Supplementary: Material-aware Gaussian Splatting for High-fidelity Mesh Reconstruction

D. M. Nguyen[1], M. Avenhaus[2], and T. Lindemeier[2]

[1]Norwegian University of Science and Technology, Norway
[2]Carl Zeiss AG, Germany

1. Losses

3DGS [KKLD23] proposed the RGB photometric loss $\mathcal{L}_{\text{rgb}}$, comprising the pixel-wise mean absolute error and structural similarity index [BPH24] between the rendered image $\hat{\mathcal{I}}$ and its ground truth $\mathcal{I}$, as shown in Equation 1. We set $\lambda_{\text{SSIM}} = 0.2$, aligning with previous works.

$$\mathcal{L}_{\text{rgb}} = (1-\lambda_{\text{SSIM}})L_1 + \lambda_{\text{SSIM}} L_{\text{D-SSIM}} \tag{1}$$

To regulate the shape of Gaussians for unbiased depth rendering, we adopt the plane loss $\mathcal{L}_{\text{plane}}$ from [CLY*25]. Equation 2 defines $\mathcal{L}_{\text{plane}}$, where min selects the smallest component of an input vector, and $\mathcal{V}$ denotes the set of Gaussians that are visible during the rasterization process. We set $\lambda_{\text{plane}} = 100.0$, following [CLY*25].

$$\mathcal{L}_{\text{plane}} = \frac{\lambda_{\text{plane}}}{|\mathcal{V}|} \sum_{\mathbf{s}} ||\min(\mathbf{s}_i)||_1 \tag{2}$$

We follow previous works [HYC*24, HSG24, DXX*24, YSG24, WLW*24, CLY*25, JTL*24, LZF*24, ZWY25] in defining the depth-normal consistency loss $\mathcal{L}_{\text{dn}}$. A normal map $\hat{\mathcal{N}}$ can be estimated for a viewpoint by applying a Sobel-like operator to the rendered depth map $\mathcal{D}$. The process first uses depth values from $\mathcal{D}$ and camera intrinsics K to find 3D points $\mathbf{p}_c$ in camera space for all pixel coordinates $\mathbf{p}$. It then computes for each $\mathbf{p}_c$ two directional vectors, $\vec{g}_u$ and $\vec{g}_v$, estimating the local gradients by using 4 neighbors of $\mathbf{p}_c$. The normal direction at $\mathbf{p}_c$ can now be derived by taking the cross product of these gradients, as described in Equation 3. Equation 4 derives $\mathcal{L}_{\text{dn}}$, where the norm-1 error is weighted with gradients of $\mathcal{I}$ for edge-aware consistency. We set $\lambda_{\text{dn}} = 0.03$, determined via empirical experiments.

$$\mathbf{p}_c = \mathcal{D}(\mathbf{p})K^{-1}\tilde{\mathbf{p}}, \quad \hat{\mathcal{N}} = \frac{\vec{g}_u \times \vec{g}_v}{|\vec{g}_u \times \vec{g}_v|} \tag{3}$$

$$\mathcal{L}_{\text{dn}} = \frac{\lambda_{\text{dn}}}{|\mathcal{N}|} \sum_{\mathbf{p}} |\nabla \mathcal{I}(\mathbf{p})|^2 ||\mathbf{n}_{\mathbf{p}} - \hat{\mathbf{n}}_{\mathbf{p}}||_1 \tag{4}$$

Until recently, multi-view consistency constraints received credit for boosting mesh reconstruction performance [CLY*25, WLW*24, HSG24]. $\mathcal{L}_{\text{mv}}$ consists of multi-view geometric and photometric errors, denoted L_{g} and L_{p}, applied on grayscale ground truth $\mathcal{I}_{\text{G}}$. The loss is computed as their weighted sum, as shown in Equation 5, where λ_{g} and λ_{p} are set to 10^{-3} and 0.15, respectively.

$$\mathcal{L}_{\text{mv}} = \frac{1}{|\mathcal{I}_{\text{G}}|} \sum_{\mathbf{p}} (\lambda_{\text{g}} L_{\text{g}}(\mathbf{p}) + \lambda_{\text{p}} L_{\text{p}}(\mathbf{p})) \tag{5}$$

To stimulate geometric consistency across views, we perform forward and backward projections of pixels in a reference view $\mathcal{C}_{\text{r}}$ with a neighbor view $\mathcal{C}_{\text{n}}$ and minimize the reprojection error. Unlike PGSR [CLY*25], we incorporate a decay rate $r_{\text{g}} = 3.0$ into the exponential weight involved in L_{g} computation, as shown in Equation 6. Here, $\hat{\mathbf{p}}_{\text{n}}$ and $\hat{\mathbf{n}}_{\text{n}}$ are pixels and their normals in a neighboring view corresponding to pixels $\mathbf{p}_{\text{r}}$ and normals $\mathbf{n}_{\text{r}}$ in the reference view, respectively. These correspondences are identified through forward and backward projections with occlusion-aware filtering.

$$L_{\text{g}}(\mathbf{p}) = \exp(-r_{\text{g}}||\hat{\mathbf{p}}_{\text{n}} - \mathbf{p}_{\text{r}}||_1)(||\hat{\mathbf{p}}_{\text{n}} - \mathbf{p}_{\text{r}}||_1 + \arccos(\hat{\mathbf{n}}_{\text{n}} \cdot \mathbf{n}_{\text{r}})) \tag{6}$$

For the multi-view photometric consistency, we compute the L_{p} term identically to [CLY*25]'s implementation, i.e., the normalized cross-correlation (NCC) [YH09] between warped patches $\mathcal{P}_{\text{r}}$ and $\hat{\mathcal{P}}_{\text{n}}$ sampled from gray images of reference and neighbor views, respectively. As before, corresponding patches $\hat{\mathcal{P}}_{\text{n}}$ in neighbor views are found via homography matrices H_{rn} that transform pixels from $\mathcal{C}_{\text{r}}$ to $\mathcal{C}_{\text{n}}$. In particular, Equation 7 defines L_{p}, where exponential weighting is applied as with L_{g}.

