Neural Lumigraph Rendering

Petr Kellnhofer[1,2], Lars C. Jebe[1], Andrew Jones[1], Ryan Spicer[1], Kari Pulli[1], Gordon Wetzstein[2,1]
[1]Raxium [2]Stanford University
{pkellnhofer,ljebe,ajones,rspicer,kpulli,gwetzstein}@raxium.com

Abstract

Novel view synthesis is a challenging and ill-posed inverse rendering problem. Neural rendering techniques have recently achieved photorealistic image quality for this task. State-of-the-art (SOTA) neural volume rendering approaches, however, are slow to train and require minutes of inference (i.e., rendering) time for high image resolutions. We adopt high-capacity neural scene representations with periodic activations for jointly optimizing an implicit surface and a radiance field of a scene supervised exclusively with posed 2D images. Our neural rendering pipeline accelerates SOTA neural volume rendering by about two orders of magnitude and our implicit surface representation is unique in allowing us to export a mesh with view-dependent texture information. Thus, like other implicit surface representations, ours is compatible with traditional graphics pipelines, enabling real-time rendering rates, while achieving unprecedented image quality compared to other surface methods. We assess the quality of our approach using existing datasets as well as high-quality 3D face data captured with a custom multi-camera rig.

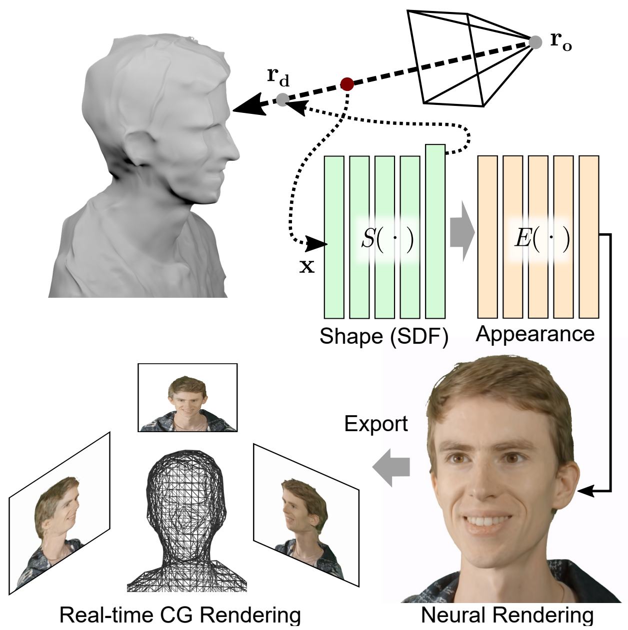


Figure 1. Overview of our framework. Given a set of multi-view images, we optimize representation networks modeling shape and appearance of a scene end to end using a differentiable sphere tracer. The resulting models can be exported to enable view-dependent real-time rendering using traditional graphics pipelines.

1. Introduction

Novel view synthesis and 3D shape estimation from 2D images are inverse problems of fundamental importance in applications as diverse as photogrammetry, remote sensing, visualization, AR/VR, teleconferencing, visual effects, and games. While traditional 3D computer vision pipelines have been studied for decades, only emerging neural rendering techniques have been able to achieve photorealistic quality for novel view synthesis (e.g., [38, 56]).

State-of-the-art neural rendering approaches, such as neural radiance fields [38], however, do not offer real-time framerates, which severely limits their applicability to the aforementioned problems. This limitation is primarily imposed by the choice of implicit neural scene representation and rendering algorithm, namely a volumetric representation that requires a custom neural volume renderer. Neural surface representations, for example using signed distance functions (SDFs) [41, 18, 2, 60], occupancy fields [35], or feature-based representations [54], on the other hand implicitly model the surface of objects, which can be extracted using the marching cubes algorithm [32] and exported into traditional mesh-based representations for real-time rendering. Although implicit neural surface representations have recently demonstrated impressive performance on shape reconstruction [60], their performance on view interpolation and synthesis tasks is limited. Thus, SOTA neural rendering approaches either perform well for view synthesis [38] or 3D shape estimation [60], but not both.

Here, we adopt an SDF-based sinusoidal representation network (SIREN) as the backbone of our neural rendering system. While these representations have recently demonstrated impressive performance on representing shapes via direct 3D supervision with point clouds [52], we are the first to demonstrate how to leverage SIREN's extreme capacity in the context of learning 3D shapes using 2D supervision with images via neural rendering. For this purpose, we devise a novel loss function that maintains SIREN's high-capacity encoding for the supervised images while constraining it in the angular domain to prevent overfitting on these views. This training procedure allows us to robustly fit a SIREN-based SDF directly to a sparse set of multi-view images. Our 2D-supervised implicit neural scene representation and rendering approach performs on par with NeRF on view interpolation tasks while providing a high-quality 3D surface that can be directly exported for real-time rendering at test time.

Specifically, we make the following contributions:

- We develop a neural rendering framework comprising an implicit neural 3D scene representation, a neural renderer, and a custom loss function for training. This approach achieves $10\times$ higher rendering rates than NeRF while providing comparable, SOTA image quality with the additional benefit of optimizing an implicitly defined surface.
- We demonstrate how both shape and view-dependent appearance of our neural scene representation can be exported and rendered in real time using traditional graphics pipelines.
- We also build a custom camera array and capture several datasets of faces and heads for evaluating our approach and baselines. These data are available on the project website.[1]

2. Related Work

Traditional 3D computer vision pipelines use structure-from-motion and multi-view-stereo algorithms to estimate sparse point clouds, camera poses, and textured meshes from 2D input views (e.g., [55, 49, 7, 47, 48]). Re-rendering these scene representations, however, does not achieve photorealistic image quality. As an alternative, image-based rendering techniques have been explored for decades [50]. Lumigraph rendering [17, 4] stands out among these methods as an approach that leverages proxy scene geometry to interpolate the captured views better. Still, these traditional approaches have not demonstrated photorealistic view synthesis for general 3D scenes.

Emerging neural scene representations often model an object or scene explicitly using some 3D proxy geometry, such as an imperfect mesh [21, 57, 45, 62] or depth map [44] estimated by multi-view stereo or other means, an object-specific shape template [25], a multi-plane [63, 37, 11] or multi-sphere [3, 1] image, or a volume [53, 31]. An overview of recent neural rendering techniques, including extensive discussions of explicit representations, is provided by Tewari et al. [56].

