Supplementary Material: Structured Low-Rank Matrix Factorization for Point-Cloud Denoising

Kripasindhu Sarkar[1,2], Florian Bernard[3,4], Kiran Varanasi[1], Christian Theobalt[3,4], and Didier Stricker[1,2]

[1]DFKI Kaiserslautern, [2]TU Kaiserslautern, [3]MPI Informatics, [4]Saarland Informatics Campus

1. Introduction

In this document we provide supplementary information for our paper 'Structured Low-Rank Matrix Factorization for Point-Cloud Denoising', referred to as *Main Paper*.

2. Theoretical Motivation of Dictionary Learning Framework

Please see *Section 3.3* in the *Main paper* for the meaning of the variables and the context of the following equations.
Matrix factorization:

$$\min_{\Phi, A} \; \ell(\mathbf{Y}, \Phi A) + \Omega(\Phi, A)\,, \tag{1}$$

Loss function with unobserved data:

$$\ell(\mathbf{Y}, \Phi A) := \|M \odot (\mathbf{Y} - \Phi A)\|_F^2\,, \tag{2}$$

Regularizer:

$$\Omega(\Phi, A) = \lambda \sum_{i=1}^{r} \|\Phi_i\|_\phi \|A_{i,:}^T\|_a\,, \tag{3}$$

where

$$\|y\|_\phi = \lambda_2 \|y\|_2 + \lambda_E \|Ey\|_2, \text{ and} \tag{4}$$

$$\|z\|_a = \lambda_1 \|z\|_1\,. \tag{5}$$

Motivation for the regularizer (Eq. 4, Main paper): The motivation for using this particular form of the regularizer in (3) (*Eq. 4, Main paper*), i.e. the sum of the product of two vector norms, lies in its relation to the *projective tensor norm* [2, 7], which plays an important role for the theoretical motivation of the employed framework [2, 7, 3]. Roughly speaking, one can introduce a lifted optimization problem that is closely related to Problem (1), where one combines the loss function (2) (defined for the lifted variable) with the projective tensor norm as regularizer for the lifted variable. In this case, one directly optimizes over the product $\Phi A := \mathbf{X}$, rather than over Φ and A individually. Since the loss function (2) is convex in $\mathbf{X}$ $(= \Phi A)$, and the projective tensor norm is also convex in $\mathbf{X}$ (since it is a *norm*), the so-introduced lifted problem is convex. While one can now, in principal, directly find the global minimizer $\mathbf{X}^*$ of the lifted convex problem, this does not yield the desired factorization (Φ, A). Moreover, depending on the choice of norms, cf. (3), computing the projective tensor norm may be very difficult. Instead, based on the much-celebrated Burer-Monteiro approach [5], one can consider a (non-convex) factorized version of the lifted problem, where, under certain conditions, a *rank-deficient* local optimum is also a global optimum. For further details and background we refer the interested reader to [5, 2, 7, 3].

3. Results

Table 1 shows the full quantitative results on 9 meshes, whose summary is provided in *Table 1, Main paper*.

In addition we provide zoomed version of some of the figures from the *Main paper* in Fig.s 1–4.

References

[1] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silva. Computing and rendering point set surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 9(1):3–15, Jan 2003. 2, 6

[2] F. Bach, J. Mairal, and J. Ponce. Convex sparse matrix factorizations. *arXiv.org*, 2008. 1

[3] F. Bernard, P. Gemmar, F. Hertel, J. Goncalves, and J. Thunberg. Linear shape deformation models with local support using graph-based structured matrix factorisation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5629–5638, 2016. 1

[4] J. Bohg, J. Romero, A. Herzog, and S. Schaal. Robot arm pose estimation through pixel-wise part classification. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3143–3150, May 2014. 5

[5] S. Burer and R. D. Monteiro. Local minima and convergence in low-rank semidefinite programming. *Mathematical Programming*, 103(3):427–444, 2005. 1

