Stereo Radiance Fields (SRF): Learning View Synthesis for Sparse Views of Novel Scenes

Julian Chibane[1,2] Aayush Bansal[3] Verica Lazova[1,2] Gerard Pons-Moll[1,2]
[1]University of Tübingen, Germany, [2]Max Planck Institute for Informatics, Germany [3]Carnegie Mellon University, USA
{jchibane, vlazova, gpons}@mpi-inf.mpg.de, aayushb@cs.cmu.edu

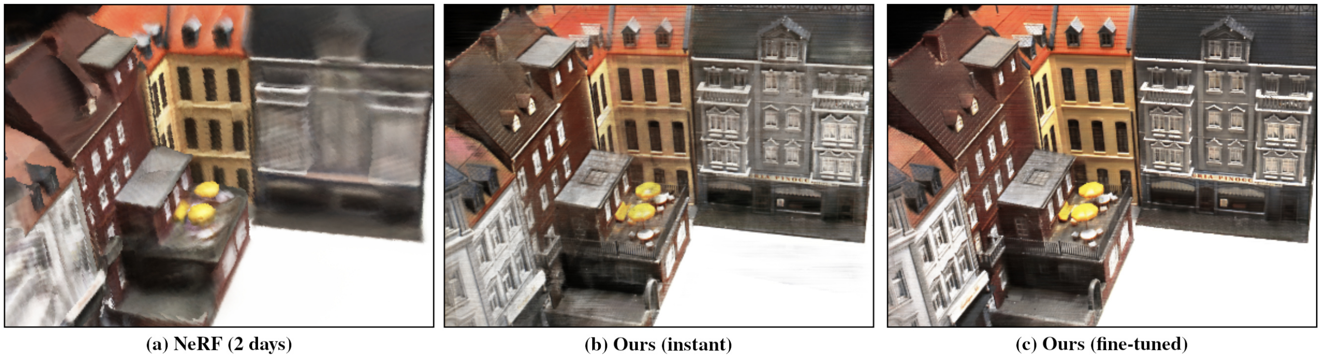


Figure 1. Our method can synthesize new views with a single network forward pass from 10 sparse and spread-out views of a novel scene. Here we synthesize a new view with **(a) NeRF** [34], which requires scene-specific training for 2 days; with our method **(b)** which produces the result instantaneously and **(c)** our improved result after fine-tuning our pre-trained model for 15 minutes on the 10 views.

Abstract

Recent neural view synthesis methods have achieved impressive quality and realism, surpassing classical pipelines which rely on multi-view reconstruction. State-of-the-Art methods, such as NeRF [34], are designed to learn a single scene with a neural network and require dense multi-view inputs. Testing on a new scene requires re-training from scratch, which takes 2-3 days. In this work, we introduce Stereo Radiance Fields (SRF), a neural view synthesis approach that is trained end-to-end, generalizes to new scenes, and requires only sparse views at test time. The core idea is a neural architecture inspired by classical multi-view stereo methods, which estimates surface points by finding similar image regions in stereo images. In SRF, we predict color and density for each 3D point given an encoding of its stereo correspondence in the input images. The encoding is implicitly learned by an ensemble of pair-wise similarities – emulating classical stereo. Experiments show that SRF learns structure instead of over-fitting on a scene. We train on multiple scenes of the DTU dataset and generalize to new ones without re-training, requiring only 10 *sparse and spread-out views as input. We show that 10-15 minutes of fine-tuning further improve the results, achieving significantly sharper, more detailed results than scene-specific models. The code, model, and videos are available –* https://virtualhumans.mpi-inf.mpg.de/srf/.

1. Introduction

We introduce a neural multi-view view synthesis approach which is trained end-to-end, generalizes to novel scenes, and requires only sparse views at test time (Fig. 1-(b)). This is in stark contrast to State-of-the-Art (SOTA) view synthesis methods like NeRF [34], which are trained for a specific scene and require dense multi-views to produce sharp results.

On one end of the view synthesis spectrum of methods, we have pure data-driven methods such as NeRF [34], which have shown impressive results. NeRF takes a radical data-driven approach by learning a mapping from a location and direction to the emitted radiance. This mapping is specifically trained for a scene (Fig. 2-(a)). Generalization to a new scene requires retraining for 2 days and results are blurry when trained on sparse and spread-out views (Fig. 1-(a)). On the other end of the spectrum, popular classical image-based rendering techniques [46] use geometry [8, 29, 44, 45]. These approaches warp pixels to the desired target view via correspondences [39, 41, 49] or multi-view 3D reconstruction [42, 43]. Consequently, these methods rely on high-quality 3D reconstruction or dense per-pixel correspondence, which requires dense multi-views. Recent work [5, 38] combines classical methods with data-driven approaches by learning to correct the warped views of classical methods. The sequential pipeline in these methods [5, 38] do not allow end-to-end learning.

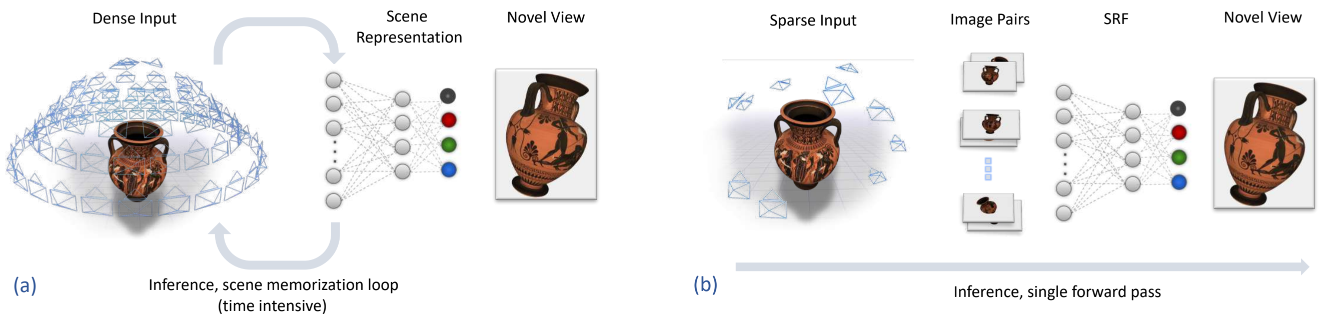


Figure 2. **Pure data-driven view synthesis and SRF (ours).** Existing methods achieve remarkable realism representing scenes with a neural network. A model is trained specifically for a scene to synthesize high-quality novel views. However, this requires dense views and 2 days of training per scene. In this work, we address the more challenging task of novel view synthesis from sparse and spread-out views, using a single forward pass through the network, to instantly obtain the result.