As opposed to explicit representations, emerging neural implicit scene representations promise 3D-structure-aware, continuous, memory-efficient representations for shape parts [15, 14], objects [41, 36, 2, 18, 60, 8, 5], or scenes [10, 54, 23, 43, 52]. These representations implicitly define an object or a scene using a neural network and can be supervised directly with 3D data, such as point clouds, or with 2D multi-view images [46, 54, 40, 39, 38, 60, 29, 24, 30, 27]. It is important to distinguish between neural implicit representations that use implicitly defined volumes [38, 29, 28] from those using implicitly defined surfaces, for example represented as signed distance functions (SDFs) [41, 36, 2, 18, 54, 23, 43] or occupancy networks [35, 6, 39]. Surface-based representations allow for traditional mesh representations to be extracted and rendered efficiently with traditional computer graphics pipelines.

The neural rendering methods closest to our work are neural radiance fields (NeRF) [38], which provide the best image quality for view synthesis to date but do not directly model object shape, and implicit differentiable renderer (IDR) [60], which recently demonstrated SOTA performance for shape estimation but which does not achieve the same quality as NeRF for view synthesis. Our work leverages emerging sinusoidal representation networks (SIREN) [52] to achieve both of these capabilities simultaneously. This is crucial for the neural rendering pipeline we propose, which first learns a neural implicit surface representation and then exports it into a format that is compatible with existing real-time graphics pipelines. Although SIRENs have been proposed in prior work [52], we are the first to demonstrate how to leverage their impressive capacity with 2D multi-view image supervision – a non-trivial task due to SIREN's extreme overfitting behavior.

3. Neural Rendering Pipeline

In this section, we describe our differentiable neural rendering pipeline, which is illustrated in Fig. 1.

3.1. Representation

We represent both shape and appearance of 3D objects using implicit functions in a framework similar to IDR [60]. Unlike previous work, however, our network architecture builds on sinusoidal representation networks (SIREN) [52], which allow us to represent signals of significantly higher complexity within the same number of learnable parame-

[1] http://www.computationalimaging.org/publications/nlr/

ters compared with common non-periodic multilayer perceptrons (MLP).

We express the continuous shapes of a scene as the zero-level set $S_0 = \{\mathbf{x} \mid S(\mathbf{x}) = 0\}$ of a signed distance function (SDF)

$$S(\mathbf{x};\theta) : \mathbb{R}^3 \to \mathbb{R}, \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^3$ is a location in 3D space and θ are the learnable parameters of our SIREN-based SDF representation.

Next, we model appearance as a spatially varying emission function, or radiance field, E for directions $\mathbf{r_d} \in \mathbb{R}^3$ defined in a global coordinate system. This formulation does not allow for relighting but it enables photorealistic reconstruction of the appearance of a scene under fixed lighting conditions. We leave the problem of modeling lighting and shading as an avenue of future work.

We additionally condition E by the local normal direction $\mathbf{n} = \nabla_{\mathbf{x}} S(\mathbf{x})$ as computed by automatic differentiation. This does not constrain any degrees of freedom but it has been shown to improve the training performance [60]. Finally, we also reuse θ to increase the network capacity and allow for modeling of fine spatial details and micro-reflections that are of a notably higher spatial complexity than the underlying shape. Together, we express the radiance field as

$$E(\mathbf{x}, \mathbf{r_d}, \mathbf{n}; \theta, \phi) : \mathbb{R}^9 \to \mathbb{R}^3, \tag{2}$$

to represent RGB appearance using the additional learnable parameters ϕ.

3.2. Neural Rendering

The goal of neural rendering is to project a 3D neural scene representation into one or multiple 2D images. We solve this task in two steps: 1) We find the 3D surface as the zero-level set S_0 closest to the camera origin along each ray; 2) We resolve the appearance by sampling the local radiance E.

To address 1), we sphere trace the SDF to find S_0 [20]. For this purpose, we define a view and a projection matrix, $V \in \mathbb{R}^{4\times4}$ and $P \in \mathbb{R}^{4\times4}$, similar to OpenGL's rendering API [59]. A ray origin $\mathbf{r_o}$ and direction $\mathbf{r_d}$ for an output pixel at relative projection plane location $\mathbf{u} \in [-1,1]^2$ is then

$$\mathbf{r_o} = (V^{-1} \cdot [0,0,1,0]^T)_{x,y,z}, \tag{3}$$

$$\mathbf{r_d} = \nu\left((P \cdot V)^{-1} \cdot [\mathbf{u}_x, \mathbf{u}_y, 0, 1]^T\right), \tag{4}$$

where $(\cdot)_{x,y,z}$ are vector components and $\nu(\omega) = \omega_{x,y,z}/||\omega_{x,y,z}||$ is vector normalization.

The sphere-tracing algorithm minimizes $|S(\mathbf{x},\theta)|$ along each ray using iterative updates of the form

$$\mathbf{x}_0 = \mathbf{r_o}, \quad \mathbf{x}_{i+1} = \mathbf{x}_i + S(\mathbf{x}_i)\mathbf{r_d}. \tag{5}$$

Finally, $S_0 = \{\mathbf{x}_n \mid S(\mathbf{x}_n) = 0\}$ is the zero-set of rays converged to a foreground object for the step count $n = 16$. A small residual $|S(\mathbf{x}_n)| < 0.005$ is tolerated in practice. As proposed in recent work [60, 24, 30], we only retain gradients in the last step rather than for all steps of the sphere tracer. This approach makes sphere tracing memory efficient. Please refer to the supplemental materials for additional details.

The appearance is directly sampled from our radiance field as $E(S_0, \mathbf{r_d}, \nabla S(S_0); \theta, \phi)$.

3.3. Loss Function

We supervise our 3D representation using a set of m multi-view 2D images $\mathbf{I} = \mathbb{R}^{m\times w\times h\times 3}$ with known object masks $\mathbf{M} = \mathbb{R}^{m\times w\times h}$ where 1 marks foreground. Our unique approach to leveraging SIREN as a neural representation in this setting is challenging, because of SIREN's tendency to overfit the signal to the supervised views.