$$L_{\text{p}}(\mathbf{p}) = \frac{1}{|\mathcal{P}|} \sum_{p} \exp(-||\hat{\mathbf{p}}_{\text{n}} - \mathbf{p}_{\text{r}}||_1)(1 - L_{\text{NCC}}(\hat{\mathcal{P}}_{\text{n}}(p), \mathcal{P}_{\text{r}}(p))) \tag{7}$$

The total variance measured for a feature map $\mathcal{X}$ and finite difference calculated for a weight map $\mathcal{W}$ (defined for all image pixels $\mathbf{p} = (u, v)$) are given in Equation 8 and 9, respectively.

$$\begin{aligned}\Delta_{uv}^{\mathcal{X}} = &\exp(-||\mathcal{I}_{u,v} - \mathcal{I}_{u-1,v}||_1)||\mathcal{X}_{u,v} - \mathcal{X}_{u-1,v}||_1 \\ &+\exp(-||\mathcal{I}_{u,v} - \mathcal{I}_{u,v-1}||_1)||\mathcal{X}_{u,v} - \mathcal{X}_{u,v-1}||_1\end{aligned} \tag{8}$$

$$\Omega_{uv}^{\mathcal{W}} = 0.5 * (\mathcal{W}_{u,v} + \mathcal{W}_{u-1,v} + \mathcal{W}_{u,v} + \mathcal{W}_{u,v-1}) \tag{9}$$

We then provide the normal total variance loss $\mathcal{L}_{\text{tv}}$ and the BRDF smoothness loss $\mathcal{L}_{\text{sm}}$ in Equation 10 and 11, respectively, with $\lambda_{\text{normal}} = 0.1$, $\lambda_{\text{smooth}} = 0.01$, and $k_{\text{ro}} = 8.0$.

$$\begin{aligned}\mathcal{W}_{\mathcal{R}} &= 0.5 \times \tanh(k_{\text{ro}}(0.5 - \mathcal{R})) + 0.5 \\ \mathcal{L}_{\text{tv}} &= \frac{\lambda_{\text{normal}}}{|\mathcal{N}|} \sum_{u,v} \Omega_{uv}^{\mathcal{W}_{\mathcal{R}}} \Delta_{uv}^{\mathcal{N}}\end{aligned} \tag{10}$$

$$\mathcal{L}_{\text{sm}} = \frac{\lambda_{\text{smooth}}}{|\mathcal{X}|} \left(\Delta_{uv}^{\mathcal{A}} + \Delta_{uv}^{\mathcal{R}}\right) \tag{11}$$

If ground-truth masks are provided, we apply the alpha loss $\mathcal{L}_{\text{alpha}}$ as shown in Equation 12. Here, $\hat{\mathcal{T}}$ and $\mathcal{T}$ are the blended alpha map and ground truth mask defined for all pixels $\mathbf{p}$, with $\lambda_{\text{alpha}} = 0.2$.

$$\mathcal{L}_{\text{alpha}} = -\frac{\lambda_{\text{alpha}}}{|\mathcal{T}|} \sum_{\mathbf{p}} [\alpha_{\mathbf{p}} \log \hat{\alpha}_{\mathbf{p}} + (1 - \alpha_{\mathbf{p}}) \log(1 - \hat{\alpha}_{\mathbf{p}})] \tag{12}$$

The total loss $\mathcal{L}$ is finally given in Equation 13, where $\mathcal{L}_{\text{pbr}}$ is essentially $\mathcal{L}_{\text{rgb}}$ with $\hat{\mathcal{I}}$ replaced with $\bar{\mathcal{I}}$—the PBR image. Please refer to the main paper for the detailed implementation of $\mathcal{L}_{\text{ro}}$ and the joint optimization process.

$$\mathcal{L} = \mathcal{L}_{\text{plane}} + \mathcal{L}_{\text{alpha}} + \mathcal{L}_{\text{dn}} + \mathcal{L}_{\text{mv}} + \mathcal{L}_{\text{tv}} + \mathcal{L}_{\text{sm}} + \mathcal{L}_{\text{ro}} + \mathcal{L}_{\text{pbr}} \tag{13}$$

2. Physically based rendering

To recap, [Kaj86] introduced the rendering equation calculating the outgoing radiance $L_{\text{o}}(\mathbf{x}, \omega_{\text{o}})$ at a surface point $\mathbf{x}$ in direction ω_{o} as in Equation 14.

$$L_{\text{o}}(\mathbf{x}, \omega_{\text{o}}) = L_{\text{e}}(\mathbf{x}, \omega_{\text{o}}) + \int_{S^2} f_r(\mathbf{x}, \omega_{\text{i}}, \omega_{\text{o}}) L_{\text{i}}(\mathbf{x}, \omega_{\text{i}}) |\cos\theta_{\text{i}}| d\omega_{\text{i}} \tag{14}$$

The outgoing radiance is described as the sum of the emitted radiance $L_{\text{e}}(\mathbf{x}, \omega_{\text{o}})$ at $\mathbf{x}$ along ω_{o}, and the incident radiance from all directions on the sphere S^2 centered at $\mathbf{x}$ scaled by the bidirectional reflectance distribution function (BRDF) $f_r(\mathbf{x}, \omega_{\text{i}}, \omega_{\text{o}})$ and a cosine term. We employ the Cook-Torrance microfacet model [CT82, WMLT07] to formulate the BRDF given an arbitrary surface point, as shown in Equation 15.

$$f_r(\omega_{\text{i}}, \omega_{\text{o}}) = (1 - m)\frac{\mathbf{a}}{\pi} + \frac{D \times F \times G}{4(\mathbf{n} \cdot \omega_{\text{o}})(\mathbf{n} \cdot \omega_{\text{i}})} \tag{15}$$

