ProbeSDF: Light Field Probes For Neural Surface Reconstruction

Briac Toussaint, Diego Thomas, Jean-Sébastien Franco

ProbeSDF: Light Field Probes For Neural Surface Reconstruction

Briac Toussaint
Univ. Grenoble Alpes, CNRS, Inria, Grenoble INP, LJK, France
briac.toussaint@inria.fr

Diego Thomas
Kyushu University, Japan
thomas@ait.kyushu-u.ac.jp

Jean-Sébastien Franco
Univ. Grenoble Alpes, CNRS, Inria, Grenoble INP, LJK, France
jean-sebastien.franco@inria.fr

Abstract

SDF-based differential rendering frameworks have achieved state-of-the-art multiview 3D shape reconstruction. In this work, we re-examine this family of approaches by minimally reformulating its core appearance model in a way that simultaneously yields faster computation and increased performance. To this goal, we exhibit a physically-inspired minimal radiance parametrization decoupling angular and spatial contributions, by encoding them with a small number of features stored in two respective volumetric grids of different resolutions. Requiring as little as four parameters per voxel, and a tiny MLP call inside a single fully fused kernel, our approach allows to enhance performance with both surface and image (PSNR) metrics, while providing a significant training speedup and real-time rendering. We show this performance to be consistently achieved on real data over two widely different and popular application fields, generic object and human subject shape reconstruction, using four representative and challenging datasets.

1. Introduction

Neural radiance fields have established a new milestone for the task of novel view synthesis and have led to a plethora of variants of broad applicability. We here take particular interest in SDF surface-based radiance fields, which have proven to be a highly performing variant for the tasks of 3D modeling from images [27, 30], with wide reaching applications ranging from object or performance capture, to immersive and 3D content production.

At the heart of such methods usually lies a parametrization of the light field, in the form of a 5-dimensional function. This function maps the spatial coordinates of a point in space and the observed direction to a color and a signed distance to the reconstructed surface. On one hand, the mapping can be global and requires a complex scene-wide MLP as implicit 5D decoder, as is the case for most seminal approaches [16, 27]. On the other hand, recent popular variants that target higher speed and memory efficiency use explicit representations. Global representations are based on a grid encoding [3, 28, 30] where interpolated spatial and angular features are run through a global shallow MLP. Local representations use the sparse 3D Gaussian Splatting [11] sample-based encoding approach, where each separate Gaussian comes with its own opacity and adjoining set of angular color features. While grids or local approaches use a single deep MLP to densely encode the complete light field function, they almost always colocate storage and decoding of color and angular features.

But is this really a necessary or even desirable feature of any light field encoding? We question these assertions by postulating that, contrary to the spatial components of the light field that mainly encode intrinsic surface texture, the angular components encode phenomena that are predominantly extrinsic to the surface, i.e. lighting and environment, and as such can be encoded at a lower spatial resolution as they are locally almost invariant to parallax. This hypothesis is in fact exploited by the rendering community, where angular and environment components contributing to the radiance equation used to compute rendered colors, are typically precomputed at sparse 3D locations coined *light field probes* [21], and interpolated in-between.

Leveraging this intuition, our approach borrows the spherical harmonic parametrization popular with explicit methods *e.g.* [3, 11] to represent angular components, but instead decorrelates high density spatial variation from lower density angular variation using a mid-resolution probe grid of angular features. Each of the components encode abstract features that are fed to a tiny MLP that acts as a minimal BRDF decoder of spatial and angular varying components, while we explicitly store SDF values. This scheme advantageously replaces the core appearance and decoding model of SDF approaches [13, 27, 28, 30] to pre-

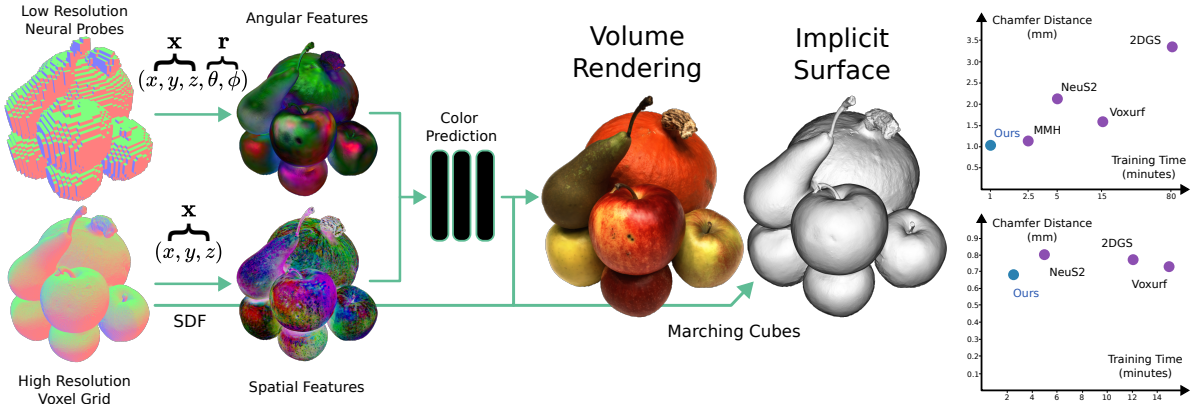

Figure 1. We design a new appearance model for neural surface approaches, which combines spatial and angular features for improved reconstruction quality, training and inference speed. We plot the chamfer distance as a function of training speed for several baselines on MVMannequins (top) and DTU (bottom).

dict ray colors. We are careful in our model to also include power terms of the cosine angle between the view direction and the estimated surface normal, allowing our lightweight model to effectively account for increased surface reflectivity at grazing angles. This yields an unprecedented performance combination, superseding both 3D and image metric performances with four popular benchmarks for general object reconstruction [7, 31] and human reconstuction [6, 24] while simultaneously achieving a significant training and rendering performance boost. In summary, our contributions are:

- An explicit embedding of the angular dependency that is both local, smooth and efficient.
- A simpler color prediction MLP that is agnostic to the surface position and orientation.

2. Related Works

In this work we leverage disentangled spatial and angular parameterization of radiance towards high quality appearance and surface modeling. Here we review closely related works on surface-based 3D scene representations and existing parametrizations of color.

Neural Surface Reconstruction. Through the use of implicit surface representations, neural implicit fields have demonstrated increasing capabilities to generate high-quality 3D models [20, 32, 33]. In NeuS [27], Wang et al. proposed to parametrize the surface with a signed distance function that is modeled with a multi layer perceptron (MLP). The approach opened new frontiers in the generation of accurate 3D scenes from images but at the cost of prohibitive training times (several hours). Follow-up works have thus focused on boosting both accuracy and computation time by employing explicit grids of features and smaller MLPs. Notably, Wu et al. proposed in Voxurf [30] to use a dense grid of voxels, and Neus2 was also proposed that uses hash grids [28]. Recently, Millimetric humans [24] have demonstrated remarkably fast and accurate reconstructions in the case of human datasets by using sparse and dynamic grids. However, all the methods discussed above rely on a co-located encoding of position and view direction to parametrize the appearance. As a consequence these approaches either fail to model local radiance effects like interreflections or require larger MLPs and longer training.

Angular parametrization Deep MLPs have become ubiquitous to model view dependent colors, taking as input the view direction [16], the view and normal [27, 28, 30, 32] or the reflected direction [24, 25]. Non-neural approaches such as plenoxels [3] or the Gaussian splatting methods [5, 11] favor a spherical harmonics decomposition. The main advantage compared to a deep MLP is the computational efficiency, at the cost of a large number of appearance parameters: 27 per voxel for [3] and 48 per Gaussian for [11]. Note that some neural architectures [24, 25, 28] use spherical harmonics as a positional encoding of the directional vectors, but this is completely distinct from encoding the signal into spherical harmonic coefficients. In this work we propose a lighter yet efficient parametrization of color by using two decorrelated voxel structures to encode different components of the observed color.

In the field of real-time rendering, where lighting and BRDF are known in advance, the idea of caching the incoming light at discrete locations in space, known as light field probes, has become a standard [14, 15, 21]. We draw inspiration from this technique for our proposed approach to multiview 3D reconstruction, with the difference in our case that the probe parameters are optimized along with the voxel parameters.

High Frequency Reflections. A number of approaches specifically deal with high frequency reflections, which requires dedicated representations and algorithms. Guo et al.

[4] proposed to model a 3D scene with an additional nerf that predicts the reflected light at each point in space. The reflection image is obtained by marching into the second NeRF and then composited with the main image. However, this approach is underconstrained and thus requires strong priors such as the reflective surfaces being flat.

Verbin et al. [25] proposed to learn a per-point roughness value that directly modulates the amplitude of the high frequencies of the positional encoding of the input vector. This allows to reuse the information encoded in the environment for the whole surface despite the spatially varying roughness. This approach enables editing the roughness and the base color of the object but suffers from a very long training time since the environment lighting is still encoded into a large MLP.

In contrast, recent works [26, 29] recover highly detailed reflections using cone tracing, which inherently solves the problem posed by self-reflections. During cone tracing, Verbin et al. [26] proposed to sample a multi-resolution hash grid and down-weighs the higher spatial frequencies based on the roughness while Wu et al. [29] proposed to sample a mip-mapped grid of features. Both approaches give impressive results even when confronted with curved, mirror-like objects but at a large training cost exceeding 10 GPU hours per scene. In contrast, we target objects with mild specularities which alleviates the need to rely on cone tracing and show that both fast training and real-time rendering can be achieved. More specifically, our neural light field probes can learn information about the surroundings whereas cone tracing gathers it at a greater cost. This assumption proves to be valid for standard acquisition scenes, with the four datasets tested.

Minimal Parametrization. Explicitly disentangling the material parameters from the lighting is particularly challenging but also rewarding because it can yield a minimal parametrization of the appearance. Munkberg et al. [19] show that the approximate rendering equation developed in [10] is sufficiently accurate to recover the geometry using a differentiable rasterizer, along with an approximate material decomposition. Jiang et al. [8] applies a similar strategy but with the 3D gaussian splatting volume rendering. The NeRFactor pipeline [34] starts from a pre-trained NeRF that is distilled into lighting, albedo and neural BRDF parameters. Jin et al. proposed TensoIR [9] that achieves a more accurate decomposition by evaluating the rendering equation with importance sampling. An efficient tensor representation helps to accelerate the rendering of the secondary rays but the complete training still takes five hours on a single GPU, with decent results available after half an hour of training. Our proposed method converges in just a few minutes and we find that only 4 coefficients per voxel are sufficient in most cases. The dimensionality is equivalent to the albedo and roughness values of a physically based material parametrization, but without the cost associated with the inversion of the rendering equation.

3. Method

We use an SDF grid as the implicit representation of the 3D shape. SDF values in the grid are optimized via our newly proposed differentiable appearance model and the derived surface normal vectors $\mathbf{n}$.

3.1. Decoupled Parametrization of Radiance

Volumetric neural rendering as proposed in the seminal work NeRF computes the color at a given pixel by integrating radiances along camera-pixel rays weighted by local opacity values. Follow-up papers condition the opacity on the distance to the surface to improve the reconstruction quality [27, 33]. The radiance C represents the amount of energy that is emitted by a point in space. In physics it is expressed as the integral over the hemisphere Ω of the spatially varying bidirectional reflectance distribution function (SVBRDF) f_r multiplied by the incoming radiance L and by the cosine between the incoming direction ω and the surface normal $\mathbf{n}$.

$$C = \int_{\Omega} f_r(\mathbf{v}, \omega, \mathbf{x}) L(\omega, \mathbf{x})(\omega \cdot \mathbf{n}) d\omega. \tag{1}$$

The (spatially varying) BRDF depends on the point's position $\mathbf{x}$, direction of incoming light ω and viewing direction $\mathbf{v}$. $L(\omega, \mathbf{x})$ is the intensity of light at point $\mathbf{x}$ that comes from the direction ω and $\mathbf{n}$ is the normal vector of the surface at position $\mathbf{x}$. In equation 1 both f_r and L are unknown and disentangling these two functions during optimization is extremely difficult. As a consequence the main stream of research employs an MLP to directly approximate the result of this integral and obtain the outgoing radiance with the following equation.

$$C = MLP(\mathbf{F}_s(\mathbf{x}), \mathbf{v}, \mathbf{n}, \mathbf{r}), \tag{2}$$

where $\mathbf{F}_s(\mathbf{x})$ represents spatial features encoded at a point position $\mathbf{x}$ and $\mathbf{r}$ is the reflected vector at point $\mathbf{x}$ for the viewing direction $\mathbf{v}$. This approximation works well if the incoming light is positionally invariant, but its accuracy degrades in the presence of local lighting effects such as close point lights, interreflections and self shadows. In such cases, the spatial features must additionally encode some information about the local lighting and thus require a larger dimensionality than one would expect. A large and deep MLP is also necessary to decode this compressed information, which lowers the computational efficiency.

We propose to decouple the radiance model into spatial and angular features. We model a 3D scene with a combination of a high resolution sparse voxel grid that models the SDF field and a coarser grid that models the angular features

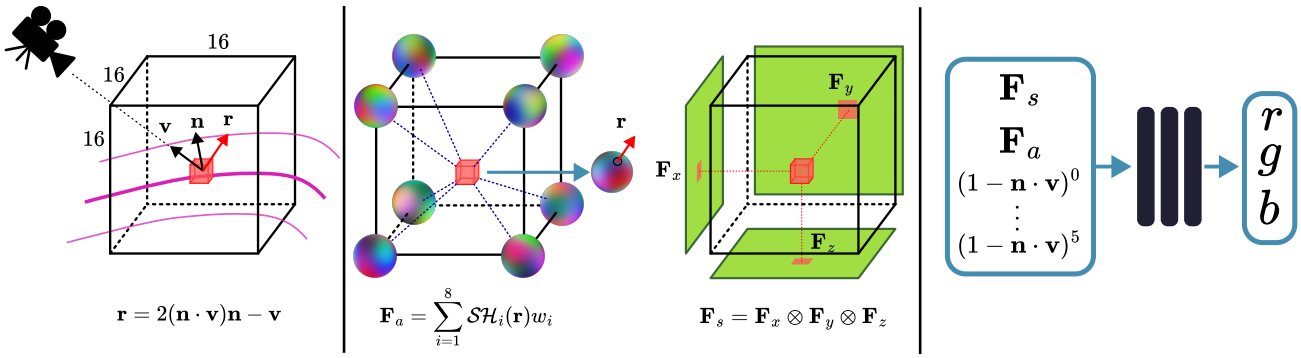

Figure 2. Overview of the color prediction for a single voxel inside a 16^3 tile. The angular features F_a are computed by interpolating and evaluating the spherical harmonics from the 8 nearest probes with the reflected vector $\mathbf{r}$. The spatial features F_s are computed as the outer product of three orthogonal planes. A small neural network decodes these inputs into a color.

as shown in fig. 2. For each point in space, the spatial features are obtained with a planar factorization in the coarse resolution tile with $\mathbf{F}_s = \mathbf{F}_x \otimes \mathbf{F}_y \otimes \mathbf{F}_z$ for dimensionality reduction [2]. The angular features are obtained by trilinearly interpolating light field probes as explained in sec. 3.1. In addition, a dependency on the angle of incidence must be added to account for differences in reflectivity, as shown in fig. 3.

