Pacific Graphics 2018
H. Fu, A. Ghosh, and J. Kopf
(Guest Editors)

Volume 37 (2018), Number 7

Supplementary material:

Non-Local Low-Rank Normal Filtering for Mesh Denoising

Xianzhi Li, Lei Zhu, Chi-Wing Fu, and Pheng-Ann Heng

There are five parts in this supplementary material:

- The first part (Part 1) presents details about the optimization procedure described in Section 4.3 of the main paper.
- The second part (Part 2) presents additional comparison results against the state-of-the-art methods on synthetic models.
- The third part (Part 3) presents additional comparison results against the state-of-the-art methods on real scanned models.
- The fourth part (Part 4) presents our results on inputs with different noise levels to evaluate its robustness.
- The last part (Part 5) presents the parameter values for producing the results (shown in the main paper) of our method and the compared methods.

Note that both the main paper and this supplementary material have many equations. To make it clear, all the equations from the main paper are highlighted in bold.

Part 1. More Optimization Details

Part 1-A:

To derive **Eq. (8)** (main paper) from **Eq. (7)** (main paper), we first need to establish the following inequality, which generalizes Theorem 3.1 in [ZHY*12]:

To prove: For any given matrix $\mathbb{X} \in \mathcal{R}^{m\times n}$, any matrices $\mathbb{A} \in \mathcal{R}^{\lambda\times m}$ and $\mathbb{B} \in \mathcal{R}^{\lambda\times n}$, such that $\mathbb{A}\mathbb{A}^T{=}\mathcal{I}$ and $\mathbb{B}\mathbb{B}^T{=}\mathcal{I}$, for non-negative integer λ=rank($\mathbb{A}$)=rank($\mathbb{B}$), where $0 \le \lambda \le \min(m,n)$, we have

$$f(\mathrm{Tr}(\mathbb{A}\mathbb{X}\mathbb{B}^T)) \le \sum_{i=1}^{\lambda} f(\sigma_i(\mathbb{X})) \ , \tag{1}$$

where $f(x){=}\frac{(1+\gamma)x}{\gamma+x}$; $\sigma_i(\mathbb{X})$ is the i-th singular value of $\mathbb{X}$; γ is a positive scalar; and $\mathcal{I}$ is a $\lambda\times\lambda$ identity matrix.

Proof: We start from the left hand side of Eq. (1). According to Theorem 3.1 in [ZHY*12], we have the inequality below:

$$\mathrm{Tr}(\mathbb{A}\mathbb{X}\mathbb{B}^T) \le \sum_{i=1}^{\lambda} \sigma_i(\mathbb{X}) \ . \tag{2}$$

Since $f(x)$ is a monotonically increasing function, it has the following property: $f(x_1+x_2) \le f(x_1)+f(x_2)$ for any scalar $x_1 \ge 0$ and $x_2 \ge 0$. Hence, by Eq. (2), we can obtain

$$\begin{aligned} f(\mathrm{Tr}(\mathbb{A}\mathbb{X}\mathbb{B}^T)) &\le f(\sum_{i=1}^{\lambda} \sigma_i(\mathbb{X})) \ \ (\because f \text{ is monotonically increasing}) \\ &\le \sum_{i=1}^{\lambda} f(\sigma_i(\mathbb{X})) \ \ (\text{by } f(x_1{+}x_2) \le f(x_1){+}f(x_2)) \ . \ \blacksquare \end{aligned}$$

Part 1-B:

With the inequality we proved in Appendix Part 1-A, we can re-write the truncated γ norm $||\Omega_{\mathcal{D}}||_{tg}$ in **Eq. (6)** (main paper) as

$$\begin{aligned} ||\Omega_{\mathcal{D}}||_{tg} &= \sum_{i=\lambda+1}^{N} \frac{(1+\gamma)\sigma_i(\Omega_{\mathcal{D}})}{\gamma+\sigma_i(\Omega_{\mathcal{D}})} \quad (\text{from } \textbf{Eq. (6)} \text{ of the main paper}) \\ &= \sum_{i=1}^{N} \frac{(1+\gamma)\sigma_i(\Omega_{\mathcal{D}})}{\gamma+\sigma_i(\Omega_{\mathcal{D}})} - \sum_{i=1}^{\lambda} \frac{(1+\gamma)\sigma_i(\Omega_{\mathcal{D}})}{\gamma+\sigma_i(\Omega_{\mathcal{D}})} \\ &= ||\Omega_{\mathcal{D}}||_{\gamma} - \sum_{i=1}^{\lambda} f(\sigma_i(\Omega_{\mathcal{D}})) \ \ (\text{by definition of } f \text{ in Eq. (1)}) \\ &= ||\Omega_{\mathcal{D}}||_{\gamma} - \max_{\mathbb{A}\mathbb{A}^T=\mathcal{I},\mathbb{B}\mathbb{B}^T=\mathcal{I}} f(\mathrm{Tr}(\mathbb{A}\Omega_{\mathcal{D}}\mathbb{B}^T)), \ \ (\text{by Eq. (1)}) \end{aligned} \tag{3}$$

