

Feature-Preserving Mesh Decimation for Normal Integration

– Supplementary Material –

Moritz Heep
PhenoRob
University of Bonn
mheep@uni-bonn.de

Sven Behnke
Autonomous Intelligent Systems
University of Bonn
behnke@ais.uni-bonn.de

Eduard Zell
Independent Researcher
ezell@hotmail.de

Overview

This supplementary material, provides additional technical details on the normal integration, the final algorithm and additional evaluation results that have been removed from the main document due to page constraints. For easier navigation and cross-referencing, we follow the section titles of the submitted paper. For a quick overview, we briefly list the content of the Appendix:

- Appendix A provides additional details on the integration, improving further the surface reconstruction accuracy compared to previous work[5].
- Appendix B describes additional details on the computation of quadrics and the linear system to solve.
- Appendix C provides additional comparisons to previous work, quantitative analysis on the reconstruction accuracy and qualitative examples.
- Appendix D is a list of all the datasets we used together with the urls to find them.

Finally, we included samples of the reconstructed surfaces using our method (`ours.obj`) and using [5] (`adaptive.obj`) for

- Michelangelo’s David (Fig. 1 of the manuscript) and
- all five objects in the DiLiGenT-MV [6] dataset (reconstructed from the first view each).

These samples can be found in the respective subdirectories of `meshes/` of this supplementary material.

A reference implementation of our method is available under <https://moritzheep.github.io/anisotropic-screen-meshing>.

A. Screen Space Mesh Decimation

In the following, we will provide more detail on our mesh-based integration and the underlying sparse linear system, that slightly differs from [5] but improves the reconstruction accuracy even further.

A.1. Details on Mesh-Based Normal Integration

The unified functional for normal integration is

$$E_{\text{Int}} = \int_{\Omega} \left\| \langle \vec{n}, \vec{r} \rangle \cdot \begin{pmatrix} \partial_u z \\ \partial_v z \end{pmatrix} + D \cdot \begin{pmatrix} n_x \\ n_y \end{pmatrix} \right\|^2 d^2 u. \quad (1)$$

with the respective choices for \vec{r} and D depending on the projection type – orthographic or perspective. In the case of pixel-based integration, the partial derivatives are discretised by using each pixel and its immediate neighbourhood. In the case of mesh-based integration, the pixel neighbourhood is replaced by adjacent vertices.

Previous work [5] showed that the minimiser of E_{Int} is found by solving the sparse linear system

$$\sum_{w \in \mathcal{V}_v} \sum_{f \in \mathcal{F}(v,w)} \omega_{f,vw} \cdot (m_f \cdot \delta z_{vw} + \langle \vec{b}_f, \delta \vec{u}_{vw} \rangle) = 0 \quad (2)$$

for each vertex $v \in \mathcal{V}$. All vertices w that are connected by an edge with vertex v have an impact on depth z_v . We denote these vertices by \mathcal{V}_v . The faces adjacent to the edge (v, w) , denoted as $\mathcal{F}_{u,v}$, define the strength of the influence of w . The edge weights $\omega_{f,vw} := \cot(\alpha_{f,vw})$ are given by the cotangent of the angle in f that is opposite to (v, w) . The weights are identical to those of the Cotan-Laplacian [8]. Together with the two parameters [5]

$$m_f = \int_f \langle \vec{r}, \vec{n} \rangle^2 d\Omega \quad (3)$$

$$\vec{b}_f = D \cdot \int_f \langle \vec{r}, \vec{n} \rangle \cdot \begin{pmatrix} n_x \\ n_y \end{pmatrix} d\Omega \quad (4)$$

they ensure that the integration yields the same results for different triangulations. Previously, these parameters have been calculated by assuming a constant face normal \vec{n}_f . In contrast, we found that

$$m_f = \frac{1}{|\mathcal{P}_f|} \sum_{p \in \mathcal{P}_f} \langle \vec{r}_p, \vec{n}_p \rangle^2 \quad (5)$$

$$\vec{b}_f = \frac{D}{|\mathcal{P}_f|} \langle \vec{r}_p, \vec{n}_p \rangle \cdot \begin{pmatrix} n_x \\ n_y \end{pmatrix}_p \quad (6)$$

		[1]	Isotropic [5]			Our Decimation, [5]’s Integration			Ours		
	Dataset	Ref	low	mid	high	low	mid	high	low	mid	high
Orthographic	Bear	2.97	3.95	3.65	3.37	3.67	3.33	3.19	3.84	3.38	3.04
	Buddha	6.74	7.74	7.54	7.33	7.30	7.10	7.08	6.86	6.68	6.61
	Cow	2.45	3.42	3.12	2.96	3.23	3.00	2.86	3.07	2.85	2.74
	Pot2	5.15	5.89	5.77	5.65	5.72	5.59	5.48	5.63	5.47	5.29
	Reading	6.34	7.08	6.93	6.83	6.88	6.76	6.64	6.82	6.67	6.50
Perspective	Bear	2.91	3.94	3.72	3.47	3.64	3.48	3.33	3.69	3.22	2.90
	Buddha	6.75	7.74	7.53	7.40	7.31	7.13	7.09	6.83	6.68	6.62
	Cow	2.35	3.49	3.24	3.07	3.29	3.09	2.99	2.97	2.77	2.63
	Pot2	4.99	6.04	5.86	5.76	5.81	5.69	5.61	5.48	5.32	5.16
	Reading	6.28	7.19	6.94	6.85	6.91	6.77	6.69	6.74	6.52	6.45

Table 1. Comparison of the average RMSE over all 20 views of DiLiGenT-MV: Isotropic remeshing and integration from [5], our decimation combined with the integrator of [5] and our decimation with our integrator. Our finer approximation of the integration parameters yields to a tighter approximation of the underlying surface.

is a tighter approximation that considers variations of the normals within the faces and yields more accurate surface integrations, see Tab. 1.

B. Algorithm

In this section, we want to express the quadrics in a more familiar form of a quadratic function and derive the linear system to solve during vertex alignment.

B.1. Explicit Form of the Quadrics

In Eq. (9) of the main paper, we defined the quadric for the continuous case as follows:

$$Q_v(\delta \vec{x}_v) := \sum_{f \in \mathcal{F}_v} \int_f \|J_f(\vec{u}_v - \vec{u}) + \delta \vec{x}_v\|_{M(\vec{x})}^2 d\Omega, \quad (7)$$

where both energy terms can be unified into one term by the norm induced by

$$M(\vec{x}) = \vec{n}(\vec{x}) \cdot \vec{n}(\vec{x})^t + \lambda \cdot \mathbb{1}, \quad (8)$$

cf. Eq. (13) of the main work. To obtain the known form of a quadratic problem,

$$Q_v(\delta \vec{x}_v) = \langle \delta \vec{x}_v, A_v \delta \vec{x}_v \rangle - 2 \langle \vec{b}_v, \delta \vec{x}_v \rangle + c_v \quad (9)$$

we have to apply the binomial formula to the integrand and rearrange the addends. In the end, we get

$$A_v = \sum_{f \in \mathcal{F}_v} \int_f M(\vec{x}) d\Omega \quad (10)$$

for the quadratic part,

$$\vec{b}_v = \sum_{f \in \mathcal{F}_v} \int_f M(\vec{x}) \cdot J_f(\vec{u}_v - \vec{u}) d\Omega$$

for the linear part and

$$c_v = \sum_{f \in \mathcal{F}_v} \int_f \|J_f(\vec{u}_v - \vec{u})\|_{M(\vec{x})} d\Omega \quad (11)$$

for the constant part.

In the discretised version (Eq. 12 of the main document), we replace the integral by a sum:

$$Q_v(\delta \vec{x}) = \sum_{f \in \mathcal{F}_v} \frac{A_f^{(3)}}{|\mathcal{P}_f|} \sum_{p \in \mathcal{P}_f} \|J_f \delta \vec{u}_{vp} + \delta \vec{x}\|_{M_p}^2. \quad (12)$$

Similarly, the integral is replaced by a sum in the coefficients of the linear system:

$$A_v = \sum_{f \in \mathcal{F}_v} \frac{A_f^{(3)}}{|\mathcal{P}_f|} M_p, \quad (13)$$

$$\vec{b}_f = \sum_{f \in \mathcal{F}_v} \frac{A_f^{(3)}}{|\mathcal{P}_f|} \cdot M_p \cdot J_f(\vec{u}_v - \vec{u}), \quad (14)$$

$$c_f = \sum_{f \in \mathcal{F}_v} \frac{A_f^{(3)}}{|\mathcal{P}_f|} \|J_f(\vec{u}_v - \vec{u})\|_{M_p}, \quad (15)$$

where $M_p = \vec{n}_p \cdot \vec{n}_p^t + \lambda \mathbb{1}$ from the pixel normals \vec{n}_p . However, when solving for the optimal vertex positions, we consider the following quadric:

$$\tilde{Q}_v(\delta \vec{u}_v) := Q_v(J_f \cdot \delta \vec{x}_v) \quad (16)$$

which is now in \mathbb{R}^2 since $J_f : \mathbb{R}^2 \rightarrow \mathbb{R}^3$. Hence, we must replace $\delta \vec{x} \mapsto J_f \cdot \delta \vec{x}_v$ in Eq. (9). As a result, the coefficients

of this quadratic function are:

$$\tilde{A}_v = J_f^t A_v J_f, \quad (17)$$

$$\tilde{b}_f = J_f^t \vec{b}_v, \quad (18)$$

$$\tilde{c}_v = c_v. \quad (19)$$

To find the final displacement $\delta \vec{u}_v$ that moves vertex v into its optimal position, we solve

$$\tilde{A}_v \delta \vec{u} = \tilde{b}_v. \quad (20)$$

By doing so, we neglect the influence of a vertex displacement on the adjacent vertices and their quadrics, which is a common approximation in mesh-processing [2, 10].