Here, the albedo $\mathbf{a}$, metallic m, and normal $\mathbf{n}$ values are retrieved from the rendered albedo $\mathcal{A}$, metallic $\mathcal{M}$, and normal $\mathcal{N}$ map, respectively, as the shading is performed during the deferred rendering step. The Fresnel term F quantifies how reflectance changes with viewing angle and is computed using Schlick's approximation [Sch94]. The geometry term G accounts for self-shadowing and masking, where microfacets occlude light or block reflected rays. The normal distribution function D describes the statistical orientation of microfacets relative to the surface normal. It determines the concentration of microfacets aligned with the halfway vector $\mathbf{h}$ between light and view directions, i.e, $\mathbf{h} = (\omega_{\text{o}} + \omega_{\text{i}})/|\omega_{\text{o}} + \omega_{\text{i}}|$. Equation 16 details the D term, where $\theta_{\mathbf{h}}$ is the angle between $\mathbf{h}$ and the surface normal $\mathbf{n}$, and roughness values ρ are extracted from the rendered roughness map $\mathcal{R}$.

$$D(\rho, \theta_{\mathbf{h}}) = \frac{\rho^2}{\pi(\cos^2\theta_{\mathbf{h}}(\rho^2 - 1) + 1)^2} \tag{16}$$

Since the integral term in Equation 14 is intractable in practice and requires specialized ray-tracing hardware for efficient sampling, [KG13] proposed an approximation to the rendering equation. Specifically, when substituting the BRDF formulation of Equation 15 into Equation 14, the rendering equation can be decomposed into a diffuse L_{d} and specular L_{s} component. As light-emitting surfaces are not within the scope of our method, the L_{e} term is dropped, and Equation 14 can be rewritten as Equation 17.

$$L_{\text{o}}(\mathbf{x}, \omega_{\text{o}}) = L_{\text{d}}(\mathbf{x}, \omega_{\text{o}}) + L_{\text{s}}(\mathbf{x}, \omega_{\text{o}}) \tag{17}$$

The diffuse L_{d} term is defined in Equation 18, where the integral domain S^2 is replaced with the hemisphere Ω about the surface point $\mathbf{x}$ and normal $\mathbf{n}$ as we do not model transmission. The diffuse lighting (the integral term) can be interpreted as incoming irradiance to $\mathbf{x}$ given an outgoing direction ω_{o}. As a result, it can be prefiltered by convolving a cubemap storing the environment radiance. At runtime, the cubemap is sampled from ω_{o} direction to retrieve its corresponding irradiance value.

$$L_{\text{d}}(\mathbf{x}, \omega_{\text{o}}) = (1 - m)\frac{\mathbf{a}}{\pi} \int_{\Omega} L_{\text{i}}(\mathbf{x}, \omega_{\text{i}})(\mathbf{n} \cdot \omega_{\text{i}}) d\omega_{\text{i}} \tag{18}$$

$$L_{\text{s}}(\mathbf{x}, \omega_{\text{o}}) = \int_{\Omega} \frac{D \times F \times G}{4(\mathbf{n} \cdot \omega_{\text{o}})(\mathbf{n} \cdot \omega_{\text{i}})} L_{\text{i}}(\mathbf{x}, \omega_{\text{i}})(\mathbf{n} \cdot \omega_{\text{i}}) d\omega_{\text{i}} \tag{19}$$

The specular L_{s} term, defined in Equation 19, has the integral term depending on ω_o and thus cannot be pre-computed like L_{d}. Therefore, the split-sum approximation proposed by [KG13] treats L_{s} as two components following Equation 20.

$$L_{\text{s}}(\mathbf{x}, \omega_{\text{o}}) \approx \int_{\Omega} L_{\text{i}}(\mathbf{x}, \omega_{\text{i}}) d\omega_{\text{i}} \int_{\Omega} \frac{D \times F \times G}{4(\mathbf{n} \cdot \omega_{\text{o}})(\mathbf{n} \cdot \omega_{\text{i}})} (\mathbf{n} \cdot \omega_{\text{i}}) d\omega_{\text{i}} \tag{20}$$

The first component of this approximation assumes constant incoming lighting and is fairly accurate for common environments.

This component can be prefiltered in the same manner as L_d, but it must take roughness into account, with increasing roughness values gradually blurring the cubemap. As a result, the environment lighting is precomputed at multiple mip resolutions, each corresponding to a roughness level. At runtime, the supplied roughness ρ will select the closest environment mipmap to sample irradiance from, avoiding the need for expensive computation. Thanks to this splitting, the BRDF's response values in the remaining term can now be pre-computed and stored in a 2D lookup texture (LUT). Similar to the first term, the 2D LUT encodes information for each roughness level and can be efficiently sampled during runtime to retrieve the integral value. The prefiltered cubemap and its mip scales make up the environment light map $\mathcal{E}$ of the scene.

In our implementation, the above formulation is realized as a fully differentiable deferred shading module operating on the rendered G-buffer. Given per-pixel normals $\mathcal{N}$, view directions ω_o, albedo $\mathcal{A}$, roughness $\mathcal{R}$, and metallic $\mathcal{M}$, we first compute the reflection direction $\omega_r = 2(\mathbf{n}\cdot\omega_o)\mathbf{n} - \omega_o$ and use it to query a prefiltered specular environment map. The diffuse irradiance is obtained by sampling a low-frequency cubemap using surface normals, while the specular radiance is retrieved from a roughness-dependent mip level of a high-resolution environment cubemap, following the split-sum approximation [KG13]. This design avoids explicit Monte Carlo integration and enables efficient shading during optimization.

To approximate the BRDF integral in Equation 20, we employ a precomputed 2D lookup texture (LUT) indexed by the dot product $N\cdot V$ and the roughness ρ. The LUT stores the integrated Fresnel and geometry response terms, allowing the specular reflectance to be expressed as

$$\mathbf{R}_s = \mathbf{F}_0 \cdot \mathrm{LUT}_x(N\cdot V, \rho) + \mathrm{LUT}_y(N\cdot V, \rho), \tag{21}$$

where $\mathbf{F}_0$ is the base reflectance at normal incidence. We set $\mathbf{F}_0 = 0.04$ for dielectric materials and interpolate it with the albedo for metallic surfaces as $\mathbf{F}_0 = (1-m)\cdot 0.04 + m\cdot\mathbf{a}$. This formulation mirrors standard real-time PBR pipelines and ensures stable gradients during backpropagation.