In total, we use four different constraints to optimize the end-to-end representation using mini-batches of image pixels $\mathbf{U}$ with RGB values $\mathbf{I_U}$ and object masks $\mathbf{M_U}$.

First, we minimize an $L1$ image reconstruction error for the true foreground pixels $\mathbf{U}_f = \mathbf{U} \cap S_0 \cap \{\mathbf{U} \mid \mathbf{M_U} = 1\}$ as

$$\mathcal{L}_R = \frac{1}{|\mathbf{U}|} \sum_{\mathbf{c} \in \mathbf{I}_{\mathbf{U}_f}} |E(\mathbf{x}, \mathbf{r_d}, \mathbf{n}; \theta, \phi) - \mathbf{c}|, \tag{6}$$

where $\mathbf{c}$ is an RGB value of a foreground pixel in a mini-batch. Both $L1$ and $L2$ work well but we have found $L1$ to produce marginally sharper images.

Second, we regularize the S by an eikonal constraint

$$\mathcal{L}_E = \frac{1}{|\mathbf{U}|} \sum_{\mathbf{x}_r} \left\|(\|\nabla_{\mathbf{x}} S(\mathbf{x}_r;\theta)\|_2 - 1)\right\|_2^2 \tag{7}$$

to enforce its metric properties important for efficient sphere tracing [19, 24, 52, 60]. Random points $\mathbf{x}_r$ are uniformly sampled from a cube which encapsulates the object's bounding unit radius sphere.

Third, we restrict the coarse shape by enforcing its projected pattern to fall within the boundaries of the object masks. For this purpose, we adopt the soft mask loss proposed in [60] defined for the non-foreground pixels and softness parameter α as

$$\mathcal{L}_M = \frac{1}{\alpha|\mathbf{U}|} \sum_{m \in \mathbf{M}_{\mathbf{U}\setminus\mathbf{U}_f}} \mathrm{BCE}(\mathrm{sigmoid}(-\alpha S_{min}), m), \tag{8}$$

where BCE is the binary cross entropy and $S_{min} = \arg\min_t S(\mathbf{r_o} + t\mathbf{r_d}; \theta)$ is the minimum S value along the entire ray approximated by dense sampling of t.

Finally, we regularize the radiance field E to avoid overfitting to training views. SIRENs have a remarkable regressive potential, which biases them to overfit the appearance to the training views. We leverage this power to allow for encoding of photorealistic surface details, but we need to restrict the behavior of the E in the angular domain conditioned by $\mathbf{r_d}$ to achieve favorable interpolation behavior. Inspired by multi-view projective texture mapping [9], we linearize the angular behavior using a smoothness term

$$\mathcal{L}_S = \frac{1}{|\mathbf{U}|}\sum ||\nabla^2_{\mathbf{r_d}} E(\mathbf{x}, \mathbf{r_d}, \mathbf{n}; \theta, \phi)||_2^2. \tag{9}$$

Note that such level of control is unique to SIREN and related architectures as they are C^∞ differentiable.

Together, we optimize parameters θ and ϕ as

$$\arg\min_{\theta,\phi} \mathcal{L}_R + w_E\mathcal{L}_E + w_M\mathcal{L}_M + w_S\mathcal{L}_S, \tag{10}$$

with weights $w_E = 0.1$, $w_M = 100$, and $w_S = 0.01$ for all of our experiments. We have not found the performance to be very sensitive to this choice with the exception of w_S where large values cause high-frequency artifacts in S.

3.4. Additional Training Details

We optimize the loss in mini-batches of 50,000 individual rays sampled uniformly across the entire training dataset. We have found a large batch size and uniform ray distribution to be critical to prevent local overfitting of SIREN, especially for the high-frequency function E.

We implement the MLPs representing S and E as SIRENs with 5 layers using 256 hidden units each. Additionally, we use Fourier features $\{\sin(2k\pi\mathbf{r_d}), \cos(2k\pi\mathbf{r_d}) \,|\, k \in 1\dots4\}$ in E to further support angular resolution [38, 60]. This strategy is necessary to fit the sparsely supervised rays well while $\mathcal{L}_S$ enhances interpolation between them.

We initialize S to a unit sphere of radius 0.5 by pretraining to a procedural shape as described in [52]. We trace the object rays in a larger sphere of radius 1, but we have found that the smaller initial radius improves the initial fit as well as the consequent convergence rate.

We implement our method in PyTorch [42] and optimize the loss using the Adam solver [26] with an initial learning rate of 10^{-4} decreased by a factor of 2 every 40,000 batches for the overall training length of 150,000 batches on a single Nvidia GPU RTX 2080Ti.

4. Real-time Rendering Pipeline

While we show that SIREN is remarkably efficient in shape and appearance representation with 2D supervision, the required sphere tracer does not run at real-time rates for moderate to high image resolutions. To overcome this challenge, we show how to leverage the compactness of our surface-based representation and convert our neural model to a triangular mesh suitable for real-time computer graphics applications. For this purpose, we leverage unstructured lumigraph rendering, which preserves view-dependent effects learned by our neural representation [4].

4.1. Mesh extraction

First, we use the marching cubes algorithm [33] to extract a high-resolution surface mesh from the SDF S voxelized at a resolution of 512^3. Instead of extracting the zero-level set, we found that offsetting the iso-surface of S by 0.5% of the object radius in the outside direction optimizes the resulting image quality in practice. To export the appearance, we resample the optimized emissivity function E to synthesize projective textures $\mathbf{T}_i$ for N camera poses and corresponding projection matrices. The ability to resample the camera poses for efficient viewing space coverage is a key feature of our method and we explore the choice of N and camera distributions in the supplement.