where $||\mathbb{N}_{\mathcal{D}}||_\gamma = \sum_{i=1}^{N} \frac{(1+\gamma)\sigma_i(\mathbb{N}_{\mathcal{D}})}{\gamma+\sigma_i(\mathbb{N}_{\mathcal{D}})}$ is the γ norm [KPC15]. Note that $\max_{\mathbb{A}\mathbb{A}^T=\mathcal{I},\mathbb{B}\mathbb{B}^T=\mathcal{I}} f(\mathrm{Tr}(\mathbb{A}\mathbb{N}_{\mathcal{D}}\mathbb{B}^T)) = \sum_{i=1}^{\lambda} f(\sigma_i(\mathbb{N}_{\mathcal{D}}))$, where $\mathbb{A}$ and $\mathbb{B}$ are defined as in **Eq. (9)** (main paper). By replacing the truncated γ norm in **Eq. (6)** (main paper) with the two terms in the last row of Eq. (3), the optimization model for computing $\mathbb{N}_{\mathcal{D}}$ in **Eq. (7)** (main paper) can be reformulated as

$$\min_{\mathbb{N}_{\mathcal{D}}} ||\mathbb{N}_{\mathcal{D}}||_\gamma - \max_{\mathbb{A}\mathbb{A}^T=\mathcal{I},\mathbb{B}\mathbb{B}^T=\mathcal{I}} f(\mathrm{Tr}(\mathbb{A}\mathbb{N}_{\mathcal{D}}\mathbb{B}^T)) + \frac{\mu}{2}||\mathbb{N}_{\mathcal{M}} - \mathbb{N}_{\mathcal{D}}||_F^2 ,$$

which is **Eq. (8)** in Section 4.3 of the main paper. ■

Part 1-C:

This part provides the details on how we derive a tractable optimization procedure to solve **Eq. (13)** (main paper) based on the iteratively re-weighted least squares (IRLS) technique. Specifically, we first decompose the non-linear term $f(\mathrm{Tr}(\mathbb{A}^t\mathbb{N}_{\mathcal{D}}(\mathbb{B}^t)^T))$ in **Eq. (13)** (main paper) as

$$\begin{aligned} f(\mathrm{Tr}(\mathbb{A}^t\mathbb{N}_{\mathcal{D}}(\mathbb{B}^t)^T)) &= \mathrm{Tr}(f(\mathbb{A}^t\mathbb{N}_{\mathcal{D}}(\mathbb{B}^t)^T)) \\ &= \mathrm{Tr}(\frac{(1+\gamma)\mathbb{A}^t\mathbb{N}_{\mathcal{D}}(\mathbb{B}^t)^T}{\gamma+\mathbb{A}^t\mathbb{N}_{\mathcal{D}}(\mathbb{B}^t)^T}) \\ &= \mathrm{Tr}(\frac{1+\gamma}{\gamma+\mathbb{A}^t\mathbb{N}_{\mathcal{D}}(\mathbb{B}^t)^T}\cdot(\mathbb{A}^t\mathbb{N}_{\mathcal{D}}(\mathbb{B}^t)^T)) \\ &= \mathrm{Tr}(\Phi\cdot(\mathbb{A}^t\mathbb{N}_{\mathcal{D}}(\mathbb{B}^t)^T)) , \end{aligned} \tag{4}$$

where matrix Φ is $(1+\gamma)/(\gamma+\mathbb{A}^t\mathbb{N}_{\mathcal{D}}(\mathbb{B}^t)^T)$. Thanks to the decomposition, we can solve the optimization in **Eq. (13)** (main paper) using a two-phase scheme:

Phase 1: we take the current estimate of $\mathbb{N}_{\mathcal{D}}$ (together with $\mathbb{A}^t$, $\mathbb{B}^t$, and γ) to compute matrix Φ.

Phase 2: we compute $\mathbb{N}_{\mathcal{D}}$ in the next iteration by

$$\min_{\mathbb{N}_{\mathcal{D}}} -\mathrm{Tr}(\Phi\cdot(\mathbb{A}^t\mathbb{N}_{\mathcal{D}}(\mathbb{B}^t)^T)) + \frac{\mu}{2}||\mathbb{N}_{\mathcal{M}} - \mathbb{N}_{\mathcal{D}}||_F^2 + \frac{\beta}{2}||\mathbb{N}_{\mathcal{D}} - \mathbb{W}^k + \frac{1}{\beta}\mathbb{Z}^k)||_F^2 , \tag{5}$$

which admits a closed-form solution:

$$\mathbb{N}_{\mathcal{D}} = \frac{\mu\mathbb{N}_{\mathcal{M}} + (\mathbb{A}^t)^T\Phi^T\mathbb{B}^t + \beta\mathbb{W}^k - \mathbb{Z}^k}{\mu+\beta} . \tag{6}$$

The advantage of the above two-phase scheme is its fast computation. We empirically set the number of iterations in the scheme as two in all the experiments to tradeoff between computation time and mesh denoising quality.

