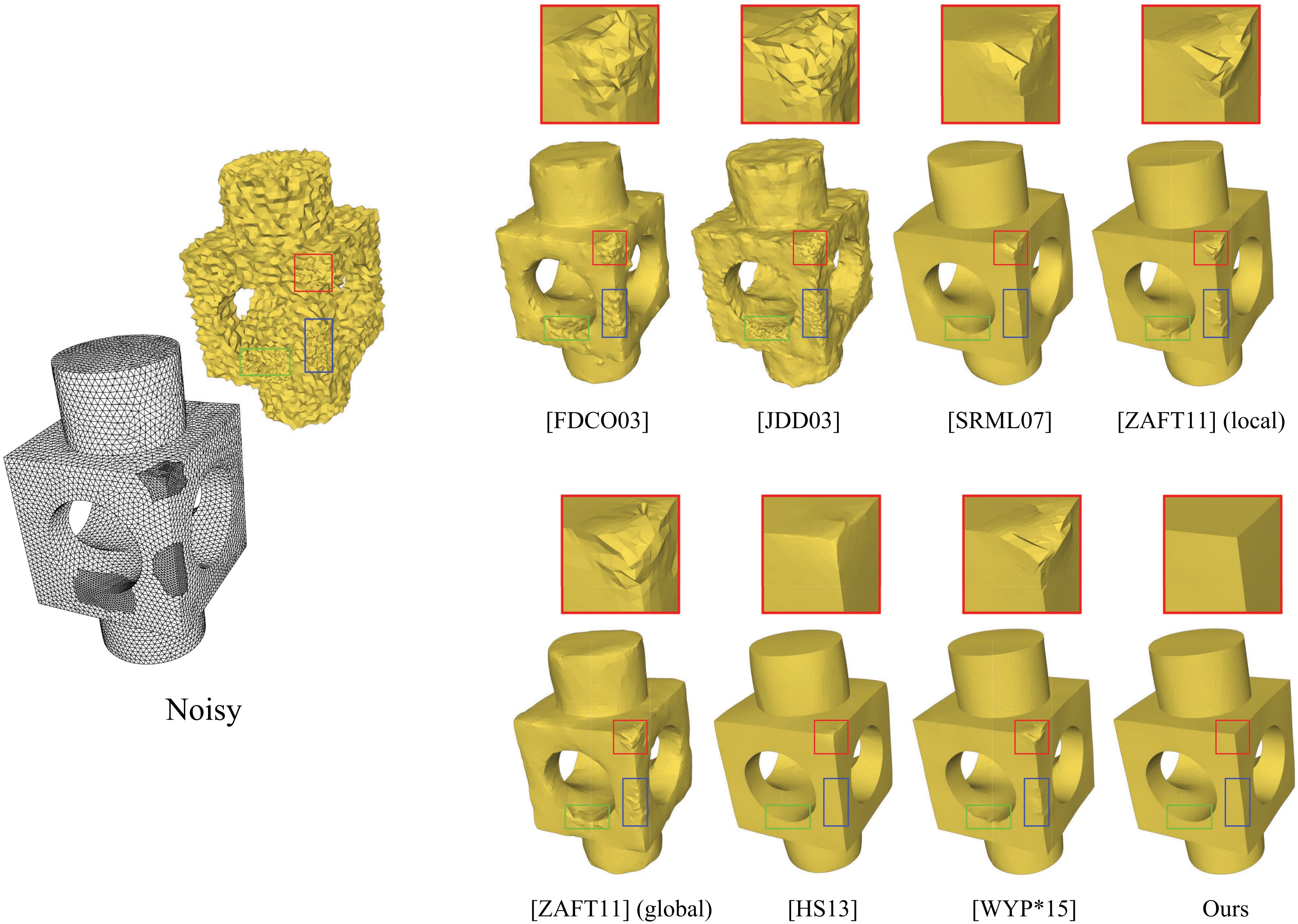




Results: vs bilateral filter



Results: comparisons



Time statistics

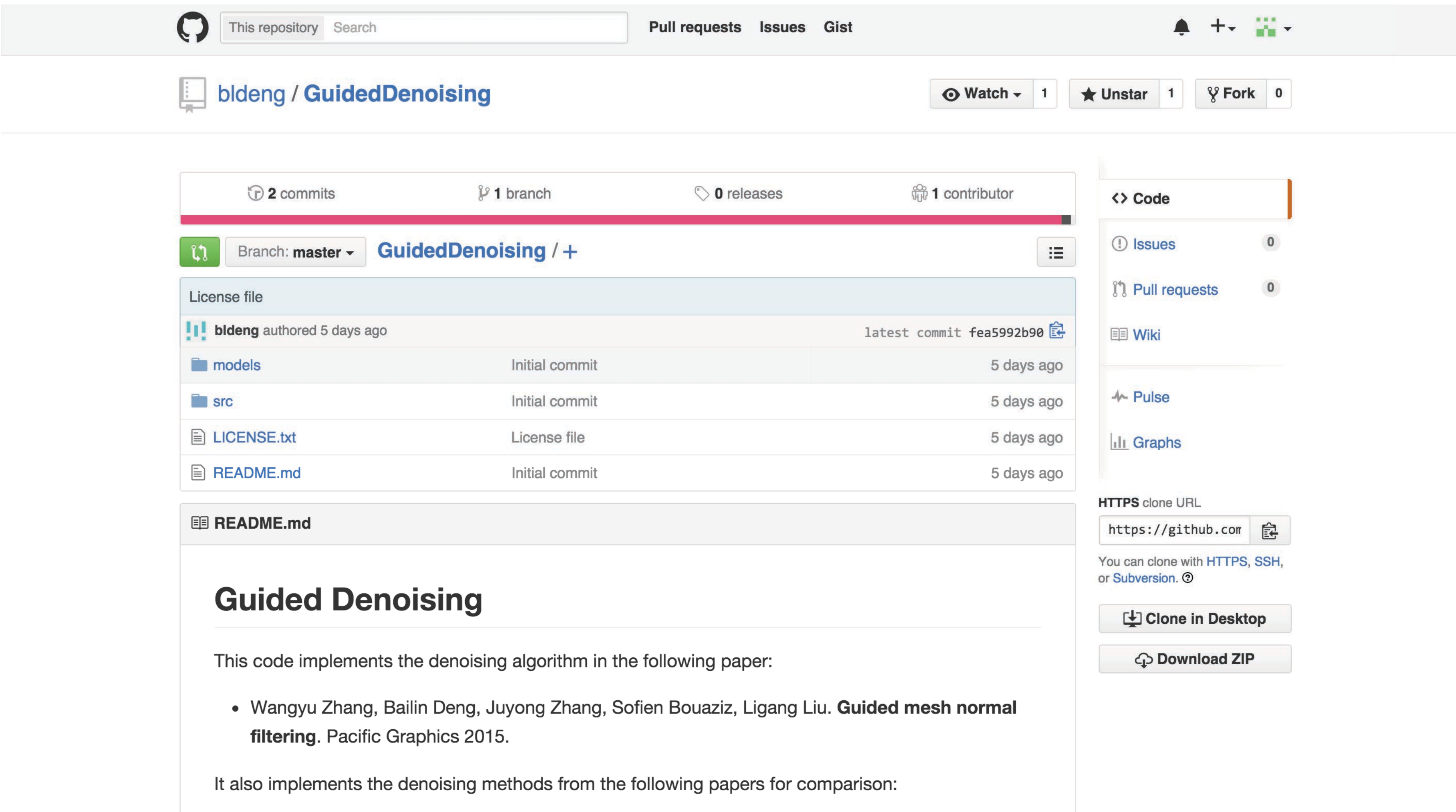
Model	*#Vertices*	*#Faces*	*Time(s)/Iter*
Fandisk	***6475***	***12946***	***0.076***
Block	***8771***	***17550***	***0.104***
Bunny	***34834***	***69451***	***0.698***
Iron	***85574***	***168285***	***1.571***

Conclusion

- A joint bilateral filter for mesh processing
    - ✦ A novel method to construct the guidance signal
    - ✦ Effective and efficient, simple to implement
    - ✦ Much better denoising results than state of the art

The source code is available:

https://github.com/bldeng/GuidedDenoising



Thank you!