A generic and flexible regularization framework for NeRFs

Thibaud Ehret Roger Marí Gabriele Facciolo
Université Paris-Saclay, CNRS, ENS Paris-Saclay, Centre Borelli, 91190, Gif-sur-Yvette, France
thibaud.ehret@ens-paris-saclay.fr

Abstract

Neural radiance fields, or NeRF, represent a breakthrough in the field of novel view synthesis and 3D modeling of complex scenes from multi-view image collections. Numerous recent works have shown the importance of making NeRF models more robust, by means of regularization, in order to train with possibly inconsistent and/or very sparse data. In this work, we explore how differential geometry can provide elegant regularization tools for robustly training NeRF-like models, which are modified so as to represent continuous and infinitely differentiable functions. In particular, we present a generic framework for regularizing different types of NeRFs observations to improve the performance in challenging conditions. We also show how the same formalism can also be used to natively encourage the regularity of surfaces by means of Gaussian or mean curvatures.

1. Introduction

Realistic rendering of new views of a 3D scene or a given volume is a long standing problem in computer graphics. The interest in this problem has been rekindled by the growth of augmented and virtual reality. Traditionally, 3D scenes were estimated from a set of images using classic Structure-from-Motion (SfM) and Multi-View Stereo (MVS) tools such as COLMAP [36] or [12, 25, 34, 39].

Recently, Mildenhall *et al*. [24] have shown that differentiable volume rendering operations can be plugged into a neural network to learn a neural radiance field (NeRF) volumetric representation of a scene encoding its geometry and appearance. Starting from a sparse, yet nonetheless large, set of views of the scene, NeRF learns in a self-supervised manner, by maximizing the photo-consistency across the predicted renderings corresponding to the available viewpoints. After convergence, the network is able to render realistic novel views by querying the NeRF function at unseen viewpoints.

This breakthrough led to a very active research field focused on pushing back the limits of the initial models. Notably, Martin-Brualla *et al*. [21] showed that it is possible to learn scenes from unconstrained photos with moving transient objects or different lightning conditions. Other works deal with dynamic or deformable scenes [19, 28–30, 43], complex illumination models [3, 20, 40, 49] or very few training views [7, 16, 18, 47]. In other words, the goal is to make NeRF more robust to be able to train reliably even in the most adverse conditions. For example, imposing regularity constraints on the scene seems to be a promising way to reduce reliance on large datasets [26].

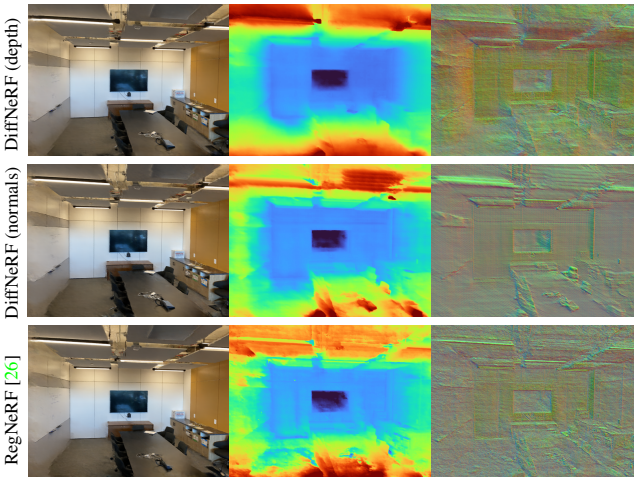

Figure 1. We propose a generic regularization framework for NeRF that outperforms previous state-of-the-art methods when training with only three input views. We compare here the proposed DiffNeRF with depth regularization (top), DiffNeRF with normals regularization (middle) and RegNeRF [26]. Left to right: RGB prediction, depth map, map of normals.

The objective of this work is to show how one can adapt differential geometry concepts to elegantly incorporate regularizers that make NeRF more robust. The advantages are twofold: first, differential geometry is mathematically formalized and its literature is vast with many suitable tools already available and, second, neural representations are perfectly adapted to represent continuous infinitely differentiable volumetric functions in which differential geometry operators are naturally defined.

To this aim, we present a generic framework based on

differential geometry to regularize different types of NeRFs observations. We derive the two specific cases of depth regularization, thus linking to the previously proposed Reg-NeRF [26], as well as normals regularization in Section 3. We also show in Section 4 that this approach is not only competitive with the state of the art for the problem training a NeRF model when few images (for example only three) are available but also that it produces smoother and more accurate depth maps. Finally, we straightforwardly extends the proposed framework to surfaces regularization in Section 5 showing that generality of the proposed approach.

2. Related Work

2.1. Fundamentals of Neural Radiance Fields

NeRF was originally introduced as a continuous volumetric function $\mathcal{F}$, learned by a multi-layer perceptron (MLP), to model the geometry and appearance of a 3D scene [24, 42]. Given a 3D point $\mathbf{x} \in \mathbb{R}^3$ of the scene and a viewing direction $\mathbf{v} \in \mathbb{R}^2$, NeRF predicts an associated RGB color $\mathbf{c} \in [0,1]^3$ and a scalar volume density $\sigma \in [0, \infty)$, *i.e.*

$$\mathcal{F} : (\mathbf{x}, \mathbf{v}) \mapsto (\mathbf{c}, \sigma). \tag{1}$$

The value of σ defines the geometry of the scene and is learned exclusively from the spatial coordinates $\mathbf{x}$, while the value of $\mathbf{c}$ is also dependent on the viewing direction $\mathbf{d}$, which allows to recreate non-Lambertian surface reflectance.