Part 1-D:

This part provides the details on how we minimize the objective function in **Eq. (14)** (main paper) using difference of convex (DC) programming [TA97] [KPC15].

Specifically, we let $\mathbb{L} = \mathbb{N}_{\mathcal{D}}^{k+1} + \frac{1}{\beta}\mathbb{Z}^k$, and $\widehat{\mathbb{U}}\mathrm{diag}\{\sigma_{\mathbb{L}}\}\widehat{\mathbb{V}}^T$ to be the SVD of matrix $\mathbb{L}$. Then, $\mathbb{W}^{k+1}$, the solution of the optimization in **Eq. (14)** (main paper), is given by $\mathbb{W}^{k+1}=\widehat{\mathbb{U}}\mathrm{diag}\{\sigma^*\}\widehat{\mathbb{V}}^T$, where σ^* is the converged local optimal of variable σ after H iterations (H=2 in all the experiments). At the $(h+1)$-th

($h \in [1,H]$) inner iteration, σ^{h+1} is computed by the minimization below:

$$\min_{\sigma \geq 0} < y_h, \sigma > + \frac{\beta}{2} ||\sigma - \sigma_{\mathbb{L}}||^2 , \tag{7}$$

whose closed-form solution is given by

$$\sigma^{h+1} = \max(\sigma_{\mathbb{L}} - \frac{y_h}{\beta}, 0) , \tag{8}$$

where $y_h = \partial f(\sigma^h)$ is the gradient of $f()$ with respect to σ^h.

Part 2. Additional Denoising Results on Synthetic Models

In this part, we show additional denoising results (Figures A-D) on synthetic models (prepared with various noise) not presented in the main paper.

Figures A and B show the denoising results for input models corrupted with Gaussian noise; while in Figures C and D, the input models are corrupted by impulsive noise. These comparisons further demonstrate that our approach can better preserve fine details than the other methods regardless of the noise distributions and noise levels. Even for input meshes with low triangle counts (e.g., PIERROT), our method can still generate promising results.

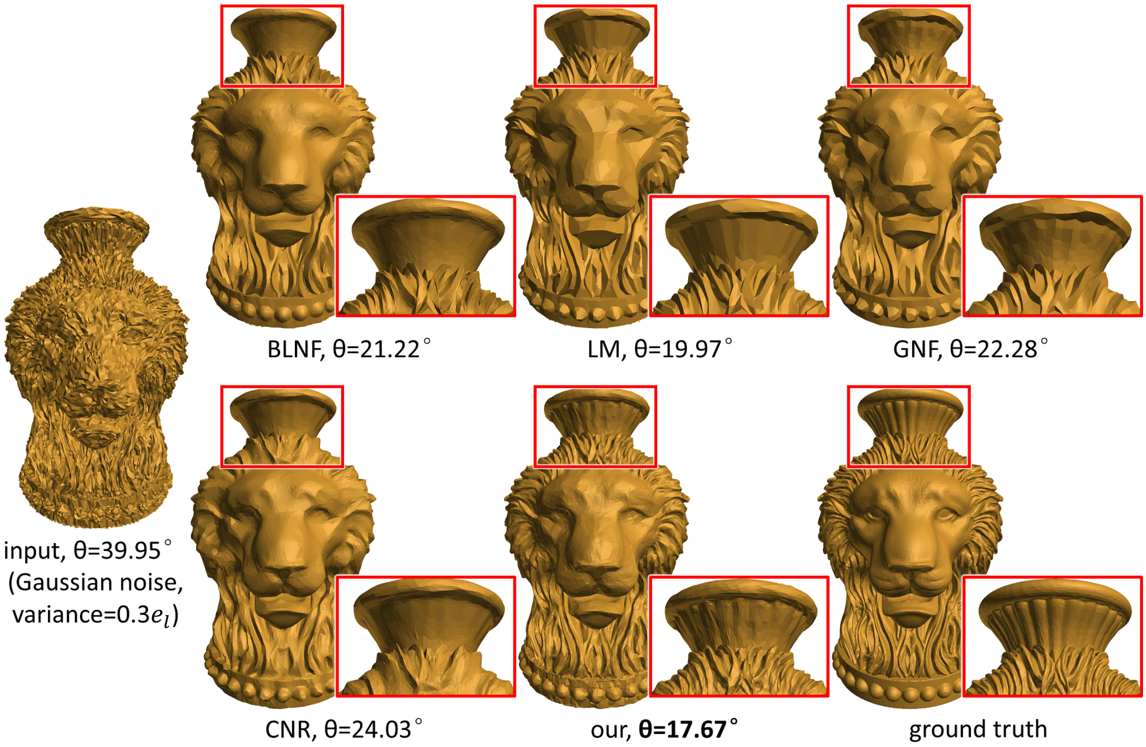

Figure A: Additional comparison results on VASELION prepared with Gaussian noise level of $0.3e_l$.

Table 1: Method parameters for generating the results shown in Figure A.