	σ	noisy	plane	[6]	[1]	ours
Bunny	0.0025	2.49e-03	2.91e-03	1.57e-03	1.93e-03	**7.17e-04**
	0.0050	4.98e-03	3.35e-03	4.25e-03	2.58e-03	**1.15e-03**
	0.0075	7.44e-03	3.24e-03	7.02e-03	3.23e-03	**1.62e-03**
MilkBottle	0.0025	2.50e-03	2.66e-03	2.51e-02	1.66e-03	**6.46e-04**
	0.0050	4.99e-03	3.41e-03	2.44e-02	2.24e-03	**1.08e-03**
	0.0075	7.45e-03	3.34e-03	2.24e-02	2.96e-03	**1.63e-03**
Baseball	0.0025	2.72e-03	1.98e-03	1.64e-03	2.17e-03	**1.12e-03**
	0.0050	5.11e-03	2.43e-03	3.92e-03	2.70e-03	**1.16e-03**
	0.0075	7.58e-03	2.43e-03	7.21e-03	3.24e-03	**1.28e-03**
Fandisk	0.0025	2.49e-03	2.27e-03	1.44e-03	1.89e-03	**8.12e-04**
	0.0050	4.96e-03	2.56e-03	3.59e-03	2.52e-03	**1.46e-03**
	0.0075	7.39e-03	2.92e-03	6.03e-03	3.17e-03	**2.05e-03**
Supernova	0.0025	2.48e-03	3.71e-03	2.26e-03	1.41e-03	**1.09e-03**
	0.0050	4.88e-03	3.79e-03	5.02e-03	2.17e-03	**1.70e-03**
	0.0075	7.22e-03	3.82e-03	7.05e-03	3.05e-03	**2.25e-03**
Terrex	0.0025	2.48e-03	3.01e-03	2.08e-03	1.50e-03	**1.01e-03**
	0.0050	4.90e-03	3.32e-03	4.49e-03	2.21e-03	**1.63e-03**
	0.0075	7.27e-03	3.53e-03	6.30e-03	3.08e-03	**2.44e-03**
Wander	0.0025	2.46e-03	3.27e-03	2.11e-03	1.55e-03	**1.54e-03**
	0.0050	4.81e-03	3.81e-03	4.06e-03	**2.42e-03**	2.52e-03
	0.0075	6.99e-03	4.45e-03	5.78e-03	**3.48e-03**	3.61e-03
Leather-Shoe	0.0025	2.48e-03	3.05e-03	2.25e-03	1.49e-03	**1.02e-03**
	0.0050	4.93e-03	3.57e-03	4.97e-03	2.21e-03	**1.74e-03**
	0.0075	7.31e-03	4.00e-03	6.92e-03	3.13e-03	**2.79e-03**
Brain	0.0025	2.50e-03	2.89e-03	1.98e-03	1.57e-03	**8.83e-04**
	0.0050	4.98e-03	3.14e-03	4.26e-03	2.23e-03	**1.57e-03**
	0.0075	7.42e-03	3.74e-03	6.37e-03	3.11e-03	**2.56e-03**

Table 1. Comparison of RMS point-to-mesh error (relative to bounding box) of different algorithms for various Gaussian noise levels with standard deviation σ.

[6] S. Cheng and M. Lau. Denoising a point cloud for surface reconstruction. *CoRR*, abs/1704.04038, 2017. 2, 6

[7] B. Haeffele, E. Young, and R. Vidal. Structured low-rank matrix factorization: Optimality, algorithm, and applications to image processing. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 2007–2015, 2014. 1