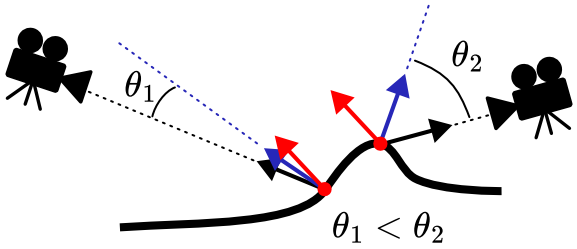

Figure 3. The reflectivity cannot be encoded in $\mathbf{F}_a$ because different values of θ can map to the same reflected vector (in red). In that case, similar angular features will be obtained for the two viewpoints due to the spatial and angular proximity of the lookups. The angle of incidence disambiguates the two situations.

The reflectivity, also called the Fresnel term, is commonly approximated as follows for rendering purposes [22]:

$$R(\theta) = R_0 + (1 - R_0)(1 - \cos\theta)^5. \tag{3}$$

R_0 is the reflectivity at normal incidence and θ is the angle between the view direction and the half-way vector to the light source. In our case, R_0 is unknown and there is no explicit light source which means that we cannot use this formula directly. Instead, we approximate $\cos\theta$ by $\mathbf{n}\cdot\mathbf{v}$ and let the neural network learn its own polynomial approximation of the Fresnel term. Our revised parametrization is given by eq. 4:

$$C = MLP(\mathbf{F}_s(\mathbf{x}), \mathbf{F}_a(\mathbf{x}, \mathbf{r}), (1 - \mathbf{n}\cdot\mathbf{v})^0, \ldots, (1 - \mathbf{n}\cdot\mathbf{v})^5). \tag{4}$$

We denote the dimensionality of the features by n_s and n_a such that $\mathbf{F}_s \in \mathbb{R}^{n_s}$ and $\mathbf{F}_a \in \mathbb{R}^{n_a}$. Most materials reflect the light in a cone around the reflected vector which is why we condition $\mathbf{F}_a$ on $\mathbf{r}$ but some materials have retro-reflective properties that break this assumption. In that case, conditioning on the view direction would be more appropriate. Lastly, eq. 4 assumes isotropic materials since the reflectivity depends only on $\mathbf{n}\cdot\mathbf{v}$. Anisotropic materials could be handled by adding a dependency on the azimuthal angle.

3.2. Light Field Probes on a Coarse Grid

In this work, we replace the generic angular inputs $\mathbf{v}$, $\mathbf{n}$, $\mathbf{r}$ with new angular inputs encoding the local illumination (the light field probes), with the goal to achieve a more efficient parametrization of the appearance in terms of parameter count, training and inference speed. Similarly to previous works [3, 11], we use a spherical harmonics decomposition to model this dependency, but with several important differences: first, we encode abstract features on the sphere rather than the outgoing radiance which lets a neural network handle the non-linearities. Second, our directional embedding has a low spatial resolution which takes advantage of the smooth spatial variations of the lighting. Third, we sample this embedding with the reflected vector instead of the view vector to link the local surface orientation with the shading function, which has been showed to be beneficial [25] for improved surface quality. The angular features encoded into a single neural probe are given by eq. 5:

$$\mathbf{F}_a = \mathcal{SH}(\mathbf{r}) = \sum_{j=1}^{l^2} \mathbf{b}_j Y_j(\mathbf{r}). \tag{5}$$

The polynomials of the SH basis are denoted with Y_j. There are l^2 coefficients[1] $\mathbf{b}_j \in \mathbb{R}^{n_a}$. Increasing values of l unlock signals of increasing angular frequency which gives an explicit control on the representable amount of specularities. We observe two guiding principles: (1) The shinier the material, the higher the frequency of $\mathbf{F}_a$ should be in the angular domain. (2) The closer the light source, the higher the frequency $\mathbf{F}_a$ should be in the spatial domain. See fig. 4 for an illustration.

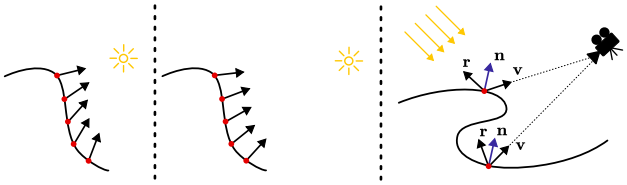

Figure 4. Left and center: The lighting direction changes at a lower rate on the surface for distant light sources. Right: Self shadows impose variations in lighting even with directional illumination.

For the case of common scenes such as humans or everyday objects, assuming materials with high to medium roughness so that $\mathbf{F}_a$ does not need to be of very high angular frequency is generally sufficient. We also assume that the lights are far enough, so that probes at a low spatial frequency can explain the parallax. The probes are stored at 1/16th the resolution of the main voxel grid, as shown in fig. 2. The voxels tri-linearly interpolate the 8 nearest probes according to eq. 6 and 7, with w_i the interpolation weight of the i-th corner.

$$\mathbf{F}_a = \sum_{i=1}^{8} \mathcal{SH}_i(\mathbf{r}) w_i = \sum_{i=1}^{8} \sum_{j=1}^{l^2} \mathbf{b}_{ij} Y_j(\mathbf{r}) w_i \tag{6}$$

$$= \sum_{j=1}^{l^2} \hat{\mathbf{b}}_j Y_j(\mathbf{r}) \quad \text{with } \hat{\mathbf{b}}_j = \sum_{i=1}^{8} \mathbf{b}_{ij} w_i \tag{7}$$

By linearity, the interpolation and the evaluation of the SH basis can be swapped so that only one SH basis needs to be evaluated instead of 8. The partial derivatives needed for gradient descent are given in eq. 8 and 9. The partial derivatives with respect to $\mathbf{r}$ as well as $(1 - \mathbf{n} \cdot \mathbf{v})^p$ in eq. 4 are propagated up to the SDF through $\mathbf{n}$.

$$\frac{\partial \mathbf{F}_a}{\partial \mathbf{b}_{ij}} = diag(w_i Y_j(\mathbf{r})) \tag{8}$$

$$\frac{\partial \mathbf{F}_a}{\partial \mathbf{r}} = \sum_{j=1}^{l^2} \hat{\mathbf{b}}_j \nabla Y_j(\mathbf{r}). \tag{9}$$

[1]We use a one-indexed notation for the SH order l instead of the traditional zero-indexing so that we have l^2 coefficients.

3.3. Neural SDF Estimation Approach

We here discuss how to embed the new proposed color decoder in an implicit surface reconstruction pipeline. During the forward pass, for a pixel (u, v) in a given viewpoint, the color is computed by a transmittance T_i-weighted average of the colors C_i of N points sampled from the current scene volume estimate along the pixel aligned ray, where T_i is the product of obstructing opacities α_j. We use the NeuS equation to relate volumetric opacities to the SDF s_i at each sample point [27]:

$$c(u,v) = \sum_{i=1}^{N} T_i \alpha_i C_i, \qquad T_i = \prod_{j<i} (1 - \alpha_j),$$
$$\alpha_i = max(\frac{\Phi_\tau(s_i) - \Phi_\tau(s_{i+1})}{\Phi_\tau(s_i)}, 0),$$
$$\Phi_\tau(s_i) = \frac{1}{1 + \exp^{-\tau s_i}}. \tag{10}$$

where τ is a scale factor that increases during optimization. The SDF field can then be optimized by backpropagating updates from the observed color loss through the rendering equation, and from the set of regularization losses.

Our losses and regularizations are detailed in equations 11 to 16. The raw SDF $\hat{s}$ is convolved by a 5^3 gaussian kernel G into a smoother SDF s to stabilize training [30]. Note that s is used for all subsequent operations such as normal computation and rendering. Eq. 11 maintains s and $\hat{s}$ close to each other, which also promotes smoothness. The Eikonal regularization 12 ensures that s is a signed distance. Equations 13, 14, 15 promote spatial smoothness for the normals, for the factorized spatial features and for the probes coefficients respectively. In eq. 15, $\mathcal{V}_i$ is the set of neighboring probes adjacent to the i-th one and N is the total number of probes. The regularizations are applied on each voxel, factorized feature, probe or image pixel when appropriate.

$$\mathcal{L}_{\text{sdf}} = \sum_{\text{voxel}} |s - \hat{s}|^2, \quad s = G(\hat{s}) \tag{11}$$

$$\mathcal{L}_{\text{Eik}} = \sum_{\text{voxel}} (||\nabla s|| - 1)^2 \tag{12}$$

$$\mathcal{L}_{\text{normal}} = \sum_{\text{voxel}} ||\nabla \mathbf{n}||^2, \quad \mathbf{n} = \nabla s / ||\nabla s|| \tag{13}$$

$$\mathcal{L}_{\text{features}} = \sum_{\text{texel}} ||\nabla \mathbf{F}_x||^2 + ||\nabla \mathbf{F}_y||^2 + ||\nabla \mathbf{F}_z||^2 \tag{14}$$

$$\mathcal{L}_{\text{probes}} = \sum_{i=1}^{N} \sum_{j=1}^{l^2} \sum_{k \in \mathcal{V}_i} ||\mathbf{b}_{ij} - \mathbf{b}_{kj}||^2 \tag{15}$$

$$\mathcal{L}_{\text{photo}} = \sum_{u,v} ||c(u,v) - c_{\text{gt}}(u,v)||^2 \tag{16}$$

In eq. 16, c is the rendered image and c_{gt} is the ground truth image. We apply a weight of $(\max(c, c_{\text{gt}}) + \epsilon)^{-1}$ on the photometric gradient of each pixel in eq. 16 to penalize relative differences rather than absolute differences [17]. We apply an empirical factor of $(1 + |s| * 5)^{-1}$ on the gradients of the per-voxel regularizations to lower their importance away from the zero-crossing. The idea is to keep the regularizations and the data term balanced both near and far the surface.

3.4. Training

Our complete loss is a weighted sum of eq. 11 to 16 that we minimize with the Adam optimizer [12]. In summary, the trained parameters are the raw SDF values $\hat{s}$, the factorized features $\mathbf{F}_x, \mathbf{F}_y, \mathbf{F}_z$, the probes coefficients $\mathbf{b}$ and the MLP weights. We follow a coarse-to-fine learning strategy, both in the spatial and angular domains. We start with coarse voxels and $l = 2$, under the supervision of low resolution images. We iteratively subdivide the voxels and switch to higher resolution images, and introduce higher angular frequencies by increasing l. Starting the optimization with too many degrees of freedom for the appearance parametrization can lead to local minima that are not easy to overcome, as noted by [11].

4. Implementation Details

We use a 2 hidden layer MLP with 32 neurons and ReLU activations with a sigmoid for the last layer. The probes at the shared corners of adjacent tiles are duplicated in memory (but keep shared values) to benefit from hardware-accelerated interpolation. We support up to 16 spherical harmonic coefficients ($l = 4$) per probe. The computation of $\mathbf{F}_a$, $\mathbf{F}_s$ and the MLP call are joined in a single CUDA kernel for efficiency, in the style of [18] and [24]. Each 16^3 tile contains $3 \times 16 \times 16 \times n_s$ surface coefficients and $8 \times l^2 \times n_a$ spherical harmonic features. Assuming $l = 4$ and $n_s = n_a$, the probes only require $(8 \times 16)/(3 \times 16 \times 16) = 1/6$th of the memory taken by the spatial features. The sparse voxel grid is based on the code of [24], with targeted modifications to implement our probes.

5. Experiments

We evaluate our method on 4 datasets : MVMannequins [24], ActorsHQ [6], DTU [7] and BlendedMVS [31]. Unless stated otherwise, we train and evaluate the baselines using all available images at full resolution on a workstation equipped with an RTX A6000 GPU.

Notation The dimensionality of the spatial and angular features and the maximum spherical harmonic order can be summarized as a configuration triplet (n_s, n_a, l). Some datasets may contain inconsistent shadows or large exposure changes (DTU and BMVS) that are hard to model by our representation so we optionally train per-camera bias vectors in our MLP. Configurations using the per-camera bias vectors are annotated with a ✓-symbol and an ✗-symbol is used otherwise.

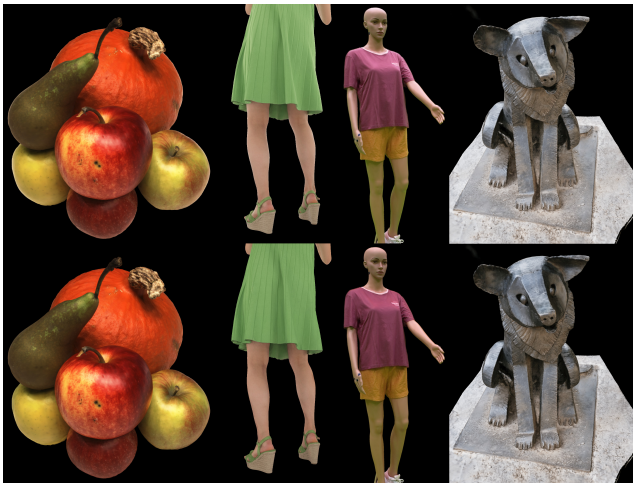

Figure 5. Top: Default rendering. Bottom: We disable the Fresnel factor, as if the camera is looking at normal incidence (set $\mathbf{n} \cdot \mathbf{v} = 1$) but we still compute $\mathbf{F}_a$ with the reflected vector. Notice the lack of reflectivity at grazing viewing angles, especially visible on the apples and on the green dress.