Finally, the diffuse and specular components are combined as

$$\tilde{\mathcal{I}} = (1-m)\,\mathbf{a}\cdot\mathcal{E}_d(\mathbf{n}) \;+\; \mathcal{E}_s(\omega_r,\rho)\cdot\mathbf{R}_s, \tag{22}$$

where $\mathcal{E}_d$ and $\mathcal{E}_s$ denote the diffuse and specular environment maps, respectively. Optional ambient occlusion and tone mapping can be applied as post-processing steps. This formulation yields a physically yet computationally efficient shading model that integrates seamlessly into our joint optimization framework and remains fully differentiable with respect to all BRDF and G-buffer parameters.

3. More experimental results

This supplementary section provides additional reconstruction results we conducted with our and other methods for comparison, in particular:

- Figure 1, 2, and 3 provide qualitative comparisons of reconstructed meshes for the DTU dataset. The figures compare 2DGS [HYC*24], PGSR [CLY*25], and two variants of our model.
- Figure 4 compares the reconstructed normals of our full model with three SoTA in surface reconstruction for the Shiny Blender Synthetic dataset [VHM*22].
- Figure 5 and 6 showcase the reconstructed meshes of the Barn and Truck scenes from the TnT dataset [KPZK17], respectively.

References

[BPH24] BAKER A. H., PINARD A., HAMMERLING D. M.: On a structural similarity index approach for floating-point data. *IEEE Transactions on Visualization and Computer Graphics 30*, 9 (Sept. 2024), 6261–6274. URL: https://doi.org/10.1109/TVCG.2023.3332843, doi:10.1109/TVCG.2023.3332843. 1

[CLY*25] CHEN D., LI H., YE W., WANG Y., XIE W., ZHAI S., WANG N., LIU H., BAO H., ZHANG G.: Pgsr: Planar-based gaussian splatting for efficient and high-fidelity surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics 31*, 9 (Sept. 2025), 6100–6111. URL: http://dx.doi.org/10.1109/TVCG.2024.3494046, doi:10.1109/tvcg.2024.3494046. 1, 3, 7

[CT82] COOK R. L., TORRANCE K. E.: A reflectance model for computer graphics. *ACM Trans. Graph. 1*, 1 (Jan. 1982), 7–24. URL: https://doi.org/10.1145/357290.357293, doi:10.1145/357290.357293. 2

[DXX*24] DAI P., XU J., XIE W., LIU X., WANG H., XU W.: High-quality surface reconstruction using gaussian surfels. In *ACM SIGGRAPH 2024 Conference Papers* (2024), Association for Computing Machinery. 1

[HSG24] HUANG Z., SHI Y., GONG M.: Visibility-aware pixelwise view selection for multi-view stereo matching. In *Pattern Recognition: 27th International Conference, ICPR 2024, Kolkata, India, December 1–5, 2024, Proceedings, Part XVIII* (Berlin, Heidelberg, 2024), Springer-Verlag, p. 130–144. URL: https://doi.org/10.1007/978-3-031-78456-9_9, doi:10.1007/978-3-031-78456-9_9. 1

[HYC*24] HUANG B., YU Z., CHEN A., GEIGER A., GAO S.: 2d gaussian splatting for geometrically accurate radiance fields. In *SIGGRAPH 2024 Conference Papers* (2024), Association for Computing Machinery. doi:10.1145/3641519.3657428. 1, 3, 7

[JTL*24] JIANG Y., TU J., LIU Y., GAO X., LONG X., WANG W., MA Y.: Gaussianshader: 3d gaussian splatting with shading functions for reflective surfaces. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2024), pp. 5322–5332. doi:10.1109/CVPR52733.2024.00509. 1

[Kaj86] KAJIYA J. T.: The rendering equation. *SIGGRAPH Comput. Graph. 20*, 4 (Aug. 1986), 143–150. URL: https://doi.org/10.1145/15886.15902, doi:10.1145/15886.15902. 2

[KG13] KARIS B., GAMES E.: Real shading in unreal engine 4. In *Proceedings of Physically Based Shading Theory Practice* (2013), vol. 4, p. 1. 2, 3

[KKLD23] KERBL B., KOPANAS G., LEIMKUEHLER T., DRETTAKIS G.: 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph. 42*, 4 (July 2023). URL: https://doi.org/10.1145/3592433, doi:10.1145/3592433. 1

[KPZK17] KNAPITSCH A., PARK J., ZHOU Q.-Y., KOLTUN V.: Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics 36*, 4 (2017). 3

[LZF*24] LIANG Z., ZHANG Q., FENG Y., SHAN Y., JIA K.: GS-IR: 3D Gaussian Splatting for Inverse Rendering . In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Los Alamitos, CA, USA, June 2024), IEEE Computer Society, pp. 21644–21653. URL: https://doi.ieeecomputersociety.org/10.1109/CVPR52733.2024.02045, doi:10.1109/CVPR52733.2024.02045. 1