4.2. Rendering

First, we rasterize the extracted mesh using OpenGL [59] and project the vertex positions to each pixel. Next, we compute angles $\tau_{1\dots N}$ between the ray towards the current rendering camera and the rays towards each of the N projective texture map viewpoints. We then apply the unstructured lumigraph rendering technique of Buehler et al. [4] to blend contributions from the first $k = 5$ textures, sorted by τ_i in ascending order, yielding a rendered image

$$\mathbf{R} = \textstyle\sum_{i=1\dots k} w_i \mathbf{T}_i, \tag{11}$$

where the weights w_i are computed as

$$\hat{w}_i = 1/\tau_i(1 - \tau_i/\tau_k), \tag{12}$$

$$w_i = \hat{w}_i / \textstyle\sum_{i=1\dots k} \hat{w}_i. \tag{13}$$

This formulation satisfies the epipolar consistency by converging to an exclusive mapping by texture $\mathbf{T}_j$ when $\tau_j \rightarrow 0$ [4]. Additionally, we discard samples from occluded textures by setting their w_i to zero. Occlusions are detected by a comparison between the pre-rendered depth associated with a texture and the distance between the mesh voxel and the texture viewpoint. The same technique is commonly used in real-time graphics for shadow mapping.

4.3. Evaluation

Efficiency We compare the efficiency of our real-time rasterized neural lumigraph renderer (NLR-RAS) with our sphere-traced renderer (NLR-ST, Sec. 3) along with other baselines in Table 1. We observe, that although both NLR-ST and IDR are based on sphere tracing, the capacity of SIREN allows for a smaller and faster model, which is evident by the model size. Furthermore, the results show how costly the implicit volumetric rendering is. In conclusion, only the explicit representations of Colmap and our NLR-RAS allow for truly real-time performance with framerates over 60 fps at HD resolution on commodity hardware.

Method	Render time [s]	Model size [MB]
Colmap [48]	**Real-time**	30.39
IDR [60]	45	11.13
NV [31]	0.65	438.36
NeRF [37]	150	2.27
NLR-ST	13	**2.07**
NLR-RAS	**Real-time**	34.68

Table 1. Rendering time and representation size comparison for the DTU scan 65 [22] at 1600 $\times$ 1200 pixel resolution. "Real-time" denotes framerates of at least 60 fps.

Figure 2. Our custom camera array comprising 16 GoPro HERO7 and 6 central Back-Bone H7PRO cameras (large circular lenses).

Image quality Both the quantitative comparisons in Tables 2, 3 and qualitative examples in Figures 3, 4 demonstrate the high NLR-RAS rendering quality. While lower than that of our NLR-ST renderer, the NLR-RAS still achieves PSNRs far superior to other explicit (Colmap) and implicit (IDR) surface representations.

5. Camera Array and Data

Human Head Video Dataset Our dataset consists of 7 multiview captures showing a person performing facial expressions.

Camera Array Our custom camera array that was used to capture the dataset is comprised of 16 GoPro HERO7 action cameras and 6 Back-Bone H7PRO cameras. The Back-Bone cameras are modified GoPro cameras that can fit a standard C-Mount lens. Compared to the unmodified GoPro cameras, the Back-Bone cameras have a narrower field-of-view (FoV) and are thus able to capture the subject in more detail. We capture at 4k / 30 fps in portrait orientation with the Back-Bone cameras and at 1080p / 60 fps in landscape orientation with the GoPro cameras. Figure 2 shows a frontal view of the camera array with the six Back-Bone cameras in the center of the array and the GoPro cameras placed around them. We capture our subjects from 60 cm distance and cover approximately 100°.

Synchronization We trigger the camera shutter with a WiFi remote. The cameras do not support a generator lock, so during capture they are only loosely synchronized. We always capture videos for our dataset, even in the cases in which we only use a static frame. To improve synchronization, we flash an ArUco marker [13] on a cellphone before each take. We then detect the first frame that sees the marker in each video which allows us to synchronize the cameras with an accuracy of 1 frame or better.

6. Experiments

In this section we show that our method is able to achieve state-of-the-art image reconstruction quality on-par with volumetric methods such as NeRF [38] while allowing for efficient surface reconstruction utilized for real-time rendering in Sec. 4.

Baselines We compare our method to novel view synthesis techniques with various scene representations. Specifically, we compare to the traditional multi-view stereo of Colmap [48], the explicit volumetric representation of Neural Volumes (NV) [31], the implicit volume representation of NeRF [38], and the implicit signed distance function of IDR [60]. Please refer to the supplemental material for implementation details.

Regression performance We have used the popular DTU MVS dataset [22] with 49 or 64 calibrated camera images along with object masks provided by previous work [60, 39] to measure the image reconstruction error metrics. We held out three views for testing. Table 2 shows that our method achieves SOTA training error comparable with NeRF. Moreover, our image quality is significantly better than that of our closest competitor, IDR. We attribute this major separation to the unparalleled representation capacity of SIRENs. A qualitative comparison is available in Fig. 3.

Additionally, we report the shape reconstruction error as Chamfer distance from the ground-truth provided in the dataset. Although the shape reconstruction is not our explicit goal, we note that our error is on par with other techniques, though worse than IDR which explicitly focuses on this problem [60]. We observe that this emerges as a trade-off between the accuracy of view-dependent and high-frequency details in the image reconstruction on one hand, and the view consistency reflected in the geometry on the other one.

View interpolation Our angular smoothness loss $\mathcal{L}_S$ is specifically designed to avoid collapse of the emissivity function E for interpolated views. We tested its efficiency quantitatively by measuring the image reconstruction error on test views the held out from results in Table 2. Please refer to the supplement for details. As expected, there is a measurable quality drop when compared to the training views observed consistently for all of the methods. However, the interpolated views produced by our method maintain many of the favorable characteristics from the regres-

Scan	Colmap [48]				IDR [60]				NeRF [37]				NLR-ST				NLR-RAS		
	CD↓	PSNR↑	SSIM↑	LPIPS↓	CD↓	PSNR↑	SSIM↑	LPIPS↓	CD↓	PSNR↑	SSIM↑	LPIPS↓	CD↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
65	2.42	15.25	0.862	0.143	**0.70**	23.87	0.948	0.094	**1.02**	**33.57**	**0.962**	**0.055**	1.03	32.13	0.961	0.063	31.46	0.960	0.066
97	**0.72**	11.93	0.843	0.154	1.09	23.02	0.921	0.117	1.43	28.28	0.933	0.089	**1.16**	**28.48**	**0.939**	**0.088**	28.37	0.939	0.089
106	0.78	15.75	0.887	0.141	**0.58**	20.97	0.907	0.183	**0.84**	**33.50**	0.947	0.092	0.82	32.98	**0.951**	**0.083**	31.32	0.951	0.090
118	**0.44**	22.73	0.921	0.091	0.50	22.62	0.944	0.139	**0.88**	**35.62**	**0.966**	**0.070**	1.71	34.87	0.964	0.075	33.33	0.963	0.078

Table 2. Image error metrics PNSR, SSIM [58] and LPIPS [61] computed on DTU [22] for the supervised views. See Supp. for metrics of the held-out views. The Chamfer distance (CD) is computed based on the scripts from [22]. Best scores in bold, second best underlined.