MVMannequins is a dataset of 14 dressed mannequins for multi-view reconstruction benchmarking with 68 cameras at 2048^2 resolution. We use the official evaluation code and report the metrics in table 2. We outperform all the baselines while taking only about a minute of training time, almost 3x faster than [24], and 5GB of VRAM. We achieve a rendering speed of 300Hz to 400Hz and a model size of 30MB with 2mm voxels. A visual ablation of the components of our parametrization is presented in fig. 6 on the *kino/jea* mannequin, clearly showing the role of each parameter group.

ActorsHQ is a multiview dataset of humans in motion with 160 cameras at a high resolution of 4088×2990. We evaluate on the 1000th temporal frame of 13 sequences using full resolution images. We compare against Voxurf [30] and NeuS2 [28], that we trained with half resolution images since we ran into issues when trying the full resolution images. We report the PSNR in table 3. Voxurf takes about one hour to converge (we tripled the number of training iterations compared to the DTU configuration). NeuS2 converges fast but with a lower PSNR. In comparison, our method only takes $\sim$4 minutes to converge on the full resolution images, and we also outperform both when training at the same resolution. We maintain a rendering speed above 100Hz despite each voxel having a side length of 0.6mm at the highest resolution, with a model size of 230MB. This

Metrics	(4,4,4)✗	(4,4,4)✓	(8,8,4)✓	(12,12,4)✓	(4,4,1)✓	(4,4,2)✓	(4,4,3)✓
Chamfer	2.47	2.37	**2.22**	2.31	2.58	2.34	2.37
PSNR	34.76	35.19	35.89	**36.17**	34.44	34.86	35.05

Table 1. We ablate the number of spatial features n_s, angular features n_a and spherical harmonics order l on BlendedMVS. Each triplet denotes a configuration (n_s, n_a, l). The ✓-symbol indicates that per-camera bias vectors are trained.

Metrics	(4,4,4)✗	(8,8,4)✗	MMH [24]	Voxurf [30]	Neus2 [28]	Colmap [23]	2DGS [5]
Chamfer (mm)	1.04	**1.03**	1.14	1.59	2.13	3.52	3.35
PSNR	36.81	**36.90**	36.33	35.51	34.22	-	34.89
Training	1min	1min	2-3min	15min	5min	>1h	>1h

Table 2. MVMannequins averages

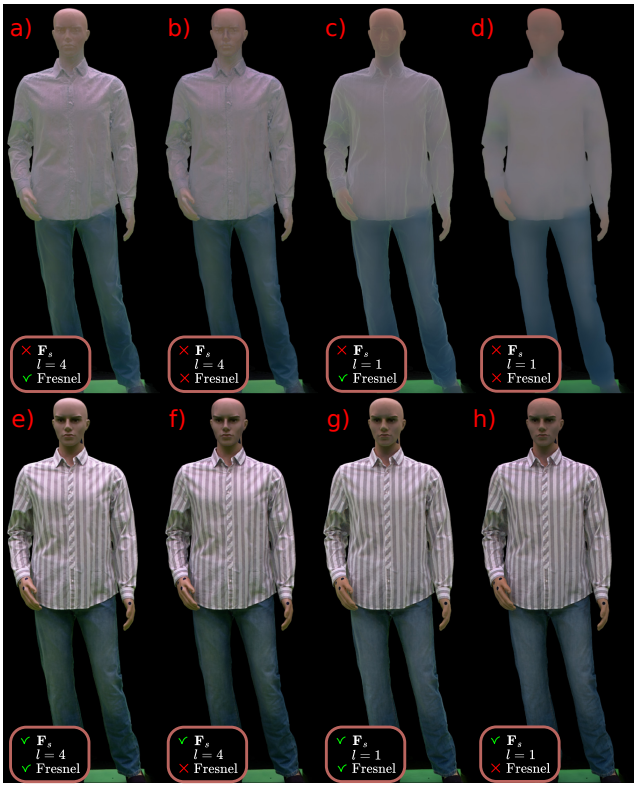

Figure 6. We disable some components of a pre-trained example (bottom left, trained with $n_s = n_a = 4$ and $l = 4$) and visualize the results. In turn, we remove the spatial features (set $\mathbf{F}_s = 0$), only keep the constant coefficient of the SH basis ($l = 1$) and disable the incident angle embedding (set $\mathbf{n} \cdot \mathbf{v} = 1$). The learned Fresnel factor is important for grazing reflections (c vs d), the angular features provide the shading (b vs d) and the spatial features contain the high frequency details (e vs a).

benchmark highlights the capability of our method to gracefully scale to high resolution scenes, both in terms of training time and rendering speed.

	(4,4,4)✗			Voxurf	NeuS2
Resolution	r/1	r/2	r/4	r/2	r/2
PSNR	**37.48**	36.62	34.74	36.56	34.53
Training	250s	110s	53s	1h	250s
Model size	232MB	57MB	15MB	500MB	25MB

Table 3. ActorsHQ averages

Metrics	(4,4,4)✓	(8,8,4)✓	Voxurf	Neus2	2DGS
Chamfer	**0.68**	0.71	0.73	0.80	0.76
PSNR	37.03	**37.74**	37.08	36	36.03
Training	150s	160s	15min	5min	12min
Model size	56MB	88MB	500MB	25MB	49MB

Table 4. DTU averages

Metrics	(4,4,4)✓	(8,8,4)✓	Voxurf	Neus2
Chamfer	2.37	**2.22**	2.64	2.93
PSNR	35.19	**35.89**	35.11	33.62
Training	110s	120s	15min	5min

Table 5. BMVS averages

DTU provides 49 to 64 images of various objects at 1600×1200 resolution, along with evaluation point clouds. We use the foreground masks provided by IDR [32] for training and testing all the baselines. 2DGS [5] does not support masks for training so it uses complete images instead, cropped and at half resolution according to its evaluation protocol. We compare against Voxurf [30], Neus2 [28] and 2DGS [5] on DTU in table 4, using the official python script for geometric evaluation. We outperform all three in term of chamfer distance and PSNR, twice as fast as NeuS2. Our rendering speed ranges from 200Hz to 300Hz depending on the object.

BlendedMVS. Finally, we test our method on 8 scenes

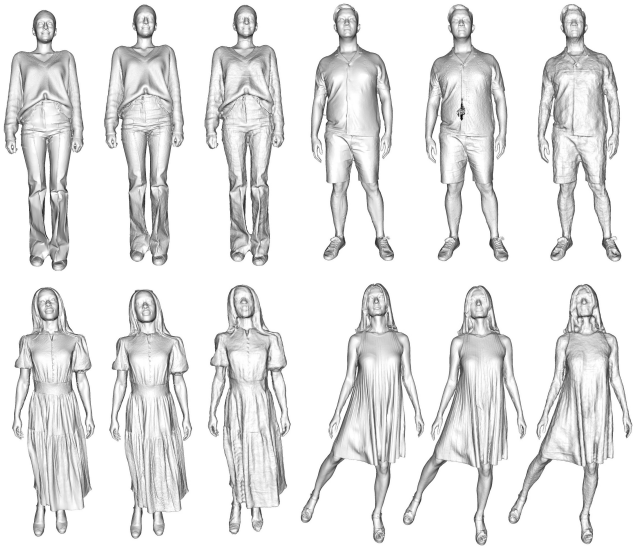

Figure 7. Results on 4 actors of the ActorsHQ dataset: Ours (left), Voxurf (middle) and NeuS2 (right)

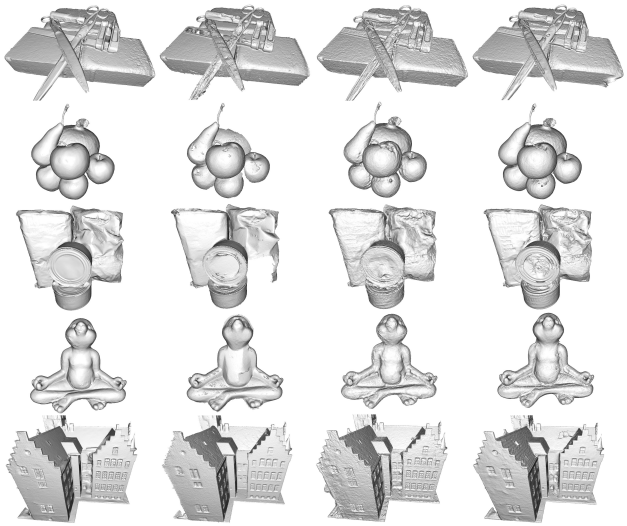

Figure 8. Reconstructions on challenging examples of DTU. From left to right: Our method, 2DGS, NeuS2, Voxurf.

of the BlendedMVS dataset [31] and report the metrics in table 5. The BlendedMVS is a semi-synthetic dataset were objects have been first reconstructed then reprojected in the images in order to produce reliable ground truth geometry and masks. There is no official script for geometric evaluation so we measure two-ways point to mesh distances in the reference frame used for training in [27], scaled by 1000x. We also ablate various choices of configurations in table 1.

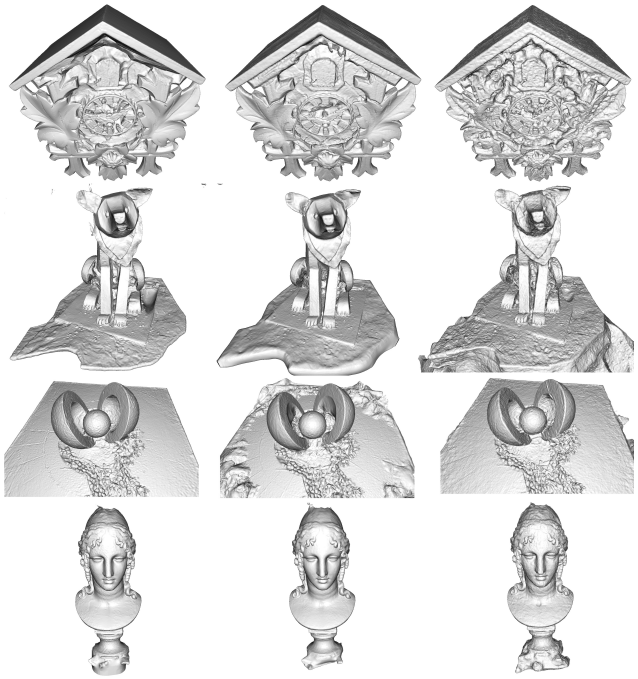

Figure 9. Reconstructions examples on BlendedMVS. From left to right: Our method, Voxurf, NeuS2.

6. Limitations

We identify several limitations to our approach. First, the fact that the lighting is encoded locally can degrade the extrapolation capabilities compared to modeling the environment with a global representation. This becomes apparent on the ActorsHQ dataset where there is a good overlap of the cameras field of views along the longitude, but very little overlap along the latitude on the legs. This creates shadows artifacts when looking from above or below rather that at a level angle. Second, although our decomposition is physically-inspired, we do not explicitly simulate light transport which can result in local minima due to shape-radiance ambiguities.

7. Conclusion and Future Work

We have presented a novel approach to the modeling of the view dependent appearance, enabling both fast reconstruction and real-time rendering of humans and objects. Crucially, we do not compromise on the geometric and photometric quality, even surpassing the state of the art in many instances. Our approach can easily scale to high resolution inputs and maintains its high performance. We have reached a parsimonious parametrization of the surface properties thanks to a simple formulation. Disentangling the spatial from the angular components also paves the way towards temporal reconstruction. We identify new exciting research directions, such as handling high frequency reflections at

a lower cost than current dedicated approaches and reducing the gap between neural rendering and physically-based rendering. Other embeddings of the angular dependency representations can be investigated, such as cubemaps or mixtures of spherical gaussians.

8. Acknowledgments

This work was supported by French government funding managed by the National Research Agency under the Investments for the Future program (PIA) grant ANR-21-ESRE-0030 (CONTINUUM). Research conducted at Kyushu University was supported by JSPS/KAKENHI JP23H03439 and AMED JP24wm0625404. We thank Laurence Boissieux for her valuable help generating 3D renderings on short notice.

References

[1] Realitycapture. https://www.capturingreality.com/. Accessed: 2024-11-21. 1

[2] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *European Conference on Computer Vision (ECCV)*, 2022. 4

[3] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *CVPR*, 2022. 1, 2, 4

[4] Yuan-Chen Guo, Di Kang, Linchao Bao, Yu He, and Song-Hai Zhang. Nerfren: Neural radiance fields with reflections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18409–18418, 2022. 3

[5] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *SIGGRAPH 2024 Conference Papers*. Association for Computing Machinery, 2024. 2, 7

[6] Mustafa Işık, Martin Rünz, Markos Georgopoulos, Taras Khakhulin, Jonathan Starck, Lourdes Agapito, and Matthias Nießner. Humanrf: High-fidelity neural radiance fields for humans in motion. *ACM Transactions on Graphics (TOG)*, 42(4):1–12, 2023. 2, 6, 1

[7] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engil Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 406–413. IEEE, 2014. 2, 6, 1

[8] Yingwenqi Jiang, Jiadong Tu, Yuan Liu, Xifeng Gao, Xiaoxiao Long, Wenping Wang, and Yuexin Ma. Gaussianshader: 3d gaussian splatting with shading functions for reflective surfaces. *arXiv preprint arXiv:2311.17977*, 2023. 3

[9] Haian Jin, Isabella Liu, Peijia Xu, Xiaoshuai Zhang, Songfang Han, Sai Bi, Xiaowei Zhou, Zexiang Xu, and Hao Su. Tensoir: Tensorial inverse rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 3

[10] Brian Karis and Epic Games. Real shading in unreal engine 4. *Proc. Physically Based Shading Theory Practice*, 4(3):1, 2013. 3

[11] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42 (4), 2023. 1, 2, 4, 6

[12] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. 6

[13] Zhaoshuo Li, Thomas Müller, Alex Evans, Russell H Taylor, Mathias Unberath, Ming-Yu Liu, and Chen-Hsuan Lin. Neuralangelo: High-fidelity neural surface reconstruction. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2023. 1

[14] Zander Majercik, Adam Marrs, Josef Spjut, and Morgan McGuire. Scaling probe-based real-time dynamic global illumination for production. *arXiv preprint arXiv:2009.10796*, 2020. 2