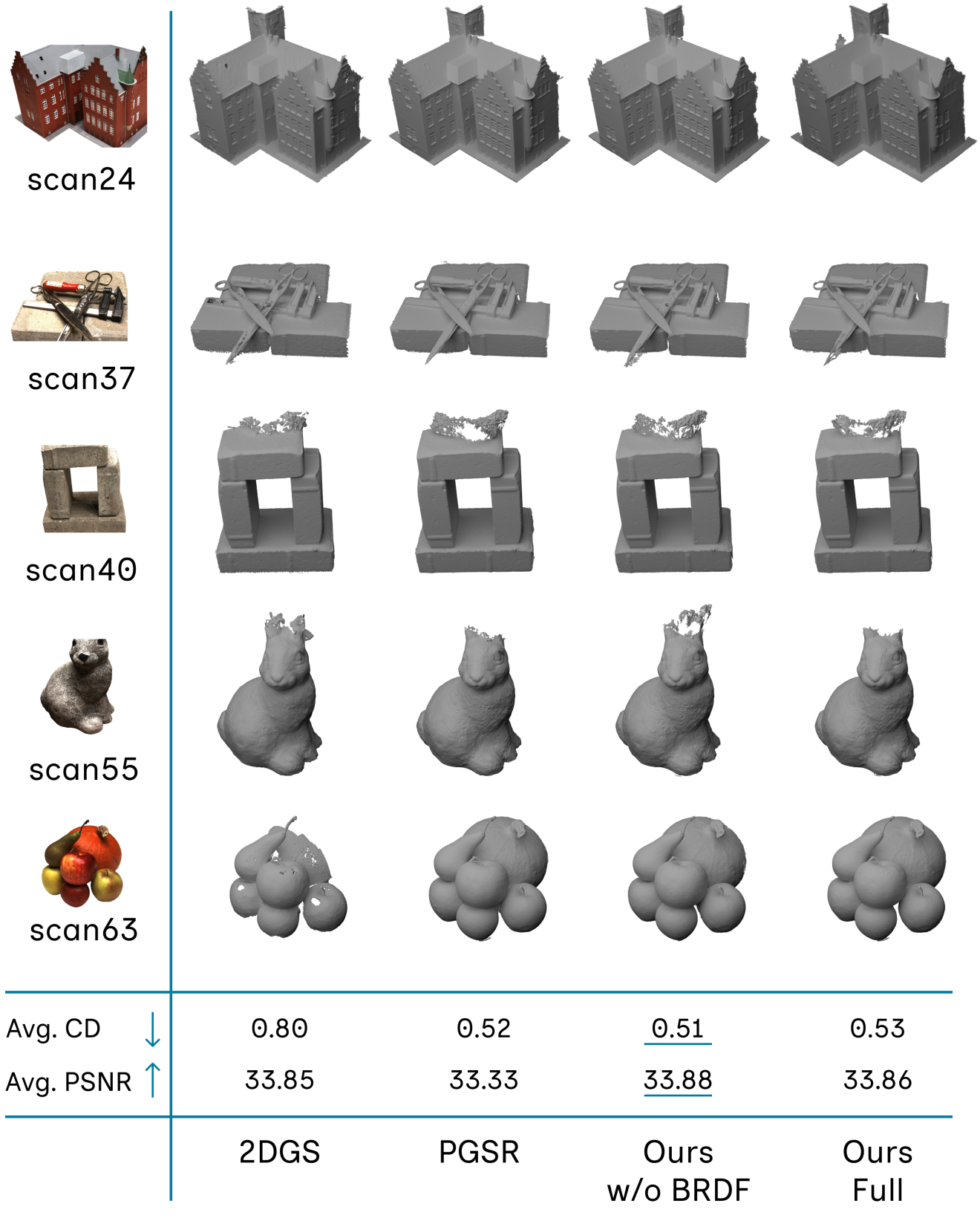

Figure 1: *Qualitative comparison of reconstructed meshes from the DTU dataset - Part 1*

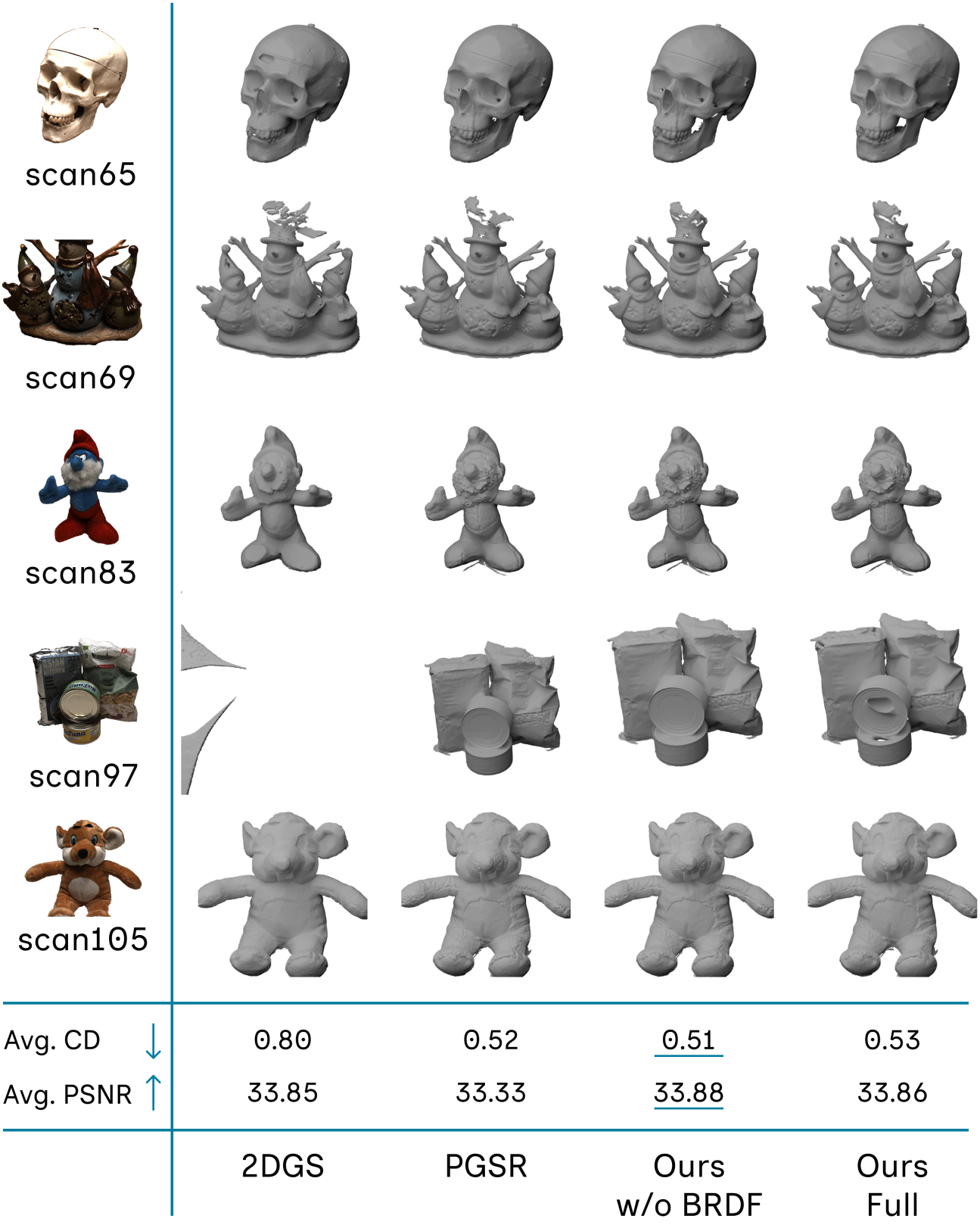

Figure 2: *Qualitative comparison of reconstructed meshes from the DTU dataset - Part 2*