Model	Methods									
	BLNF			LM	GNF			Our method		
	σ_s	k_{iter}	v_{iter}	λ	σ_r	k_{iter}	v_{iter}	$\sigma_{\mathcal{M}}$	N_k	v_{iter}
VASELION	0.40	20	15	0.00001	0.20	20	10	0.40	12	5

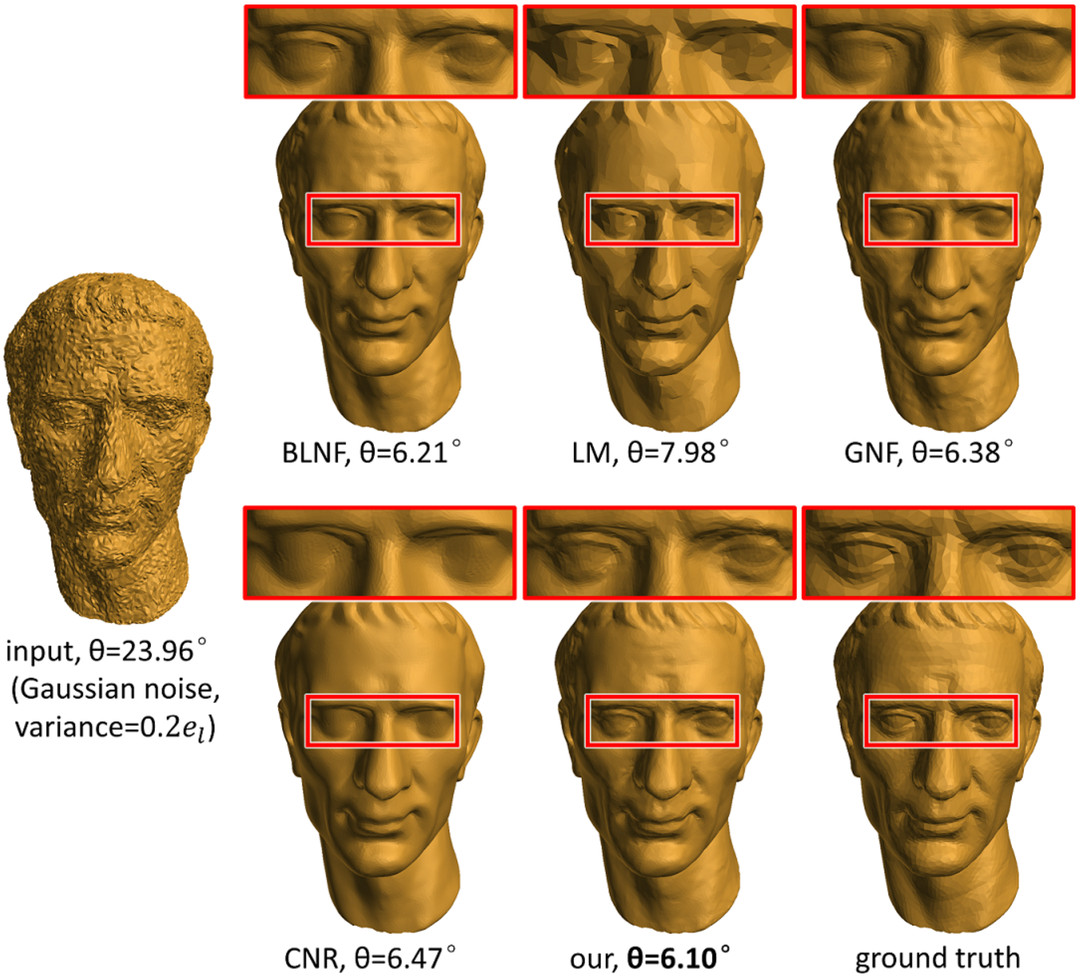

Figure B: Additional comparison results on JULIUS prepared with Gaussian noise level of $0.2e_l$.

Table 2: Method parameters for generating the results shown in Figure B.

Model	Methods									
	BLNF			LM	GNF			Our method		
	σ_s	k_{iter}	v_{iter}	λ	σ_r	k_{iter}	v_{iter}	$\sigma_{\mathcal{M}}$	N_k	v_{iter}
JULIUS	0.45	6	4	0.000003	0.45	5	4	0.33	8	8