[15] Morgan McGuire, Mike Mara, Derek Nowrouzezahrai, and David Luebke. Real-time global illumination using precomputed light field probes. In *Proceedings of the 21st ACM SIGGRAPH symposium on interactive 3D graphics and games*, pages 1–11, 2017. 2

[16] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 1, 2

[17] Ben Mildenhall, Peter Hedman, Ricardo Martin-Brualla, Pratul P. Srinivasan, and Jonathan T. Barron. Nerf in the dark: High dynamic range view synthesis from noisy raw images. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16169–16178, 2021. 6

[18] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, 2022. 6

[19] Jacob Munkberg, Jon Hasselgren, Tianchang Shen, Jun Gao, Wenzheng Chen, Alex Evans, Thomas Müller, and Sanja Fidler. Extracting Triangular 3D Models, Materials, and Lighting From Images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8280–8290, 2022. 3

[20] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3504–3515, 2020. 2

[21] David Neubelt & Matt Pettineo. Siggraph 2015 course: Physically based shading in theory and practice - advanced lighting r&d at ready at dawn studios. https://blog.selfshadow.com/publications/s2015-shading-course/. Accessed: 2024-11-07. 1, 2

[22] Christopher M. Schlick. An inexpensive brdf model for physically-based rendering. *Computer Graphics Forum*, 13, 1994. 4

[23] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016. 7

[24] Briac Toussaint, Laurence Boissieux, Diego Thomas, Edmond Boyer, and Franco Jean-Sébastien. Millimetric Human

Surface Capture in Minutes. In *SIGGRAPH Asia 2024 - 17th ACM SIGGRAPH Conference and Exhibition on Computer Graphics and Interactive Techniques in Asia*, pages 1–12, Tokyo, Japan, 2024. ACM. 2, 6, 7

[25] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T. Barron, and Pratul P. Srinivasan. Ref-NeRF: Structured view-dependent appearance for neural radiance fields. *CVPR*, 2022. 2, 3, 4

[26] Dor Verbin, Pratul P Srinivasan, Peter Hedman, Ben Mildenhall, Benjamin Attal, Richard Szeliski, and Jonathan T Barron. Nerf-casting: Improved view-dependent appearance with consistent reflections. *arXiv preprint arXiv:2405.14871*, 2024. 3

[27] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *NeurIPS*, 2021. 1, 2, 3, 5, 8

[28] Yiming Wang, Qin Han, Marc Habermann, Kostas Daniilidis, Christian Theobalt, and Lingjie Liu. Neus2: Fast learning of neural implicit surfaces for multi-view reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 1, 2, 6, 7

[29] Liwen Wu, Sai Bi, Zexiang Xu, Fujun Luan, Kai Zhang, Iliyan Georgiev, Kalyan Sunkavalli, and Ravi Ramamoorthi. Neural directional encoding for efficient and accurate view-dependent appearance modeling. In *CVPR*, 2024. 3

[30] Tong Wu, Jiaqi Wang, Xingang Pan, Xudong Xu, Christian Theobalt, Ziwei Liu, and Dahua Lin. Voxurf: Voxel-based efficient and accurate neural surface reconstruction. In *International Conference on Learning Representations (ICLR)*, 2022. 1, 2, 5, 6, 7

[31] Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. *Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 6, 8, 1

[32] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. *Advances in Neural Information Processing Systems*, 33, 2020. 2, 7

[33] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021. 2, 3

[34] Xiuming Zhang, Pratul P. Srinivasan, Boyang Deng, Paul Debevec, William T. Freeman, and Jonathan T. Barron. Nerfactor: neural factorization of shape and reflectance under an unknown illumination. *ACM Trans. Graph.*, 40(6), 2021. 3

ProbeSDF: Light Field Probes For Neural Surface Reconstruction

Supplementary Material

9. Visual ablation

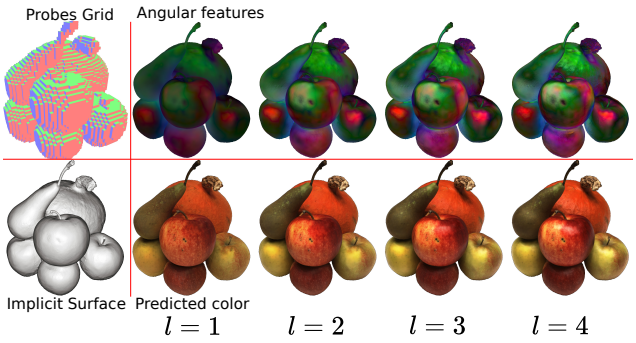

Figure 10. We train a scene with $l = 4$, that we then visualize with l varying from 1 to 4, left to right. The corresponding angular features are on the top row, and the predicted color is on the bottom row. We observe that the specularities can be removed simply by disabling the high order spherical harmonics coefficients.

10. Detailed tables

Tables 8 to 20 contain detailed metrics from all our experiments. The dimensionality of the spatial features $\mathbf{F}_s$ is n_s and the dimensionality of the angular features $\mathbf{F}_a$ is n_a. We denote a training configuration with a triplet (n_s, n_a, l). A ✓-symbol denotes the training of per-camera bias vectors, an ✗-symbol is used otherwise.

11. Neural Architecture comparison

Table 6 gives a comparison of several neural architectures used in the context of implicit surface reconstruction. Thanks to our light field probes, our MLP is entirely agnostic to the surface orientation and position, hence we can reduce its size and obtain high quality renderings, in real-time.

Method	Grid Type	SDF MLP Layers / Neurons	Color MLP Layers / Neurons
NeuS	✗	8 / 256	4 / 256
NeuS2	hash-grid	1 / 64	2 / 64
Voxurf	dense	✗	4 / 192
Ours	sparse	✗	2 / 32

Table 6. Neural architectures in the literature.

12. Comparisons on ActorsHQ

Geometric comparisons on the ActorsHQ dataset [6] are shown in figures 12, 13, 14, 15. The dataset comes with meshes reconstructed by RealityCapture [1], a multi-view stereo reconstruction software. We cannot compute geometric metrics since there is no ground truth obtained independently from the images. A qualitative comparison of the volume rendering quality is shown in figures 16, 17 and 18. We used the (4,4,4) configuration here. Note that the input images come with pre-baked segmentation masks, as shown in figure 11, that tend to have poorer accuracy on the arms and hands. This results in both geometric and photometric artifacts that are difficult to eliminate.

Figure 11. Imprecise segmentation example.

13. Comparisons on DTU

Figure 19 presents a comparison of some of the reconstruction results on DTU [7]. Close-ups of the volume rendered images are shown in figures 20 and 21. Our results are obtained with the (4,4,4) configuration.

14. Comparisons on BlendedMVS

Geometric comparisons on the BlendedMVS dataset [31] are shown in figures 22 and 23. Qualitative comparisons of the volume rendering are shown in figure 24. We used the (8,8,4) configuration here as it performed a little better on this dataset (see table 17).

15. Performance Analysis

We record inference timings on the apples example (scan 63 of DTU, (4,4,4) configuration), with a window of resolution 1920×1163 and present the results in table 7. We

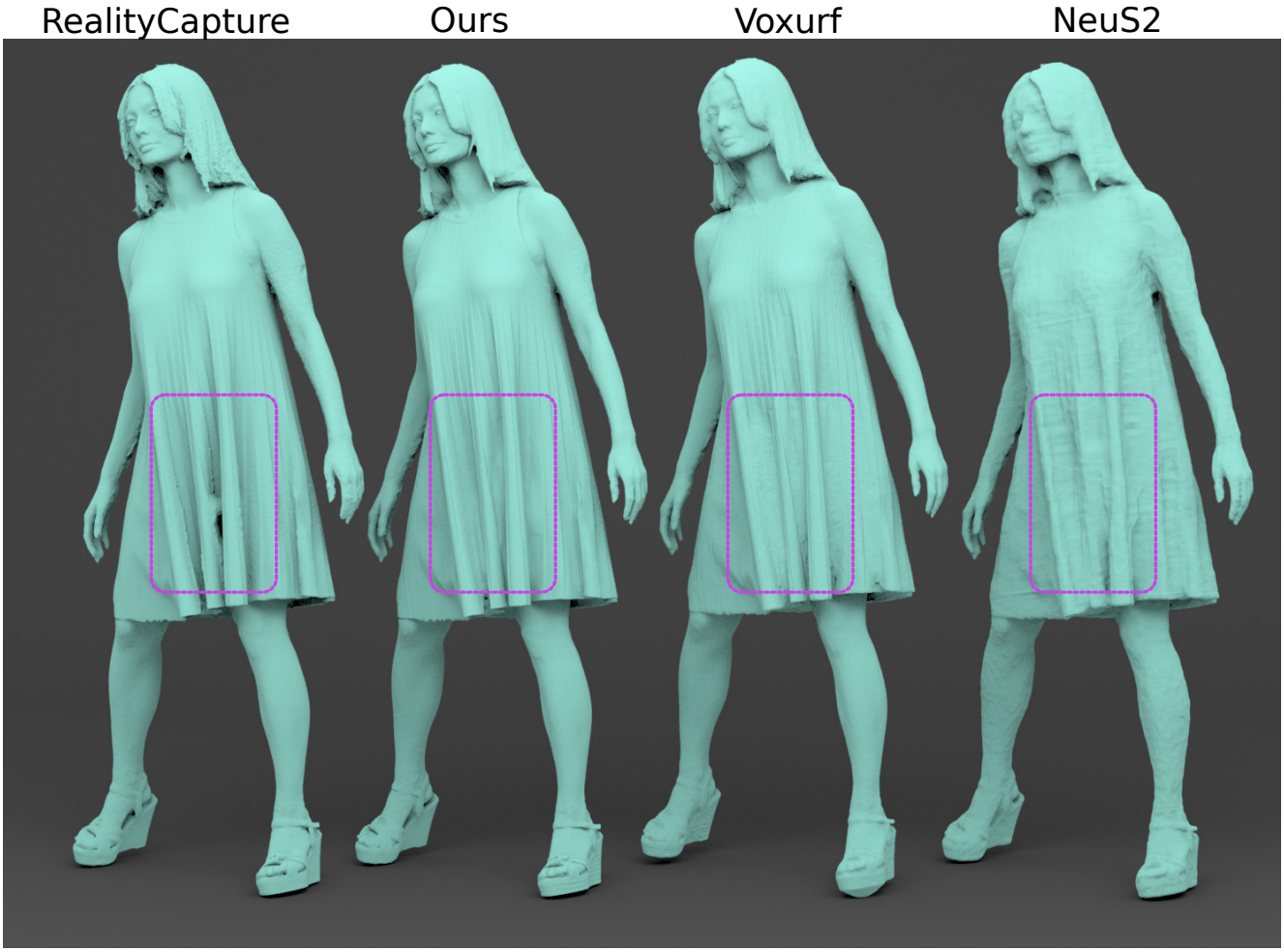

Figure 12. Reconstruction results on ActorsHQ. Left to right: RealityCapture, Ours, Voxurf, NeuS2.

lock the memory clock to 5001MHz and the gpu clock to 1500MHz to obtain stable performance measurements. The shading column corresponds to the assignment of a color to each voxel. The render column corresponds to the volume rendering kernel, which samples the SDF and color fields along rays to generate the final image. The first 4 rows correspond to the fully-fused color prediction kernel, with the computation of the spatial and angular features enabled or disabled. Thus, the 4th row corresponds to the MLP inference only whereas the 1st row corresponds to the full model. The last two rows correspond to the computation of $\mathbf{F}_s$ or $\mathbf{F}_a$ on their own, in separate kernels, and whose result is interpreted as a per-voxel color for visualization.

We observe that just evaluating the MLP (4th row) or computing the angular features on their own (5th row) roughly takes the same amount of time (1.80ms and 1.96ms) but that fusing the two operations together only takes 2.32ms, much less than the sum of the two ($1.80 + 1.96 = 3.76$ms). Including the computation of the spatial features gives the full model at 2.47ms.

Type	Shading	Render
MLP ✓, $\mathbf{F}_s$ ✓, $\mathbf{F}_a$ ✓	2.47	1.34
MLP ✓, $\mathbf{F}_s$ ✓, $\mathbf{F}_a$ ✗	1.92	1.34
MLP ✓, $\mathbf{F}_s$ ✗, $\mathbf{F}_a$ ✓	2.32	1.34
MLP ✓, $\mathbf{F}_s$ ✗, $\mathbf{F}_a$ ✗	1.80	1.34
$\mathbf{F}_a$ only	1.96	1.34
$\mathbf{F}_s$ only	0.82	1.34

Table 7. Timings in ms

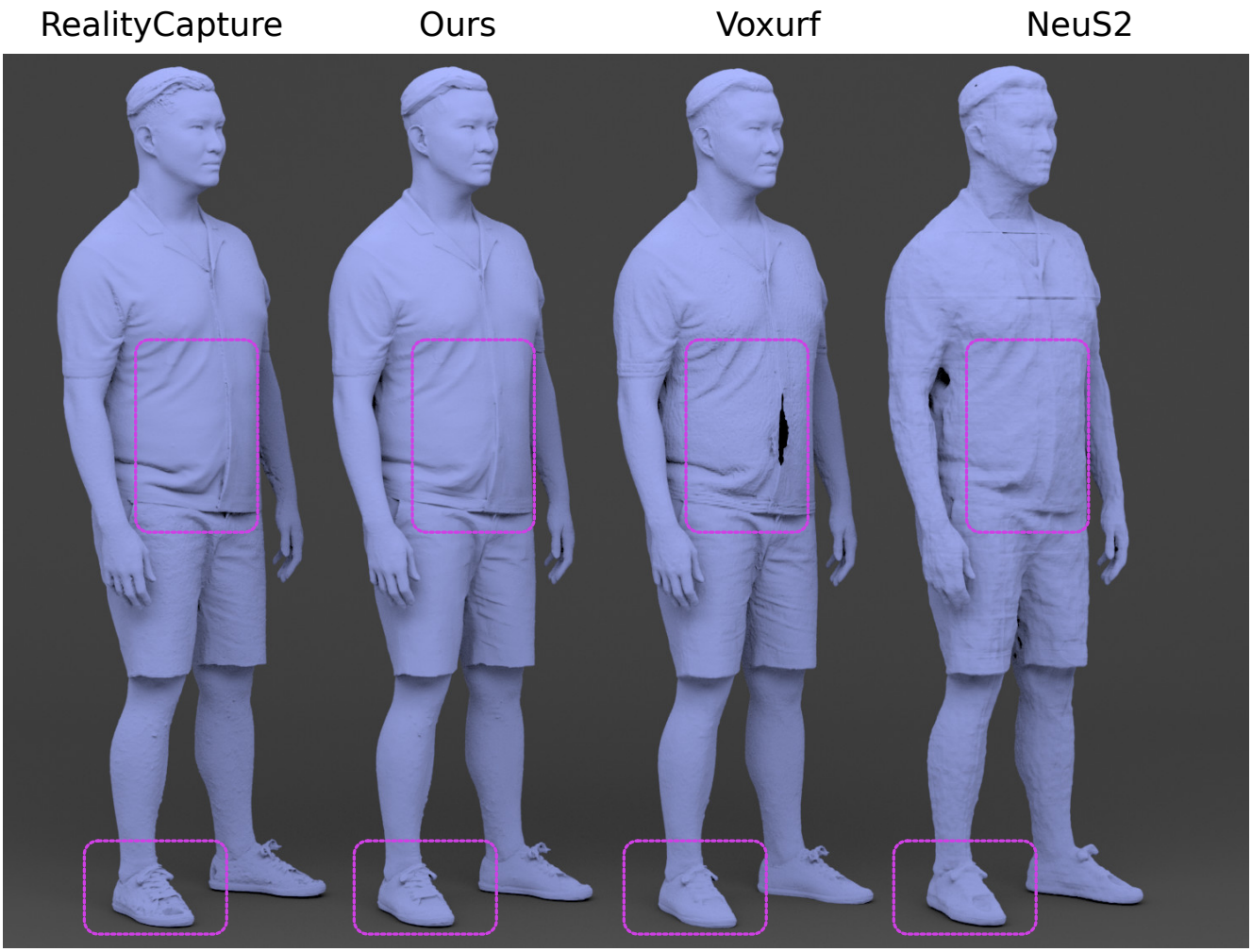

Figure 13. Reconstruction results on ActorsHQ. Left to right: RealityCapture, Ours, Voxurf, NeuS2.