NeRF models are trained based on a classic differentiable volume rendering operation [22], which establishes the resulting color of any ray passing through the scene volume and projected onto a camera system. Each ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{v}$ with $t \in \mathbb{R}^+$, defined by a point of origin $\mathbf{o}$ and a direction vector $\mathbf{v}$, is discretized into N 3D points $\{\mathbf{x}_i\}_{i=1}^N$, where $\mathbf{x}_i = \mathbf{o} + t_i\mathbf{v}$ with t_i sampled between the minimum and maximum depth. The sampling depends on the method. The rendered color $\mathbf{c}(\mathbf{r})$ of a ray $\mathbf{r}$ is obtained as

$$\mathbf{c}(\mathbf{r}) = \sum_{i=1}^{N} T_i \alpha_i \mathbf{c}_i \tag{2}$$

where

$$T_i = \textstyle\prod_{j=1}^{i-1} (1 - \alpha_j) \quad \text{and} \quad \alpha_i = 1 - \exp(-\sigma_i(t_{i+1} - t_i)). \tag{3}$$

In (2), $\mathbf{c}_i$ and σ_i correspond to the color and volume density output by the MLP at the i-th point of the ray, *i.e.* $\mathcal{F}(\mathbf{x}_i, \mathbf{v})$ as per (1). The transmittance T_i and opacity α_i are two factors that weight the contribution of the i-th point to the rendered color according to the geometry described by σ_i and $\sigma_j : j < i$, to handle occlusions.

Using the transmittance T_i and opacity α_i, the observed depth $d(\mathbf{r})$ in the direction of a ray $\mathbf{r}$ can be rendered in a similar manner to (2) [11, 32] as

$$d(\mathbf{r}) = \sum_{i=1}^{N} T_i \alpha_i t_i. \tag{4}$$

Following this volume rendering logic, the NeRF function $\mathcal{F}$ is optimized by minimizing the squared error between the rendered color and the real colors of a batch of rays $\mathcal{R}$ that project onto a set of training views of the scene taken from different viewpoints: $L_{RGB} = \sum_{\mathbf{r} \in \mathcal{R}} \|\mathbf{c}(\mathbf{r}) - \mathbf{c}_{\text{GT}}(\mathbf{r})\|_2^2$, where $\mathbf{c}_{\text{GT}}(\mathbf{r})$ is the observed color of the pixel intersected by the ray $\mathbf{r}$, and $\mathbf{c}(\mathbf{r})$ is the NeRF prediction (2). The rays in each batch $\mathcal{R}$ are chosen randomly from the available views, encouraging gradient flow where rays casted from different viewpoints intersect with consistent scene content.

mip-NeRF: The original NeRF approach casts a single ray per pixel [24]. When the training images observe the scene at different resolutions, this can lead to blurred or aliased renderings. To prevent such situation, the mip-NeRF formulation [2] can be adopted, which casts a cone per pixel instead. As a result, mip-NeRF is queried in terms of conical frustums and not discrete points, yielding a continuous and natively multiscale representation of regions of the volume space.

2.2. Regularization in Few-shot Neural Rendering

The original NeRF methodology was demonstrated using 20 to 62 views for real world static scenes, and 100 views or more for synthetic static scenes [24]. In the absence of large datasets, the MLP usually overfits to each training image when only a few are available, resulting in unrealistic interpolation for novel view synthesis and poor geometry estimates.

A number of NeRF variants have been recently proposed to address few-shot neural rendering. The use of regularization techniques is common in these variants, to achieve smoother results in unobserved areas of the scene volume or radiometrically inconsistent observations [18].

Implicit/indirect regularization methods rely on geometry and appearance priors learned by pre-trained models. PixelNeRF [47] introduced a framework that can be trained across multiple scenes, thus acquiring the ability to generalize to unseen environments. The MLP learns generic operations while keeping the output conditioned to scene-specific content thanks to an additional input feature vector, extracted by a pre-trained convolutional neural network (CNN). Similarly, DietNeRF [16] complements the NeRF loss (2.1) with a secondary term that encourages similarity between pre-trained CNN high-level features in renderings of known and unknown viewpoints. Other approaches like GRAF [37], π-GAN [5], Pix2NeRF [4] or LOLNeRF [31] combine NeRF with generative models: latent codes are mapped to an instance of a radiance field of a certain pre-learned category (e.g. faces, cars), thus providing a reasonable 3D model regardless of the number of available of views.

Explicit/direct regularization methods can be divided into externally supervised and self-supervised. Self-supervised

variants incorporate constraints to enforce smoothness between neighboring samples in space, such as RegNeRF [26] (see Section 3). InfoNeRF [18] prevents inconsistencies due to insufficient viewpoints by minimizing a ray entropy model and the KL-divergence between the normalized ray density obtained from neighbor viewpoints. In contrast, externally supervised regularization methods usually penalize differences with respect to extrinsic geometric cues. Depth-supervised NeRF [11] encourages the rendered depth (4) to be consistent with a sparse set of 3D surface points obtained by structure from motion. A similar strategy is used in [20], based on a set of 3D points refined by bundle adjustment; or [32], where a sparse point cloud is converted into dense depth priors by means of a depth completion network.

3. A generic regularization framework

One of the major challenges when training a NeRF with insufficient data is to learn a consistent scene geometry so that the model extrapolates well on unseen views. In that case, it is common to add additional priors to the model to improve the quality of the learned models.

A classic hypothesis in depth and disparity estimation is that the target is smooth [15, 35]. The same *a priori* can be applied to the scene modeled by the NeRF. Due to the ability of NeRFs to model transparent surfaces and volumes, the predicted weights can be highly irregular. As a consequence, it is easier to regularize across different rendered viewpoints (i.e. after projection onto a given camera) rather than directly regularizing the 3D scene itself. This means that instead of using the depth function d from Eq. (4), it is more appropriate to work with the depth map $\tilde{d}$ produced by the NeRF model from a given viewpoint. This depth map $\tilde{d}$ is then indexed by its 2D coordinate (x, y) instead of a ray in the 3D space.

In image processing, a classic way of enforcing smoothness is to add a regularization term in the loss function based on the gradients of the image. For example, penalizing the squared L_2 norm of the gradients has the effect of removing high gradients in the depth map thus enforcing it to be smooth, as desired. In addition, it does not penalize slanted surfaces (since they have null Laplacian) as it would happen in the case of using a total variation regularization [33]. The proposed regularization term thus reads

$$L_{depth} = \sum_{(x,y)} \text{clip}(\|\nabla \tilde{d}(x, y)\|^2, g_{max}). \tag{5}$$

In practice, we add a differentiable clipping to L_{depth}, parametrized by g_{max}, to preserve sharp edges that could otherwise be over-smoothed.