C. Evaluation

In this section, we provide additional benchmark results and insights on how the compression ratio and reconstruction error depend on the user-set decimation threshold. Further error maps are found in Appendix C.3. We also discuss two interesting outliers: One, where our anisotropic decimation method is more accurate than the pixel-based reference and one, where a higher decimation threshold, *i.e.* lower resolution mesh, surpasses higher resolutions. At the end of this section, we depict reconstructions of all objects of the LUCES [7], RGBN [9] and PS [3] dataset for various values of the decimation threshold and report vertex count and compression ratios.

C.1. Benchmark Comparison

To evaluate our method against previous work on mesh-based integration [5], we listed only the orthographic projection for the DiLiGenT-MV dataset [6] in the main document. We complement this comparison with the results of the perspective projection in Tab. 1. Furthermore, we perform the same evaluation on the LUCES dataset [7], but only for the perspective projection. Since this dataset is dedicated to near-field photometric stereo, *i.e.* objects are very close to the camera, the orthographic projection is an unsuitable approximation. As in previous tests, we match the vertex count to the low, mid and high settings of previous work [5]. Results are listed in Tab. 2. Again, our method generates tighter approximations of the underlying surface.

C.1.1 Inspecting the Buddha Results in DiLiGenT-MV

We noticed that our decimation-based method at the 'high' accuracy setting outperforms the pixel-based baseline in the case of the BUDDHA figurine in the DiLiGenT-MV dataset. This is surprising as our method uses fewer vertices than the pixel-based method. To investigate the origin of this result, we examined the differences in each view. Figure 1

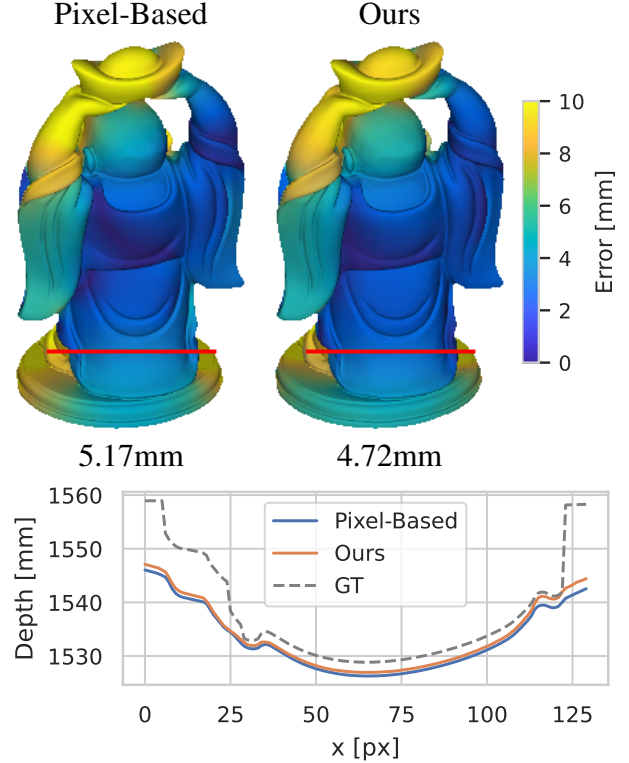


Figure 1. Top: Error map and RMSE for pixel-based [1] (left) and our (right) integration. Bottom: Slice of the aligned depth maps (along the red line indicated above).

illustrates the error map for the tenth view. While our decimation is generally lossy – except for perfectly flat regions – discontinuities and highly slanted (near-discontinuous) surfaces are problematic and error prone regions for normal integration. For the Buddha statue, such surfaces are present around the base and the lower part of the garment. It seems that our method performs slightly better in these situations. We believe this is due to differences between our integrator and the integrator in [1]. In [1] all normals are weighted equally while our method compensates for foreshortening assigning a higher weight to normals in slanted regions, as their real surface is bigger than it appears in screen space.

C.1.2 Inspecting the Buddha Results in LUCES

The BUDDHA figurine in the LUCES dataset shows an atypical behaviour: Lower mesh resolutions yield a better surface approximation. This inverted connection between resolution and reconstruction quality is also present for the previous isotropic normal integration [5], *cf.* Tab. 2. A visual inspection revealed that the normal integration places the Buddha's face too far in the front compared to the figurine's base, see Fig. 2. This is true for both pixel- and mesh-based integration. At lower mesh resolutions, there are more pix-

	[1]	Isotropic [5]			Ours		
Dataset	Ref	low	mid	high	low	mid	high
BALL	0.40	0.56	0.48	0.47	0.54	0.51	0.49
BELL	0.30	0.82	0.62	0.54	0.54	0.51	0.47
BOWL	0.08	0.35	0.22	0.15	0.15	0.14	0.12
BUDDHA	3.46	3.59	3.68	3.73	3.46	3.55	3.56
BUNNY	3.38	4.03	3.93	3.83	3.90	3.80	3.74
CUP	0.01	0.36	0.20	0.08	0.06	0.03	0.02
DIE	1.62	2.98	2.67	2.46	2.83	2.57	2.63
HIPPO	2.73	3.13	2.96	2.91	3.04	2.88	2.86
HOUSE	11.08	11.08	11.30	11.35	11.38	11.49	11.32
JAR	0.50	0.55	0.46	0.43	0.43	0.43	0.43
OWL	4.89	6.22	5.86	5.69	6.00	5.47	5.34
QUEEN	3.74	5.17	4.43	4.30	5.05	4.08	3.98
SQUIRREL	1.91	2.87	2.47	2.34	2.62	2.15	2.03
TOOL	0.91	1.04	0.96	0.93	0.91	0.90	0.90

Table 2. RMSE in mm for all objects of the LUCES dataset using the perspective projection. The vertex count is given by the respective low, mid and high settings of [5] and matched by our decimation pipeline.

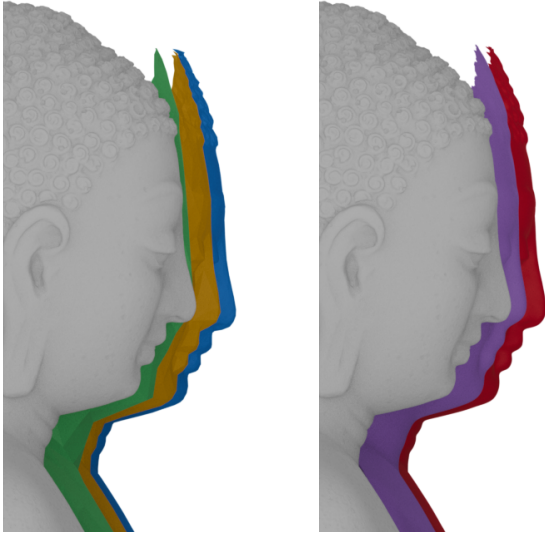


Figure 2. Side profiles of the BUDDHA figurine in LUCES [7]. Left: Our decimation with thresholds 2048 (green), 64 (yellow) and 2 (blue). Right: Pixel-based integration [4] with ground-truth normals (red) and smoothed normals (purple). All examples were aligned to the ground-truth surface (grey) at the base of the figurine.

els per triangle which implicitly smoothens the surface normals and flattens the integrated surface. This flattening coincidentally reduces the constant offset to the ground-truth surface. This hypothesis is supported by the fact that we can recreate this behaviour in pixel-based integration by applying a Gaussian kernel to the normal map, see Fig. 2.