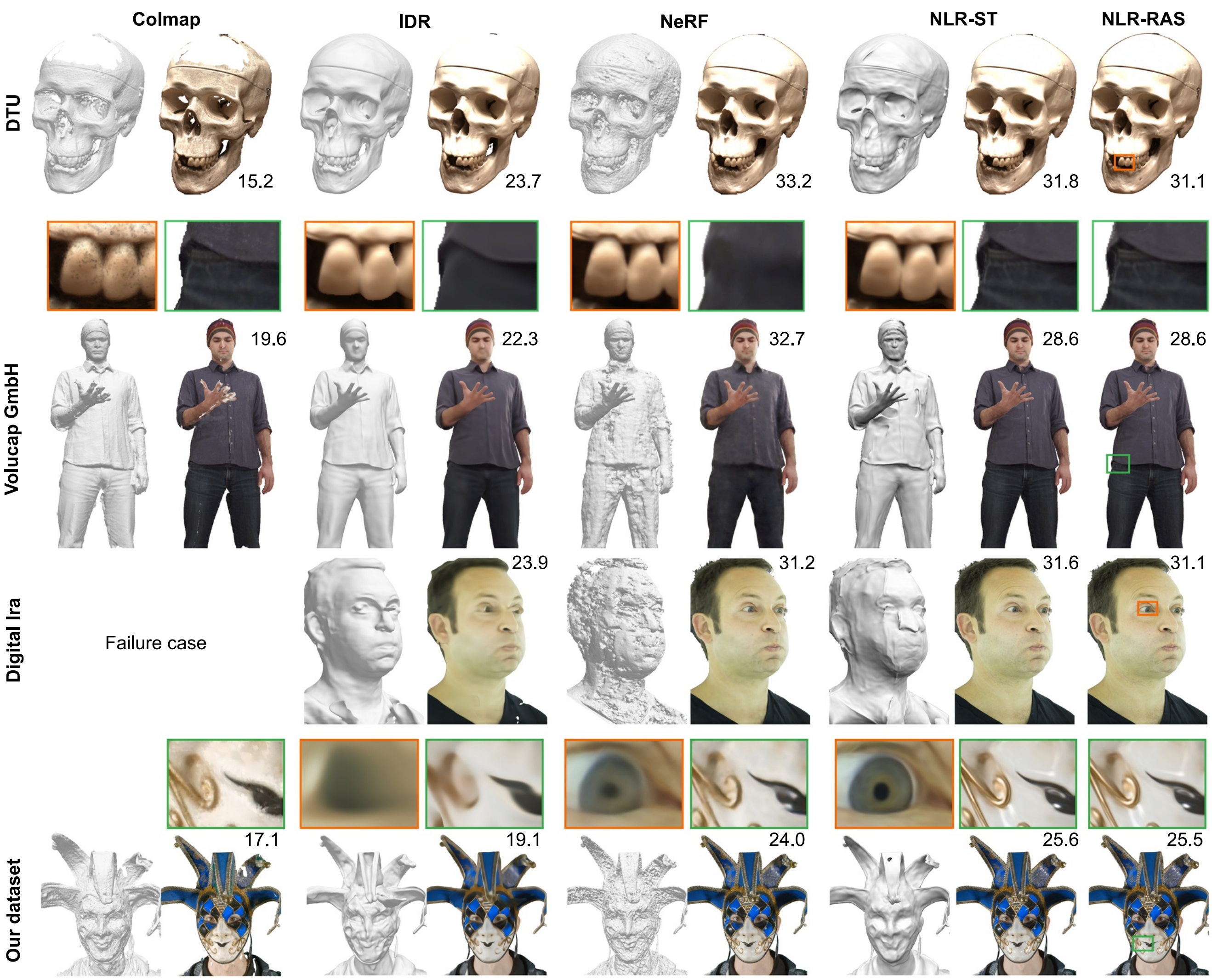


Figure 3. Reconstructed shape and image from various multi-view datasets. The reprojection error is listed in dB of PSNR.

sion case. We provide a qualitative comparison of both supervised and interpolated views in Fig. 4 and in our video.

Human representation View-synthesis of human subjects is particularly challenging due to the complex reflection properties of skin, eyes and hair, as well as a lack of high-quality multi-view data. We address the first challenge with our high-capacity representation network and the latter with our own dataset described in Sec. 5. Additionally, we provide experimental results for 360 degree human captures provided by Volucap GmbH [16] and high-resolution face captures from the Digital Ira project [12]. Refer to the supplemental for a detailed description of these data. Table 3 summarizes the reconstruction errors and Fig. 3 shows a few example scenes. Similar trends as in the DTU datasets can again be observed. Interestingly, our method achieves