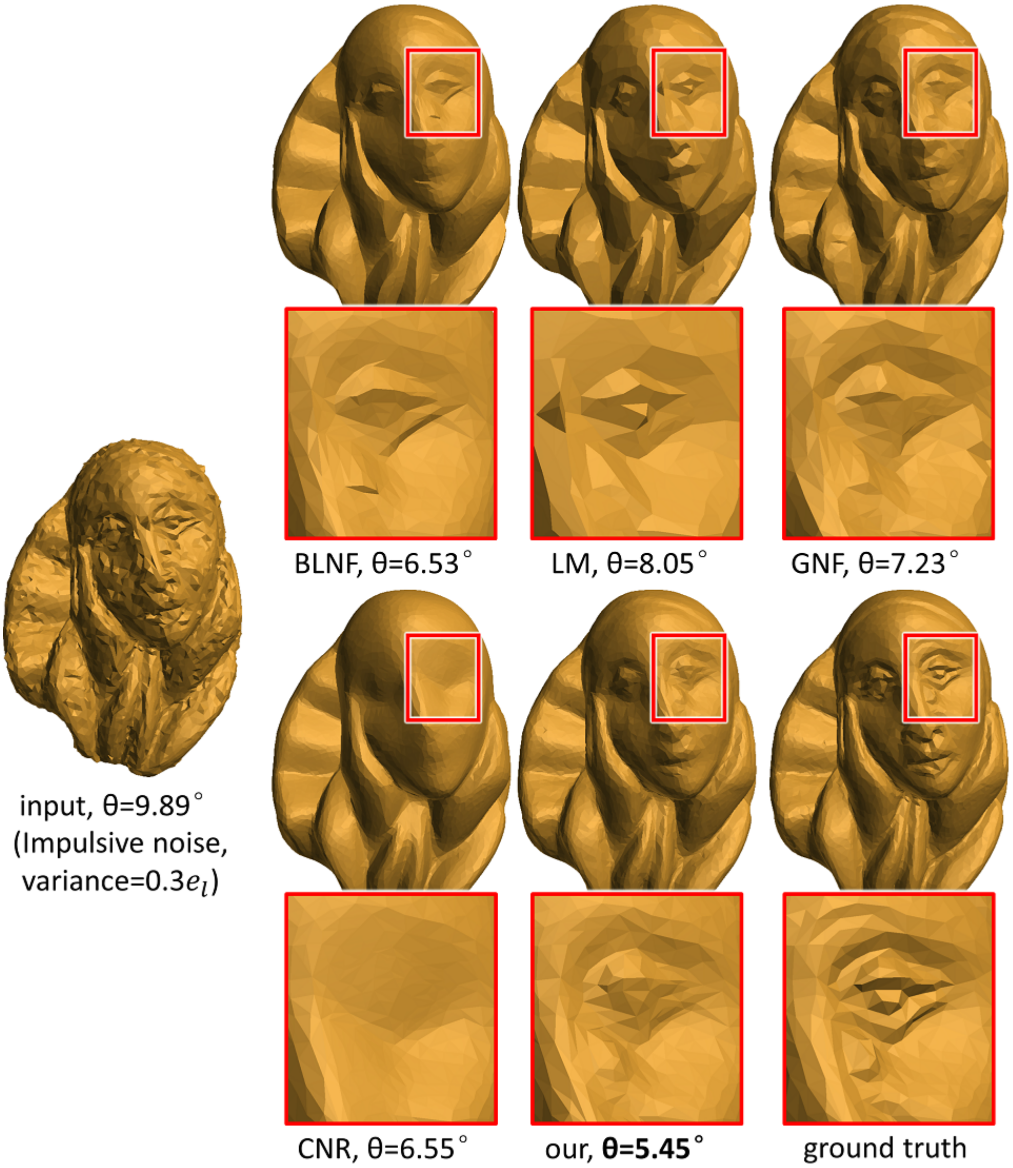

Figure C: Additional comparison results on PIERROT prepared with impulsive noise level of $0.3e_l$.

Table 3: Method parameters for generating the results shown in Figure C.

Model	Methods									
	BLNF			LM	GNF			Our method		
	σ_s	k_{iter}	v_{iter}	λ	σ_r	k_{iter}	v_{iter}	$\sigma_{\mathcal{M}}$	N_k	v_{iter}
PIERROT	0.20	10	5	0.05	0.05	5	5	0.22	6	4