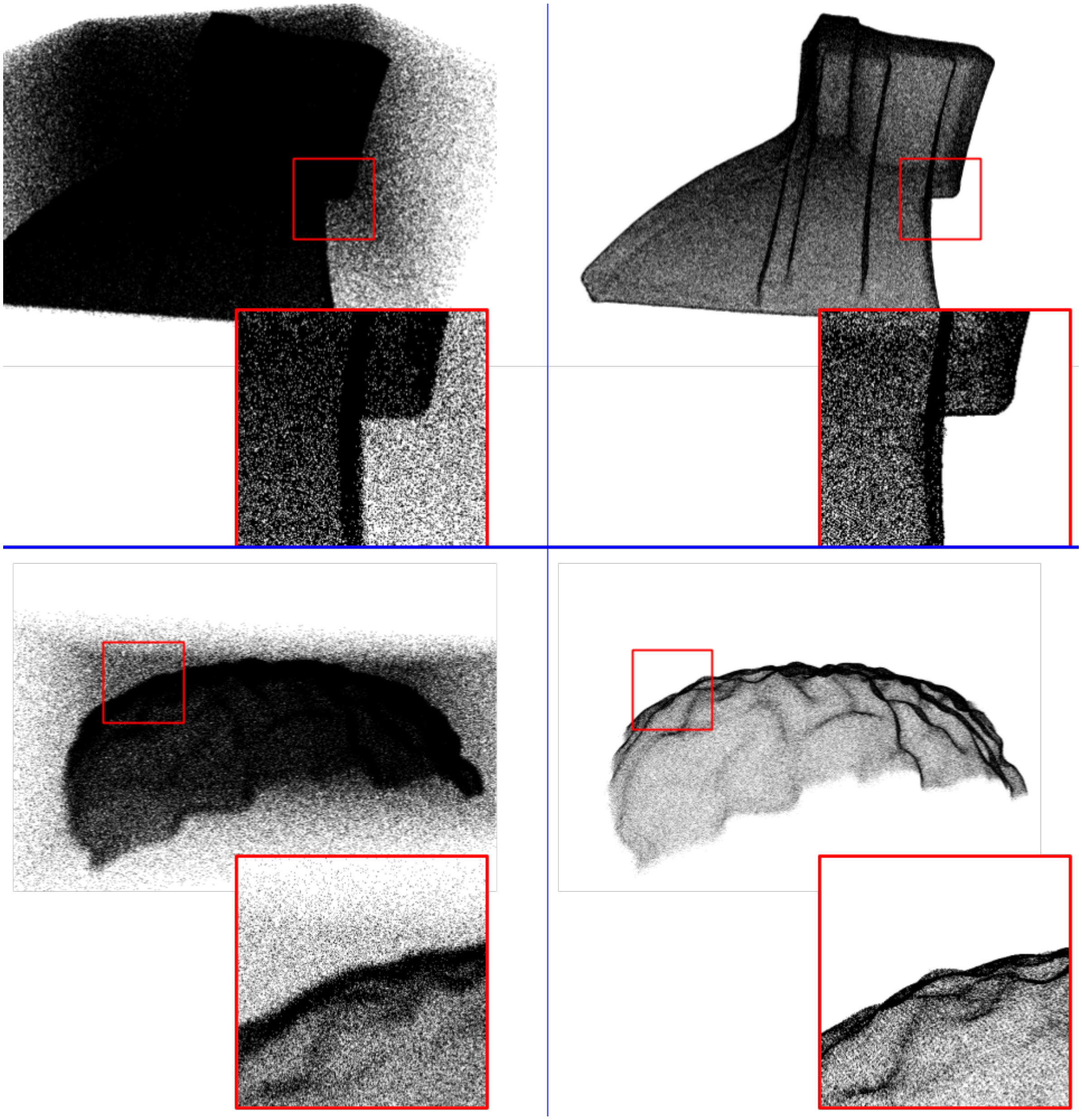

Figure 1. Result of our denoising algorithm with uniform noise (50% of input points) in addition to Gaussian noise ($\sigma = 0.0025$) for the model *Fandisk* (Top) and *Brain* (Bottom). The RMS error after denoising is 8.05e-04 and 8.70e-04 for Fandisk and Brain respectively (in comparison to 8.12e-04 and 8.83e-04 in the experiment without white noise). This shows that our denoising algorighm is highly resistant to outliers. (Zoomed version of *Figure 6, Main Paper*)

