UNISURF: Unifying Neural Implicit Surfaces and Radiance Fields for Multi-View Reconstruction

Michael Oechsle\textsuperscript{1,2,3} Songyou Peng\textsuperscript{1,4} Andreas Geiger\textsuperscript{1,2}
\textsuperscript{1}Max Planck Institute for Intelligent Systems, T"ubingen \textsuperscript{2}University of T"ubingen \textsuperscript{3}ETAS GmbH, Stuttgart \textsuperscript{4}ETH Zurich
\{firstname.lastname\}@tue.mpg.de

Abstract

Neural implicit 3D representations have emerged as a powerful paradigm for reconstructing surfaces from multi-view images and synthesizing novel views. Unfortunately, existing methods such as DVR or IDR require accurate per-pixel object masks as supervision. At the same time, neural radiance fields have revolutionized novel view synthesis. However, NeRF’s estimated volume density does not admit accurate surface reconstruction. Our key insight is that implicit surface models and radiance fields can be formulated in a unified way, enabling both surface and volume rendering using the same model. This unified perspective enables novel, more efficient sampling procedures and the ability to reconstruct accurate surfaces without input masks. We compare our method on the DTU, BlendedMVS, and a synthetic indoor dataset. Our experiments demonstrate that we outperform NeRF in terms of reconstruction quality while performing on par with IDR without requiring masks.

1. Introduction

Capturing the geometry and appearance of 3D scenes from a set of images is one of the cornerstone problems in computer vision. Towards this goal, coordinate-based neural models have emerged as a powerful tool for 3D reconstruction of geometry and appearance within the last years.

Many recent methods employ continuous implicit functions parameterized with neural networks as 3D representations of geometry \cite{ OccupancyNetworks, MarchingCubes, NeRF, IDR, DVR, DVR++, HGD, AMoV, GeomOpt} or appearance \cite{NeRF, IDR, NeRF++, NeRF++, IDR++, IDR, HGD}. These neural 3D representations have shown impressive performance on geometry reconstruction and novel view synthesis from multi-view images. Besides the choice of the 3D representation (e.g., occupancy field, unsigned or signed distance field), one key element for neural implicit multi-view reconstruction is the rendering technique. While some of these works represent the implicit surface as level set and hence render the appearance from surfaces \cite{NeRF, NeRF++, IDR}, others integrate densities by drawing samples along the viewing rays \cite{DVR, NeRF, IDR}.

In existing work, surface rendering techniques have shown impressive performance in 3D reconstruction \cite{NeRF, IDR}. However, they require per-pixel object masks as input and an appropriate network initialization since surface rendering techniques only provide gradient information locally where a surface intersects with a ray. Intuitively speaking, optimizing wrt. local gradients can be seen as an iterative deformation procedure applied to an initial neural surface which is often initialized as a sphere. Additional constraints such as mask supervision are necessary for converging to a valid surface, see Fig. 2 for an illustration. Due to their reliance on masks, surface rendering methods are limited to object-level reconstruction and do not scale to larger scenes.

On contrary, volume rendering methods like NeRF \cite{NeRF} have shown impressive results for novel view synthesis, also for larger scenes. However, surfaces extracted as level sets of the underlying volume density are usually non-smooth...
2. Related Work

In this section, we first discuss related work from the domain of 3D reconstruction from multi-view images. Next, we provide an overview of recent works on neural implicit representations as well as differentiable rendering.

3D Reconstruction from Multi-View Images: Reconstructing 3D geometry from multiple images has been a longstanding computer vision problem [14]. Before the era of deep learning, classic multi-view stereo (MVS) methods [2, 4, 5, 7, 20, 20, 48, 50, 51] focus on either matching features across views [4, 48] or representing shapes with a voxel grid [2, 5, 7, 20, 42, 50, 54, 55]. The former approaches usually have a complex pipeline requiring additional steps like fusing depth information [9, 30] and meshing [18, 19], while the latter ones are limited to low resolution due to cubic memory requirements. In contrast, neural implicit representations for 3D reconstruction do not suffer from discretization artifacts as they represent surfaces by the level set of a neural network with continuous outputs.

Recent learning-based MVS methods attempt to replace some parts of the classic MVS pipeline. For instance, some works learn to match 2D features [15, 21, 26, 56, 62], fuse depth maps [11, 46], or infer depth maps [16, 58, 59] from multi-view images. Contrary to these learning-based MVS approaches, our method only requires weak 2D supervision during optimization. Moreover, our method yields high-quality 3D geometry and synthesizes photorealistic and consistent novel views.

Neural Implicit Representations: Recently, neural implicit functions have emerged as an effective representation of 3D geometry [3, 8, 12, 31, 32, 37, 41, 43, 47, 57] and appearance [22, 24, 34, 38, 39, 40, 47, 49, 52] as they represent 3D content continuously and without discretization while simultaneously having a small memory footprint. Most of these methods require 3D supervision. However, several recent works [23, 34, 38, 52, 61] demonstrated differentiable rendering for training directly from images [23, 34, 38, 52, 61]. We divide these methods into two groups: surface rendering and volume rendering.