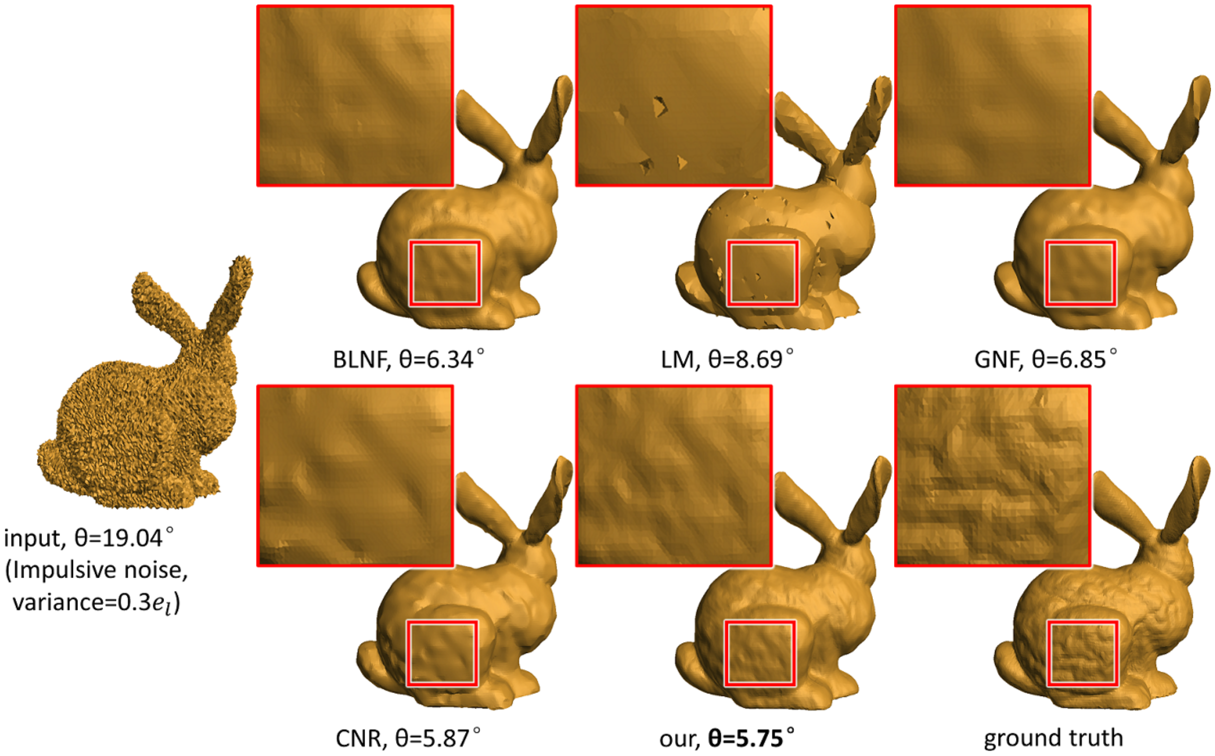

Figure D: Additional comparison results on BUNNY prepared with impulsive noise level of $0.3e_l$.

Table 4: Method parameters for generating the results shown in Figure D.

Model	Methods									
	BLNF			LM	GNF			Our method		
	σ_s	k_{iter}	v_{iter}	λ	σ_r	k_{iter}	v_{iter}	$\sigma_{\mathcal{M}}$	N_k	v_{iter}
BUNNY	0.35	18	5	0.000016	0.15	10	10	0.30	8	10

Part 3. Additional Denoising Results on Real Scanned Models

In this section, we present and compare denoising results generated by our method and other methods on more real scanned models employed by CNR.

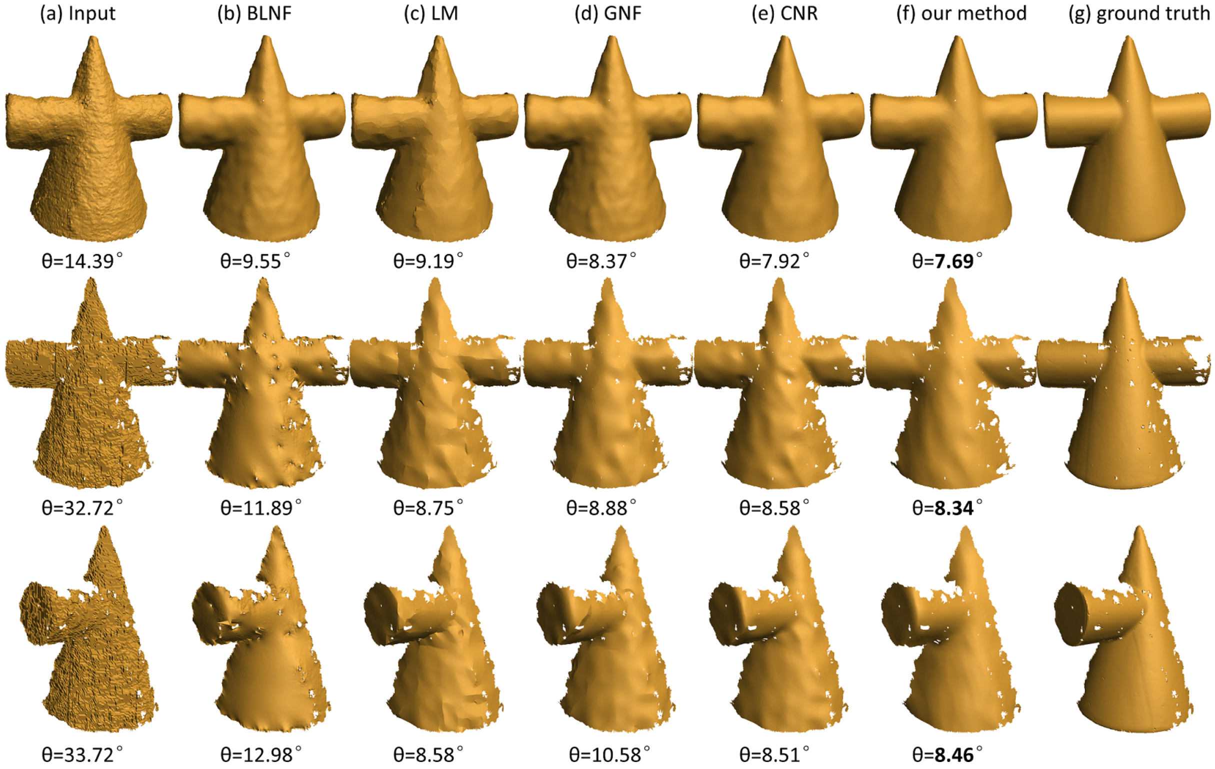

Figure E: Additional comparison results on real scanned models.