By *only* changing the ReLU activation function to a Softplus activation function, the MLP used in NeRF can be transformed into a continuous and infinitely differentiable function similarly to [14]. This allows to train directly using the gradient of the model, or even higher order operators as shown later.

Traditionally, NeRFs are defined in terms of rays, which are characterized by an origin and a viewing direction $(\mathbf{o}, \mathbf{v})$. Consequently d from (4) is parameterized by $(\mathbf{o}, \mathbf{v})$ instead of the image coordinates (x, y) as $\tilde{d}$ in (5). Let $C : \mathbb{R}^2 \to \mathbb{R}^3$ be the transformation that converts the image coordinates into the equivalent ray so that $\tilde{d}(x, y) = d(\mathbf{o}, C(x, y))$. Then the corresponding gradients are

$$\nabla_{(x,y)} \tilde{d}(x, y) = \mathbf{J}_C(x, y) \nabla_{\mathbf{v}} d(\mathbf{o}, \mathbf{v}), \tag{6}$$

where $\mathbf{v} = C(x, y)$ and $\mathbf{J}_C$ is the Jacobian matrix of C. This way, Eq. (5) can be expressed in terms of rays with the exception of $\mathbf{J}_C$ that could be computed at the same time as the corresponding rays during the dataloading process. In practice, we use a simplified regularization loss that avoids computing $\mathbf{J}_C$ (see Eq. (11)).

Link with RegNeRF. In order to improve the robustness of NeRFs when training with few data, Niemeyer *et al*. [26] proposed RegNeRF, which also uses an additional term in the loss function to regularize the predicted depth map. This work additionally proposed an appearance regularization term using a normalizing flow network trained to estimate the likelihood of a predicted patch compared to normal patches from the JFT-300M dataset [41]. While the later is not studied here, we show that their depth regularization term is simply an approximation of the more generic differential loss presented in Eq. (5).

Consider the depth map $\tilde{d}$ and the set of coordinates (x, y) that corresponds to the pixels of the depth map. RegNeRF regularizes depth by encouraging that neighboring pixels $(x + i, y + j)$ for $i, j \in \{0, 1\}^2$ and $i + j = 1$ have the same depth as the pixel (x, y) such as

$$L_{depth} = \sum_{(x,y)} \sum_{\substack{(i,j)\in\{0,1\}^2 \\ i+j=1}} (\tilde{d}(x + i, y + j) - \tilde{d}(x, y))^2, \tag{7}$$

which is a finite difference expression of the gradient of $\tilde{d}$. Thus the major difference between (7) and our approach is that (7) approximates the gradient with finite differences while we take advantage of automatic differentiation.

In practice, RegNeRF regularization is not done on the entire depth maps but rather by sampling patches. The loss (7) is computed not only for all patches corresponding to a view in the training dataset, but also for rendered patches whose observation is not available. Indeed, all views should verify this depth regularity property, not only those in the training data. As a result, RegNeRF requires modifying the dataloaders to incorporate patch-based sampling and rays corresponding to unseen views. Note that our depth regularization term (5) does not require patches and therefore can be directly applied using single rays sampling as traditionally

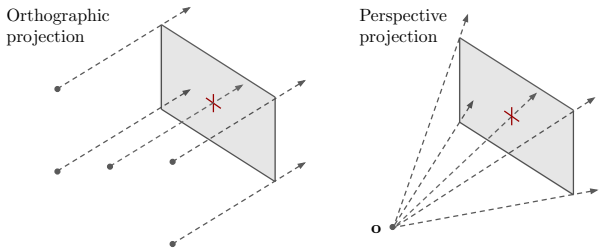

Figure 2. All perspective projection rays originate at the same center of projection **o**, located at a finite distance from the image plane. The center of projection in orthographic projection is at infinity, which can be represented by using a different origin for each ray, so that the origin points are parallel to the image plane.

done to train NeRFs. This makes the proposed framework compatible with previous single ray regularization methods, such as InfoNeRF [18], and non-uniform ray sampling, important when working with 360° images [27]. It also does not regularize unseen views as explained in Section 4.

Normals regularization. The regularization term (5) relies on depth maps. However, differential geometry also allows us to regularize other geometry-related features when training a NeRF. For example, consider n, the function that returns the scene normals for a given ray, whose projection, or map of normals, is denoted $\tilde{n}$. In that case, the regularization of the normals of the scene becomes

$$L_{normals} = \sum_{(x,y)} \|J_{\tilde{n}}(x,y)\|_F^2 \tag{8}$$

where $J_{\tilde{n}}$ is the Jacobian of the map of normals. This regularizer was applied to generate one of the results in Fig. 1.

Simplified regularization loss. The main problem with the loss presented in Eq. (5) is that it does not depend only on each individual ray, but also requires additional camera information to compute $\mathbf{J}_C$. Since this can be unpractical depending on the camera model, we propose to use a different and fixed local camera model only for the regularization process. Instead of using the usual perspective projection models associated with the training data, it is possible to regularize the scene as if the ray being processed originated from an orthographic projection camera, as illustrated in Fig. 2.

Consider a ray defined by its origin $\mathbf{o}$ and its direction $\mathbf{v}$). Let $(\mathbf{i}, \mathbf{j})$ be a local orthonormal basis of the plane defined by $\mathbf{o}$ and $\mathbf{v}$. Using an orthographic projection camera, the direction is fixed and only the origin changes to obtain other rays from the same camera. Therefore C, defined such that $\tilde{d}(x,y) = d(C(x,y), \mathbf{v})$, is explicit and $C(x,y) = x\mathbf{i} + y\mathbf{j}$. This leads to $\mathbf{J}_C(x,y) = \begin{pmatrix} \mathbf{i} \\ \mathbf{j} \end{pmatrix} \in \mathbb{R}^{2\times 3}$. Therefore

$$\nabla_{(x,y)} \tilde{d}(x,y) = \begin{pmatrix} \langle \nabla_{\mathbf{o}} d(\mathbf{o},\mathbf{v}), \mathbf{i} \rangle \\ \langle \nabla_{\mathbf{o}} d(\mathbf{o},\mathbf{v}), \mathbf{j} \rangle \end{pmatrix} \tag{9}$$