We take inspiration from both classical and pure data-driven methods. Like NeRF, we also learn a neural network to predict radiance (specifically color and density). However, instead of memorizing the scene radiance at 3D locations, we use an image-based feature encoding, which allows the network to reason about scene geometry (Fig. 2-(b)). In classical stereo reconstruction [41, 49], correspondences across views are found by computing a similarity score. We devise an architecture, called *Stereo Radiance Fields* (*SRF*), which mimics the classical approach without computing explicit correspondences, but can be trained end-to-end. A 3D point is projected to each available view to extract point-wise view features. Then view features are *processed in pairs* by a bank of filters, which emulate correspondence finding in classical methods (Fig. 3). The resulting matrix of pair-wise scores is further processed with a Convolutional Neural Network [21] (CNN), which agglomerates information from the available views to predict the desired radiance at that point.

Our experiments demonstrate that incorporating multi-view reconstruction ideas within the architecture significantly boosts generalization ability. When training on a *single scene* and testing on a *new scene*, SRF can produce reasonable results. This indicates that the network does not memorize the scene, but learns to reason about structure. When trained on multiple scenes (100 or more), SRF can generalize to novel scenes, even when only 10 sparse and spread-out views are available as input. Further improvements can be obtained by fine-tuning on the 10 views (Fig. 1-(c)), which typically takes about 15 minutes, which is much less than the $2-3$ days required by methods that re-train from scratch [34, 47]. SRF results are sharper, validating that multi-view reconstruction structure not only helps to generalize but also constrains the learning problem. We encourage the reader to view our results as videos available on our project page. To summarize, our contributions are:

- We introduce Stereo Radiance Fields (SRF), an end-to-end, self-supervised architecture for multi-view view synthesis. We bring together insights from classical multi-view reconstruction pipelines and neural rendering approaches.

- Experiments demonstrate that SRF generalize to *novel scenes* given *sparse and spread-out views* as input. Further, fine-tuning a pre-trained SRF for a few minutes on test distribution improve results.

- We show how to combine recent paradigms into one model, often treated in isolation in novel view synthesis: SRF builds on classical multi-view *3D reconstruction* and *learning* from multiple scenes.

- In the sparse and spread-out view setting, SRF produces much sharper results than SOTA baselines like NeRF [34]. We achieve even better results when we fine-tune for only 15 minutes in contrast to NeRF trained on the 10 test views for 2 days.

2. Multi-View View Synthesis

Given N camera views, our goal is to synthesize a view for a new virtual camera. This is a long-standing problem [17, 50]. Historically [46], the problem has been studied under three possible directions depending on the geometric information used: (1) rendering without geometry [2, 16, 22, 30, 35] by modelling a plenoptic function to compute intensity of light rays for a given camera at every possible angle; (2) rendering using correspondences [8, 44] which requires knowledge of positional correspondences across multi-views; and (3) rendering with explicit geometry [29, 45] which requires explicit 3D information in the form of depth or point clouds. In this work, we bring together insights from neural rendering with classical reconstruction pipelines. We encourage our network to reason about correspondences across pairs of views by computing an ensemble of pair-wise scores within the network. Although we never explicitly compute correspondences, this geometric reasoning, allow us to generalize to new scenes.

Correspondences across multi-views: Classic approaches [8, 13, 17, 44, 51] in multi-view stereo rely on correspondences across views. In this work, we bring together the insights from classical multi-view stereo [17, 46] and contemporary learning-based approaches [10, 34, 40]. We use an encoder network that inputs 10 multi-views and extracts multi-scale features [4, 40]. We replace classical block or feature matching with a multi-layer perceptron (MLP) which outputs an ensemble of similarity scores. Like us, recent work can do view synthesis from sparse views [5] incorporating explicit correspondences. However, explicitly computing correspondences is hard due to differences in illumination, zoom, scale, and occlusion. A scene-specific model is trained to correct artifacts. In our method, the network reasons about correspondences driven by the view synthesis loss, but they are never explicitly computed. Importantly, our model is not specific to a scene.

Neural Rendering and Plenoptic Modeling: State-of-the-art neural rendering [52] approaches have enabled creation of photo-realistic visual content using deep neural networks [21]. There are three popular directions for multi-view view synthesis: (1) using plane-sweep stereo [13, 15] or multi-plane image (MPI) representation [61]. MPI-based approaches [7, 14, 15, 20, 33, 48] have shown remarkable results on continuous view synthesis for small baseline shifts, but fail for large ones as it assumes accurate multi-plane imaging; (2) explicitly incorporating 3D reconstruction using SfM [42, 43] or multi-view stereo [19] for view synthesis [3, 12, 18, 32, 38]. These approaches assume a reasonably dense 3D point cloud used in conjunction with a neural network for view synthesis. The role of the neural network is to correct the imperfections in the 3D reconstruction. However, these approaches struggle when the views are sparse with small overlap because explicit 3D reconstruction fails; and (3) recent approaches [23, 24, 27, 34, 37, 47, 57, 58] learn a 3D representation that can be combined with differentiable-ray marching operations to synthesize a new view. These approaches by design require scene-specific modeling. This restricts: (1) an instant and online visualization of a new capture because it requires $2-3$ days to train a model; and (2) utilizing large amounts of diverse visual data, which has been the driving force for progress in other areas of vision such as recognition, semantic segmentation, and detection.