C.2. Controllability

In the main paper, we studied the controllability using the LUCES [7] dataset. For completeness, we list all results in Tab. 3. For a more extensive study of the influence of the decimation threshold on the final mesh quality, we tested all normal maps of the DiLiGenT-MV dataset [6] for threshold values ranging from 0.25 to 512 and evaluated both root mean square error (RMSE) and mean absolute deviation (MADE) – both after the appropriate rigid-alignment to absolve the scale ambiguity – as well as mean angular error (MAE) and vertex count. The results are depicted in Fig. 3. As expected, there is virtually no difference between the results for orthographic and perspective projection. All objects are far away from the camera, *i.e.* the orthographic projection is a good approximation of the true perspective projection. The compression ratios reflect how our algorithm adapts to the complexity of the datasets: The BEAR and COW mostly consist of smooth featureless surfaces and achieve higher compression ratios than the more complex BUDDHA dataset, especially for lower thresholds.

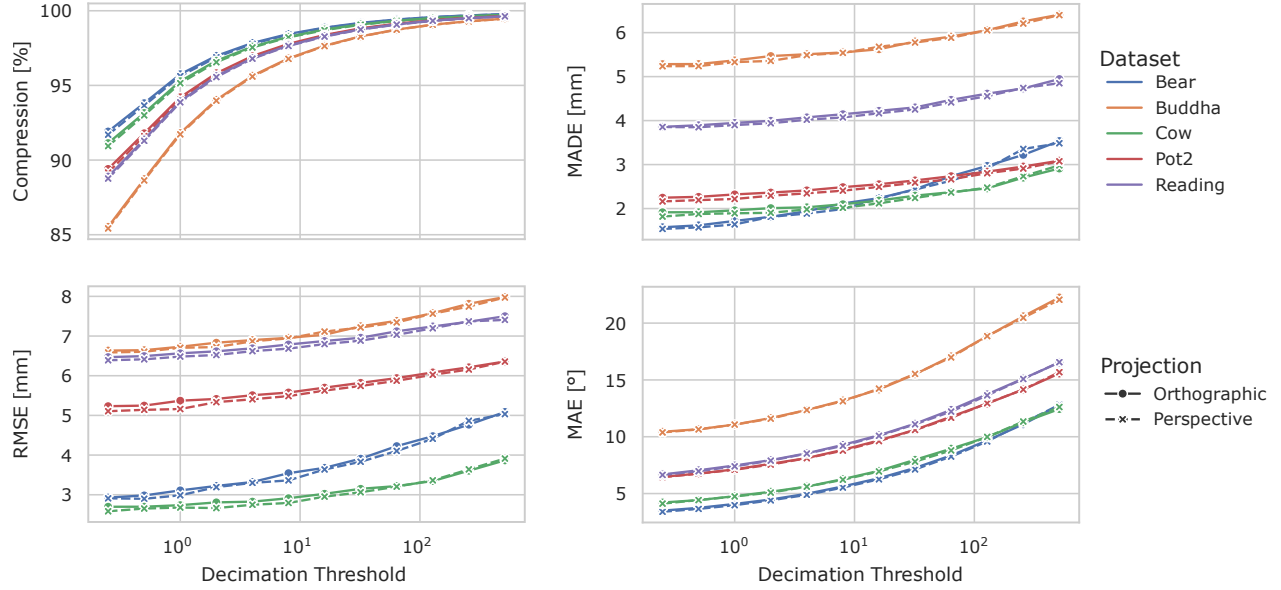


Figure 3. Influence of the decimation threshold on compression ratio, RMSE, MADE and MAE. All numbers are averages over the 20 views for each object. We investigate both orthographic and perspective projection. Please note the logarithmic x -axis.

Dataset	Ref	Threshold		
	[4]	2	2^6	2^{11}
BALL	0.40	0.48	0.54	0.60
BELL	0.30	0.33	0.49	0.72
BOWL	0.08	0.10	0.13	0.17
BUDDHA	3.46	3.58	3.51	3.32
BUNNY	3.38	3.67	3.83	4.15
CUP	0.01	0.01	0.04	0.14
DIE	1.62	1.65	2.11	3.09
HIPPO	2.73	2.85	2.91	3.09
HOUSE	11.08	11.30	11.56	11.39
JAR	0.50	0.45	0.45	0.46
OWL	4.89	5.41	5.63	5.94
QUEEN	3.74	4.02	4.41	4.98
SQUIRREL	1.91	2.07	2.37	3.00
TOOL	0.91	0.98	0.98	1.04

Table 3. RMSE in mm for all of the objects in LUCES [7] with increasing decimation threshold. The chosen thresholds should lead to a constant reduction rate of the RMSE.

C.3. Error Maps for the Benchmark Comparisons

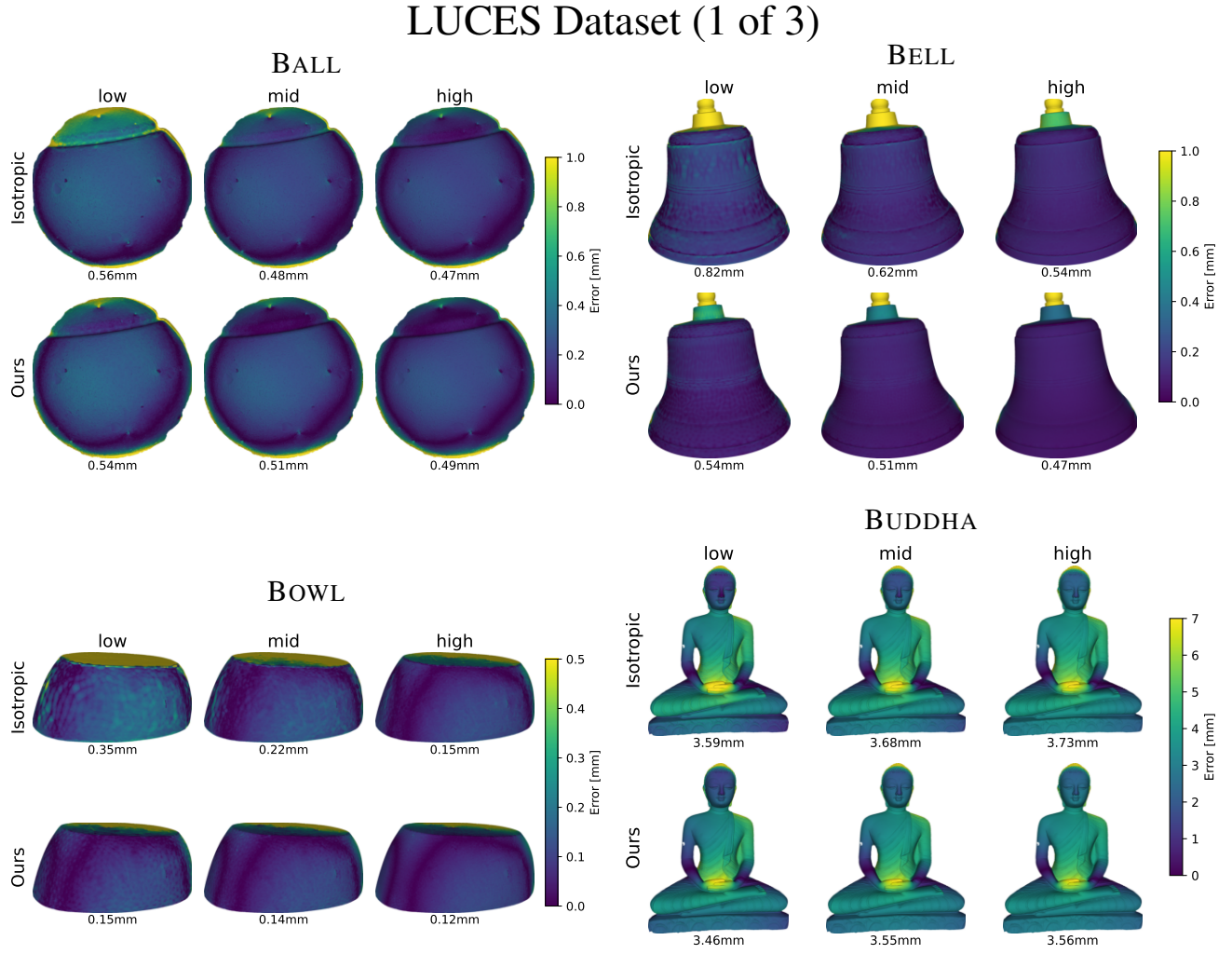


Figure 4. Error maps of the LUCES dataset after rigid alignment. We show results for all three quality settings in [5] and match the respective vertex number for our method. Pictured are the results of the perspective projection.