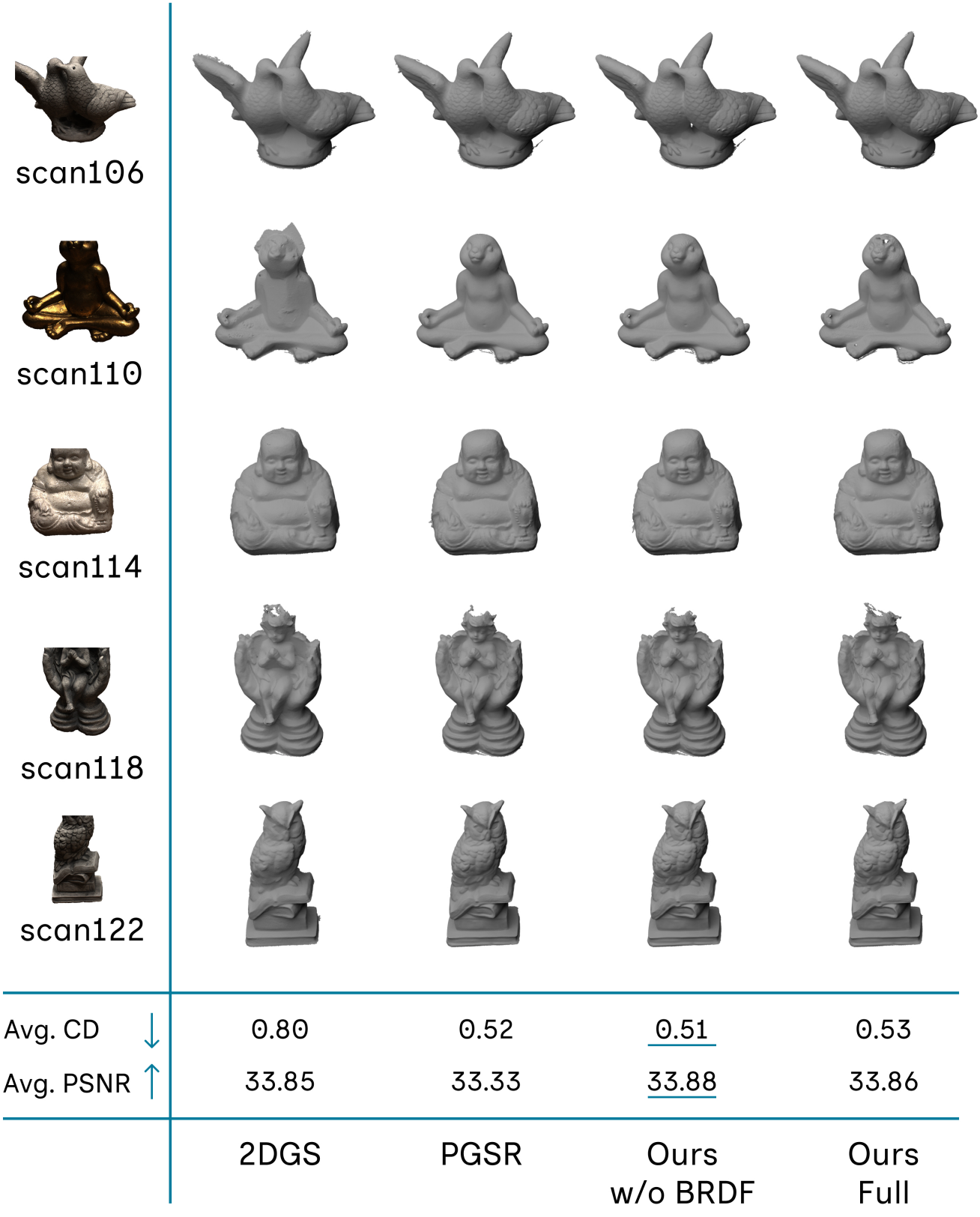

Figure 3: *Qualitative comparison of reconstructed meshes from the DTU dataset - Part 3*