Our work is deeply inspired by recent neural rendering approaches. Like NeRF [34], we predict radiance at continuous locations and use volume rendering to generate the target image. Instead of predicting based on point coordinates and radiance, we predict based on point image features and an ensemble of similarity functions that emulate classical stereo matching. Hence, our work brings together contemporary neural rendering with classical computer vision within an end-to-end architecture. SRF is similar in spirit to previous work on 3D implicit shape reconstruction, Implicit Feature Networks (IF-Nets) [6, 10] and Neural Distance Fields (NDF) [11], where we decode occupancy or unsigned distances based on volumetric deep features computed from the input, instead of the originally proposed point coordinates [9, 31, 36]. Our work also shares insights with contemporary approaches [55, 56, 59]. Finally, our work is inspired by lifelong learning [53, 54] that aims to learn a generic representation that can be easily adapted to a new task with a few examples. We learn a generic view synthesis network that readily generalizes to new scenes. Our results further improve when we adapt it to the new scene with simple fine-tuning on test examples.

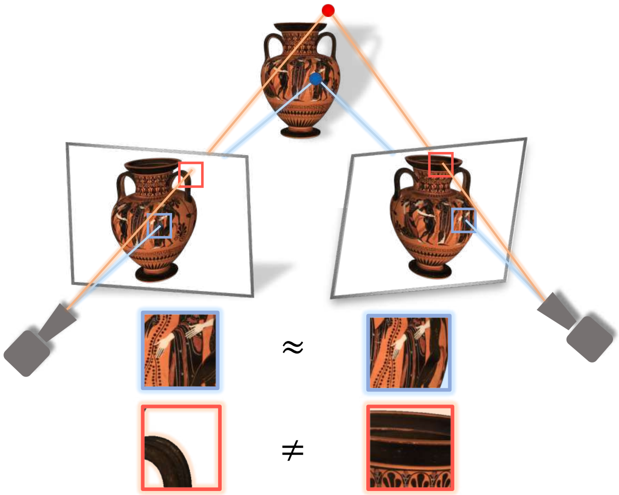


Figure 3. **Intuition of our Method:** We structure our model inspired by a geometric observation: 3D points in a scene that are on a surface will project to similar-looking regions when viewed from different perspectives (blue). We call this a photo-consistent point. A point in free space, however, will not be photo consistent (red). This holds for opaque, non-occluded surface points.

3. Method

In this section, we present our method *Stereo Radiance Fields* (SRF) for novel view synthesis given sparse and spread-out input views of objects unseen during training. We first give a background in Section 3.1 and then build SRF on these insights in Section 3.2.

3.1. Background

3.1.1 Generalizing Neural Radiance Fields (NeRF)

To produce color at a pixel of the target view, we shoot a ray from the camera position through the pixel into the scene. We binarize the ray into equal length bins and randomly

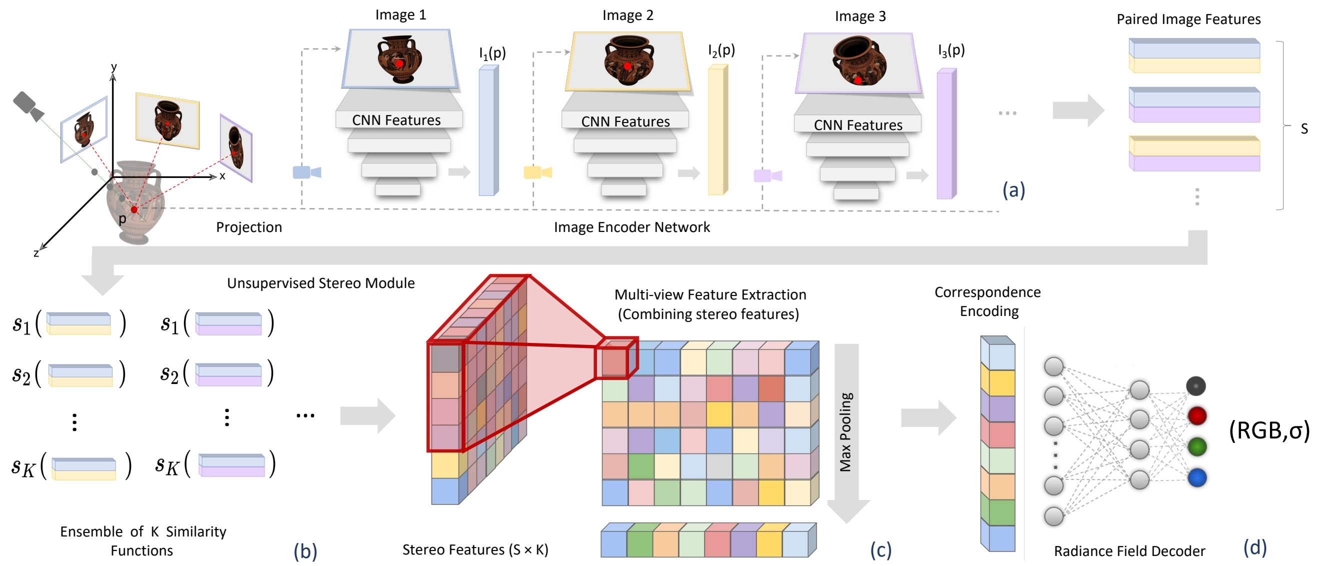


Figure 4. **Our Approach:** For a target view (left camera) we predict RGB color for each pixel. For a pixel, we project a ray into the scene and sample points along it. For a point, $\mathbf{p} \in \mathbb{R}^3$, our goal is to estimate its color, $\mathbf{c}$, and density, σ, where density encodes surface regions. (a) First, to encode the location of point $\mathbf{p}$, we project it into each reference view, $\mathbf{I}_i$ and extract features, $\mathbf{I}_i(\mathbf{p})$, generated by a 2D CNN at the location of projection. (b) If $\mathbf{p}$ is on a surface and photo-consistent, $\mathbf{I}_1(\mathbf{p}), \ldots, \mathbf{I}_N(\mathbf{p})$ will match (see Fig. 3). We emulate the process of finding photo-consistency by applying a learned similarity function $s_k(\cdot, \cdot)$ on all possible combinations. We learn an ensemble of similarity K functions, and obtain a Stereo Features matrix. (c) To aggregate multi-view information beyond pairs, we apply a 2D convolutional CNN to obtain a Multi-view Feature matrix. The matrix is Max Pooled to obtain a compact encoding of correspondence and color, which is decoded by an MLP into color and density (d). Weighted by the density, the color values along the target camera ray are fused into the final pixel color by volume rendering. We train the model end-to-end with image supervision alone.

sample one 3D point within each bin. At each point $\mathbf{p} \in \mathbb{R}^3$ we predict color $\mathbf{c} \in \{0, \ldots, 255\}^3$ and density $\sigma \in \mathbb{R}$. Density encodes regions of surface (high where there is surface, low elsewhere). Weighted by the density, the color values along the ray are fused into the final pixel color by volume rendering, following NeRF [34].