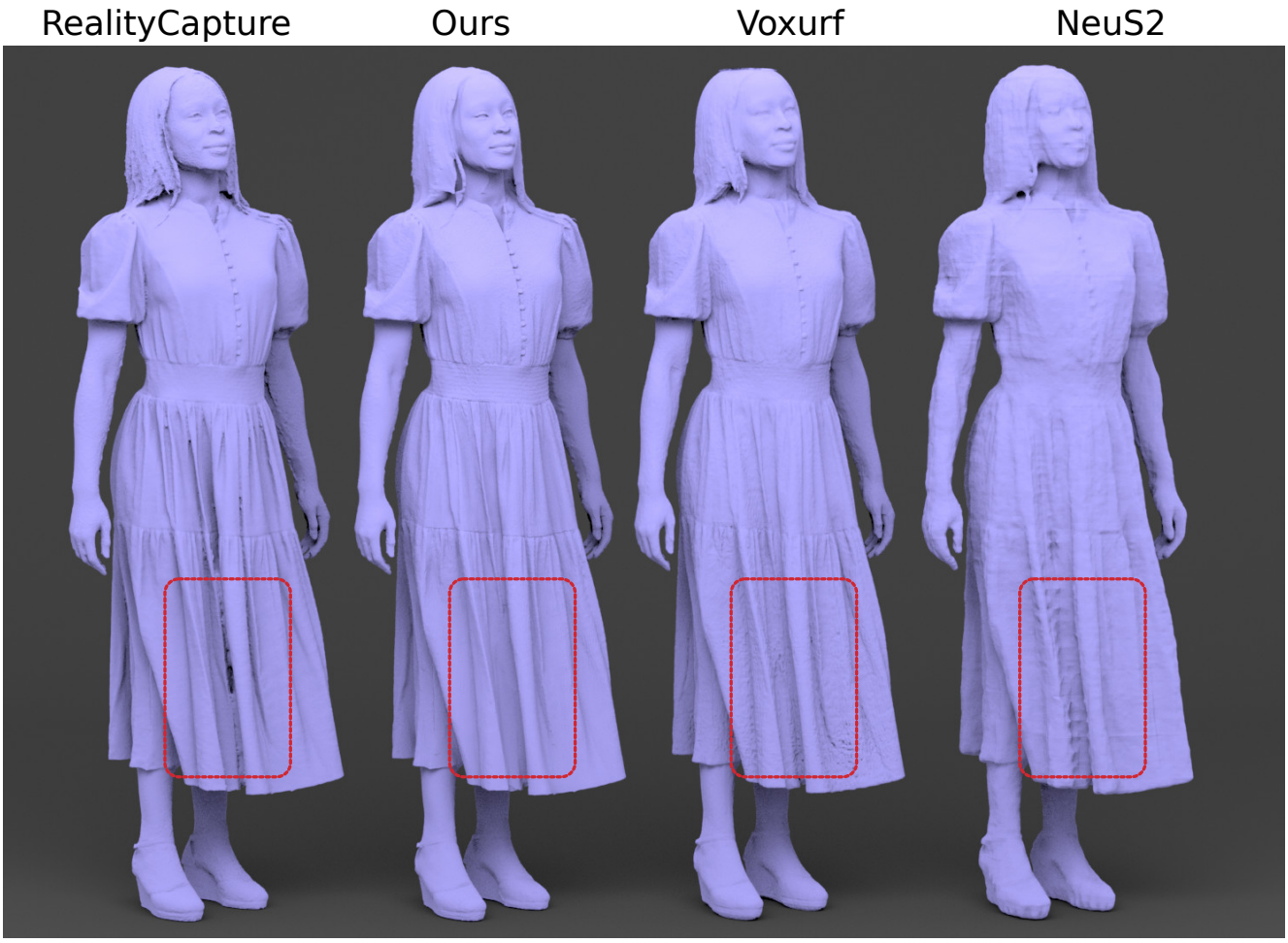

Figure 14. Reconstruction results on ActorsHQ. Left to right: RealityCapture, Ours, Voxurf, NeuS2.

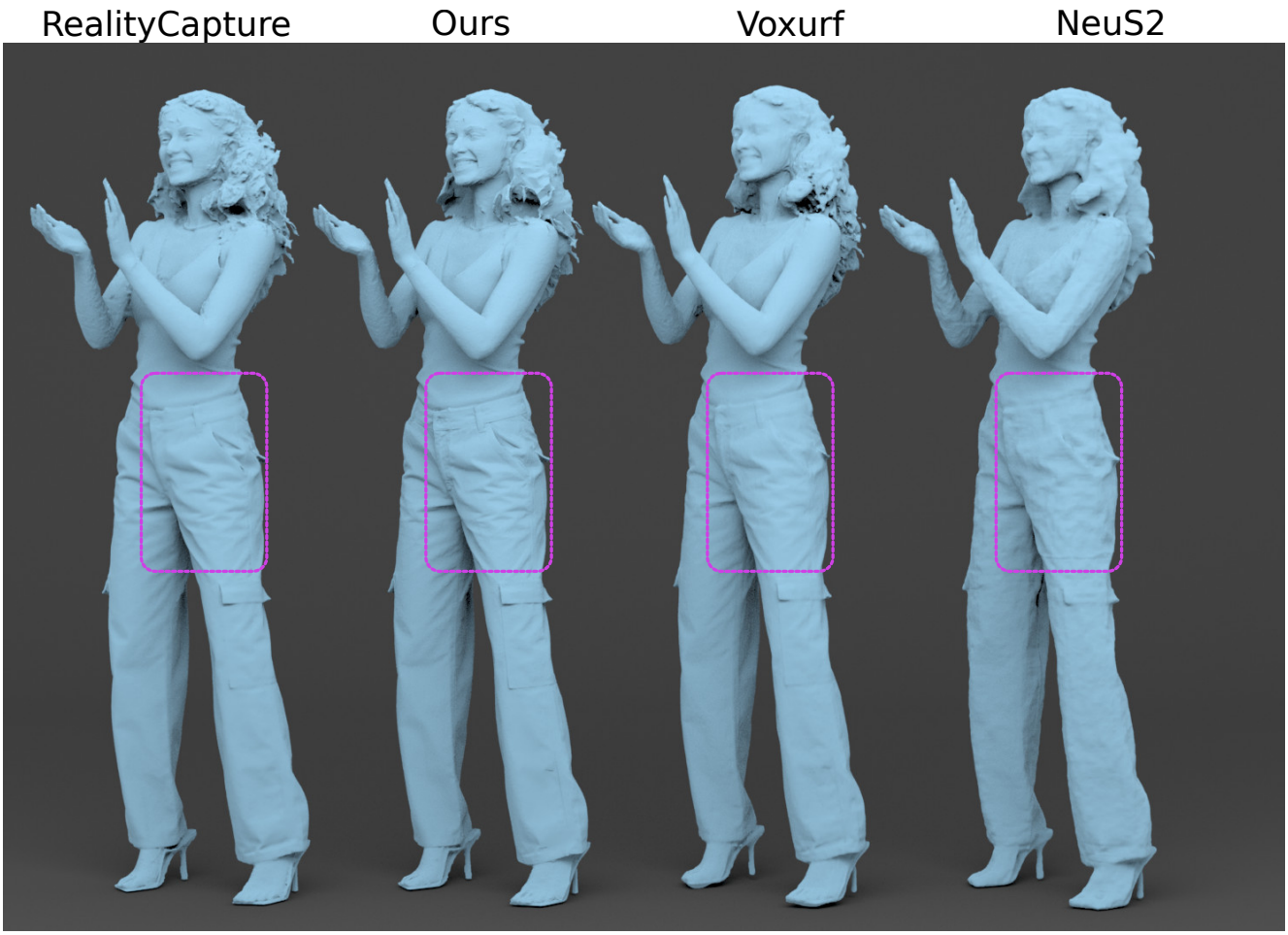

Figure 15. Reconstruction results on ActorsHQ. Left to right: RealityCapture, Ours, Voxurf, NeuS2.

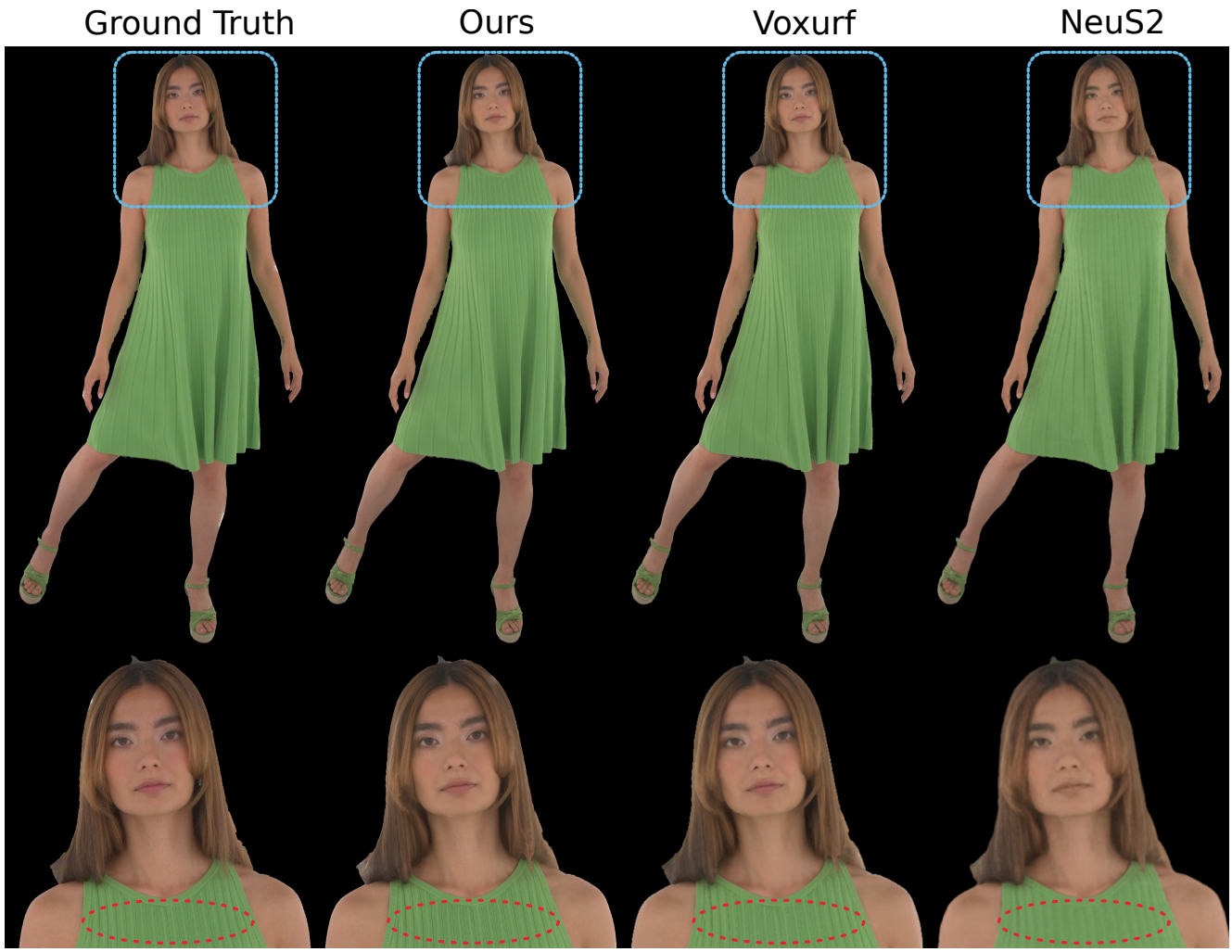

Figure 16. Qualitative comparison on ActorsHQ. Left to right: ground truth, Ours, Voxurf, NeuS2. Our method is able to handle the full resolution images, which enables to reconstruct the sewing patterns at a sub-millimetric scale.