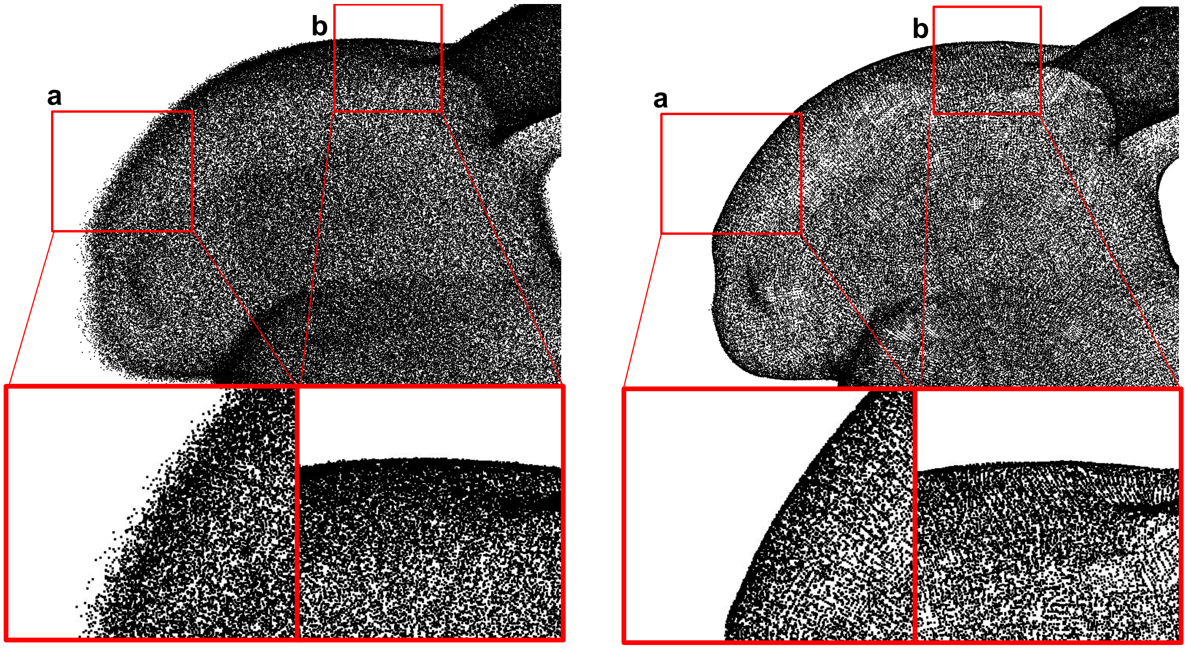

Figure 2. Result of our denoising algorithm when the noise is added in the direction of camera. The left part shows the noisy data and the right part shows the denoised output. It can be seen that our method consistently removes the noise from the surface perpendicular to the camera direction (where the maximum amount of noise occurs, see **region a**) and that it does not alter much when the surface is parallel to the camera direction (places where the surface is mostly unaffected by the added noise, see **region b**) (Zoomed version of *Figure 7, Main Paper*).