and $\|\nabla \tilde{d}(x,y)\|^2 = \langle \nabla_{\mathbf{o}} d(\mathbf{o},\mathbf{v}), \mathbf{i} \rangle^2 + \langle \nabla_{\mathbf{o}} d(\mathbf{o},\mathbf{v}), \mathbf{j} \rangle^2$. Since $(\mathbf{i}, \mathbf{j}, \mathbf{v})$ is, by construction, an orthonormal basis of the space, we also have that $\|\nabla_{\mathbf{o}} d(\mathbf{o},\mathbf{v})\|^2 = \langle \nabla_{\mathbf{o}} d(\mathbf{o},\mathbf{v}), \mathbf{i} \rangle^2 + \langle \nabla_{\mathbf{o}} d(\mathbf{o},\mathbf{v}), \mathbf{j} \rangle^2 + \langle \nabla_{\mathbf{o}} d(\mathbf{o},\mathbf{v}), \mathbf{v} \rangle^2$ thus

$$L_{depth} = \sum_{(\mathbf{o},\mathbf{v})\in\mathcal{R}} \|\nabla_{\mathbf{o}} d(\mathbf{o},\mathbf{v})\|^2 - \langle \nabla_{\mathbf{o}} d(\mathbf{o},\mathbf{v}), \mathbf{v} \rangle^2 \tag{10}$$

$$= \sum_{(\mathbf{o},\mathbf{v})\in\mathcal{R}} \|\nabla_{\mathbf{o}} d(\mathbf{o},\mathbf{v}) - \langle \nabla_{\mathbf{o}} d(\mathbf{o},\mathbf{v}), \mathbf{v} \rangle \mathbf{v}\|^2. \tag{11}$$

Note how Eq. (11) does not depend on the choice of $(\mathbf{i}, \mathbf{j})$, is entirely defined by the knowledge of the ray $(\mathbf{o}, \mathbf{v})$ and is independent from $\mathbf{J}_C(x,y)$.

4. Experimental results

We test the impact of the proposed differential regularization on the task of scene estimation using only three input views. This an extreme test case and, as such, it is highly reliant on the quality of the regularization to avoid catastrophic collapse as shown by Niemeyer *et al*. [26] for mip-NeRF [2]. In order to compare the proposed formalization of RegNeRF [26] to its original version, we modified the code of the authors and replaced their depth loss by the one in (11). We refer to our approach as DiffNeRF. The code to reproduce the results presented in this section is available at `https://github.com/tehret/diffnerf`.

Results on LLFF [23]. In Table 1, we compare the results of the original RegNeRF (using the models trained by the authors) with our DiffNeRF formalization (11). Since the code released by the authors does not contain the additional appearance loss, we added another comparison that corresponds to RegNeRF without the additional appearance regularization (*i.e*. training from scratch using the available code). The proposed DiffNeRF not only improves by 1dB the PSNR of reconstructed unseen views compared to the equivalent RegNeRF version, it also outperforms RegNeRF with appearance regularization by almost 0.5dB. This is also the case for other metrics such as SSIM and LPIPS.

Visual results on two examples of the LLFF dataset are shown in Fig. 3. In both cases, we compare the proposed version with the models trained by Niemeyer *et al*. [26]. The *horns* scene in Fig. 3 shows a first example where our formalization outperforms RegNeRF across all evaluation metrics. The proposed method is able to learn a better geometry of the image, leading to a more complete reconstruction of the triceratops skull (see the horn on the right), but also of the rest of the scene, such as the sign panel in the foreground or the handrails in the background. Similar improvements can be observed in the *trex* scene.

Fig. 1 shows another result, with the *room* scene of the LLFF dataset where the PSNR obtained with DiffNeRF is worse with respect to RegNeRF with appearance regularization. However, the depth map estimated by our formalism

		fern	flower	fortress	horns	leaves	orchids	room	trex	avg.
PSNR	PixelNeRF ft [47]	-	-	-	-	-	-	-	-	16.17
	SRF ft [8]	-	-	-	-	-	-	-	-	17.07
	MVSNeRF ft [6]	-	-	-	-	-	-	-	-	17.88
	RegNeRF (w/o app. reg.)	19.85	19.64	22.28	13.05	16.46	15.43	20.62	20.37	18.46
	DiffNeRF (ours)	**20.15**	**20.27**	**23.68**	**17.80**	**16.88**	15.50	21.04	**20.58**	**19.49**
	RegNeRF [26]	19.87	19.93	23.32	15.65	16.60	**15.56**	**21.53**	20.17	19.08
SSIM	RegNeRF (w/o app. reg.)	0.694	0.677	0.706	0.486	0.599	0.483	0.843	0.774	0.658
	DiffNeRF (ours)	**0.703**	**0.707**	**0.761**	**0.680**	**0.645**	0.487	**0.864**	**0.791**	**0.705**
	RegNeRF [26]	0.697	0.688	0.743	0.610	0.613	**0.502**	0.861	0.766	0.685
LPIPS	RegNeRF (w/o app. reg.)	0.323	0.243	0.294	0.341	0.229	0.259	0.204	0.197	0.261
	DiffNeRF (ours)	**0.290**	**0.223**	**0.219**	**0.293**	**0.186**	**0.247**	**0.171**	**0.166**	**0.224**
	RegNeRF [26]	0.304	0.234	0.258	0.356	0.222	0.251	0.185	0.197	0.251

Table 1. Quantitative comparison of novel view synthesis for different NeRF regularization on the LLFF dataset. All models were trained using only three input views. *RegNeRF (w/o app. reg.)* corresponds to the original RegNeRF without appearance regularization, while the proposed framework is *DiffNeRF*. The results using RegNerf with appearance regularization are also provided for reference. The proposed regularization almost systematically achieves the best results across all metrics without requiring any additional appearance regularization. The LPIPS metric is computed using the official implementation provided by Zhang *et al*. [48]. Best results are shown in bold.