LUCES Dataset (2 of 3)

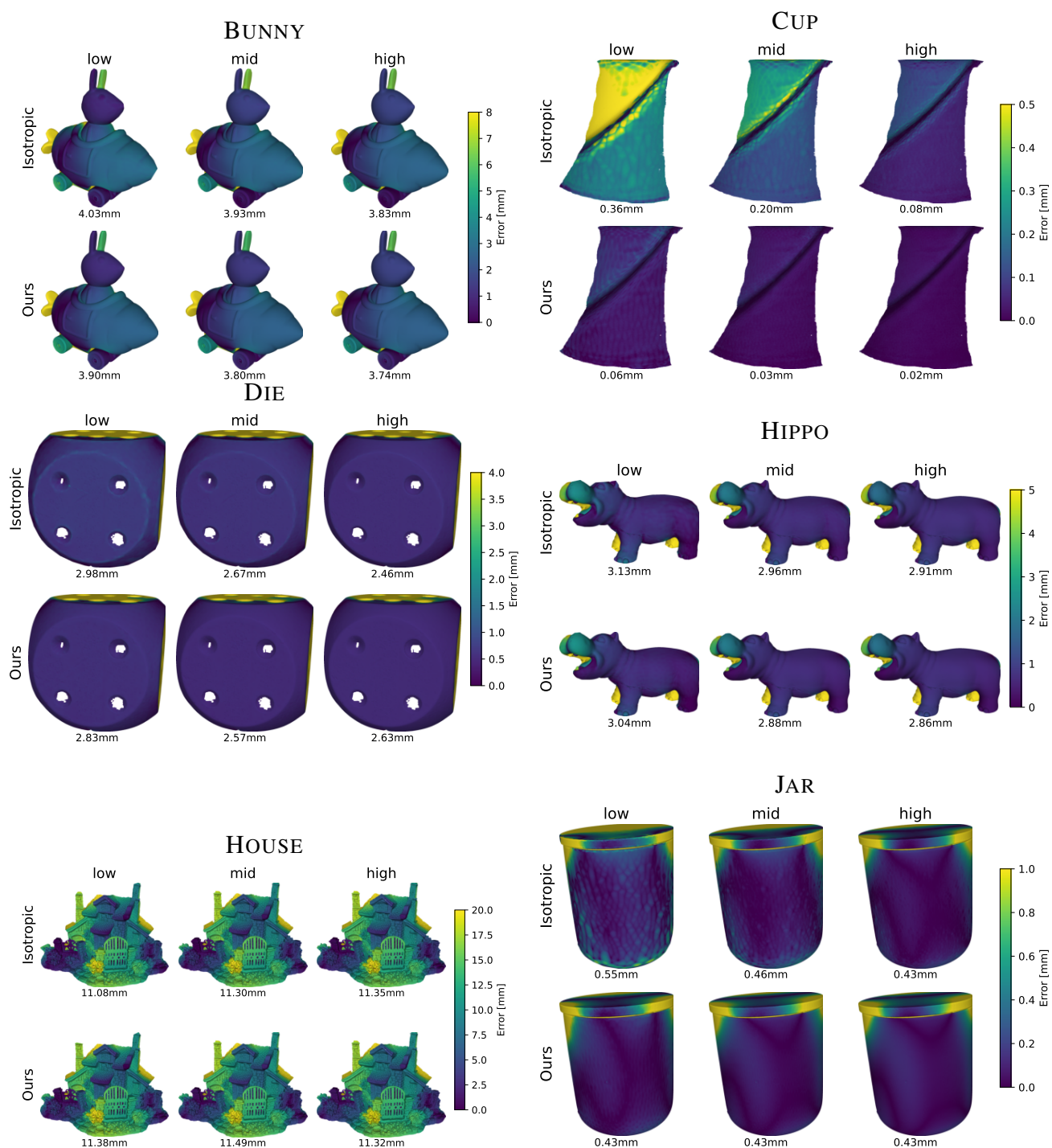


Figure 5. Error maps of the LUCES dataset after rigid alignment. We show results for all three quality settings in [5] and match the respective vertex number for our method. Pictured are the results of the perspective projection.

LUCES Dataset (3 of 3)

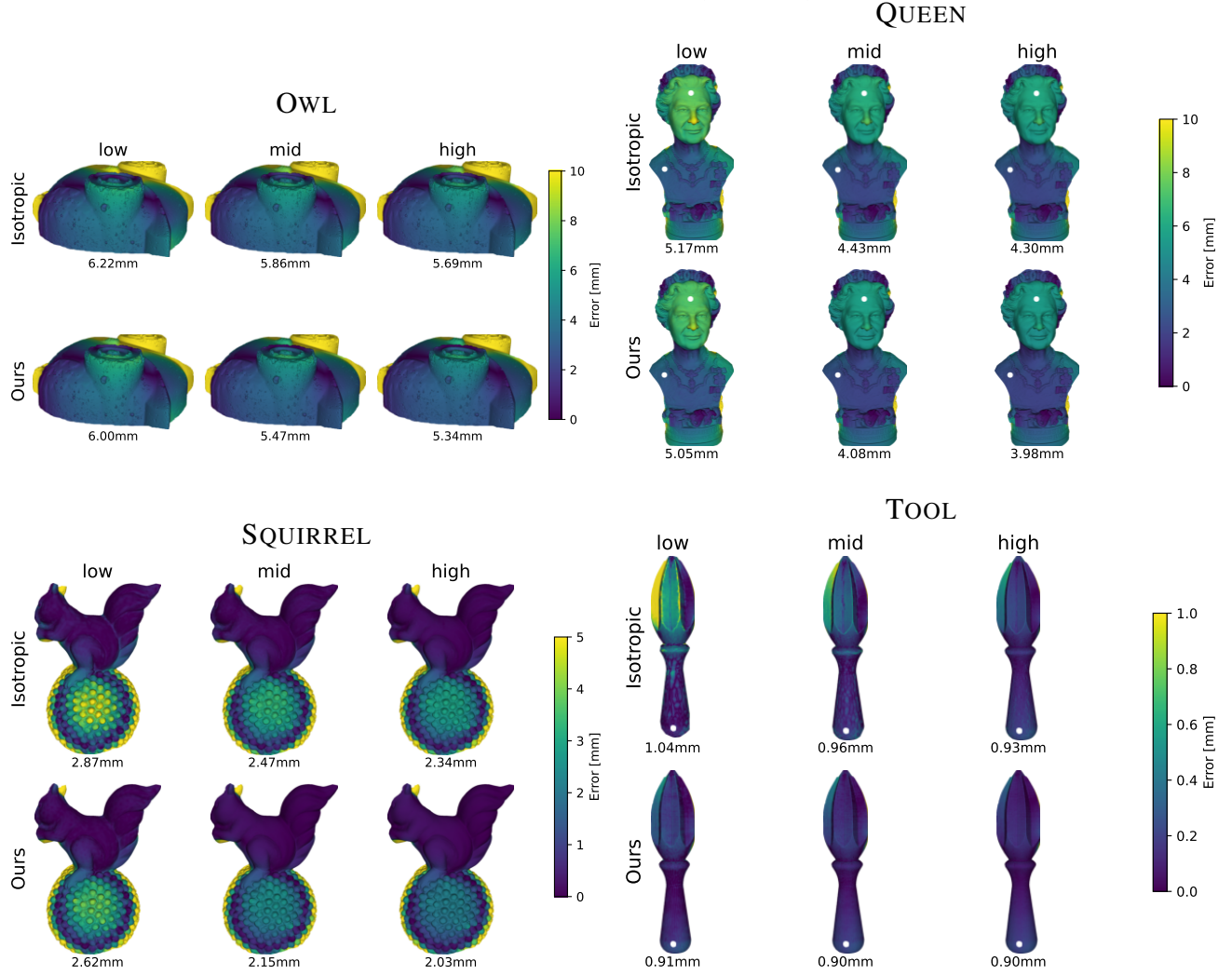


Figure 6. Error maps of the LUCES dataset after rigid alignment. We show results for all three quality settings in [5] and match the respective vertex number for our method. Pictured are the results of the perspective projection.