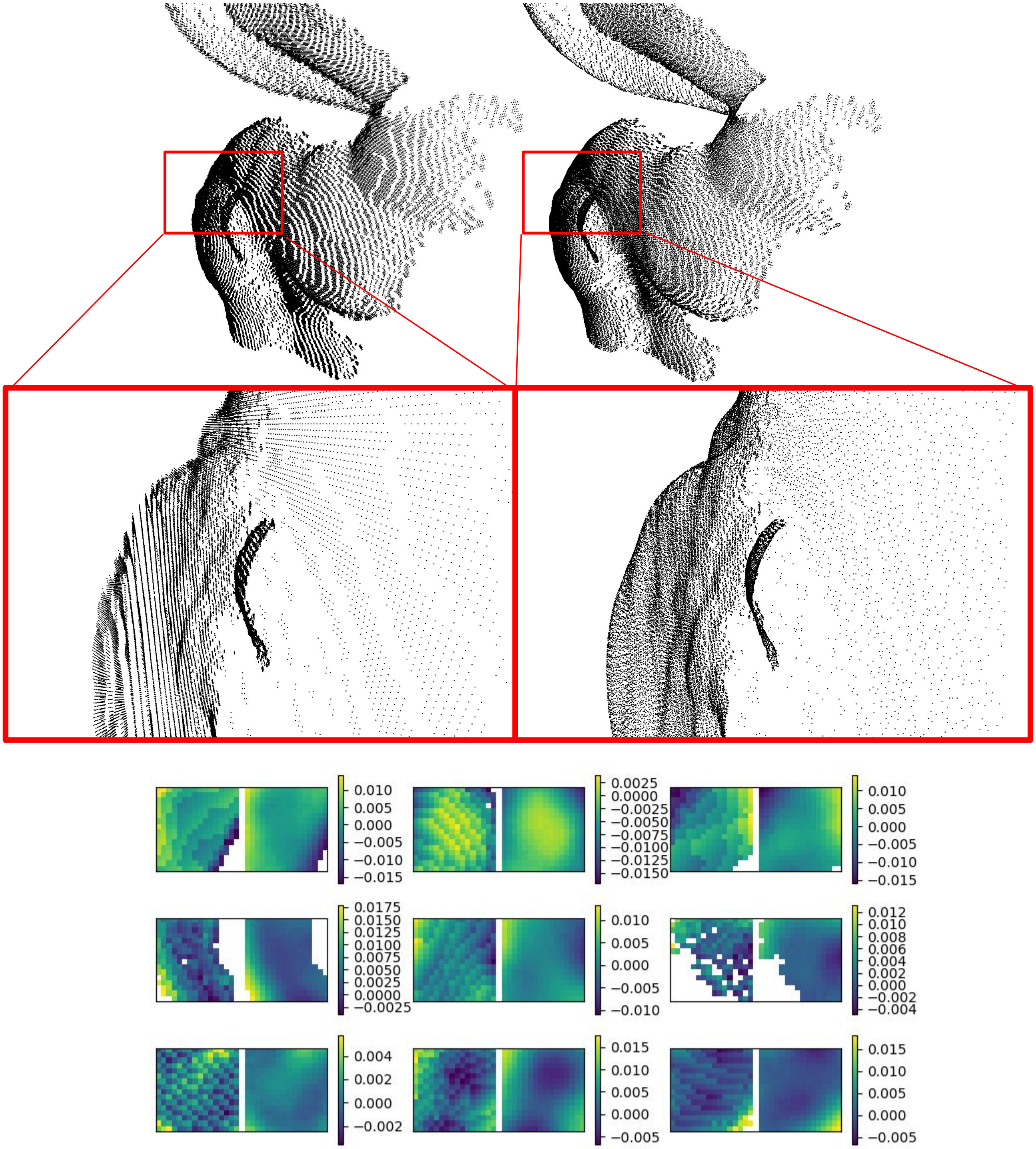

Figure 3. Top: Denoising of a point-cloud with noise obtained by a Kinect simulator [4] (parameters: $r = 0.05$, m = 16). Bottom: Visualization of the noisy and the corresponding denoised displacement maps. Note the removal of typical fringe patterns in both the raw depth scan (zoomed version) and the displacement maps (detailed version of *Figure 8, Main Paper*).

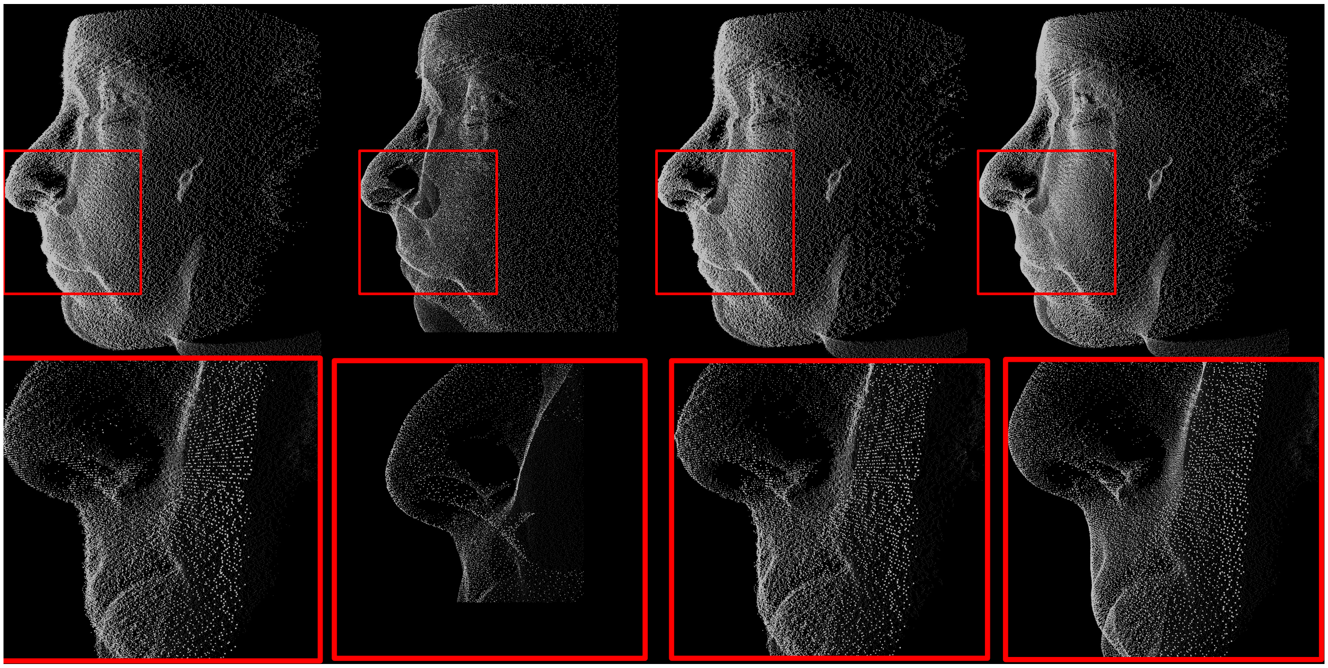

Figure 4. Denoising result from a range scanner. From left to right we show raw scanned data, denoising results from Cheng Et. al [6], moving least-squares [1] and ours. Parameters: $r = 0.25, m = 16$. (Zoomed version of *Figure 9, Main Paper*).