NeRF memorizes a scene with a neural function f, by learning to output $(\mathbf{c}, \sigma)$ given spatial location $\mathbf{p}$ and viewing direction $\mathbf{d}$

$$f_{\text{NeRF}} : (\mathbf{p}, \mathbf{d}) \mapsto (\mathbf{c}, \sigma). \tag{1}$$

This works well for a single scene with dense views. However, it fails to generalize to novel scenes as point coordinates do not carry scene-specific information. The neural network itself becomes the scene representation (Fig. 2-(a)). Instead, we aim to learn a neural model which emulates multi-view stereo reconstruction and synthesis internally and is conditioned on the scene itself at test time (Fig. 2-(b)). For this, we use a completely different point encoding architecture, which is not scene-specific

$$f : (\mathcal{I}, \mathbf{p}) \mapsto (\mathbf{c}, \sigma), \tag{2}$$

where $\mathcal{I} = \{I_i\}_{i=1}^N$ is the set of N reference images, $\mathbf{I}_i$, with known camera parameters. The design of f is inspired by classical stereo and is explained in Sec. 3.2. Note that we do not consider view dependent effects and leave it for future work. This allows us to focus on generalization to novel scenes and sparse inputs.

3.1.2 Classical Multi-View Stereo

Key to classical stereo imaging approaches (Structure-from-Motion, Multi-View Stereo) and our method is the following observation: In absence of occlusion, surface 3D points of an object project to *corresponding* photo-metrically consistent image regions in the multiple-views, whereas non-surface 3D points land on non-corresponding different regions (Fig. 3). We can invert this observation to find surfaces from images: we can find corresponding regions across views and triangulate them to find a 3D surface point. In classical works, this is done in non-differentiable, multi-step engineered pipelines. First, informative, distinctive regions of *interest* are found. Subsequently, a feature *descriptor* at the interest point is created from local image features, c.f. SIFT [26]. The descriptors from multiple images are matched based on a similarity measure. SRF internally mimics correspondence matching in an end-to-end unsupervised manner (based only on the rendering loss). Our point feature descriptors are learned by a 2D CNN image encoder network. Classical correspondence finding is emulated in SRF by processing point descriptors in pairs.

3.2. Stereo Radiance Fields (SRF)

SRF predicts color and density at a point, $\mathbf{p}$, in 3D space given, $\mathcal{I} = \{I_i\}_{i=1}^N$, a set of N reference images, $\mathbf{I}_i$, with known camera parameters. We structure *SRF*, f, in analogy to classical multi-view stereo approaches: (1) To encode the location of point $\mathbf{p}$, we project it into each reference view, $\mathbf{I}_i$, and build a local feature descriptor, $\mathbf{I}_i(\mathbf{p})$, (Sec. 3.2.1); (2) If $\mathbf{p}$ is on a surface and photo-consistent, $\mathbf{I}_1(\mathbf{p}), \ldots, \mathbf{I}_N(\mathbf{p})$ should match (Fig. 3); Feature matching is emulated with a learned function, g_{stereo}, encoding the features from all reference views (Sec. 3.2.2); (3) The encoding is decoded by a learned decoder, dec, into the NeRF [34] representation (Sec. 3.2.3). Formally, this decomposes *SRF* to:

$$f(\mathcal{I}, \mathbf{p}) = \text{dec}(g_{\text{stereo}}(\mathbf{I}_1(\mathbf{p}), \ldots, \mathbf{I}_N(\mathbf{p}))) \mapsto (\mathbf{c}, \sigma). \quad (3)$$

Figure 4 gives an overview of our method.

3.2.1 Image Encoder Network

In contrast to NeRF, where the input are point coordinates without scene-specific information, we condition our prediction on the reference images. We achieve this by projecting $\mathbf{p}$ into each reference view, $\mathbf{I}_i$, and build a local feature descriptor, $\mathbf{I}_i(\mathbf{p})$. For this, we first encode each complete reference image with a shared 2D CNN. We build $\mathbf{I}_i(\mathbf{p})$ by extracting the deep features from each CNN layer at the location of the points $\mathbf{p}$ projection. This makes $\mathbf{I}_i(\mathbf{p})$ a multi-scale feature descriptor, as 2D CNNs naturally encode local information in their first layers up to global information in later layers with a high receptive field (Fig. 4-(a) "Image Encoder Network"). Because the point projection is in continuous space, whereas the features are in a discrete grid, we use bilinear interpolation for extraction. When $\mathbf{p}$ projects outside of an image we use zero padding. See appendix for further details.

3.2.2 Unsupervised Stereo Module

We build on the intuition of Multi-View Stereo: when a 3D point $\mathbf{p}$ is projected to photo-metrically consistent regions, $\mathbf{p}$ is likely to lie on a surface, and hence a high density σ should be predicted. In order, to process an arbitrary number of views, the stereo module processes feature descriptors of views in *pairs*. Specifically, we aim at learning mappings of feature pairs $\mathbf{I}_i(\mathbf{p}), \mathbf{I}_j(\mathbf{p})$:

$$s : (\mathbf{I}_i(\mathbf{p}), \mathbf{I}_j(\mathbf{p})) \mapsto x \in \mathbb{R}_0^+, \quad (4)$$

that allow the network to learn image scores useful for correspondence finding or propagate image color. Note, although our formulation is based on pairwise processing analog to similarity computation, *correspondences are not explicitly computed*. We represent each mapping s in the network using a single neuron. In practice, each possible pair $(\mathbf{I}_i(\mathbf{p}), \mathbf{I}_j(\mathbf{p}))$ with $i, j \in 1, \ldots, N, i \neq j$ is input to a neuron with ReLU non-linearity to ensure non-negative outputs (Fig. 4-(b)). This yields a vector $\mathbf{x}$ of size $S = N^2 - N$ with one entry per pair. Instead of relying on only a single neuron, we apply a bank of neurons, $s_k(\cdot, \cdot)$, $k = 1 \ldots K$ in the same fashion. Each neuron might learn different similarities, or specialize in propagating color. We concatenate the output vector $\mathbf{x}_k$ of each neuron in the bank into a *Stereo Feature* matrix $\mathbf{X} = [\mathbf{x}_1 \ldots \mathbf{x}_K] \in \mathbb{R}^{S \times K}$ whose height is the number of pairs S and the width is the number of neurons K used (Fig. 4-(b) "Stereo Features"). The Stereo Feature matrix can be efficiently computed by arranging feature pairs in a matrix and convolving it with the neuron bank.