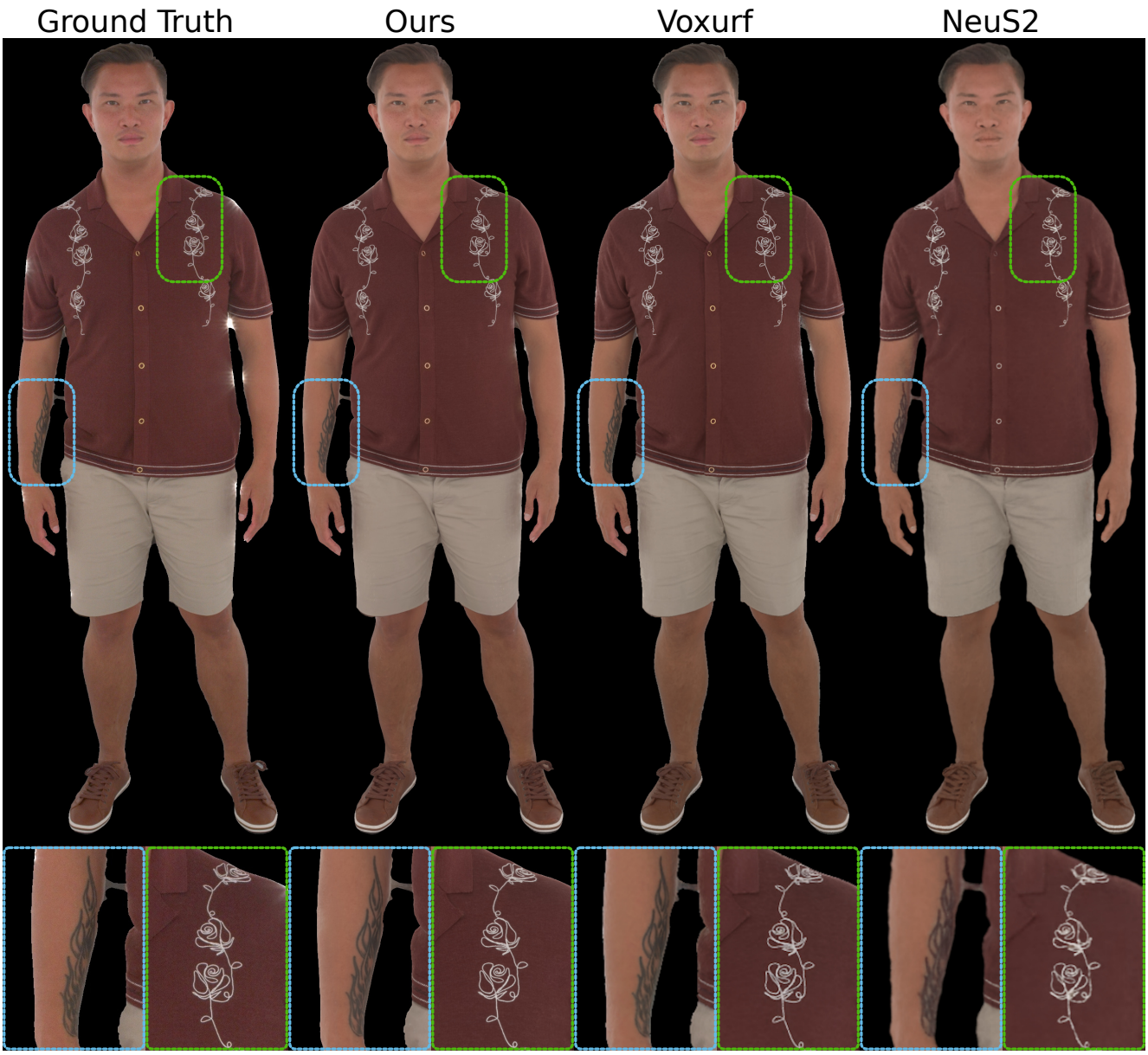

Figure 17. Qualitative comparison on ActorsHQ. Left to right: ground truth, Ours, Voxurf, NeuS2.

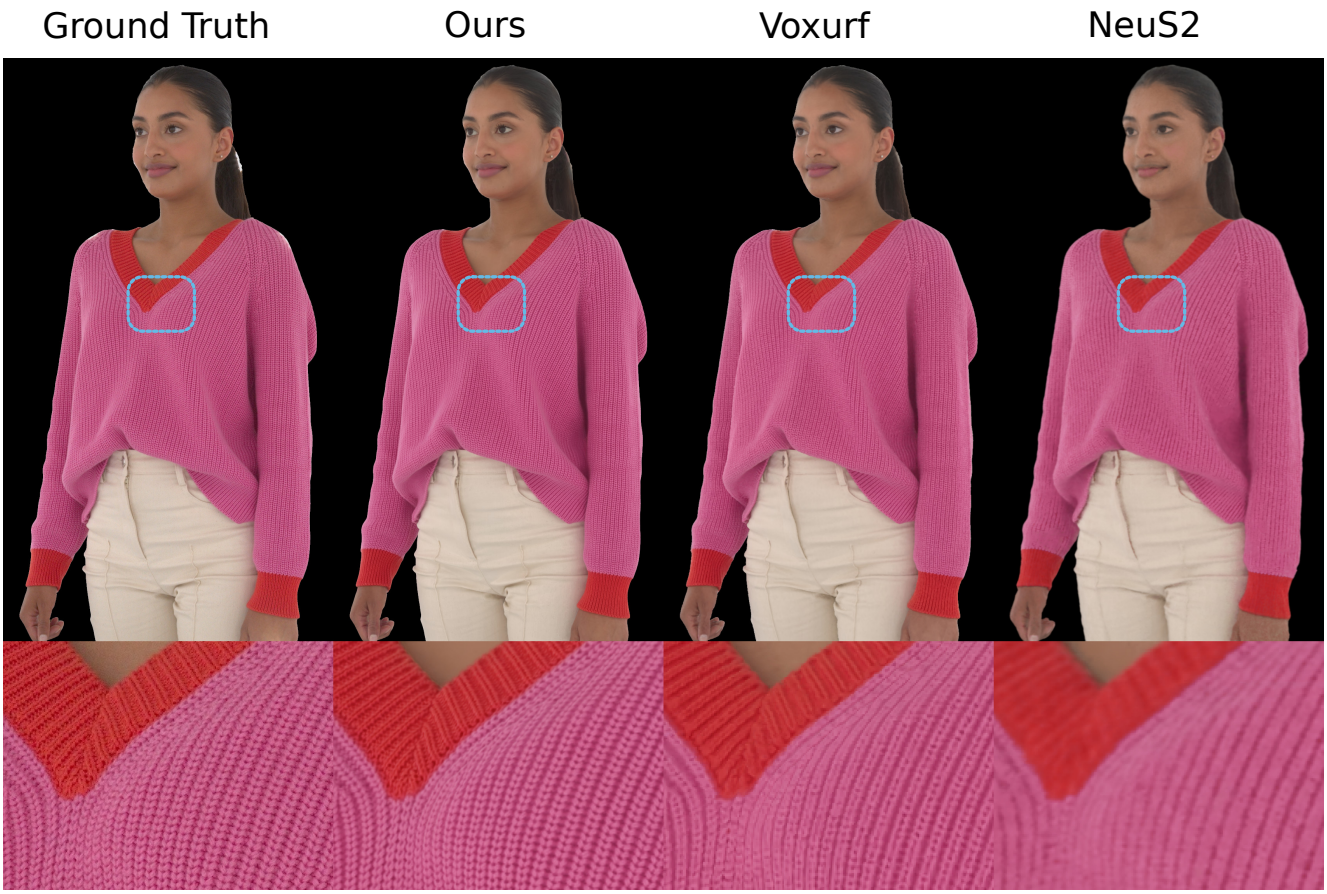

Figure 18. Qualitative comparison on ActorsHQ. Left to right: ground truth, Ours, Voxurf, NeuS2.

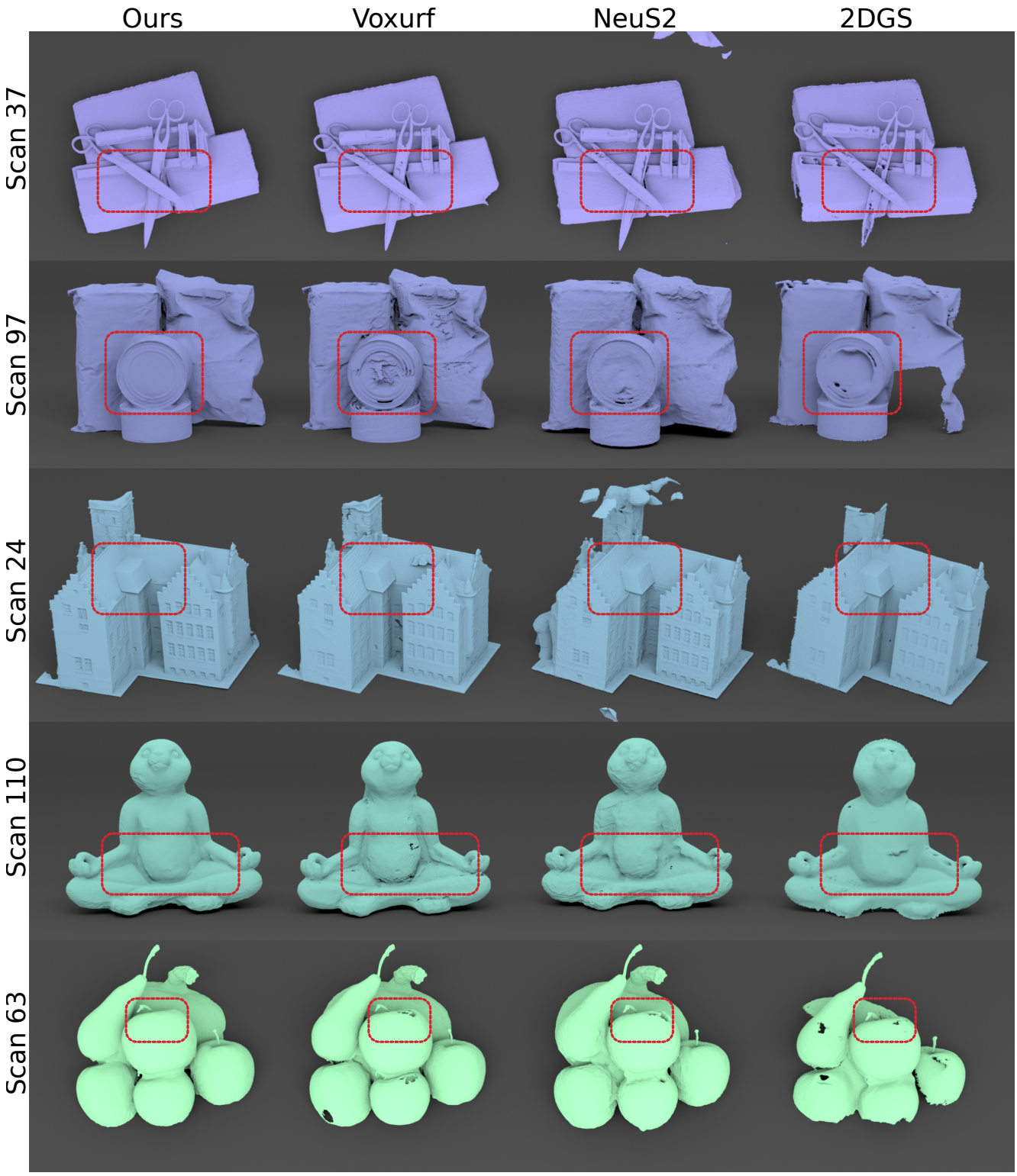

Figure 19. Reconstruction results on DTU. Left to right: Ours, Voxurf, NeuS2, 2DGS. We find that 2DGS excels at reconstructing flat surfaces (doll house roof) but tends to under-perform on reflective materials. 2DGS fails to extract geometry on some parts of the objects (scans 97 and 63). In contrast, our method recovers smooth surfaces even under strong specularities (metal scissors, tuna can and apples). Voxurf struggles on the most shiny materials despite its considerably larger MLP. NeuS2's reconstruction suffers from grid-aligned artifacts, possibly due to discontinuities in its hash-grid interpolation scheme (shoulder of the bunny in scan 110).

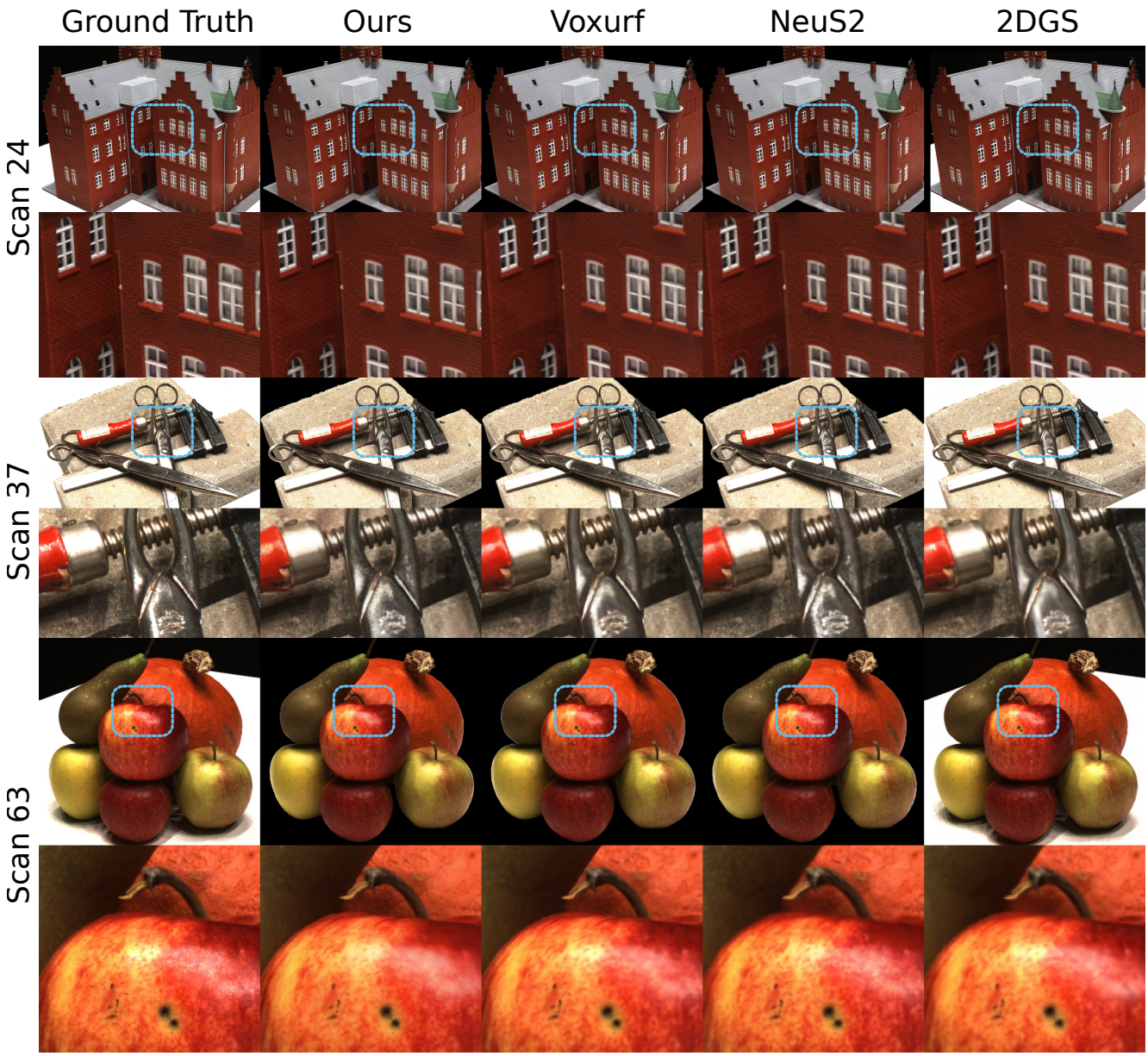

Figure 20. Qualitative comparison on DTU. Top: scan 24, middle: scan 37, bottom: scan 63. Left to right: ground truth, Ours, Voxurf, NeuS2, 2DGS.