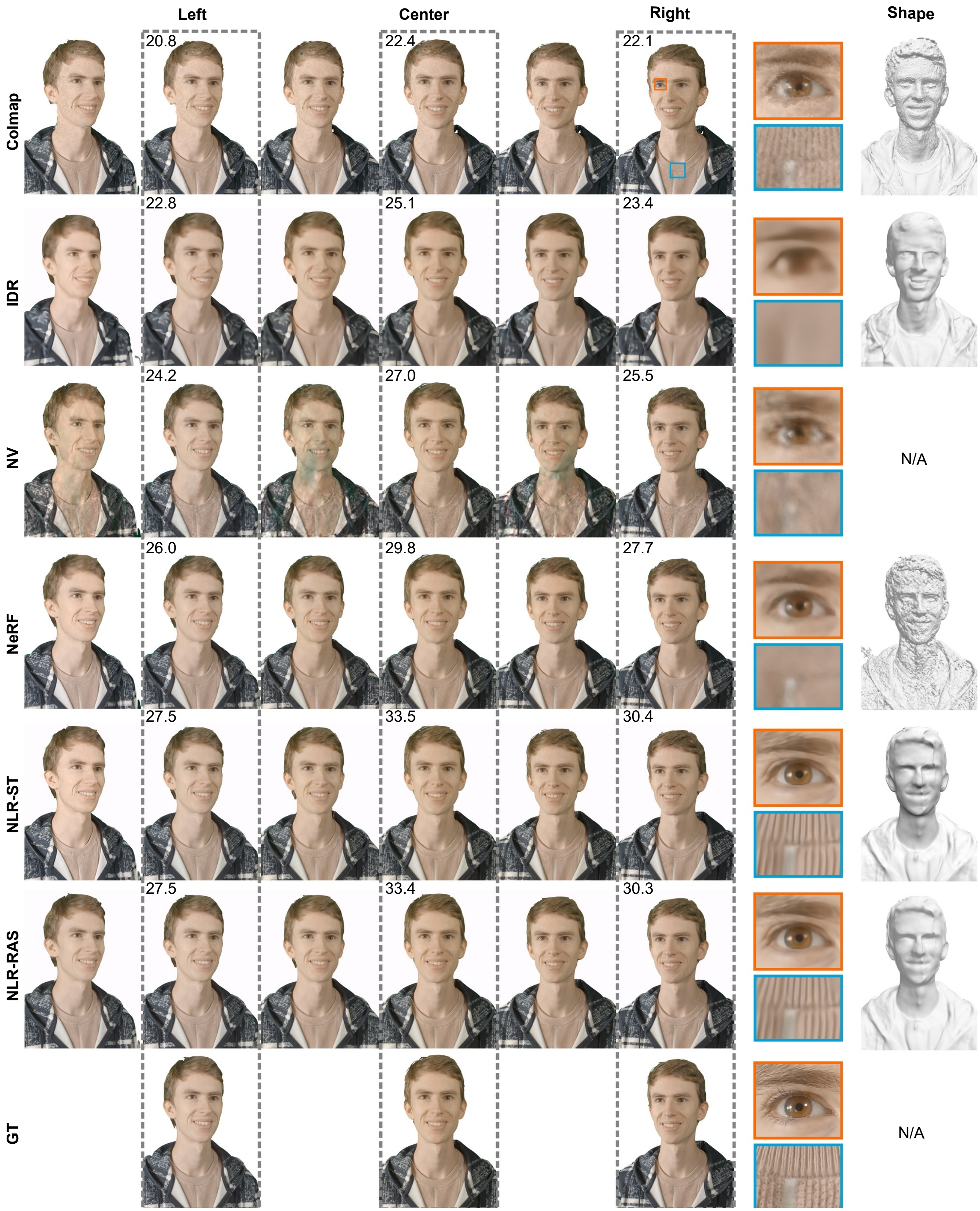
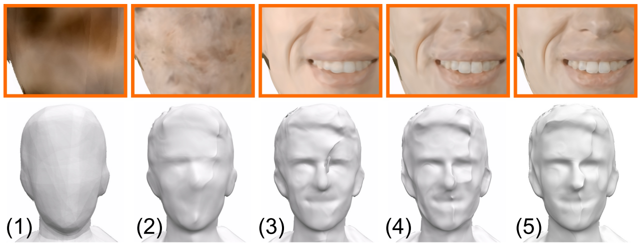


Figure 4. A qualitative comparison for a frame in our dataset. Three supervised views (dashed outline) along with three interpolated views and learned shape. Close-ups enhance detail of the rightmost view. PSNR showed for supervised views.



	Act.	$\mathcal{L}_S$	Batch size	Extra FF	PSNR↑ Train	PSNR↑ Test	LPIPS↓ [61] Train	LPIPS↓ [61] Test
(1)	Relu	No	2K	No	24.99	10.57	0.171	0.285
(2)	Sine	No	2K	No	28.49	19.27	0.161	0.239
(3)	Sine	Yes	2K	No	28.26	24.77	0.151	0.179
(4)	Sine	Yes	50K	No	30.06	25.20	0.143	0.172
(5)	**Sine**	**Yes**	**50K**	**Yes**	**30.75**	**26.04**	**0.132**	**0.151**

Figure 5. Learned shapes with interpolated close-ups (left) and reconstruction metrics for each condition (right) in our ablation study.

Dataset	Colmap	IDR	NV	NeRF	NLR-ST	N.-RAS
Volucap (1)	19.6/.037	22.3/.043	29.3/.034	**32.7**/.026	28.6/**.022**	28.6/**.022**
Dig. Ira (1)	Fail.	23.9/.286	26.5/.287	31.2/.267	**31.6/.255**	31.1/.260
Ours (7)	17.6/.187	22.3/.202	*25.1/.186*	28.5/.171	**30.5/.147**	30.3/.151

Table 3. Average reconstruction PSNR/LPIPS [61] scores computed across datasets (number of scenes in parentheses). *Italic:* Only 5 scenes tested. See Supplement for an extended version.

a bigger advantage for very high-resolution (3000×4000 px) detailed images in our own dataset. We speculate that this shows that the traditional ReLU based networks used by IDR and NeRF have reached their capacity, while the explicit representations of Colmap and NV lack easy scaling. Once again, this does not come at cost of interpolation properties as shown in Figure 4 and our videos.

Ablation study Finally, we verify that the performance of our method is based on the choice of our representation and training procedure. In Figure and Table 5, we compare several variants of our method on the scene in Fig. 4. A standard MLP with ReLU does not have the capacity to train a detailed representation (1). SIREN remedies this but tends to quickly overfit to the trained pixels (2). We resolve this first by adding our angular smoothness loss $\mathcal{L}_S$ that regularizes behavior in the angular domain (3), and then by increasing the batch size in order to achieve spatially uniform image quality (4). Additional Fourier Features [38] for the ray direction remove low frequency noise in E (5).