Pairwise photo-consistency is, however, not a sufficient condition to identify surface points. 3D points might project to photo-consistent image regions in a stereo pair when reference views are captured nearby but not on a third view. We aggregate information from multi views by convolving the Stereo Feature matrix along the direction of pairs of views. Specifically, we aggregate 4 pairs in the height direction and all similarity measures along the width direction (Fig. 4-(c) "Multi-view Feature Extraction").

To merge view-pair information into a single vector $\mathbf{y} \in \mathbb{R}^K$, we run Max Pooling in the direction of views. Note that the complete stereo module by design, is flexible for varying number of input views during training and testing: the Max Pooling step is computing a vector $\mathbf{y}$ of fixed dimension K given varying number of input views. This constitutes the unsupervised Stereo Module, denoted by

$$g_{\text{stereo}} : (\mathbf{I}_1(\mathbf{p}), \ldots, \mathbf{I}_N(\mathbf{p})) \mapsto \mathbf{y} \in \mathbb{R}^K. \quad (5)$$

3.2.3 Radiance Field Decoder

The last stage of our network is to decode the stereo encoding $\mathbf{y} = g_{\text{stereo}}(\mathbf{I}_1(\mathbf{p}), \ldots, \mathbf{I}_N(\mathbf{p}))$ of point $\mathbf{p}$ into the final color $\mathbf{c}$ and density σ. For this, we rely on a simple MLP network denoted by

$$\text{dec} : \mathbf{y} \mapsto (\mathbf{c}, \sigma). \quad (6)$$

Sampled colors along a ray are fused based on their density following volume rendering [28, 34]. The training of the network is done fully end-to-end using only multi-view images without 3D data or supervision on the stereo module (Fig. 4-(d)). We use the $L2$ loss for comparing the rendered prediction with the target image. Please consider the appendix for further architectural details.