Surface rendering approaches, including DVR [38] and IDR [61], determine the radiance directly on the surface of an object and provide a differentiable rendering formulation using implicit gradients. This allows for optimizing neural implicit surfaces from multi-view images. Conditioning on the viewing direction allows IDR to capture a high level of detail, even in the presence of non-lambertian surfaces. However, both DVR and IDR require pixel-accurate object masks for all views as input. In contrast, our method leads to similar reconstructions without requiring masks.

NeRF [34] and follow-ups [6, 28, 35, 36, 44, 45, 49, 53, 63] use volume rendering by learning alpha-compositing of a radiance field along rays. This method has shown impres-
sive results on novel view synthesis and does not require mask supervision. However, the recovered 3D geometry is far from satisfactory, see Fig. 3. Several follow-up works (Neural Body [44] D-NeRF [45] and NeRD [6]) extract meshes using the volume density from NeRF, but none of them considers optimizing surfaces directly. Unlike these works, we aim at capturing accurate geometry and propose a volume rendering formulation that provably approaches surface rendering in the limit.

### 3. Background

The two main ingredients for learning neural implicit 3D representations from multi-view images are the 3D representation and the rendering technique linking the 3D representation and the 2D observations. This section provides the relevant background on implicit surface and volumetric radiance representations which we unify in this paper for the case of solid (non-transparent) objects and scenes.

**Implicit Surface Models:** Occupancy Networks [31, 38] represent surfaces as the decision boundary of a binary occupancy classifier, parameterized by a neural network

\[
o_\theta(x) : \mathbb{R}^3 \rightarrow [0, 1]
\]

where \( x \in \mathbb{R}^3 \) is a 3D point and \( \theta \) are the model parameters. The surface is defined as the set of all 3D points where the occupancy probability is one half: \( S = \{ x \mid o_\theta(x) = 0.5 \} \).

To associate a color with every 3D point \( x \) on the surface, a color field \( c_\theta(x) \) can be learned jointly with the occupancy field \( o_\theta(x) \). The color for a particular pixel/ray \( r \) is thus predicted as follows

\[
\hat{C}(r) = c_\theta(x_s)
\]

where \( x_s \) is retrieved by root finding along ray \( r \), see [38] for details. The parameters \( \theta \) of the occupancy field \( o_\theta \) and the color field \( c_\theta \) are determined by optimizing a reconstruction loss via gradient descent as described in [24, 38, 61].

While surface rendering allows for accurately estimating geometry and appearance, existing approaches strongly rely on supervision with object masks as surface rendering methods are only able to reason about rays that intersect a surface.

**Volumetric Radiance Models:** In contrast to implicit surface models, NeRF [34] represents scenes as colorized volume densities and integrates radiance along rays via alpha blending [25, 34]. More specifically, NeRF uses a neural network to map a 3D location \( x \in \mathbb{R}^3 \) and a viewing direction \( d \in \mathbb{R}^3 \) to a volume density \( \sigma_\theta(x) \in \mathbb{R}^+ \) and a color value \( c_\theta(x, d) \in \mathbb{R}^3 \). Conditioning on the viewing direction \( d \) allows for modeling view-dependent effects such as specular reflections [34, 40] and improves reconstruction quality in case the Lambertian assumption is violated [61]. Let \( o \) denote the location of the camera center. Given \( N \) samples \( \{ x_i \} \) along ray \( r = o + t d \), NeRF approximates the color of pixel/ray \( r \) using numerical quadrature:

\[
\hat{C}(r) = \sum_{i=1}^{N} T_i \left( 1 - \exp \left( -\sigma_\theta(x_i) \delta_i \right) \right) c_\theta(x_i, d)
\]

\[
T_i = \exp \left( -\sum_{j<i} \sigma_\theta(x_j) \delta_j \right)
\]

Here, \( T_i \) is the accumulated transmittance along the ray and \( \delta_i = |x_{i+1} - x_i| \) is the distance between adjacent samples. As Eq. (3) is differentiable, the parameters \( \theta \) of the density field \( \sigma_\theta \) and the color field \( c_\theta \) can be estimated by optimizing a reconstruction loss. We refer to [34] for details.

While NeRF does not require object masks for training due to its volumetric radiance representation, extracting the scene geometry from the volume density requires careful tuning of the density threshold and leads to artifacts due to the ambiguity present in the density field, see Fig. 3.