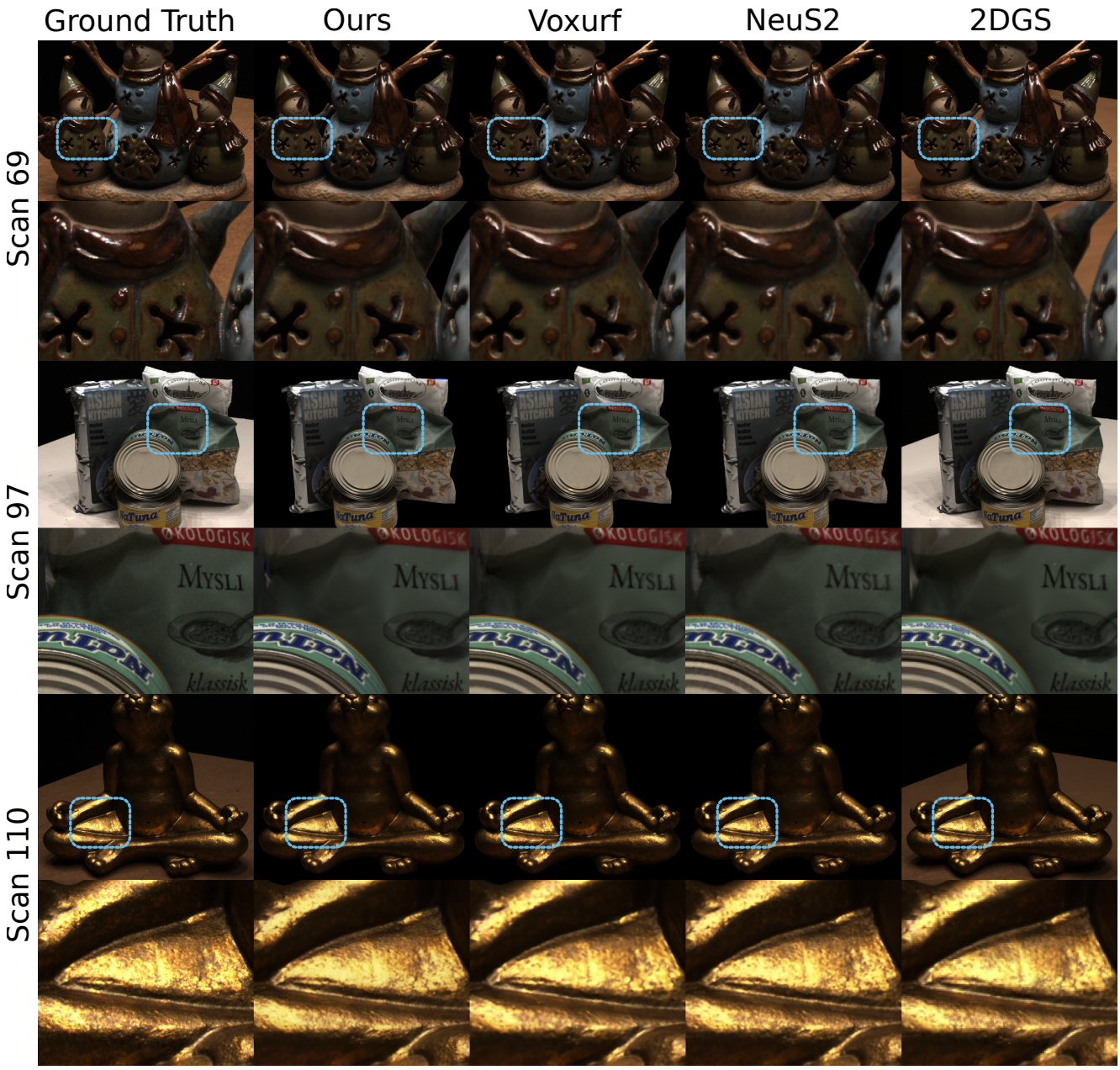

Figure 21. Qualitative comparison on DTU. Top: scan 69, middle: scan 97, bottom: scan 110. Left to right: ground truth, Ours, Voxurf, NeuS2, 2DGS.

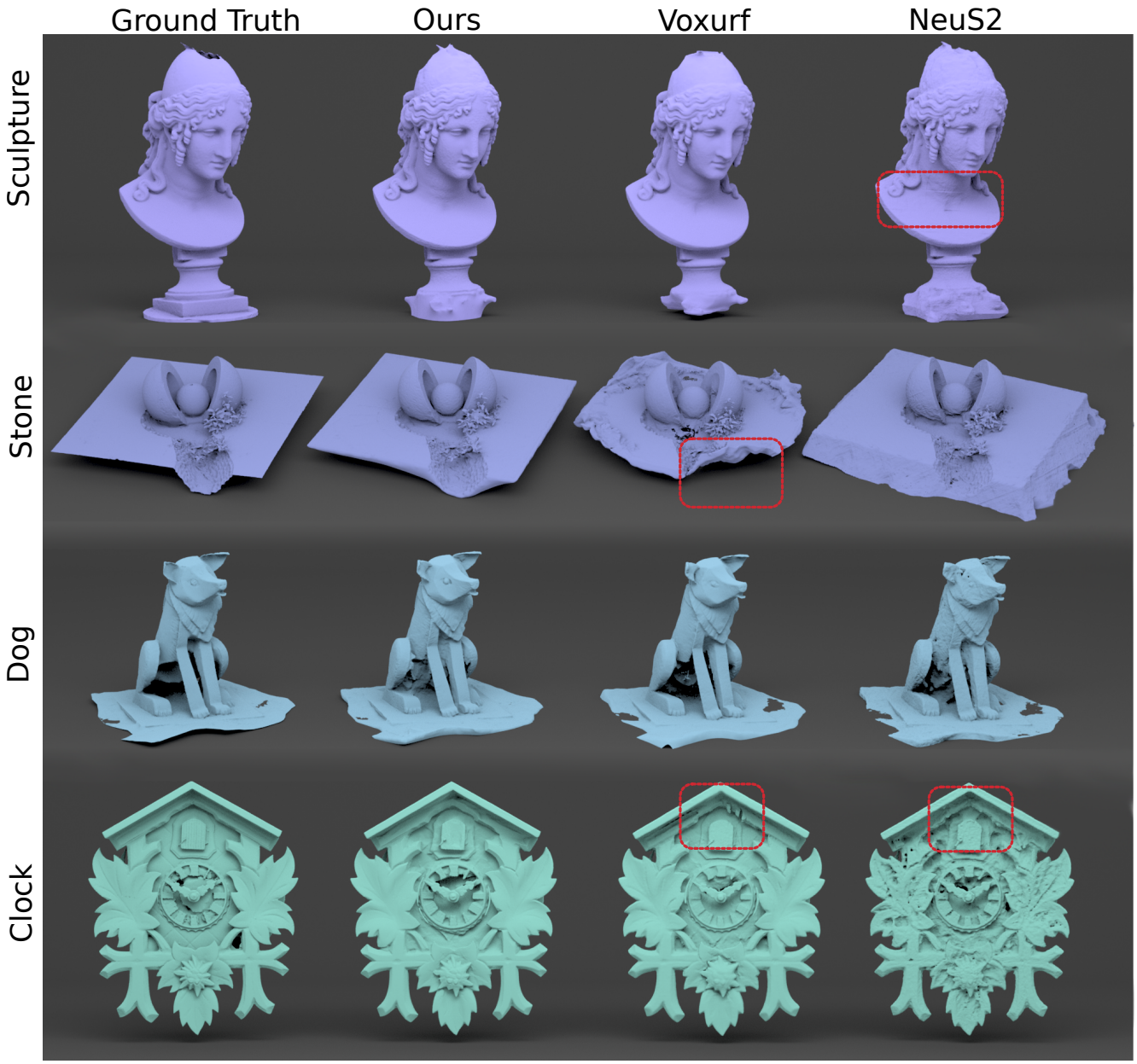

Figure 22. Reconstruction results on BlendedMVS. Left to right: Ground Truth, Ours, Voxurf, NeuS2.

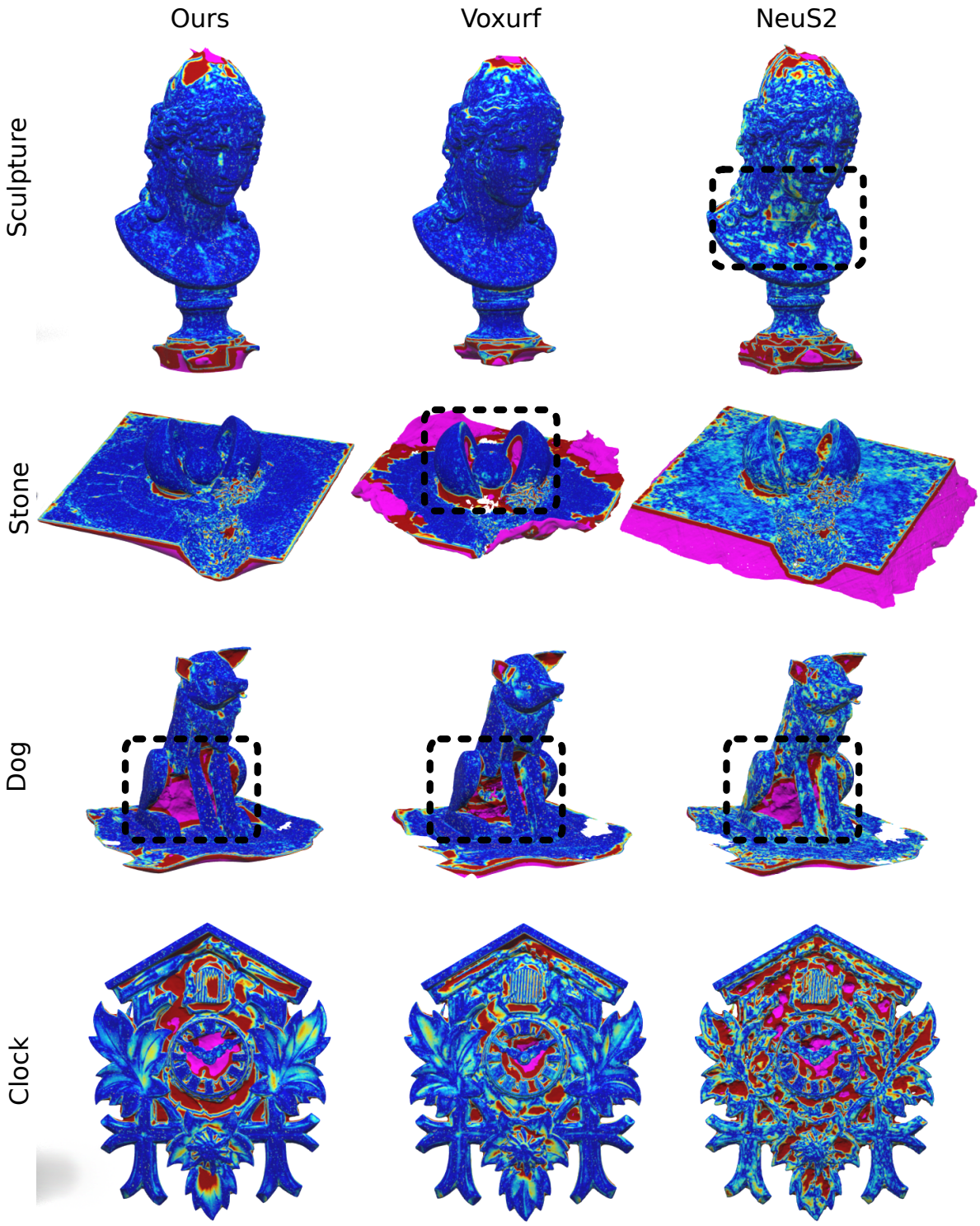

Figure 23. Accuracy heatmaps on BlendedMVS. The pink color denotes points too far away from the ground truth, which are ignored in the computation of the metrics. Left to right: Ours, Voxurf, NeuS2. Voxurf fails to carve inside the two hemispheres in the stone example and the corners of the base are missing. However, Voxurf is able to carve under the dog statue whereas both NeuS2 and our method fail on this example. NeuS2 is noticably more noisy on all examples.