Table 5: Method parameters for generating the results shown in Figure E.

Models	Methods									
	BLNF			LM	GNF			Our method		
	σ_s	k_{iter}	v_{iter}	λ	σ_r	k_{iter}	v_{iter}	$\sigma_{\mathcal{M}}$	N_k	v_{iter}
CONE (1st row)	0.60	20	20	0.10	0.60	20	20	0.60	50	300
CONE (2nd row)	3.00	150	120	1.50	3.00	60	50	3.50	30	80
CONE (3rd row)	3.00	150	200	1.30	3.00	50	60	4.00	40	60

Part 4. Robustness to noise levels

In this section, we investigate the robustness of our method by applying it to input models with varying noise levels. Figures F and G plot the filtering results for input models corrupted with Gaussian noise and impulsive noise, respectively. From the results, we can see that our method can effectively and consistently recover the underlying fine details in the presence of different amount of Gaussian noise and impulsive noise, even for large noise levels.

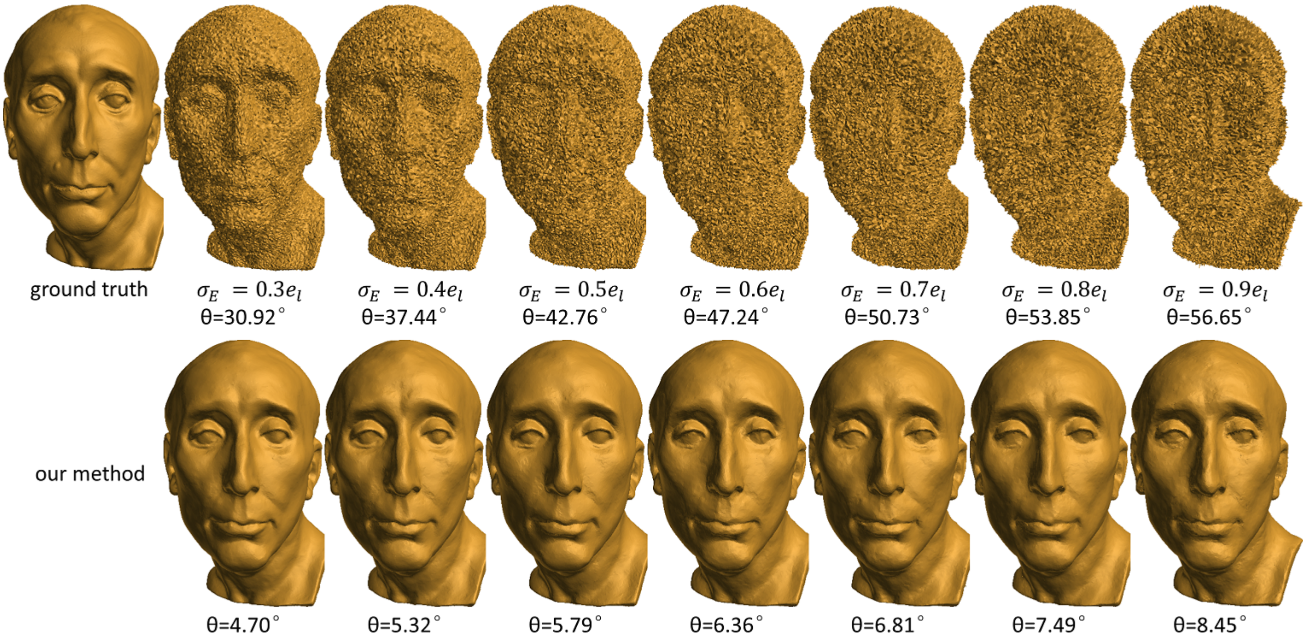

Figure F: Denoising results of our method on the synthetic noisy NICOLO models with varying amount of Gaussian noise: $0.3e_l$ to $0.9e_l$.