7. Discussion

In summary, we propose a neural rendering framework that optimizes an SDF-based implicit neural scene representation given a set of multi-view images. This framework is unique in combining a representation network architecture using periodic activations with a sphere-tracing-based neural renderer that estimates the shape and view-dependent appearance of the scene. Enabled by a novel loss function that is applied during training, our framework achieves a very high image quality that is comparable with state-of-the-art novel view synthesis methods. As opposed to those methods, our neural representation can be directly converted into a mesh with view-dependent textures that enable high-quality 3D image synthesis in real time using traditional graphics pipelines.

Our approach is not without limitations. Currently, we only consider emissive radiance functions that are adequate to model a scene under fixed lighting conditions. Future work could additionally consider dynamic lighting and shading, which some recent neural rendering approaches have started to incorporate [34, 62]. Further, similar to IDR [60], our method requires annotated object masks. Automatic image segmentation could be explored in the future to address this. Although the synthesized image quality of our approach is competitive with the state of the art, the proxy shapes produced by our method are not quite as accurate as alternative approaches [48, 60]. While this is not important for the novel view synthesis application we consider in this paper, other applications may benefit from estimating more accurate shapes. This includes occasionally visible seam artifacts caused by inaccuracies of the camera calibration. Similar to some other recent neural rendering pipelines, ours focuses on overfitting a neural representation on a single 3D scene. An interesting avenue of future work includes the learning of shape spaces, or priors, for certain types of objects, such as faces. While several methods have explored related strategies using conditioning-by-concatenation [41, 35], hypernetwork [54], or meta-learning [51] approaches using synthetic data, there is a lack of publicly available photorealistic multi-view image data. In hope of mitigating this shortcoming, we released our datasets on the project website. Still, these data may be insufficiently large for learning priors. Finally, although the inference time of our method is fast, the training time is still slow. This hurdle along with the limited computational resources at our disposal is the primary reason preventing us from exploring dynamic video sequences.

Conclusion Emerging neural rendering approaches are starting to outperform traditional vision and graphics approaches. Yet, traditional graphics pipelines still offer significant practical benefits, such as real-time rendering rates, over these neural approaches. With our work, we take a significant step towards closing this gap, which we believe to be a critical aspect for making neural rendering practical.

References

[1] Benjamin Attal, Selena Ling, Aaron Gokaslan, Christian Richardt, and James Tompkin. MatryODShka: Real-time 6DoF video view synthesis using multi-sphere images. In *Proc. ECCV*, Aug. 2020. 2

[2] Matan Atzmon and Yaron Lipman. Sal: Sign agnostic learning of shapes from raw data. In *Proc. CVPR*, 2020. 1, 2

[3] Michael Broxton, John Flynn, Ryan Overbeck, Daniel Erickson, Peter Hedman, Matthew Duvall, Jason Dourgarian, Jay Busch, Matt Whalen, and Paul Debevec. Immersive light field video with a layered mesh representation. *ACM Trans. Graph. (SIGGRAPH)*, 39(4), 2020. 2

[4] Chris Buehler, Michael Bosse, Leonard McMillan, Steven Gortler, and Michael Cohen. Unstructured lumigraph rendering. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 425–432, 2001. 2, 4

[5] Rohan Chabra, Jan Eric Lenssen, Eddy Ilg, Tanner Schmidt, Julian Straub, Steven Lovegrove, and Richard Newcombe. Deep local shapes: Learning local sdf priors for detailed 3d reconstruction. *arXiv preprint arXiv:2003.10983*, 2020. 2

[6] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proc. CVPR*, pages 5939–5948, 2019. 2

[7] Alvaro Collet, Ming Chuang, Pat Sweeney, Don Gillett, Dennis Evseev, David Calabrese, Hugues Hoppe, Adam Kirk, and Steve Sullivan. High-quality streamable free-viewpoint video. *ACM Trans. Graph. (SIGGRAPH)*, 34(4), 2015. 2

[8] Thomas Davies, Derek Nowrouzezahrai, and Alec Jacobson. Overfit neural networks as a compact shape representation, 2020. 2

[9] Paul Debevec, Yizhou Yu, and George Borshukov. Efficient view-dependent image-based rendering with projective texture-mapping. In *Eurographics Workshop on Rendering Techniques*, pages 105–116. Springer, 1998. 4

[10] SM Ali Eslami, Danilo Jimenez Rezende, Frederic Besse, Fabio Viola, Ari S Morcos, Marta Garnelo, Avraham Ruderman, Andrei A Rusu, Ivo Danihelka, Karol Gregor, et al. Neural scene representation and rendering. *Science*, 360(6394):1204–1210, 2018. 2

[11] John Flynn, Michael Broxton, Paul Debevec, Matthew DuVall, Graham Fyffe, Ryan Overbeck, Noah Snavely, and Richard Tucker. Deepview: View synthesis with learned gradient descent. In *Proc. CVPR*, pages 2367–2376, 2019. 2

[12] Graham Fyffe, Andrew Jones, Oleg Alexander, Ryosuke Ichikari, and Paul Debevec. Driving high-resolution facial scans with video performance capture. *ACM Trans. Graph.*, 34(1), Dec. 2015. 6

[13] Sergio Garrido-Jurado, Rafael Muñoz-Salinas, Francisco José Madrid-Cuevas, and Manuel Jesús Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6):2280–2292, 2014. 5

[14] Kyle Genova, Forrester Cole, Avneesh Sud, Aaron Sarna, and Thomas Funkhouser. Local deep implicit functions for 3d shape. In *Proc. CVPR*, 2020. 2

[15] Kyle Genova, Forrester Cole, Daniel Vlasic, Aaron Sarna, William T Freeman, and Thomas Funkhouser. Learning shape templates with structured implicit functions. In *Proc. ICCV*, pages 7154–7164, 2019. 2

[16] Volucap GmbH. A multiview capture data sample, 2020. 6

[17] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. The lumigraph. In *ACM SIGGRAPH*, page 43–54, 1996. 2

[18] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. In *Proc. ICML*, 2020. 1, 2

[19] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. In *Proceedings of Machine Learning and Systems 2020*, pages 3569–3579. 2020. 3

[20] John C Hart. Sphere tracing: A geometric method for the antialiased ray tracing of implicit surfaces. *The Visual Computer*, 12(10):527–545, 1996. 3