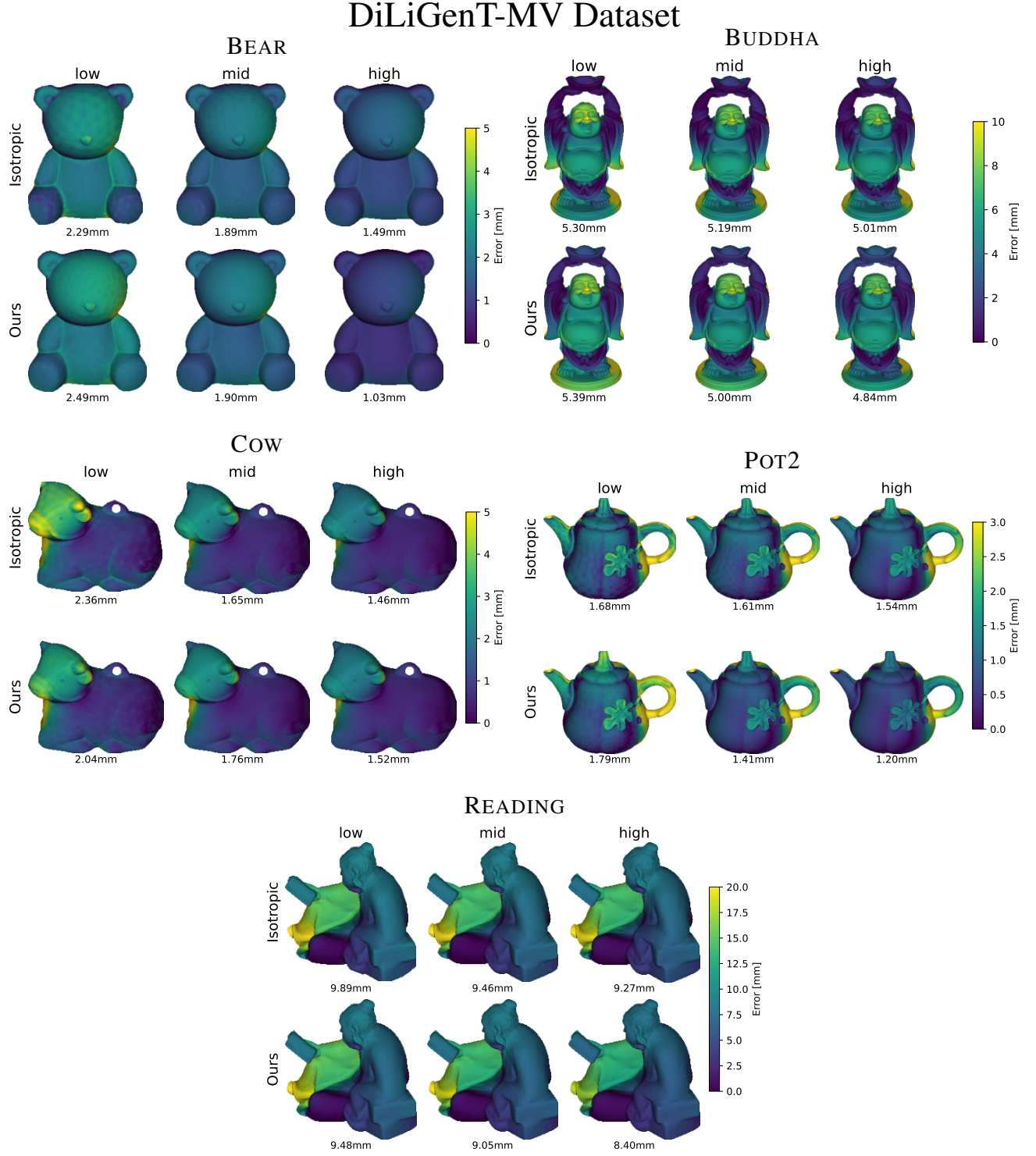


Figure 7. Error map for the first view in the DiLiGenT-MV dataset after rigid alignment. We show results for all three quality settings in [5] and match the respective vertex number for our method. Pictured are the results of the orthographic projection.

C.4. Additional Datasets

Finally, we show reconstructions from all the datasets we used. Except for LUCES [7], these datasets come without ground-truth geometry. The RMSEs for all objects in LUCES were reported in Tab. 3 which complements Tab. 2 from the submitted manuscript. For visual inspection, all objects can be found in Fig. 8 to Fig. 14. We also indicate the vertex count and compression ratio to put the results into perspective.

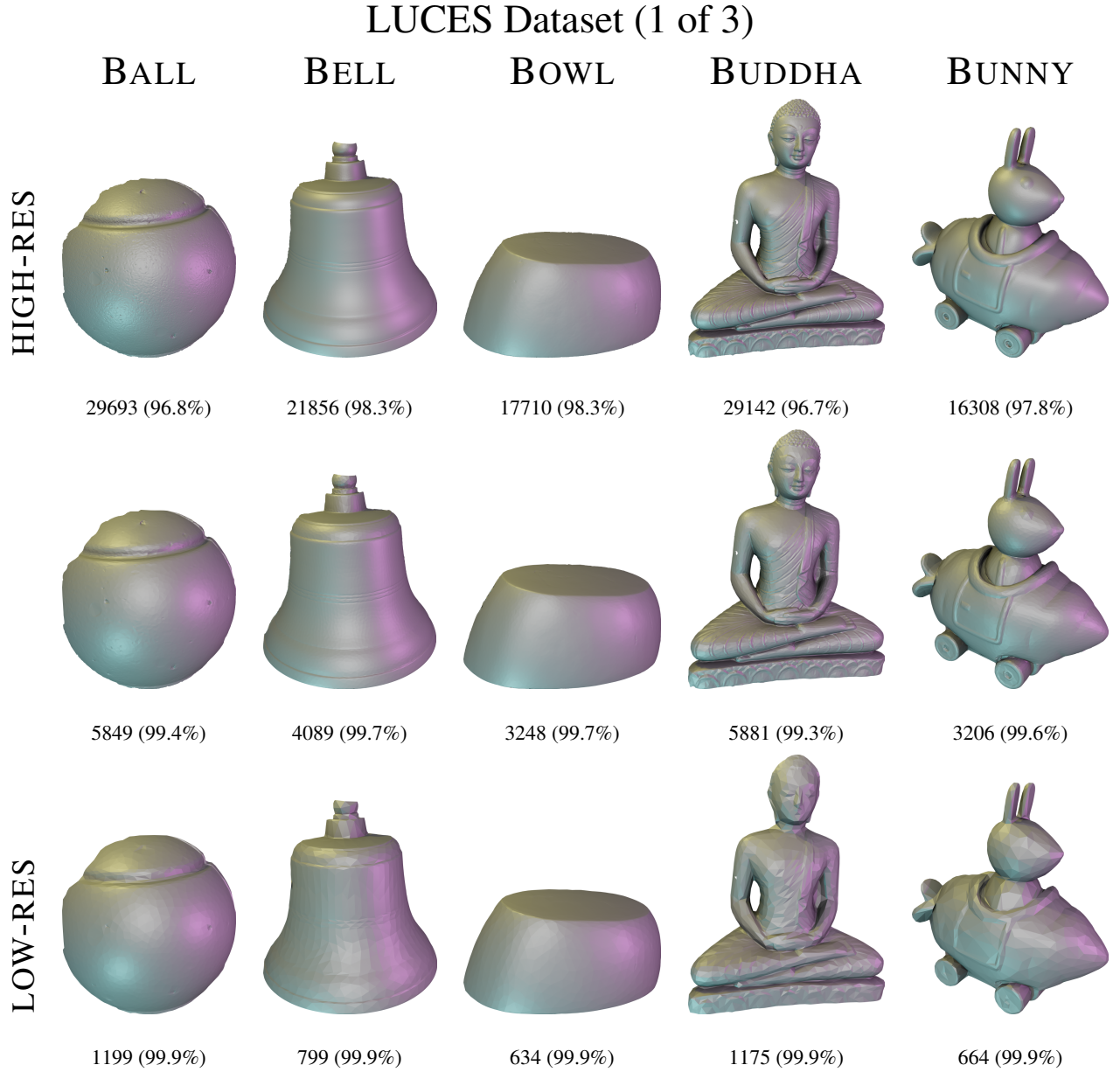


Figure 8. Reconstruction results for the LUCES dataset [7] for decimation thresholds of 2, 64 and 2048. These reconstructions correspond to the numbers reported in Tab. 2 of the manuscript. Any holes in the mesh surface are part of the provided foreground mask.