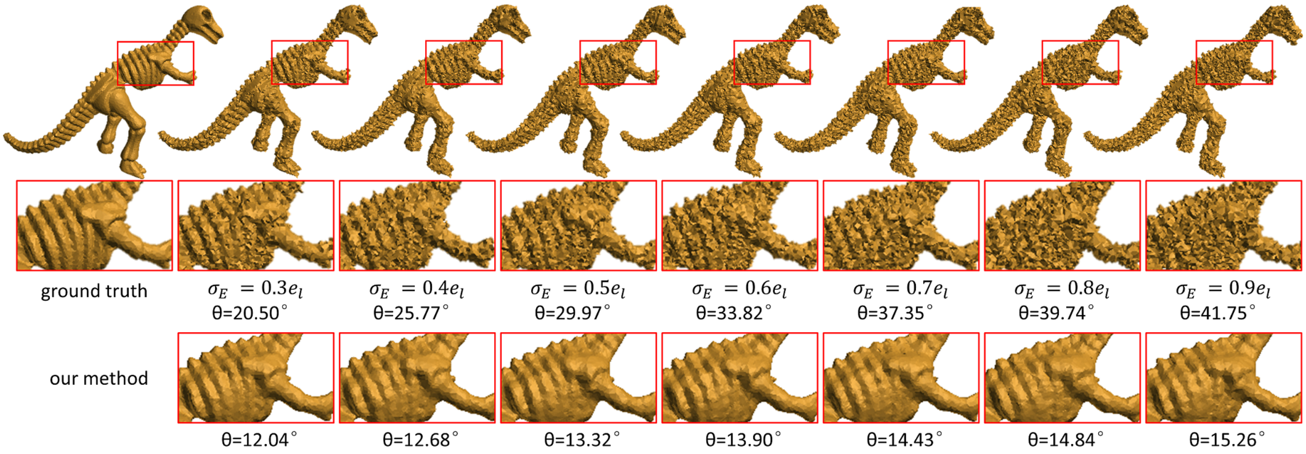

Figure G: Denoising results of our method on the synthetic noisy DINO models with varying amount of impulsive noise: $0.3e_l$ to $0.9e_l$.

Part 5. Parameters used for various results shown in Main Paper

This part presents the parameter values for the denoising results shown in our main paper. To begin with, we first explain the tuned parameters in the compared methods (including BLNF, LM, and GNF) and also in our method.

Explanation of parameters in various methods:

- BLNF
  - σ_s: the variance parameter for the spatial kernel.
  - k_{iter}: the number of iterations for normal update.
  - v_{iter}: the number of iterations for vertex update.
- LM
  - λ: weight for the L_0 term in the target function.
- GNF
  - σ_r: the variance parameter for the range kernel.
  - k_{iter}: the number of iterations for normal update.
  - v_{iter}: the number of iterations for vertex update.
- Our method
  - $\sigma_{\mathcal{M}}$: the variance parameter for the input noise.
  - N_k: the number of the most similar vertices (patches).
  - v_{iter}: the number of iterations for vertex update.

Then, Tables 6 to 9 below present specific values of the main parameters mentioned above for the denoised results (Figures 7, 8, 9, and 12 of the main paper) of different methods.

Table 6: Parameters for Figure 7 in the main paper.

Models	Methods									
	BLNF			LM	GNF			Our method		
	σ_s	k_{iter}	v_{iter}	λ	σ_r	k_{iter}	v_{iter}	$\sigma_{\mathcal{M}}$	N_k	v_{iter}
CHINESE LION	0.20	15	10	0.01	0.04	6	5	0.12	6	4
ARMADILLO	0.20	20	10	0.01	0.15	8	6	0.28	8	6
IGEA	0.35	10	5	0.00003	0.20	10	5	0.39	14	7

Table 7: Parameters for Figure 8 in the main paper.

Models	Methods									
	BLNF			LM	GNF			Our method		
	σ_s	k_{iter}	v_{iter}	λ	σ_r	k_{iter}	v_{iter}	$\sigma_{\mathcal{M}}$	N_k	v_{iter}
BUSTE	0.20	30	20	0.14	0.10	8	6	0.20	6	7
NICOLE	0.50	10	10	1.50	0.18	10	5	0.24	6	10
GARGOYLE	0.35	20	10	0.03	0.30	10	5	0.32	8	10

Table 8: Parameters for Figure 11 in the main paper.

Models	Methods									
	BLNF			LM	GNF			Our method		
	σ_s	k_{iter}	v_{iter}	λ	σ_r	k_{iter}	v_{iter}	$\sigma_{\mathcal{M}}$	N_k	v_{iter}
HAND	0.10	20	10	0.01	0.10	20	10	0.05	6	8
FACE	0.15	20	10	0.05	0.10	10	20	0.08	6	10
KITTEN	0.28	10	10	0.01	0.10	20	10	0.12	8	10

Table 9: Parameters for Figure 13 in the main paper.

Models	Methods									
	BLNF			LM	GNF			Our method		
	σ_s	k_{iter}	v_{iter}	λ	σ_r	k_{iter}	v_{iter}	$\sigma_{\mathcal{M}}$	N_k	v_{iter}
BOY	0.50	20	20	0.20	0.60	20	20	0.75	100	210
CONE	0.35	20	30	20	0.35	20	20	0.70	120	250

References

[KPC15] KANG Z., PENG C., CHENG Q.: Robust PCA via nonconvex rank approximation. *IEEE ICDM* (2015), 211–220. 4

[TA97] TAO P. D., AN L. T. H.: Convex analysis approach to dc programming: Theory, algorithms and applications. *Acta Mathematica Vietnamica 22*, 1 (1997), 289–355. 4

[ZHY*12] ZHANG D., HU Y., YE J., LI X., HE X.: Matrix completion by truncated nuclear norm regularization. *IEEE CVPR* (2012), 2192–2199. 3