	8	21	30	31	34	38	40	41	45	55	63	82	103	110	114	avg
PixelNeRF ft [47]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	**18.95**
SRF ft [8]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	15.68
MVSNeRF ft [6]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	18.54
RegNeRF (w/o app. reg.)	19.06	12.42	22.45	16.35	18.13	16.92	18.63	15.97	16.29	17.75	20.57	17.54	22.10	17.97	21.31	18.23
DiffNeRF (ours)	15.47	**13.63**	**23.18**	16.74	**18.66**	**17.28**	18.57	15.53	**16.45**	**17.94**	21.65	15.19	**23.69**	**20.32**	21.41	18.38
RegNeRF [26]	**19.45**	12.76	22.92	**16.84**	18.24	17.12	**19.09**	**18.41**	16.44	17.61	**22.91**	**19.42**	22.95	18.06	**21.52**	18.92

Table 2. Quantitative comparison of novel view synthesis for different NeRF regularization on the DTU dataset. All models were trained using only three input views. *RegNeRF (w/o app. reg.)* corresponds to the original RegNeRF without appearance regularization, while the proposed framework is *DiffNeRF*. The results using RegNerf with appearance regularization are also provided for reference. The case of scenes 41 and 82 are discussed in Section 4. Best results are shown in bold.

is still much smoother without losing details. In addition, as in the triceratops example, we can see that some details are also better reconstructed, like the audio conferencing system in the middle of the table. The LPIPS metric in Table 1 also seems to confirm that the DiffNeRF results present less visual artifacts than RegNeRF. Both Fig. 1 and Fig. 3 show that the DiffNeRF depth maps are better regularized than the original RegNeRF. In DiffNeRF we only use the input views at training time, without regularizing unseen views or requiring patch-based dataloaders with a predefined patch size (as in RegNeRF). This shows that the proposed formalism yields a better generalization. All experiments with LLFF were computed using a weight of $2e^{-4}$ for the regularization term with a clipping value $g_{max} = 20$.

Results on DTU [17]. Table 2 and Fig. 4 present results on the DTU dataset. Again, DiffNeRF produces results with a smoother scene geometry. All experiments with DTU were computed using a weight of $2e^{-4}$ with a clipping value $g_{max} = 5$.

Parameters study. We illustrate in Fig. 5 the impact of the two parameters of the proposed regularization: the weight of the regularization term in the loss and the value of the clipping. When the regularization is too weak, the surface exhibits irregular patterns. On the contrary, a regularization that is too strong can make details disappear (for example when parts of the pumpkin are merged with the background). A strong clipping allows to get back some details but can lead to visual artifacts such as staircasing. The proposed set of parameters leads to a smooth surface while keeping details.

Limitations. During the experiments, we observed two main limitations. The first one is the apparition of "floaters", groups of points with a non zero density and disjoint from the scene. These floaters can hide portions of the scene when synthesizing novel views (see Fig. 6). In the DTU dataset, we find these floating artifacts to be related to the

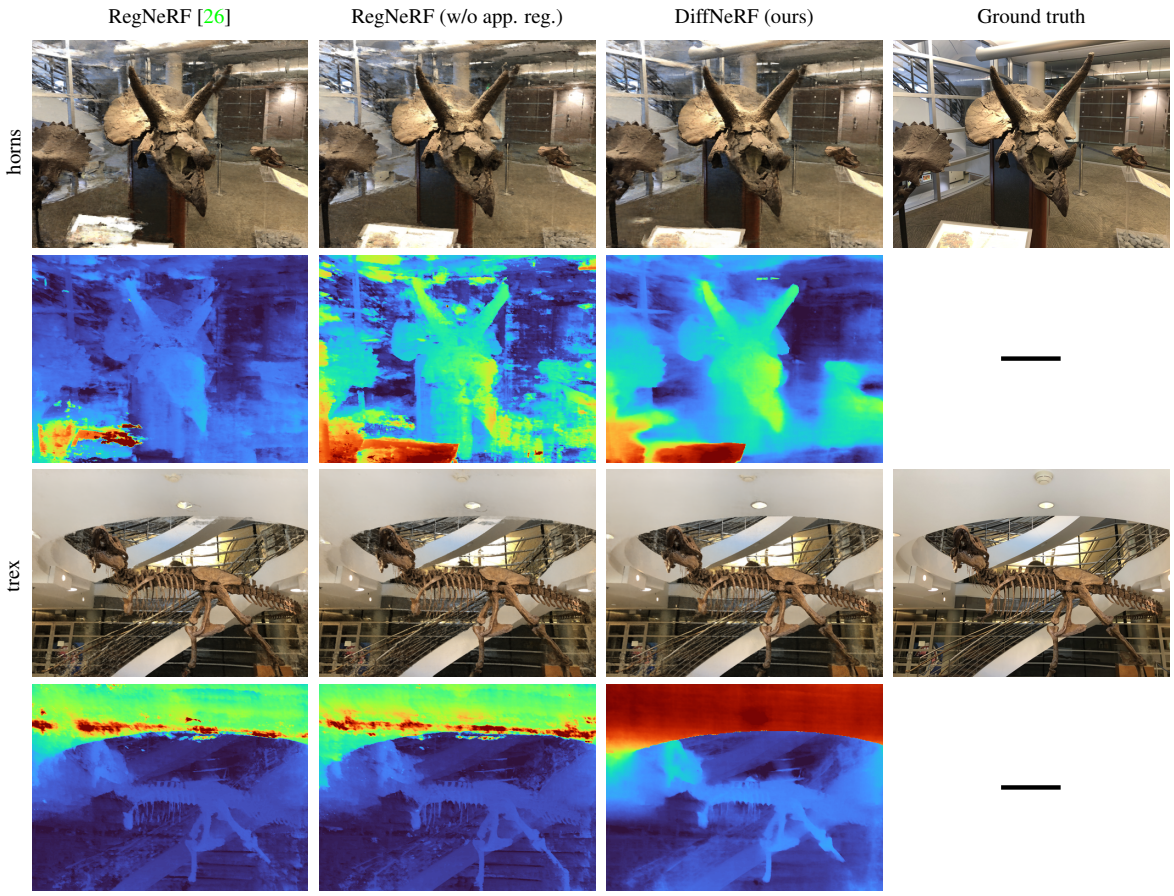

Figure 3. Visual examples of novel view synthesis for the *horns* (top) and *trex* (bottom) sequences of the LLFF dataset after training with three views. The depth map produced by the proposed DiffNeRF is more regular than those produced RegNeRF. It also recovers more details both in the foreground (see the sign panel on the left or the triceratops' left horn) but also in the background (see the glass panels and the handrails).

large textureless background regions or areas observed by a single camera (*i.e*. when the problem is not well defined, note that these regions are still regularized). We did not observe such floaters in the LLFF dataset. This also explains why regularizing unseen views, *i.e*. without any data attachment term, is not a good idea since it encourages the creation of such floaters.