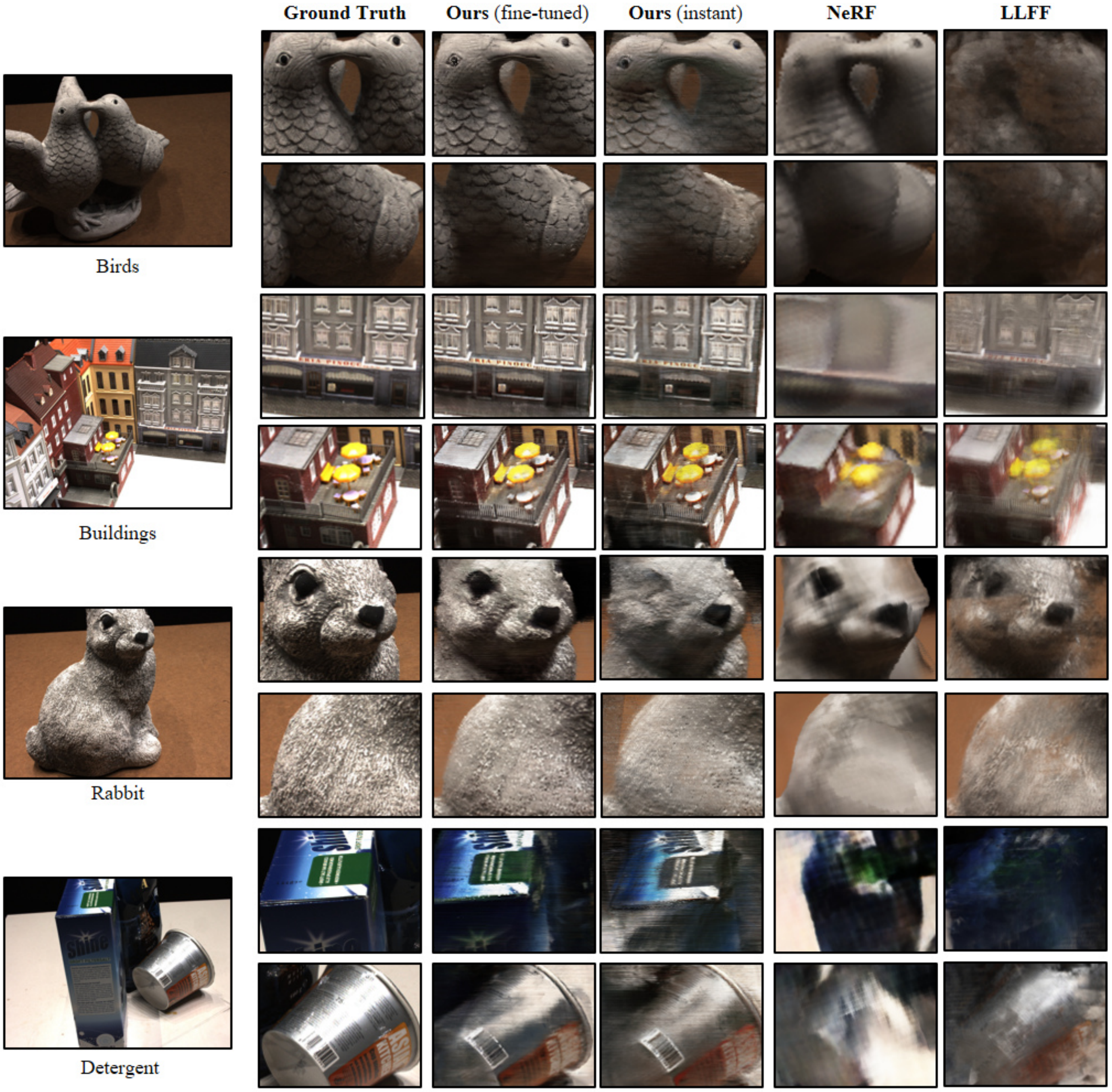


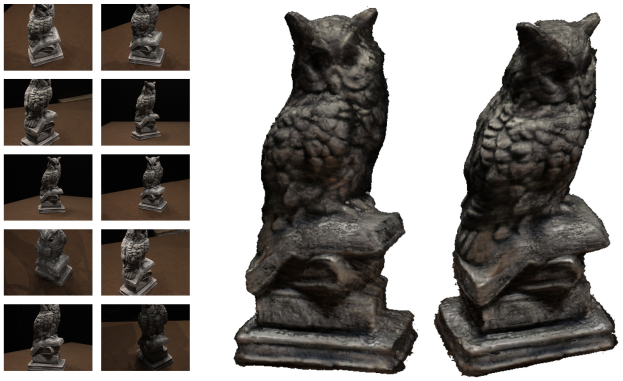
Figure 5. **Comparisons:** We compare our method on test views of scenes of DTU. Given 10 reference view images of a novel scene at test time. Our method infers sharp and detailed objects in both appearance and geometry, such as the feathers and eyes of **birds**, the letters and small benches in **buildings**, the texture of the **rabbit**, and the logo of a **detergent**. *Rabbit* and *detergent* scenes benefit most from fine-tuning. NeRF finds approximate, smooth geometry and yields blurry textures for *birds*, *buildings* and *rabbit*. For the detergent scene, it struggles to generate consistent geometry or appearance. LLFF creates some sharp image regions at the letters of *buildings* and the texture of the *rabbit* but results are usually overlaid with strong blending and ghosting effects.

4. Experiments

We, first, study the generalization ability of SRF when trained on a variety of generic objects and scenes. In Sec. 4.1, we observe that our model indeed learns generalizing structure applicable on novel scenes, given only a sparse number of views that are arbitrarily spread-out. Furthermore, we find that our model can produce 3D colored meshes from only 10 views, despite being trained for a view synthesis task as shown in Sec. 4.2. These observations suggest that incorporating *geometry and data* helps generalization. Finally, in Sec. 4.3, we show that the multi-view

	Birds			Buildings			Rabbit			Detergent		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
LLFF	18.65	0.51	0.44	15.13	0.39	0.40	17.59	0.41	0.49	14.73	0.49	0.48
NeRF	15.09	0.29	0.71	17.68	0.51	0.33	18.24	0.40	0.59	9.73	0.32	0.64
Ours	23.36	0.65	0.35	17.22	0.57	0.29	**18.79**	0.48	0.49	16.75	0.48	0.48
Ours(ft)	**24.97**	**0.72**	**0.27**	**19.71**	**0.70**	**0.18**	18.06	**0.55**	**0.40**	**16.97**	**0.60**	**0.37**

Table 1. **Quantitative Results:** Quantitative results on DTU dataset, reported in PSNR, SSIM (higher is better) and LPIPS [60] (lower is better). Ours(ft) indicates fine-tuning. We outperform all baselines consistently. SRF without fine-tuning already outperforms baselines, fine-tuned SRF produces even sharper geometry, appearance, and far fewer artifacts than all baselines.



(a) **10 Reference Views**

(b) **Meshed Prediction**
(learned without 3D supervision)

Figure 6. **Meshing Predictions.** Given only 10 images of a scene, SRF can produce colored meshes from the resulting density. We posit that SRF implicitly learns 3D reconstruction and view synthesis jointly from only 10 views even when no 3D supervision was provided during training.

structure of SRF naturally generalizes, even when learned on a *single* object for a *limited time*.

Data. We conduct our experiments on the publicly available DTU Multi-View Stereopsis Dataset [1]. It consists of 124 different scenes, including very diverse objects (E.g., buildings, statues, groceries, fruits, bricks, etc.). We split the scenes into test, validation, and training splits (see appendix for more details). We randomly sample 10 images of a scene as input to SRF. For evaluation and training purposes, we sample a different view as the target view.

Baselines We contrast our approach with **NeRF** [34]. NeRF requires scene-specific optimization. We use publicly available code to train NeRF models for each scene using 10 input images. Training a scene-specific model took 2 days. Once trained, novel views can be synthesized. We also compare to an off-the-shelf publicly available **LLFF** [33] model. Like ours, LLFF allows for generalization to test scenes[1]. Instead of a continuous 3D representation, reference images are sliced into multiple depth layers. For synthesis of a target view, neighboring reference images are warped into the target view and blended together.

4.1. Unconstrained Generalization

In this experiment, we target to learn a model that is able to perform novel view synthesis on any unseen test scene. For this, we sample a random train (109 scenes), test (10 scenes), and validation (5 scenes) split of the full DTU dataset (see appendix). We train our method until validation minimum is reached for around 3 days on a single NVIDIA Quadro RTX 8000.

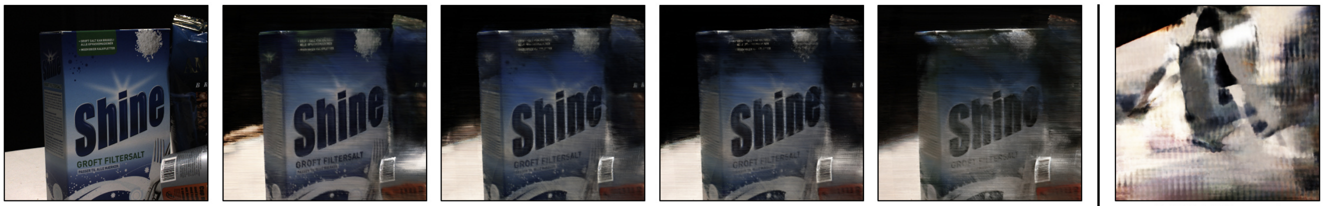
Given only 10 views of a novel scene at test time, our method is able to create sharp objects in the rendered novel views and outperforms baselines. We show qualitative analysis in Figure 5 and quantitative analysis in Table 1. Our approach generalizes to new scenes instantly and can operate on sparse and arbitrarily spread-out multi-views. Each NeRF model takes 2 days for the scene-specific optimization. Instead, our SRF can be learned from many scenes, thanks to the architecture which emulates geometric stereo matching. We find this to be key for novel view synthesis from sparse data. Moreover, we can enrich the geometric and learning concept by the idea of optimized scene representations. For this, we fine-tune our model for a short period of time. Not only does this result in sharper results as compared to the baselines, but the optimization time is also reduced from multiple days to a few minutes. We show the effect of fine-tuning our method in Fig. 7 (b)-(d) and also in Fig. 5. We observe that a NeRF model trained on the sparse and spread-out views may also lead to degenerate results as shown in Fig. 7-(f). We refer the reader to the appendix for more details.