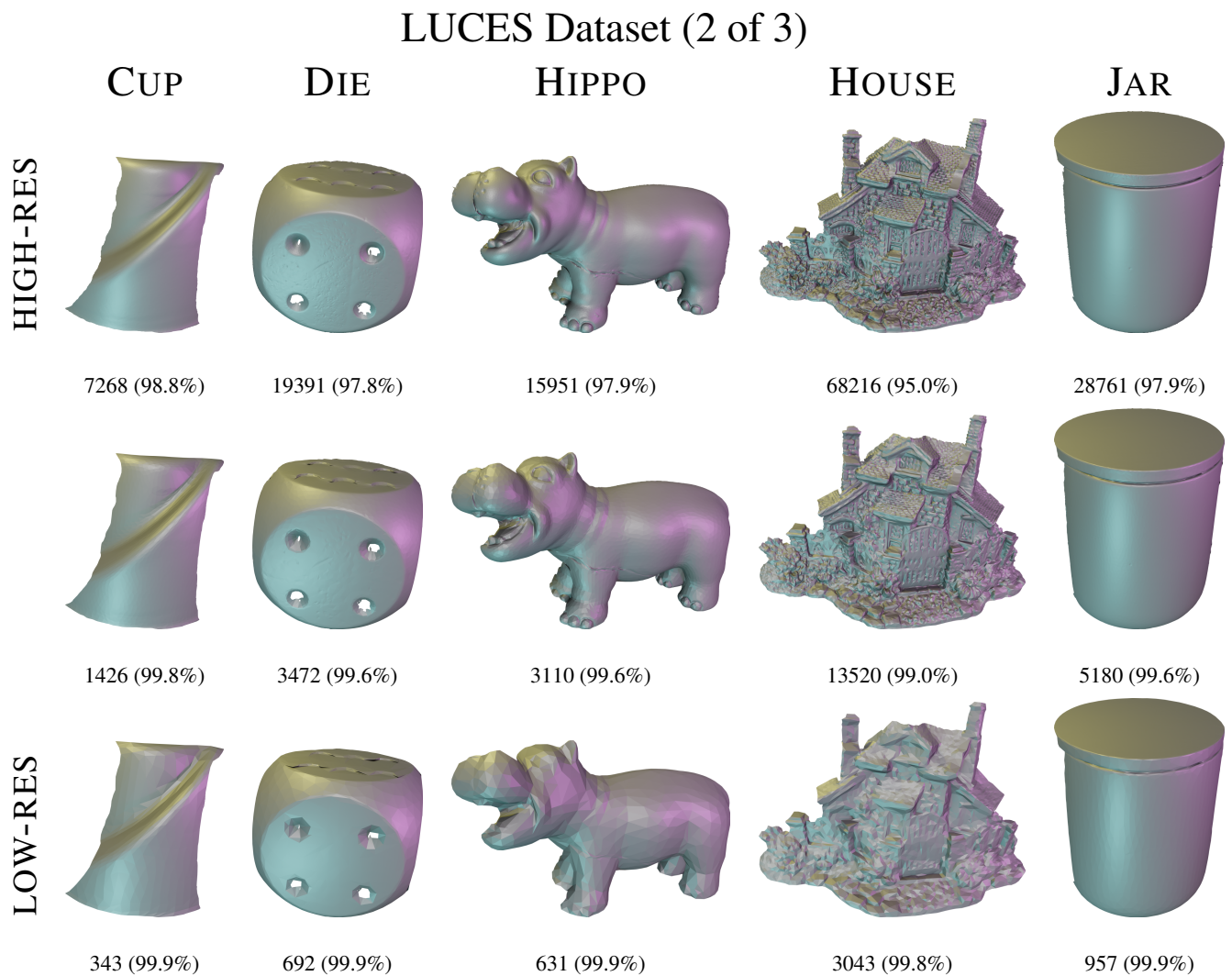


Figure 9. Reconstruction results for the LUCES dataset [7] for decimation thresholds of 2, 64 and 2048. These reconstructions correspond to the numbers reported in Tab. 2 of the main paper. Any holes in the mesh surface are part of the provided foreground mask.

LUCES Dataset (3 of 3)

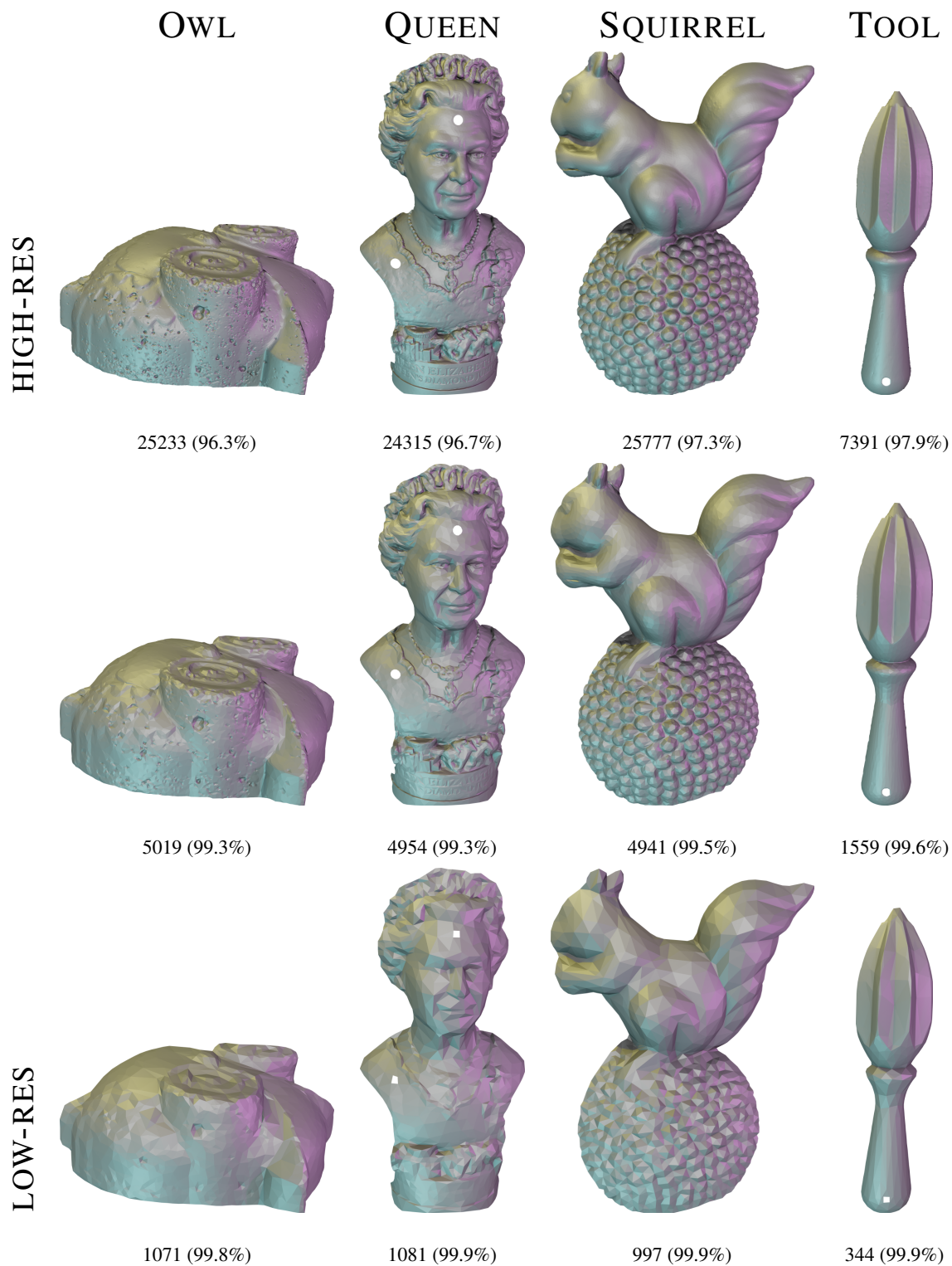


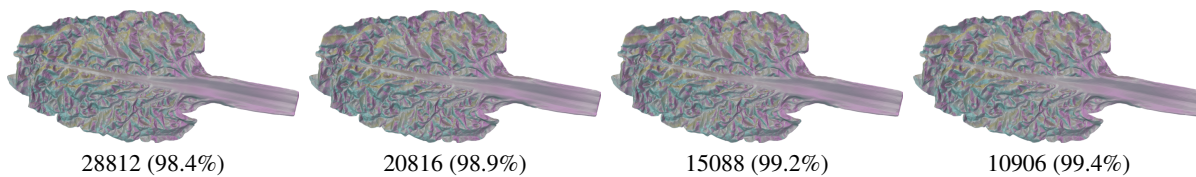
Figure 10. Reconstruction results for the LUCES dataset [7] for decimation thresholds of 2, 64 and 2048. These reconstructions correspond to the numbers reported in Tab. 2 of the main paper. Any holes in the mesh surface are part of the provided foreground mask.