The second limitations is the computation performance. Since the proposed regularization requires computing gradients, it is expected to be slower and require more memory. Nevertheless, since there is no need to regularize unseen views, the proposed method remains competitive (for the depth regularization). A comparison is shown in Table 3.

	Rays/s	Batch size
RegNeRF [26]	~ 6000	~ 2000
DiffNeRF (depth reg.)	~ 5000	~ 1000
DiffNeRF (normals reg.)	~ 1100	~ 250

Table 3. Computation speed (in rays per seconds) for the different methods on a NVIDIA V100 16Go GPU. Because diffNeRF does not require sampling additional rays from unseen views, the computation is barely slower than RegNeRF [26] ($\sim 16\%$). Higher order regularization (such as normals regularization) are much slower though.

5. Extension to surface regularization using mean and Gaussian curvatures

Another trend with NeRF-like models is to directly learn a surface model instead of a density function as shown in Section 2.1. In particular, IDR [46] and VolSDF [45] both learn the surface by means of a signed distance function (SDF). This SDF can then be used in a direct manner or as a guide to sample points as done in NeRF to learn the surface. Since this SDF can be seen as an implicit function F defining the surface $\mathcal{S}$ as the set of points $\{\mathbf{x} \in \mathbb{R}^3 \mid F(x) = 0\}$, it

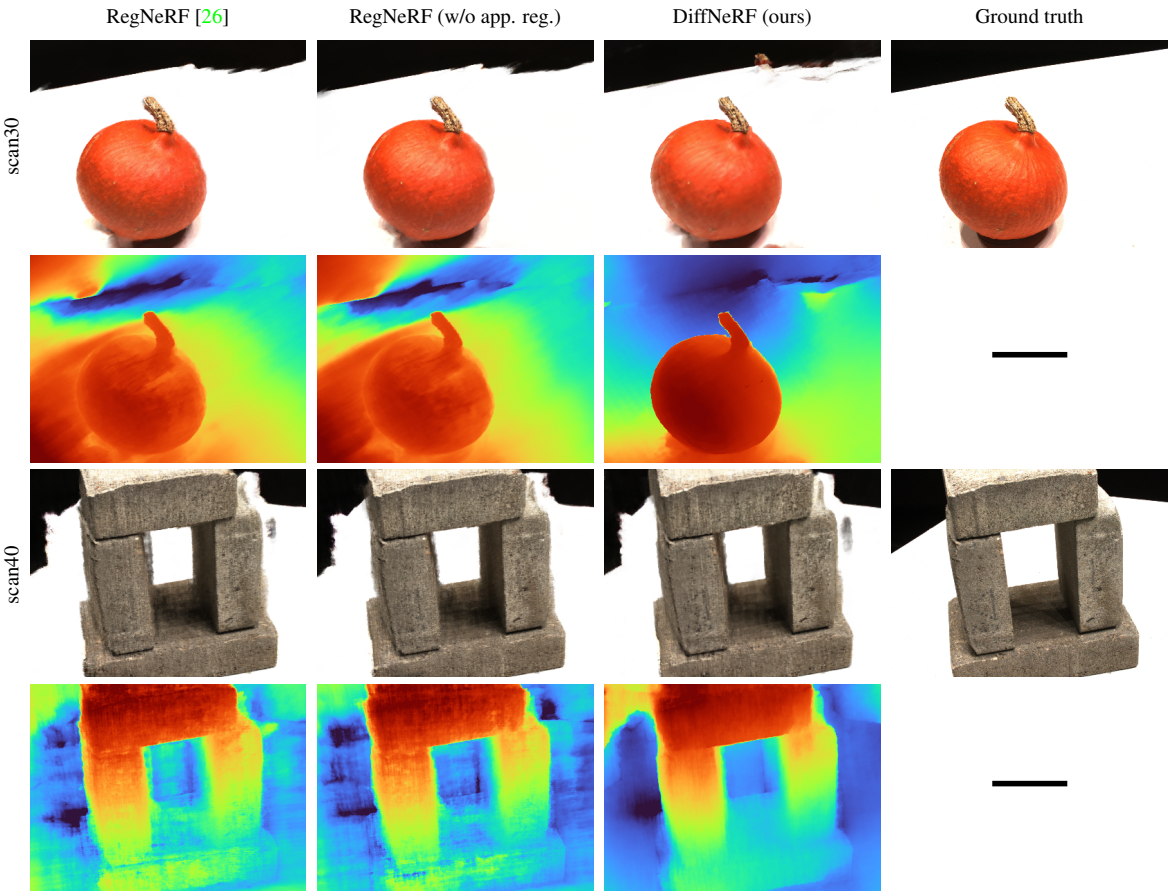

Figure 4. Visual example of novel view synthesis for scenes 30 and 40 of the DTU dataset after training with three views. The depth map produced by the proposed DiffNeRF is more regular than those produced RegNeRF. It also separates better the object from the background.

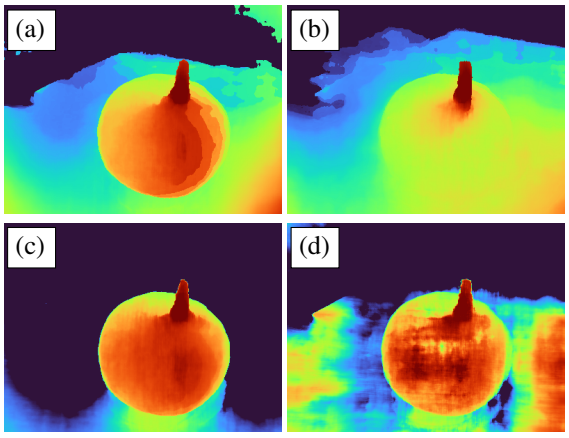

Figure 5. Visual impact of the two parameters of the regularization (reconstructions from three views). (a) strong regularization and clipping, (b) strong regularization and little clipping, (c) medium regularization and clipping, (d) little regularization and clipping.