Finally, we find that challenging BRDFs and reflective regions can pose problems for our method based on stereo matching. We observe that fine-tuning helps mitigate some issues (Fig. 9). Introducing view-dependent modeling into SRF, which we dropped in Eq. 2, may likely solve this issue and is an interesting future work direction.

4.2. Meshing Predictions

In order to mesh the prediction, we evaluate SRF conditioned on 10 images in a dense grid of points enclosing the objects. SRF predicts color and density for each point. We then threshold the density at the grid and run Marching

[1] We did not have access to the training code of LLFF. Therefore, we use an off-the-shelf model provided by the authors. It is possible that results may improve by fine-tuning the LLFF model on DTU dataset.



(a) Ground Truth

(b) Ours (fine-tuned)
Time to result: 10 hours

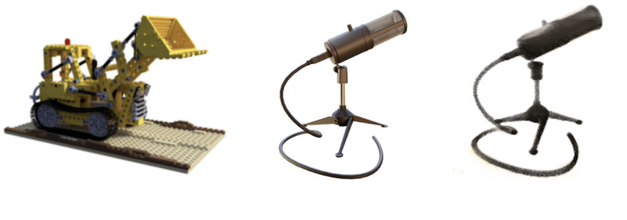
(c) Ours (fine-tuned)
Time to result: 1.5 hours

(d) **Ours** (fine-tuned)
Time to result: 15 min

(e) **Ours** (instant)
Time to result: instant

(f) **NeRF** (failiure case)
Time to result: 2 days

Figure 7. **Effect of Fine-Tuning our Method.** Our method can reconstruct geometry and appearance on challenging scenarios as it builds on *classical stereo structure* and is learned on *many scenes*. While pure NeRF struggles here **(f)**, SRF generates reasonable result **(e)**. We further improve the results by fine-tuning with the test images. We observe that around 15 minutes to be a good trade-off between quality and speed. Not only it results in sharper results as compared to the baselines, but it also reduces optimization time from 2 days to a minutes.



(a) **Single Training Object**
30 min of training

(b) **Novel Object**
ground truth

(c) **Synthesized View**
instantaneous result

Figure 8. **Natural Generalization Capability.** We train SRF only on a single object, a tractor, for as little as *30 min*, and apply it without fine-tuning to a microphone. It is apparent that geometry and some color generalize even in this extreme setting, despite the large differences in geometry and appearance between the tractor and the microphone. We attribute this to the classical stereo geometry build into our network by design.

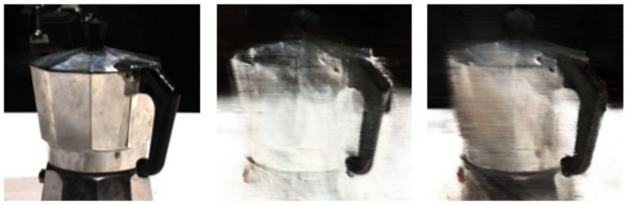
Cubes [25] to obtain a mesh. For each vertex we find on the mesh, we take its coordinates and input them to the SRF to predict color and add it to the mesh. See Fig. 6 for a result.

4.3. Natural Generalization Capability

Previously, we found that incorporating *geometry and data* helps generalization. Next, we validate that our architecture naturally generalizes by design. We take a radical setup for this: we train on a single object (a synthetic tractor [34]) for as little as 30 minutes and inspect novel view synthesis of a very different object (a microphone from NeRF data). We observe generalization despite large differences in appearance and geometry as shown in Fig. 8.

5. Discussion and Conclusion

We introduced Stereo Radiance Fields, a neural view synthesis model designed to emulate components of classical multi-view stereo. Instead of predicting radiance and color based on point-direction coordinates, we project each 3D point to multiple views, extract features, and process them in *pairs*. This learns an ensemble of scores driven only by a self-supervised rendering loss, that allow for computation of implicit correspondences. The process emulates feature matching in classical stereo within an end-to-end learnable network for view synthesis.



(a) **Reflective, textureless coffe pot**

(b) **Failure Case**
uncertain stereo matching

(c) **Fine-tunined**
reduced artifacts

Figure 9. **Limitation** Our neural architecture of SRF is strongly inspired by classical stereo matching. Modeling reflections and texture-less regions is challenging. Fine-tuning SRF ameliorate this issue, though does not totally overcome it.

Experiments demonstrate that SRF learns common structure across multiple scenes. We train a SRF model on multiple scenes from the DTU dataset, and show that SRF *generalizes*, producing realistic images. Furthermore, in contrast to prior work which requires dense views, we use arbitrarily *sparse* spread-out 10 views as input. We show that results further improve after 10-15 minutes of fine-tuning on these target 10 views. Remarkably, in the sparse view setting (10 views), our approach significantly outperforms the SOTA methods, even when we train them on the new scene for 2 days. Finally, we show that SRF implicitly compute an interpretable 3D representation allowing for colored meshing – without using 3D supervision.

In summary, SRF builds on classical multi-view stereo and recent neural rendering ideas but combines them in a unified end-to-end learnable architecture. We think the interplay of classical geometric computer vision with neural rendering is an exciting avenue, which deserves further exploration. Future work may extend them to model challenging BRDFs, and 4D space-time view synthesis of dynamic scenes from in-the-wild samples that are inherently sparse.

Acknowledgments. We thank the RVH group for their feedback. This work is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - 409792180 (Emmy Noether Programme, project: Real Virtual Humans).

References

[1] Henrik Aanæs, Rasmus Ramsbøl Jensen, George Vogiatzis, Engin Tola, and Anders Bjorholm Dahl. Large-scale data for multiple-view stereopsis. *IJCV*, 2016.