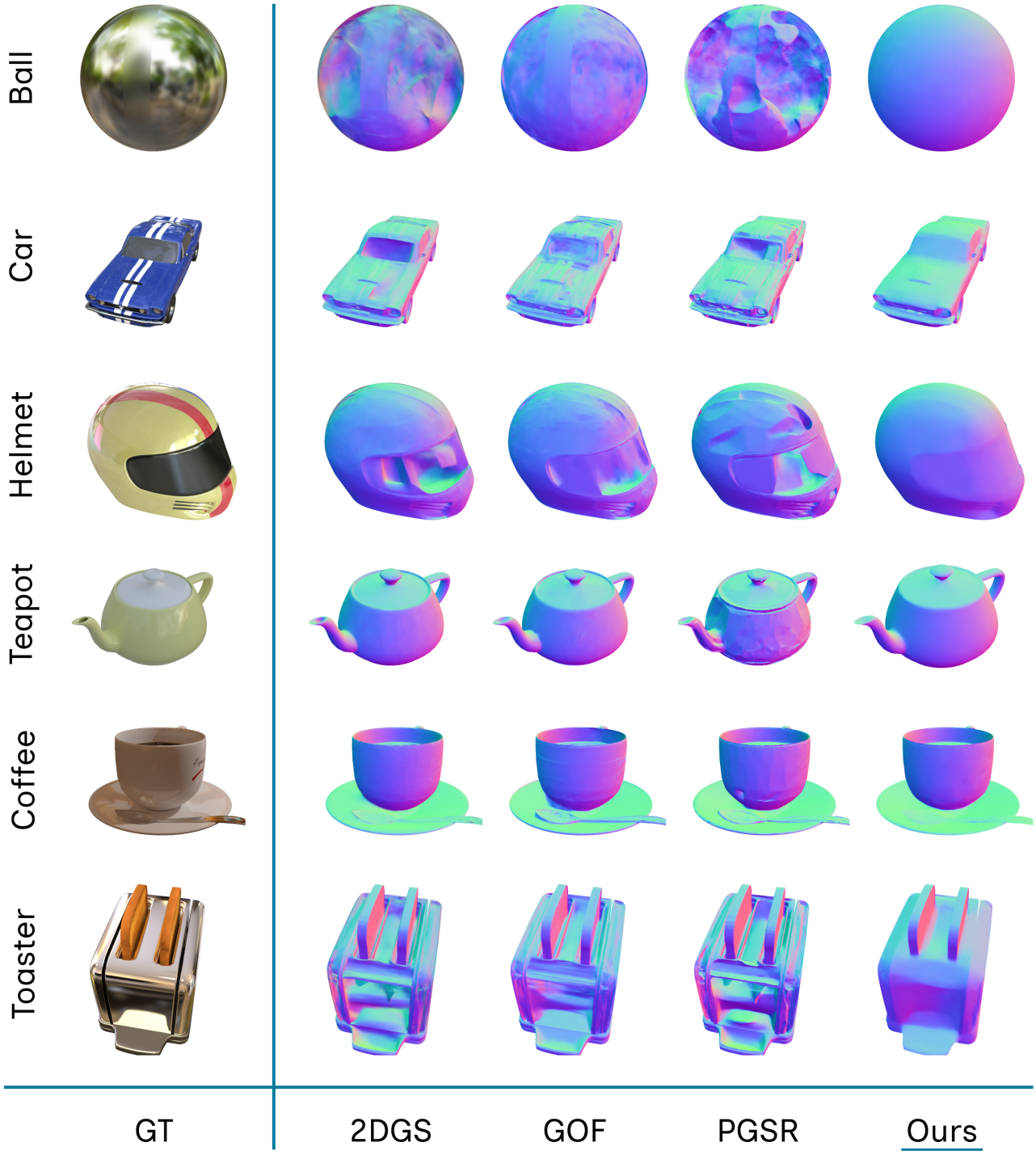

Figure 4: *We compare the reconstructed normals of our full model with three SoTA in surface reconstruction: 2DGS [HYC*24], GOF [YSG24], and PGSR [CLY*25]. SoTA methods suffer from the ambiguity caused by reflection, resulting in meshes with distorted or non-watertight geometries. Our method, on the other hand, produces more accurate reconstructed surfaces thanks to the cooperation of* $\mathcal{L}_{\text{ro}}$, $\mathcal{L}_{\text{tv}}$, *and* $\mathcal{L}_{\text{dn}}$ *during the joint optimization process.*

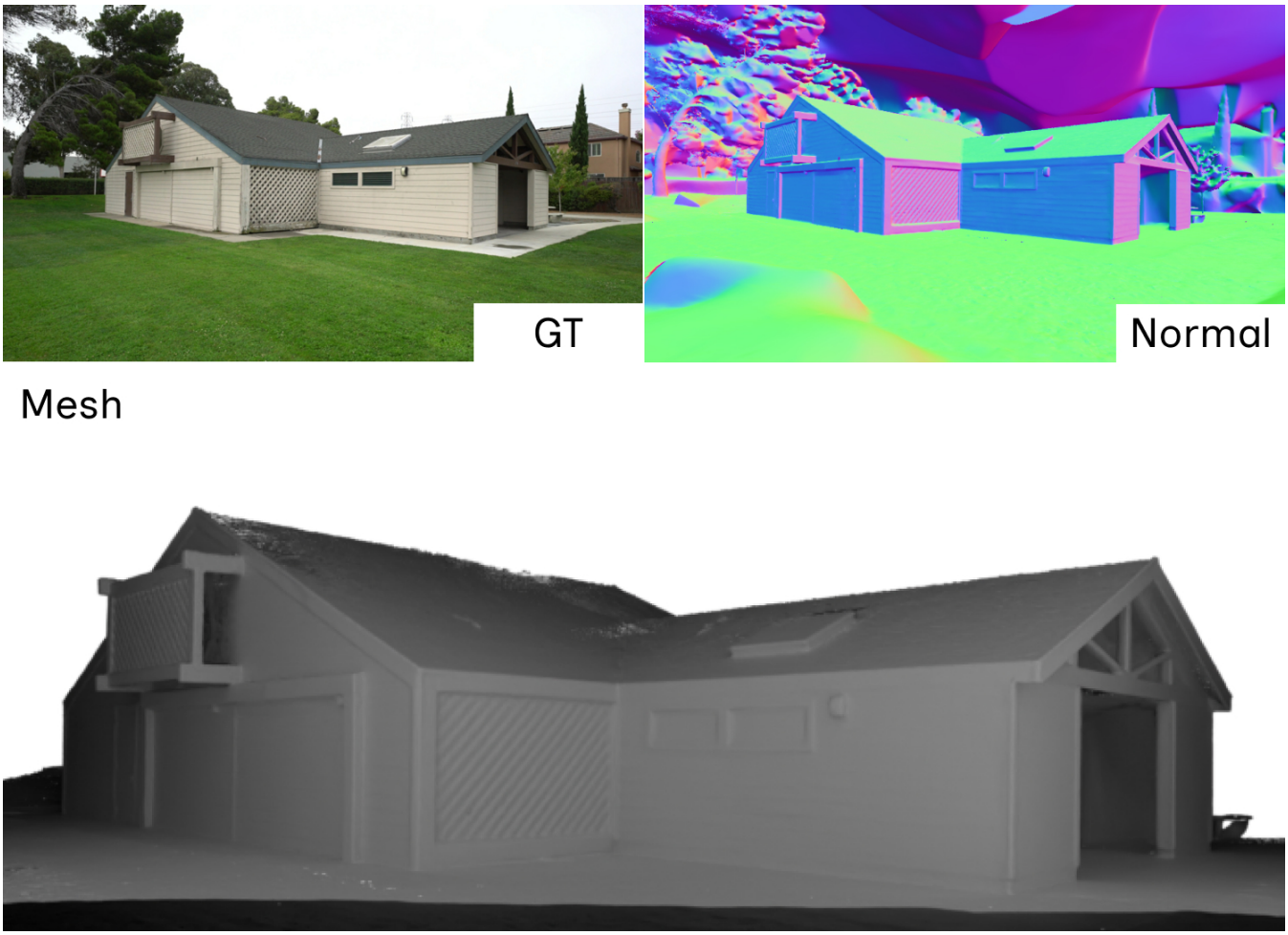

Figure 5: *Reconstruction of the Barn scene from the TnT dataset*

[Sch94] SCHLICK C.: An Inexpensive BRDF Model for Physically-based Rendering. *Computer Graphics Forum* (1994). doi:10.1111/1467-8659.1330233. 2