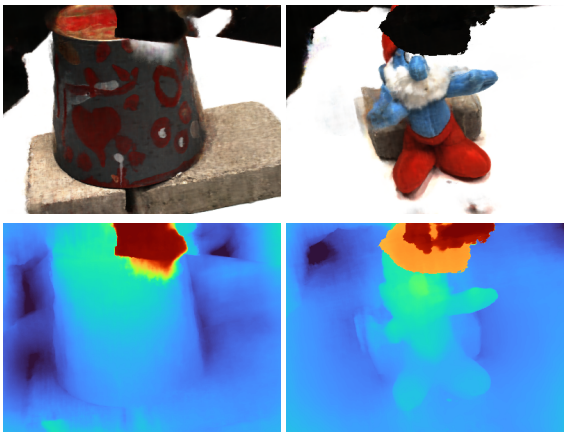

Figure 6. Failure cases for scenes 41 and 82 of the DTU dataset reconstructed from three views. "Floaters" (groups of points with a non zero density and disjoint from the scene) hide portions of the scene when synthesizing novel views.

is possible to compute other differential quantities related to surface regularity, such as the curvature. This allows to directly regularize the surface instead of regularizing the projections of the scene as shown in Section 3. We propose in this section to look at the Gaussian curvature γ_{gauss} and the mean curvature γ_{mean}, since they both have an analytical expression that can be easily implemented using the existing deep learning frameworks.

These curvatures are respectively defined as

$$\gamma_{mean} = -\textbf{div}\left(\frac{\nabla F}{\|\nabla F\|}\right) \tag{12}$$

and

$$\gamma_{gauss} = \frac{\nabla F \times H^*(F) \times \nabla F^t}{\|\nabla F\|^4}, \tag{13}$$

where H^* is the adjoint of the Hessian of F. Derivation details of these two curvatures can be found in [13]. Using (12) and (13), we can define a regularization loss similar to the one presented in Section 3 as

$$L_{curv}(\kappa_{curv}) = \mathbb{E}_{x\in\mathcal{S}}\left[\min\left(|\gamma(x)|, \kappa_{curv}\right)\right], \tag{14}$$

where γ can be either γ_{mean} or γ_{gauss}, depending on the preferred behavior, and κ_{curv} is a clipping value. The final loss to train a regularized VolSDF model using (14) becomes

$$L = L_{RGB} + \lambda_{SDF} L_{SDF} + \lambda_{curv} L_{curv}(\kappa_{curv}) \tag{15}$$

with

$$L_{SDF} = \mathbb{E}_{x\in\mathbb{R}^3}\left[\left(\|\nabla F(x)\| - 1\right)^2\right]. \tag{16}$$

As in [45], the L_{SDF} term enforces the Eikonal constraint on the implicit function F, thus learning a signed distance function. Note that (15) makes it possible to regularize the surface directly during training instead of doing it in different stages as in [44].

The regularization is characterized by the same two parameters, the regularization weight and the clipping value, than the regularization presented in Section 3. To understand the impact of these parameters, we refer to the definition of the mean and Gaussian curvature in terms of the minimum curvature γ_{min} and maximum curvature γ_{max} of the surface at a given point

$$\gamma_{mean} = \tfrac{\gamma_{min}+\gamma_{max}}{2} \quad \text{and} \quad \gamma_{gauss} = \gamma_{min}\gamma_{max}. \tag{17}$$

Although this is not a practical definition of the curvature, since it does not allow for direct computation, it shows that minimizing the mean curvature leads to surface smoothing [10]. On the other hand, minimizing the Gaussian curvature forces the minimum curvature to be zero, resulting in flat surfaces with sharp straight edges. The visual impact on the reconstructed surfaces is shown in the supplementary material. An example of a regularized reconstruction using Gaussian curvature is shown in Fig 7.

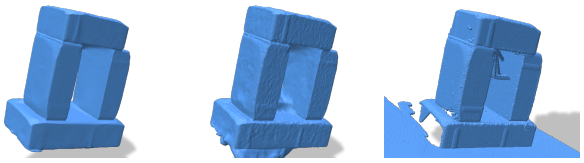

Figure 7. Visual example of a regularized reconstruction of the scene 40 of the DTU dataset. From left to right: regularized reconstruction using Gaussian curvature (13), original VolSDF results and ground truth.

6. Conclusions

With DiffNeRF, a variant of NeRF that relies on differential geometry to regularize the depth or the normals of the learned scene, we demonstrated that it is possible to achieve state-of-the-art novel view synthesis and depth estimation in few-shot neural rendering with a simple yet flexible regularization framework. This is made possible by modern deep learning frameworks, which already provide the necessary tools to implement differential geometry operators, thus facilitating their use in practice. However, the use of differential geometry is still subject to certain limitations. Higher-order operators can be costly both in memory and in computation time so a careful choice of the regularization term is essential. Operators should be chosen differently depending on the problem at hand. For example, a Gaussian curvature regularization may be appropriate for flat surfaces with strong edges, such as buildings, but could fill holes in irregular surfaces. The vast literature on differential geometry opens up many exciting opportunities to define new regularization tools with the appropriate mathematical formalism, which we hope pushes the limits of neural rendering even further. Additional studies to understand the impact of the activation function (such as softplus, squareplus [1], sine [38], Gaussian [9], *etc*.) on the results are also necessary.

Acknowledgements

Work partly financed by Office of Naval research grant N00014-17-1-2552, MENRT, and Kayrros. It was also performed using HPC resources from GENCI–IDRIS (grants AD011012453R2 and AD011011801R3) and from the "Mésocentre" computing center of CentraleSupélec and ENS Paris-Saclay supported by CNRS and Région Île-de-France (http://mesocentre.centralesupelec.fr). Centre Borelli is also with Université Paris Cité, SSA and INSERM.

References

[1] Jonathan T. Barron. Squareplus: A Softplus-Like Algebraic Rectifier, 2021. 8

[2] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan.