[2] Edward H Adelson, James R Bergen, et al. *The plenoptic function and the elements of early vision*, volume 2. Vision and Modeling Group, Media Laboratory, Massachusetts Institute of . . . , 1991.

[3] Kara-Ali Aliev, Artem Sevastopolsky, Maria Kolos, Dmitry Ulyanov, and Victor Lempitsky. Neural point-based graphics. *arXiv preprint arXiv:1906.08240*, 2019.

[4] Aayush Bansal, Xinlei Chen, Bryan Russell, Abhinav Gupta, and Deva Ramanan. Pixelnet: Representation of the pixels, by the pixels, and for the pixels. *arXiv preprint arXiv:1702.06506*, 2017.

[5] Aayush Bansal, Minh Vo, Yaser Sheikh, Deva Ramanan, and Srinivasa Narasimhan. 4d visualization of dynamic events from unconstrained multi-view videos. In *CVPR*, 2020.

[6] Bharat Lal Bhatnagar, Cristian Sminchisescu, Christian Theobalt, and Gerard Pons-Moll. Combining implicit function learning and parametric models for 3d human reconstruction. In *ECCV*. Springer, 2020.

[7] 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.*, 2020.

[8] Shenchang Eric Chen and Lance Williams. View interpolation for image synthesis. In *Conference on Computer Graphics and Interactive Techniques*, 1993.

[9] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *CVPR*, 2019.

[10] Julian Chibane, Thiemo Alldieck, and Gerard Pons-Moll. Implicit functions in feature space for 3d shape reconstruction and completion. In *CVPR*, 2020.

[11] Julian Chibane, Aymen Mir, and Gerard Pons-Moll. Neural unsigned distance fields for implicit function learning. In *NeurIPS*, 2020.

[12] Inchang Choi, Orazio Gallo, Alejandro Troccoli, Min H Kim, and Jan Kautz. Extreme view synthesis. In *CVPR*, 2019.

[13] Robert T Collins. A space-sweep approach to true multi-image matching. In *CVPR*, 1996.

[14] 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 *CVPR*, 2019.

[15] John Flynn, Ivan Neulander, James Philbin, and Noah Snavely. Deepstereo: Learning to predict new views from the world's imagery. In *CVPR*, 2016.

[16] Steven J Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F Cohen. The lumigraph. In *ACM Trans. Graph.*, 1996.

[17] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.

[18] Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. Deep blending for free-viewpoint image-based rendering. *ACM Trans. Graph.*, 2018.

[19] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. Deepmvs: Learning multi-view stereopsis. In *CVPR*, 2018.

[20] Nima Khademi Kalantari, Ting-Chun Wang, and Ravi Ramamoorthi. Learning-based view synthesis for light field cameras. *ACM Trans. Graph.*, 2016.

[21] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 2015.

[22] Marc Levoy and Pat Hanrahan. Light field rendering. In *Annual Conference on Computer Graphics and Interactive Techniques*. ACM, 1996.

[23] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *CVPR*, 2021.

[24] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *ACM Trans. Graph.*, 2019.

[25] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Computer Graphics and Interactive Techniques*, 1987.

[26] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004.

[27] 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 *CVPR*, 2021.

[28] N. Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1995.

[29] Leonard McMillan. *An image-based approach to three-dimensional computer graphics*. PhD thesis, University of North Carolina, Chapel Hill, 1997.

[30] Leonard McMillan and Gary Bishop. Plenoptic modeling: An image-based rendering system. In *Conference on Computer Graphics and Interactive Techniques*, 1995.

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

[32] Moustafa Meshry, Dan B. Goldman, Sameh Khamis, Hugues Hoppe, Rohit Pandey, Noah Snavely, and Ricardo Martin-Brualla. Neural rerendering in the wild. In *CVPR*, 2019.

[33] 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.*, 2019.

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

[35] Ren Ng. *Light field photography with a hand-held plenoptic camera*. PhD thesis, Stanford University, 2005.

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

[37] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-NeRF: Neural Radiance Fields for Dynamic Scenes. In *CVPR*, 2021.

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

[39] Ignacio Rocco, Mircea Cimpoi, Relja Arandjelović, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Neighbourhood consensus networks. In *NeurIPS*, 2018.

[40] 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 *ICCV*, 2019.

[41] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 2002.

[42] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016.

[43] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *ECCV*, 2016.

[44] Steven M Seitz and Charles R Dyer. View morphing. In *Conference on Computer Graphics and Interactive Techniques*, 1996.

[45] Jonathan Shade, Steven Gortler, Li-wei He, and Richard Szeliski. Layered depth images. In *Conference on Computer Graphics and Interactive Techniques*, 1998.

[46] Harry Shum and Sing Bing Kang. Review of image-based rendering techniques. In *Visual Communications and Image Processing*, 2000.

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

[48] Pratul P Srinivasan, Richard Tucker, Jonathan T Barron, Ravi Ramamoorthi, Ren Ng, and Noah Snavely. Pushing the boundaries of view extrapolation with multiplane images. In *CVPR*, 2019.

[49] Richard Szeliski. Image alignment and stitching: A tutorial. *Foundations and Trends in Computer Graphics and Vision*, 2006.

[50] Richard Szeliski. *Computer Vision: Algorithms and Applications*. Springer Science & Business Media, 2010.

[51] Richard Szeliski and Polina Golland. Stereo matching with transparency and matting. *IJCV*, 1999.

[52] 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. *Eurographics*, 2020.

[53] Sebastian Thrun. Is learning the n-th thing any easier than learning the first? In *NeurIPS*, 1996.

[54] Sebastian Thrun. Lifelong learning algorithms. In *Learning to learn*. Springer, 1998.

[55] Alex Trevithick and Bo Yang. Grf: Learning a general radiance field for 3d scene representation and rendering. *arXiv preprint arXiv:2010.04595*, 2020.

[56] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *CVPR*, 2021.

[57] Wenqi Xian, Jia-Bin Huang, Johannes Kopf, and Changil Kim. Space-time neural irradiance fields for free-viewpoint video. In *CVPR*, 2021.

[58] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. In *NeurIPS*, 2020.

[59] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. Pixelnerf: Neural radiance fields from one or few images. In *CVPR*, 2021.

[60] 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.

[61] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: learning view synthesis using multiplane images. *ACM Trans. Graph.*, 2018.