Figure 24. Qualitative comparison on BlendedMVS. Left to right: ground truth, Ours, Voxurf, NeuS2.

		kinette								kino					
Chamfer (mm)	Mean	cos	naked	jea	opt1	opt2	opt3	sho	tig	cos	naked	jea	opt	sho	tig
(4,4,4)✗	**1.04**	**1.51**	**0.54**	**1.09**	**1.14**	**1.75**	**1.22**	**0.95**	**0.66**	**1.53**	0.60	**0.75**	**1.51**	**0.67**	**0.67**
MMH	1.15	1.55	**0.54**	1.10	1.16	2.12	1.45	1.07	0.68	1.77	**0.59**	0.80	1.72	0.75	0.75
Voxurf	1.59	1.70	1.28	1.62	1.62	2.06	1.61	1.48	0.98	2.60	1.30	1.27	2.54	1.12	1.06
Neus2	2.13	3.71	1.43	2.10	2.13	2.90	2	2.12	1.25	3.55	1.61	1.64	2.61	1.43	1.37
Colmap	3.51	2.69	4.27	2.99	4.59	4.18	2.81	3.71	2.50	4.11	4.20	2.48	4.34	3.09	3.23
2DGS	3.35	5.05	2.27	8.30	3.25	3.40	3.63	2.66	3.52	4.18	2.25	1.75	2.80	1.98	1.87

Table 8. MVMannequins per-scene chamfer (mm)

		kinette								kino					
PSNR (db)	Mean	cos	naked	jea	opt1	opt2	opt3	sho	tig	cos	naked	jea	opt	sho	tig
(4,4,4)✗	**36.81**	**29.48**	**40.61**	**30.73**	**40.48**	**38.69**	**34.13**	**37.70**	**31.74**	**39.64**	**41.57**	**35.80**	**39.29**	**37.50**	**37.99**
MMH	36.33	29.24	40.03	30.55	39.94	38.01	33.82	37.18	31.38	39.22	40.88	35.25	38.59	37.11	37.49
Voxurf	35.51	28.39	37.51	30.20	39.26	37.34	32.82	36.73	30.81	38.17	40.09	34.45	37.57	36.69	37.16
Neus2	34.22	28.09	37.23	29.44	36.98	35.55	32.58	34.98	30.14	36.65	38.05	33.39	35.78	35.01	35.23
2DGS	34.89	27.26	37.92	28.88	37.62	36.84	32.22	36.24	29.86	37.97	38.82	34.99	37.28	36.14	36.50

Table 9. MVMannequins per-scene PSNR

		kinette								kino					
Chamfer (mm)	Mean	cos	naked	jea	opt1	opt2	opt3	sho	tig	cos	naked	jea	opt	sho	tig
(8,8,4)✗	**1.03**	**1.48**	**0.54**	**1.08**	1.12	**1.70**	1.24	**0.93**	**0.65**	1.52	**0.59**	0.74	1.51	**0.66**	0.67
(12,12,4)✗	1.04	1.52	0.54	1.09	1.13	1.72	1.23	0.96	0.66	1.52	0.59	**0.74**	1.52	0.68	**0.67**
(4,4,1)✗	1.30	1.56	1.22	1.22	1.25	1.78	1.62	1.50	1.05	1.78	1.16	0.81	1.48	0.86	0.95
(4,4,2)✗	1.04	1.48	0.59	1.09	**1.10**	1.71	**1.22**	0.94	0.67	**1.51**	0.64	0.74	**1.45**	0.69	0.69
(4,4,3)✗	1.04	1.49	0.56	1.09	1.12	1.73	1.22	0.96	0.66	1.52	0.60	0.75	1.50	0.68	0.68
(4,4,4)✗	1.04	1.51	0.54	1.09	1.14	1.75	1.22	0.95	0.66	1.53	0.60	0.75	1.51	0.67	0.67

Table 10. Detailed Ablation Table. MVMannequins per-scene chamfer (mm)

		kinette								kino					
PSNR (db)	Mean	cos	naked	jea	opt1	opt2	opt3	sho	tig	cos	naked	jea	opt	sho	tig
(8,8,4)✗	36.90	29.56	40.76	**30.80**	40.62	38.80	34.23	37.76	**31.86**	39.76	41.66	35.88	39.35	37.56	38.07
(12,12,4)✗	**36.93**	**29.57**	**40.79**	30.79	**40.63**	**38.82**	**34.24**	**37.78**	31.83	**39.79**	**41.73**	**35.88**	**39.41**	**37.58**	**38.20**
(4,4,1)✗	35.84	29.24	38.61	30.52	39.03	37.47	33.55	36.24	31.24	38.81	39.46	35.37	38.68	36.58	36.98
(4,4,2)✗	36.53	29.31	40.09	30.62	40.07	38.38	33.95	37.42	31.55	39.35	41.06	35.67	39.01	37.26	37.71
(4,4,3)✗	36.68	29.38	40.39	30.66	40.30	38.54	34.05	37.56	31.62	39.50	41.37	35.72	39.14	37.41	37.86
(4,4,4)✗	36.81	29.48	40.61	30.73	40.48	38.69	34.13	37.70	31.74	39.64	41.57	35.80	39.29	37.50	37.99

Table 11. Detailed Ablation Table. MVMannequins per-scene PSNR

PSNR	Resolution	Mean	A1S1	A2S1	A3S1	A4S1	A5S1	A6S1	A7S1	A8S1	A1S2	A4S2	A5S2	A6S2	A8S2
(4,4,4)✗	r/1	**37.48**	**37.64**	**38.21**	**37.58**	**35.78**	**38.36**	36.80	**37.80**	**38.09**	**37.91**	**35.59**	**38.56**	36.59	**38.32**
(4,4,4)✗	r/2	36.62	36.86	36.90	36.59	34.55	37.54	36.39	37.03	37.28	37.19	34.32	37.75	36.24	37.47
(4,4,4)✗	r/4	34.75	35.43	34.05	35.21	32.08	35.59	34.96	35.41	35.16	35.67	32.14	35.94	34.75	35.34
Voxurf	r/2	36.56	37.04	36.93	36.17	34.58	36.96	**36.85**	36.69	36.99	37.12	34.33	37.53	**36.81**	37.31
Neus2	r/2	34.53	35.22	34.67	33.28	32.50	35.20	33.86	35.43	35.41	35.71	32.52	35.38	34.02	35.72

Table 12. ActorsHQ per-scene PSNR

Chamfer (mm)	Mean	24	37	40	55	63	65	69	83	97	105	106	110	114	118	122
(4,4,4)✓	**0.68**	0.65	0.74	0.34	**0.34**	1.02	0.71	**0.62**	1.34	**0.94**	0.70	**0.53**	0.96	0.36	0.45	0.47
(8,8,4)✓	0.71	0.56	**0.71**	0.34	**0.34**	1.38	0.74	0.64	1.34	0.95	0.70	0.54	1.07	**0.35**	0.45	0.47
Voxurf	0.73	0.76	0.72	0.67	**0.34**	0.95	**0.62**	0.79	1.35	0.96	0.74	0.61	1.17	**0.35**	**0.44**	0.49
Neus2	0.80	0.55	0.81	1.66	0.38	**0.92**	0.72	0.79	1.31	1.07	0.80	0.61	**0.89**	0.46	0.52	0.58
2DGS	0.76	**0.47**	0.82	**0.32**	0.36	1.06	0.89	0.81	**1.30**	1.23	**0.66**	0.65	1.34	0.42	0.66	**0.46**

Table 13. DTU per-scene chamfer (mm)

PSNR (db)	Mean	24	37	40	55	63	65	69	83	97	105	106	110	114	118	122
(4,4,4)✓	37.03	35.58	30.59	35.59	35.56	38.89	38.06	34.81	39.90	33.97	38.84	40.03	36.23	34.90	40.87	41.63
(8,8,4)✓	**37.74**	**36.73**	**31.53**	**36.31**	**36.74**	39.38	38.93	35.05	40.25	**34.69**	**39.51**	**40.45**	**36.79**	**35.60**	41.59	**42.49**
Voxurf	37.08	34.97	30.70	33.82	35.02	**39.45**	**39.22**	35.48	**41.03**	34.35	38.78	39.43	35.36	35.20	**41.79**	41.69
Neus2	36.00	34.57	29.82	34.30	34.64	37.93	36.87	33.79	38.95	32.79	38.13	38.37	35.14	34.37	39.93	40.32
2DGS	36.03	35.01	30.61	34.47	33.77	38.27	36.11	**35.84**	39.53	34.26	38.33	37.86	34.82	33.46	39.10	39.05

Table 14. DTU per-scene PSNR

Chamfer (mm)	Mean	24	37	40	55	63	65	69	83	97	105	106	110	114	118	122
(4,4,4)✗	0.71	0.68	0.82	**0.34**	0.35	1.20	0.76	**0.59**	1.34	0.91	0.74	0.57	**0.91**	0.39	0.50	0.50
(8,8,4)✓	0.71	**0.56**	**0.71**	**0.34**	**0.34**	1.38	0.74	0.64	1.34	0.95	0.70	0.54	1.07	**0.35**	0.45	**0.47**
(12,12,4)✓	0.70	0.58	0.74	**0.34**	**0.34**	1.37	0.73	0.66	1.32	**0.87**	0.71	0.54	0.93	**0.35**	0.45	**0.47**
(4,4,1)✓	0.85	0.64	0.80	0.35	**0.34**	1.79	0.71	0.77	**1.30**	1.12	**0.69**	**0.52**	2.34	0.41	**0.43**	**0.47**
(4,4,2)✓	0.69	0.64	0.74	**0.34**	**0.34**	1.09	**0.69**	0.67	1.33	1.01	**0.69**	0.53	1.02	0.37	0.44	**0.47**
(4,4,3)✓	**0.67**	0.64	0.75	**0.34**	**0.34**	1.04	0.70	0.62	1.33	0.95	**0.69**	0.53	0.92	0.36	0.44	**0.47**
(4,4,4)✓	0.68	0.65	0.74	**0.34**	**0.34**	**1.02**	0.71	0.62	1.34	0.94	0.70	0.53	0.96	0.36	0.45	**0.47**

Table 15. Detailed Ablation Table. DTU per-scene chamfer (mm)

PSNR (db)	Mean	24	37	40	55	63	65	69	83	97	105	106	110	114	118	122
(4,4,4)✗	36.18	34.73	29.81	34.87	34.52	38.23	36.38	34.52	39.35	33.17	38.34	38.80	36	34.03	39.62	40.38
(8,8,4)✓	37.74	36.73	31.53	36.31	36.74	39.38	38.93	35.05	40.25	34.69	39.51	**40.45**	36.79	35.60	41.59	42.49
(12,12,4)✓	**38.03**	**37.20**	**31.96**	**36.64**	**36.95**	**39.55**	**39.33**	**35.71**	**40.42**	**35.27**	**39.58**	40.32	**36.97**	**35.89**	**41.97**	**42.75**
(4,4,1)✓	35.92	35.13	29.81	34.97	34.66	36.40	36.77	32.90	38.86	32.29	38.03	39.43	34.36	34	40.36	40.92
(4,4,2)✓	36.44	35.25	29.97	35.31	35.27	37.66	37.32	33.62	39.48	32.98	38.53	39.71	35.49	34.39	40.53	41.11
(4,4,3)✓	36.76	35.36	30.27	35.40	35.46	38.46	37.66	34.40	39.71	33.59	38.67	39.80	35.95	34.67	40.69	41.32
(4,4,4)✓	37.03	35.58	30.59	35.59	35.56	38.89	38.06	34.81	39.90	33.97	38.84	40.03	36.23	34.90	40.87	41.63

Table 16. Detailed Ablation Table. DTU per-scene PSNR

Chamfer	Mean	dog	bear	clock	durian	man	sculpture	stone	jade
(4,4,4)✓	2.36	2.51	2.27	1.90	3.63	**1.79**	**1.61**	**1.30**	3.88
(8,8,4)✓	**2.21**	**2.24**	**2.03**	**1.69**	3.26	1.81	**1.61**	1.36	**3.71**
Voxurf	2.64	2.28	2.20	1.88	**2.98**	2.11	1.75	3.96	3.99
Neus2	2.93	2.78	2.71	2.63	4.23	2.25	2.50	1.91	4.43

Table 17. BlendedMVS per-scene chamfer

PSNR	Mean	dog	bear	clock	durian	man	sculpture	stone	jade
(4,4,4)✓	35.19	35.49	30.55	34.68	31.28	42.94	40.80	30.84	34.92
(8,8,4)✓	**35.89**	**36.30**	**30.96**	**35.42**	**31.65**	**43.72**	**41.34**	31.01	**36.69**
Voxurf	35.11	35.24	30.88	34.49	29.69	43.35	41.06	30.23	35.93
Neus2	33.62	34.56	29.99	31.04	29.21	40.88	39.10	**31.36**	32.79

Table 18. BlendedMVS per-scene PSNR

Chamfer	Mean	dog	bear	clock	durian	man	sculpture	stone	jade
(4,4,4)✗	2.47	2.18	2.71	2.05	3.81	1.97	1.75	1.36	3.95
(8,8,4)✓	**2.21**	2.24	2.03	**1.69**	**3.26**	1.81	1.61	1.36	**3.71**
(12,12,4)✓	2.31	**2.10**	2.31	1.86	3.58	1.96	**1.59**	1.34	3.74
(4,4,1)✓	2.58	3.27	**2.02**	2.44	3.88	1.92	1.88	1.33	3.93
(4,4,2)✓	2.35	2.59	2.03	2.04	3.66	1.80	1.62	**1.29**	3.75
(4,4,3)✓	2.36	2.56	2.35	1.94	3.59	**1.79**	1.64	1.31	3.74
(4,4,4)✓	2.36	2.51	2.27	1.90	3.63	**1.79**	1.61	1.30	3.88

Table 19. Detailed Ablation Table. BlendedMVS per-scene chamfer

(4,4,4)✗	34.76	35.71	30.35	33.76	31	42.65	40.08	30.84	33.72
(8,8,4)✓	35.89	36.30	30.96	35.42	31.65	43.72	41.34	31.01	36.69
(12,12,4)✓	**36.17**	**36.50**	**30.97**	**36.55**	**32.12**	**43.75**	**41.41**	**31.04**	**37.00**
(4,4,1)✓	34.44	34.48	30.34	32.81	31.12	42.44	39.89	30.75	33.72
(4,4,2)✓	34.86	35	30.53	33.85	31.38	42.71	40.40	30.82	34.19
(4,4,3)✓	35.05	35.18	30.64	34.34	31.36	42.90	40.69	30.82	34.48
(4,4,4)✓	35.19	35.49	30.55	34.68	31.28	42.94	40.80	30.84	34.92

Table 20. Detailed Ablation Table. BlendedMVS per-scene PSNR