Mip-NeRF: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5855–5864, 2021. 2, 4

[3] Mark Boss, Raphael Braun, Varun Jampani, Jonathan T Barron, Ce Liu, and Hendrik Lensch. NeRD: Neural reflectance decomposition from image collections. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12684–12694, 2021. 1

[4] Shengqu Cai, Anton Obukhov, Dengxin Dai, and Luc Van Gool. Pix2NeRF: Unsupervised conditional π-GAN for single image to neural radiance fields translation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[5] Eric R Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-GAN: Periodic implicit generative adversarial networks for 3D-aware image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5799–5809, 2021. 2

[6] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14124–14133, 2021. 5

[7] Di Chen, Yu Liu, Lianghua Huang, Bin Wang, and Pan Pan. GeoAug: Data Augmentation for Few-Shot NeRF with Geometry Constraints. In *Computer Vision – ECCV 2022*, Lecture Notes in Computer Science, pages 322–337. Springer Nature Switzerland, 2022. 1

[8] Julian Chibane, Aayush Bansal, Verica Lazova, and Gerard Pons-Moll. Stereo radiance fields (srf): Learning view synthesis for sparse views of novel scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7911–7920, 2021. 5

[9] Shin-Fang Chng, Sameera Ramasinghe, Jamie Sherrah, and Simon Lucey. Gaussian activated neural radiance fields for high fidelity reconstruction and pose estimation. In *European Conference on Computer Vision*, pages 264–280. Springer, 2022. 8

[10] Ulrich Clarenz, Udo Diewald, and Martin Rumpf. Anisotropic geometric diffusion in surface processing. In *Proceedings of the IEEE Visualization Conference*, 2000. 8

[11] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised NeRF: Fewer views and faster training for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 3

[12] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multi-view stereopsis (pmvs). In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, page 3, 2007. 1

[13] Ron Goldman. Curvature formulas for implicit curves and surfaces. *Computer Aided Geometric Design*, 22(7):632–658, 2005. 8

[14] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. In *International Conference on Machine Learning*, pages 3789–3799. PMLR, 2020. 3

[15] Heiko Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on pattern analysis and machine intelligence*, 30(2):328–341, 2007. 3

[16] Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting NeRF on a diet: Semantically consistent few-shot view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5885–5894, 2021. 1, 2

[17] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engin Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 406–413, 2014. 5

[18] Mijeong Kim, Seonguk Seo, and Bohyung Han. InfoNeRF: Ray entropy minimization for few-shot neural volume rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 2, 3, 4

[19] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6498–6508, 2021. 1

[20] Roger Marí, Gabriele Facciolo, and Thibaud Ehret. Sat-NeRF: Learning multi-view satellite photogrammetry with transient objects and shadow modeling using RPC cameras. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2022. 1, 3

[21] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. NeRF in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7210–7219, 2021. 1

[22] Nelson Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):99–108, 1995. 2

[23] 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 Transactions on Graphics (TOG)*, 38(4):1–14, 2019. 4

[24] 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 *European Conference on Computer Vision*, pages 405–421, 2020. 1, 2

[25] Pierre Moulon, Pascal Monasse, Romuald Perrot, and Renaud Marlet. OpenMVG: Open multiple view geometry. In *International Workshop on Reproducible Research in Pattern Recognition*, pages 60–74. Springer, 2016. 1

[26] Michael Niemeyer, Jonathan T Barron, Ben Mildenhall, Mehdi SM Sajjadi, Andreas Geiger, and Noha Radwan. RegNeRF: Regularizing neural radiance fields for view synthesis from sparse inputs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 2, 3, 4, 5, 6, 7

[27] Takashi Otonari, Satoshi Ikehata, and Kiyoharu Aizawa. Non-uniform Sampling Strategies for NeRF on 360° images. In

33rd British Machine Vision Conference 2022, {BMVC} 2022, London, UK, November 21-24, 2022. BMVA Press, 2022. 4

[28] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5865–5874, 2021. 1

[29] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M. Seitz. HyperNeRF: A higher-dimensional representation for topologically varying neural radiance fields. *ACM Trans. Graph.*, 40(6), 2021. 1

[30] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-NeRF: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10318–10327, 2021. 1

[31] Daniel Rebain, Mark Matthews, Kwang Moo Yi, Dmitry Lagun, and Andrea Tagliasacchi. LOLNeRF: Learn from one look. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[32] Barbara Roessle, Jonathan T Barron, Ben Mildenhall, Pratul P Srinivasan, and Matthias Nießner. Dense depth priors for neural radiance fields from sparse input views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 3

[33] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 60(1-4):259–268, 1992. 3

[34] Ewelina Rupnik, Mehdi Daakir, and Marc Pierrot Deseilligny. Micmac–a free, open-source solution for photogrammetry. *Open Geospatial Data, Software and Standards*, 2(1):1–9, 2017. 1

[35] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*, 47(1):7–42, 2002. 3

[36] 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. 1

[37] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. *Advances in Neural Information Processing Systems*, 33:20154–20166, 2020. 2

[38] Vincent Sitzmann, Julien N. P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit Neural Representations with Periodic Activation Functions. volume 33, pages 7462–7473. Advances in neural information processing systems, 2020. 8

[39] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. In *ACM siggraph 2006 papers*, pages 835–846. ACM Press, 2006. 1

[40] Pratul P Srinivasan, Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T Barron. NeRV: Neural reflectance and visibility fields for relighting and view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7495–7504, 2021. 1

[41] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852, 2017. 3

[42] 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. *Computer Graphics Forum*, 39(2):701–727, 2020. 2

[43] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt. Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12959–12970, 2021. 1

[44] Guandao Yang, Serge Belongie, Bharath Hariharan, and Vladlen Koltun. Geometry processing with neural fields. *Advances in Neural Information Processing Systems*, 34:22483–22497, 2021. 8

[45] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *Advances in Neural Information Processing Systems*, 34, 2021. 6, 8

[46] 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. 6

[47] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelNeRF: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4578–4587, 2021. 1, 2, 5

[48] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 5

[49] 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 Transactions on Graphics (TOG)*, 40(6):1–18, 2021. 1