[21] Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. Deep blending for free-viewpoint image-based rendering. *ACM Transactions on Graphics (TOG)*, 37(6):1–15, 2018. 2

[22] 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. 5, 6

[23] Chiyu Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, and Thomas Funkhouser. Local implicit grid representations for 3d scenes. In *Proc. CVPR*, pages 6001–6010, 2020. 2

[24] Yue Jiang, Dantong Ji, Zhizhong Han, and Matthias Zwicker. Sdfdiff: Differentiable rendering of signed distance fields for 3d shape optimization. In *Proc. CVPR*, 2020. 2, 3

[25] Angjoo Kanazawa, Shubham Tulsiani, Alexei A Efros, and Jitendra Malik. Learning category-specific mesh reconstruction from image collections. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 371–386, 2018. 2

[26] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4

[27] Amit Kohli, Vincent Sitzmann, and Gordon Wetzstein. Semantic implicit neural scene representations with semi-supervised training. In *Proc. 3DV*, 2020. 2

[28] David B Lindell, Julien NP Martel, and Gordon Wetzstein. Autoint: Automatic integration for fast neural volume rendering. In *Proc. CVPR*, 2021. 2

[29] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *NeurIPS*, 2020. 2

[30] Shaohui Liu, Yinda Zhang, Songyou Peng, Boxin Shi, Marc Pollefeys, and Zhaopeng Cui. Dist: Rendering deep implicit signed distance function with differentiable sphere tracing. In *Proc. CVPR*, 2020. 2, 3

[31] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *ACM Trans. Graph. (SIGGRAPH)*, 38(4), 2019. 2, 5

[32] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *ACM SIGGRAPH*, page 163–169, 1987. 1

[33] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM siggraph computer graphics*, 21(4):163–169, 1987. 4

[34] Abhimitra Meka, Christian Häne, Rohit Pandey, Michael Zollhöfer, Sean Fanello, Graham Fyffe, Adarsh Kowdle, Xueming Yu, Jay Busch, Jason Dourgarian, Peter Denny, Sofien Bouaziz, Peter Lincoln, Matt Whalen, Geoff Harvey, Jonathan Taylor, Shahram Izadi, Andrea Tagliasacchi, Paul Debevec, Christian Theobalt, Julien Valentin, and Christoph Rhemann. Deep reflectance fields: High-quality facial reflectance field inference from color gradient illumination. *ACM Trans. Graph. (SIGGRAPH)*, 38(4), 2019. 8

[35] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proc. CVPR*, 2019. 1, 2, 8

[36] Mateusz Michalkiewicz, Jhony K Pontes, Dominic Jack, Mahsa Baktashmotlagh, and Anders Eriksson. Implicit surface representations as layers in neural networks. In *Proc. ICCV*, pages 4743–4752, 2019. 2

[37] Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Trans. Graph. (SIGGRAPH)*, 38(4), 2019. 2, 5, 6

[38] 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 *Proc. ECCV*, 2020. 1, 2, 4, 5, 8

[39] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proc. CVPR*, 2020. 2, 5

[40] Michael Oechsle, Lars Mescheder, Michael Niemeyer, Thilo Strauss, and Andreas Geiger. Texture fields: Learning texture representations in function space. In *Proc. ICCV*, 2019. 2

[41] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proc. CVPR*, 2019. 1, 2, 8

[42] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 4

[43] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *Proc. ECCV*, 2020. 2

[44] Eric Penner and Li Zhang. Soft 3d reconstruction for view synthesis. *ACM Trans. Graph.*, 36(6), Nov. 2017. 2

[45] Gernot Riegler and Vladlen Koltun. Free view synthesis. In *Proc. ECCV*, 2020. 2

[46] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Proc. ICCV*, pages 2304–2314, 2019. 2

[47] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4104–4113, 2016. 2

[48] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016. 2, 5, 6, 8

[49] Steven M. Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Proc. CVPR*, page 519–528, 2006. 2

[50] Harry Shum and Sing Bing Kang. Review of image-based rendering techniques. In *Visual Communications and Image Processing 2000*, volume 4067, pages 2–13. International Society for Optics and Photonics, 2000. 2

[51] Vincent Sitzmann, Eric R. Chan, Richard Tucker, Noah Snavely, and Gordon Wetzstein. Metasdf: Meta-learning signed distance functions. In *NeurIPS*, 2020. 8

[52] Vincent Sitzmann, Julien N.P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *Proc. NeurIPS*, 2020. 2, 3, 4

[53] Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhöfer. Deepvoxels: Learning persistent 3d feature embeddings. In *Proc. CVPR*, 2019. 2

[54] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *Proc. NeurIPS 2019*, 2019. 1, 2, 8

[55] Richard Szeliski. *Computer vision: algorithms and applications*. Springer, 2011 edition, 2010. 2

[56] Ayush Tewari, Ohad Fried, Justus Thies, Vincent Sitzmann, Stephen Lombardi, Kalyan Sunkavalli, Ricardo Martin-Brualla, Tomas Simon, Jason Saragih, Matthias Nießner, et al. State of the art on neural rendering. *Proc. Eurographics*, 2020. 1, 2

[57] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: Image synthesis using neural textures. *ACM Transactions on Graphics (TOG)*, 38(4):1–12, 2019. 2

[58] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 6

[59] Mason Woo, Jackie Neider, Tom Davis, and Dave Shreiner. *OpenGL programming guide: the official guide to learning OpenGL, version 1.2*. Addison-Wesley Longman Publishing Co., Inc., 1999. 3, 4

[60] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Ronen Basri, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. In *Proc. NeurIPS*, 2020. 1, 2, 3, 4, 5, 6, 8

[61] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, pages 586–595, 2018. 6, 8

[62] Xiuming Zhang, Sean Fanello, Yun-Ta Tsai, Tiancheng Sun, Tianfan Xue, Rohit Pandey, Sergio Orts-Escolano, Philip Davidson, Christoph Rhemann, Paul Debevec, Jonathan T. Barron, Ravi Ramamoorthi, and William T. Freeman. Neural light transport for relighting and view synthesis, 2020. 2, 8

[63] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *ACM Trans. Graph. (SIGGRAPH)*, 37(4), 2018. 2