[VHM*22] VERBIN D., HEDMAN P., MILDENHALL B., ZICKLER T., BARRON J. T., SRINIVASAN P. P.: Ref-nerf: Structured view-dependent appearance for neural radiance fields. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022), pp. 5481–5490. doi:10.1109/CVPR52688.2022.00541. 3

[WLW*24] WANG J., LIU Y., WANG P., LIN C., HOU J., LI X., KOMURA T., WANG W.: Gaussurf: Geometry-guided 3d gaussian splatting for surface reconstruction. *arXiv preprint arXiv:2411.19454* (2024). 1

[WMLT07] WALTER B., MARSCHNER S. R., LI H., TORRANCE K. E.: Microfacet models for refraction through rough surfaces. In *Proceedings of the 18th Eurographics Conference on Rendering Techniques* (Goslar, DEU, 2007), EGSR'07, Eurographics Association, p. 195–206. 2

[YH09] YOO J.-C., HAN T. H.: Fast normalized cross-correlation. *Circuits Syst. Signal Process. 28*, 6 (Dec. 2009), 819–843. URL: https://doi.org/10.1007/s00034-009-9130-7, doi:10.1007/s00034-009-9130-7. 1

[YSG24] YU Z., SATTLER T., GEIGER A.: Gaussian opacity fields: Efficient adaptive surface reconstruction in unbounded scenes. *ACM Transactions on Graphics* (2024). 1, 7

[ZWY25] ZHU Z.-L., WANG B., YANG J.: Gs-ror^2: Bidirectional-guided 3dgs and sdf for reflective object relighting and reconstruction, 2025. URL: https://arxiv.org/abs/2406.18544, arXiv:2406.18544. 1

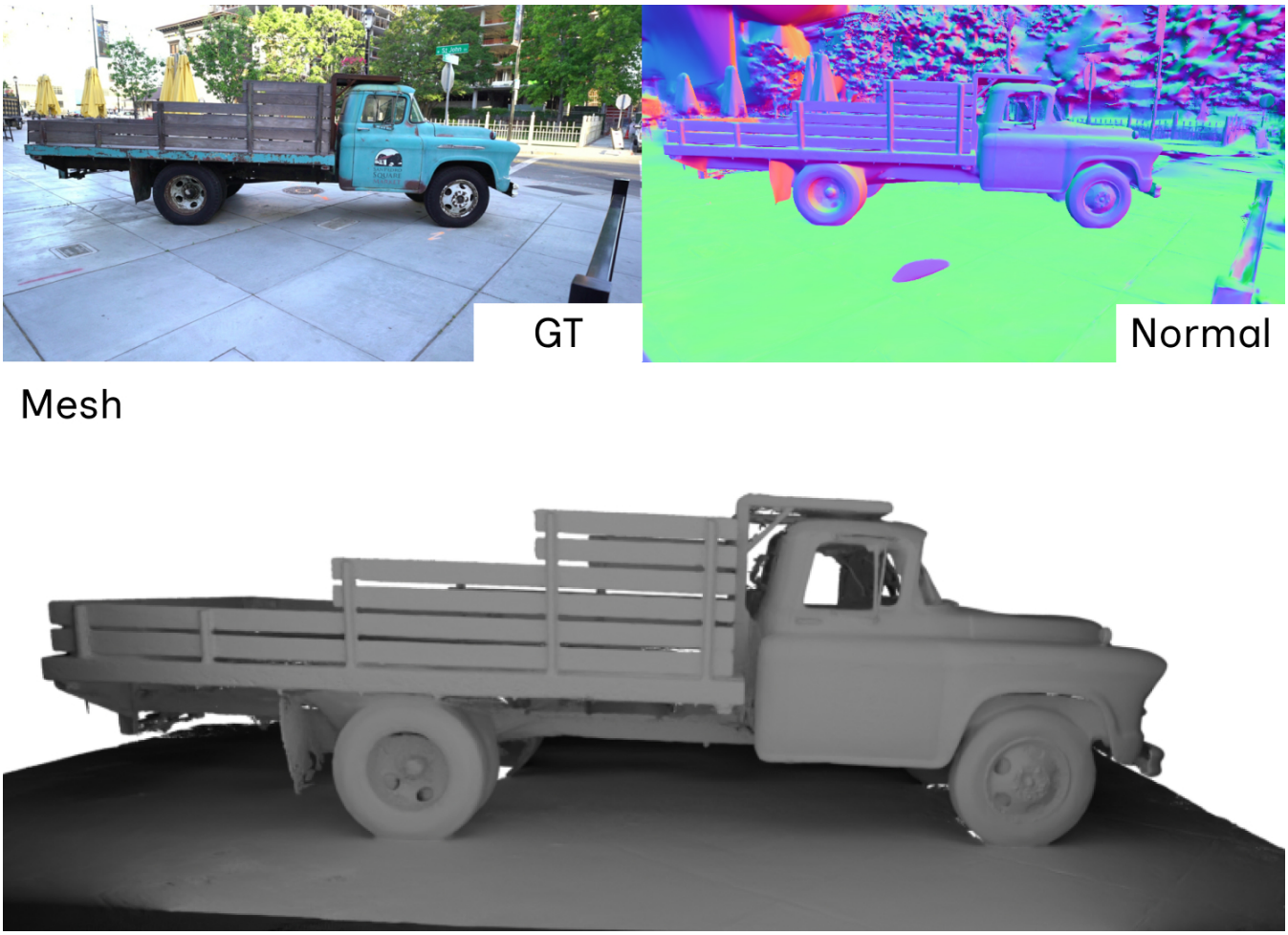

Figure 6: *Reconstruction of the Truck scene from the TnT dataset*