RGBN Dataset (1 of 2)

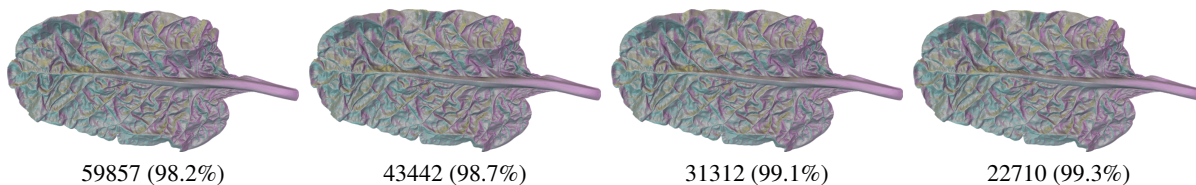
High-Res

Low-Res

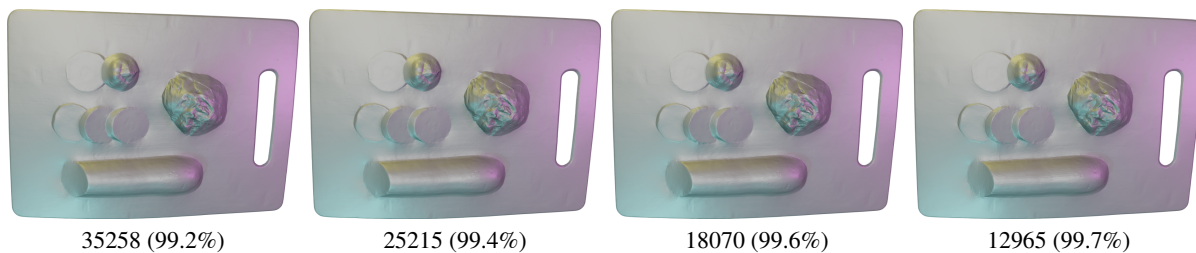
CHARD



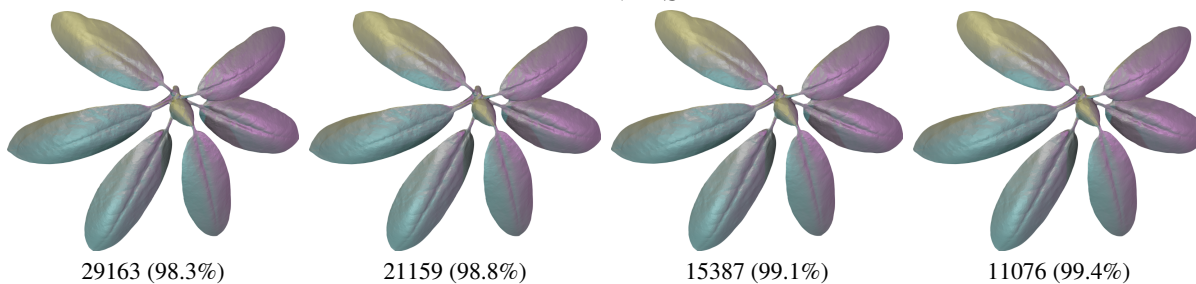
CHARD2



FOOD



LEAVES



SHELL

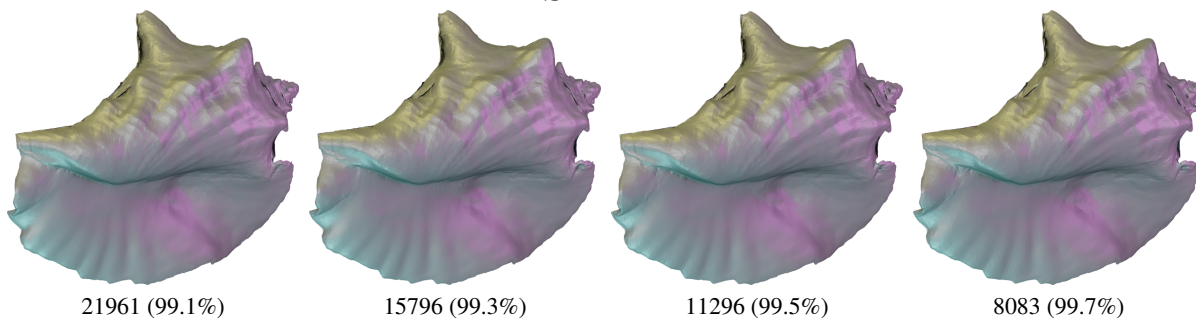


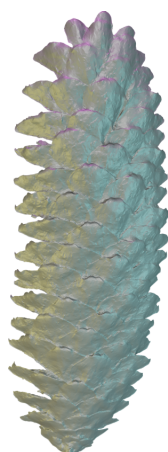
Figure 11. Reconstruction results for the RGBN dataset [9] for decimation thresholds of 8, 16, 32 and 64.

RGBN Dataset (2 of 2)

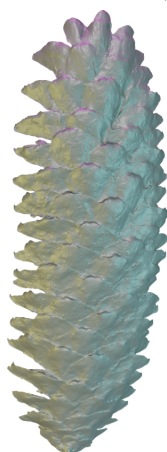
High-Res

Low-Res

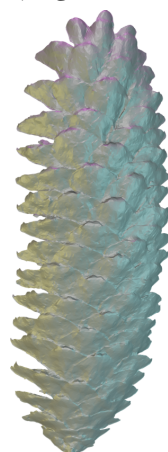
PINECONE3



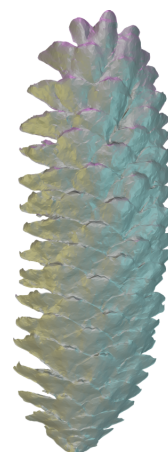
39834 (98.1%)



28919 (98.6%)



20951 (99.0%)



15176 (99.3%)

SOLDIER



49992 (98.5%)



36122 (98.9%)



26075 (99.2%)



18788 (99.4%)

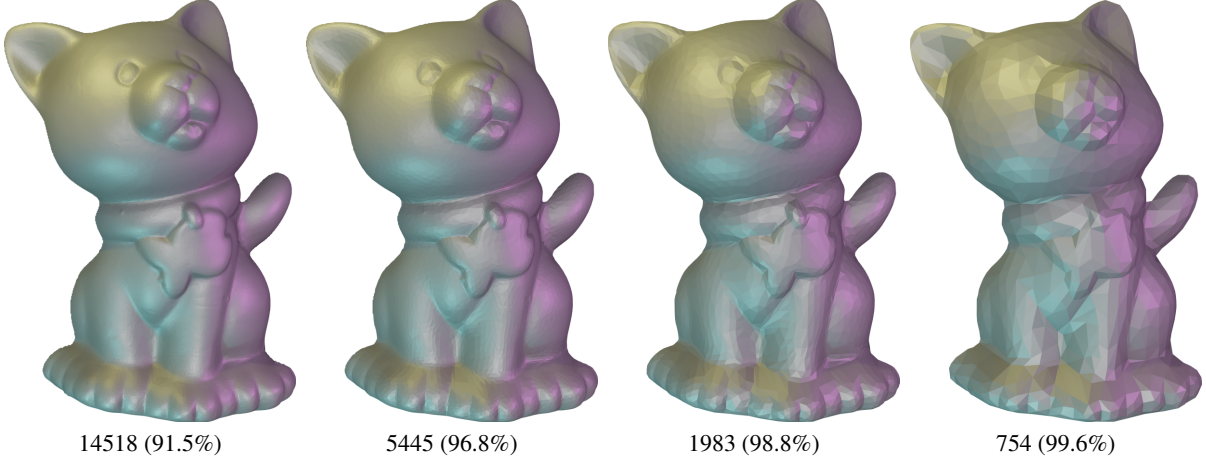
Figure 12. Reconstruction results for the RGBN dataset [9] for decimation thresholds of 8, 16, 32 and 64. Objects were rotated to the upright position.

PS Dataset (1 of 2)

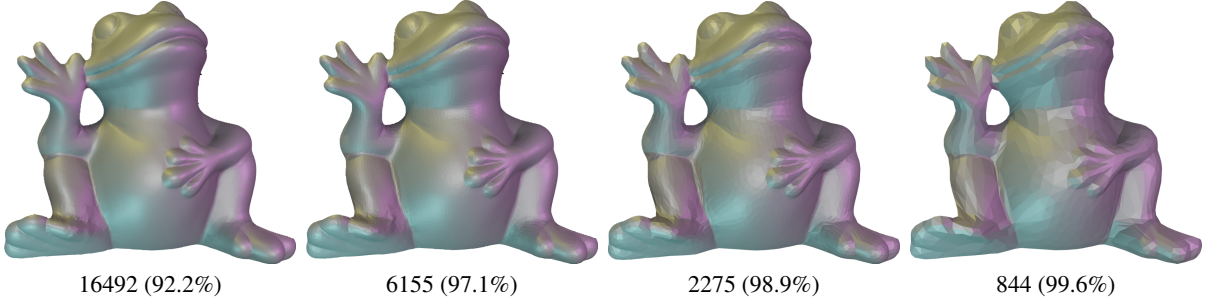
High-Res

Low-Res

CAT



FROG



HIPPO

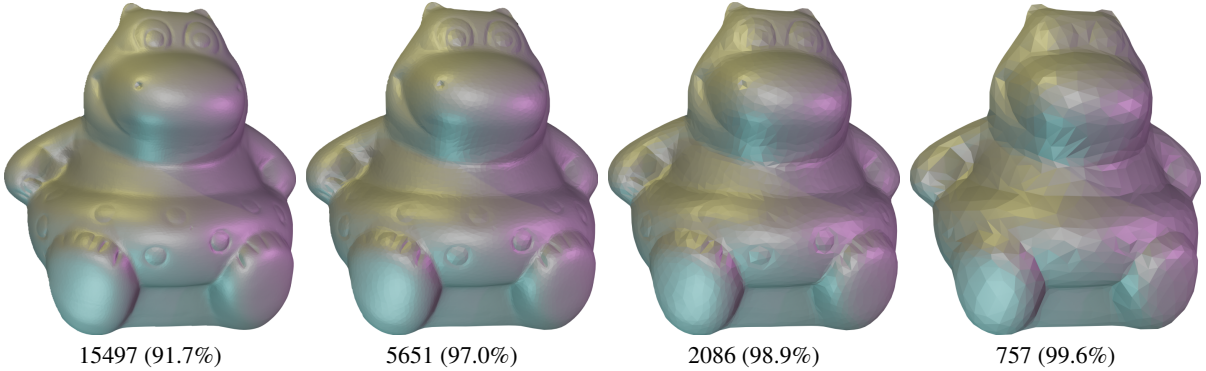


Figure 13. Reconstruction results for the PS dataset [3] with decimation thresholds of 0.125, 1, 8 and 64. The decimation threshold increases from left to right, *i.e.* mesh resolution decreases from left to right.

PS Dataset (2 of 2)

High-Res

Low-Res

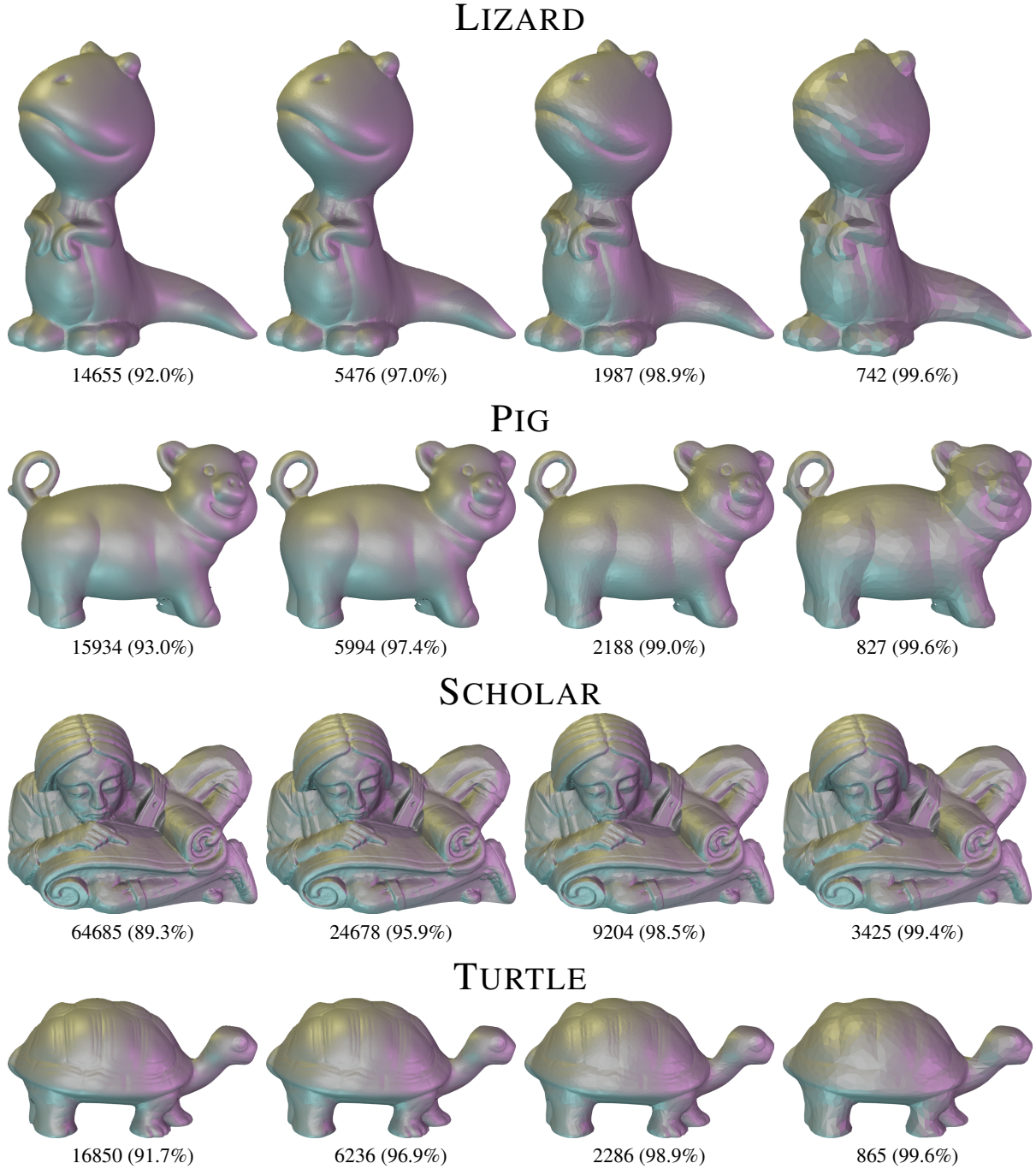


Figure 14. Reconstruction results for the PS dataset [3] with decimation thresholds of 0.125, 1, 8 and 64. The decimation threshold increases from left to right, *i.e.* mesh resolution decreases from left to right.

D. Overview of all Datasets

In this work, we used the following photometric stereo datasets:

- DiLiGenT-MV [6]: <https://sites.google.com/site/photometricstereodata/mv>
- LUCES [7]: <http://www.robertomecca.com/luces.html>
- RGBN [9]: <https://gfx.cs.princeton.edu/gfx/proj/rgbn/>
- PS Dataset [3]: <https://vision.seas.harvard.edu/qsfs/Data.html>

Furthermore, we generated synthetic datasets using the following 3D models:

- David Head [1d.inc]: <https://sketchfab.com/models/39a4d01bef37495cac8d8f0009728871/>
- Football Medal 2 [Cain]: <https://sketchfab.com/models/54d54534f11d4d23aecb945fd7eb1df4/>
- Female Head: <https://www.3dscanstore.com/3d-head-models/raw-expression-bundles/female-02-x36-expression-bundle>
- Male Head: <https://www.3dscanstore.com/3d-head-models/raw-expression-bundles/male-01-36x-expression-scan-bundle>

References

- [1] Xu Cao, Boxin Shi, Okura Fumio, and Yasuyuki Matsushita. Normal Integration via Inverse Plane Fitting With Minimum Point-to-Plane Distance. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2382–2391, 2021. [2](#), [3](#), [4](#)
- [2] Long Chen. Mesh Smoothing Schemes Based on Optimal Delaunay Triangulations. In *Proc. of International Meshing Roundtable (IMR)*, pages 109–120. Citeseer, 2004. [3](#)
- [3] Robert T. Frankot and Rama Chellappa. A method for enforcing integrability in shape from shading algorithms. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 10(4): 439–451, 1988. [3](#), [15](#), [16](#), [17](#)
- [4] Moritz Heep and Eduard Zell. ShadowPatch: Shadow Based Segmentation for Reliable Depth Discontinuities in Photometric Stereo. *Computer Graphics Forum*, 41(7):635–646, 2022. [4](#), [5](#)
- [5] Moritz Heep and Eduard Zell. An Adaptive Screen-Space Meshing Approach for Normal Integration. In *European Conference on Computer Vision, (ECCV)*, pages 445–461, Milan, Italy, 2025. Springer. [1](#), [2](#), [3](#), [4](#), [6](#), [7](#), [8](#), [9](#)
- [6] Min Li, Zhenglong Zhou, Zhe Wu, Boxin Shi, Changyu Diao, and Ping Tan. Multi-view photometric stereo: A robust solution and benchmark dataset for spatially varying isotropic materials. *IEEE Trans. on Image Processing*, 29: 4159–4173, 2020. [1](#), [3](#), [4](#), [17](#)
- [7] Roberto Mecca, Fotios Logothetis, Ignas Budvytis, and Roberto Cipolla. LUCES: A Dataset for Near-Field Point Light Source Photometric Stereo, 2021. [3](#), [4](#), [5](#), [10](#), [11](#), [12](#), [17](#)
- [8] Ulrich Pinkall and Konrad Polthier. Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics*, 2(1):15–36, 1993. [1](#)
- [9] Corey Toler-Franklin, Adam Finkelstein, and Szymon Rusinkiewicz. Illustration of complex real-world objects using images with normals. In *Proc. of Int. Symposium on Non-photorealistic Animation and Rendering NPAR*, pages 111–119, San Diego California, 2007. ACM. [3](#), [13](#), [14](#), [17](#)
- [10] Rui Xu, Longdu Liu, Ningna Wang, Shuangmin Chen, Shiqing Xin, Xiaohu Guo, Zichun Zhong, Taku Komura, Wenping Wang, and Changhe Tu. CWF: Consolidating Weak Features in High-quality Mesh Simplification. *ACM Trans. on Graphics*, 43(4):80:1–80:14, 2024. [3](#)