### 4. Method

We now describe our main contribution. In contrast to NeRF which is also applicable to non-solid scenes (e.g., fog, smoke), we restrict our focus to solid objects that can be represented by 3D surfaces and view-dependent surface colors. Our method exploits both, the power of volumetric radiance representations to learn coarse scene structure without mask supervision as well as surface rendering which acts as an inductive bias to represent objects by a set of precise 3D surfaces, leading to accurate reconstructions.
4.1. Unifying Surface and Volume Rendering

We start by noting that Eq. (3) can be rewritten as\(^3\)

\[
\hat{C}(r) = \sum_{i=1}^{N} \alpha_i(x_i) \prod_{j<i} (1 - \alpha_j(x_j)) c(x_i, d) \tag{5}
\]

with alpha values \(\alpha_i(x) = 1 - \exp(-\sigma(x) \delta_i)\). Assuming solid objects, \(\alpha\) becomes a discrete occupancy indicator variable \(o \in \{0, 1\}\) which either takes \(o = 0\) in free space and \(o = 1\) in occupied space as value:

\[
\hat{C}(r) = \sum_{i=1}^{N} o(x_i) \prod_{j<i} (1 - o(x_j)) c(x_i, d) \tag{6}
\]

We recognize this expression as the image formation model for solid objects [55] where the term \(o(x_i) \prod_{j<i} (1 - o(x_j))\) evaluates to 1 for the first occupied sample \(x_i\) along ray \(r\) and to 0 for all other samples. \(\prod_{j<i} (1 - o(x_j))\) is an indicator for visibility which is 1 if there exists no occupied sample \(x_j\) with \(j < i\) before sample \(x_i\). Thus, \(\hat{C}(r)\) takes the color \(c(x_i, d)\) of the first occupied sample along ray \(r\).

To unify implicit surface and volumetric radiance models, we parameterize \(o\) directly by a continuos occupancy field \(o_\theta(1)\) as opposed to predicting volume density \(\sigma\). Following [61], we condition the color field \(c_\theta\) on the surface normal \(n\) and a feature vector \(h\) of the geometry network which empirically induces a useful bias as also observed in [61] for the case of implicit surfaces. Importantly, our unified formulation allows for both volume and surface rendering:

\[
\hat{C}_v(r) = \sum_{i=1}^{N} o_\theta(x_i) \prod_{j<i} (1 - o_\theta(x_j)) c_\theta(x_i, n_i, h_i, d) \tag{7}
\]

\[
\hat{C}_s(r) = c_\theta(x_s, n_s, h_s, d) \tag{8}
\]

where \(x_s\) is retrieved by root-finding along ray \(r\) and \(n_s, h_s\) denote the normal and geometry features at \(x_s\), respectively. Note that \(x_s\) depends on the occupancy field \(o_\theta\), but we have dropped this dependency here for clarity. For further details, we refer the reader to the supplementary material.

The advantage of this unified formulation is that it allows for both rendering on the surface directly and rendering throughout the entire volume which enables gradually removing ambiguities during optimization. As evidenced by our experiments, it is indeed critical to combine both for obtaining accurate reconstructions without mask supervision. Being able to quickly recover the surface \(S\) via root-finding enables more effective volume rendering, successively focusing on and refining the object surfaces as we will describe in Section 4.3. Furthermore, surface rendering enables faster novel view synthesis as illustrated in Fig. 5.

\(^3\)We drop dependencies on the model parameters for clarity.

4.2. Loss Function

We optimize the following regularized loss function

\[
\mathcal{L} = \mathcal{L}_{\text{rec}} + \lambda \mathcal{L}_{\text{reg}} \tag{9}
\]

with \(\ell_1\) reconstruction loss and \(\ell_2\) surface regularization which encourages the normal of a surface point \(x_s\) and a point sampled in its neighborhood to be similar:

\[
\mathcal{L}_{\text{rec}} = \sum_{r \in \mathcal{R}} \|\hat{C}_v(r) - C(r)\|_1 \tag{10}
\]

\[
\mathcal{L}_{\text{reg}} = \sum_{x_s \in S} \|n(x_s) - n(x_s + \epsilon)\|_2 \tag{11}
\]

Here, \(\mathcal{R}\) denotes the set of all pixels/rays in the minibatch, \(S\) is the set of corresponding surface points, \(C(r)\) is the observed color for pixel/ray \(r\) and \(\epsilon\) is a small random uniform 3D perturbation. The normal at \(x_s\) is given by

\[
n(x_s) = \frac{\nabla_{x_s} o_\theta(x_s)}{\|\nabla_{x_s} o_\theta(x_s)\|_2} \tag{12}
\]

which can be computed using double backpropagation [38].

4.3. Optimization

The key hypothesis of implicit surface models [38,61] is that only the region at the first intersection point with the surface contributes to the rendering equation. However, this assumption is not true during early iterations where the surface is not well defined. Consequently, existing methods [38,61] require strong mask supervision. Conversely, during later iterations, knowledge of the approximate surface is valuable for drawing informative samples when evaluating the volume rendering equation in Eq. (7). Therefore, we utilize a training schedule with a monotonically decreasing sampling interval for drawing samples during volume rendering, as visualized in Fig. 4. In other words, during early iterations, the samples \(\{x_i\}\) cover the entire optimization volume, effectively bootstrapping the reconstruction process using volume rendering. During later iterations, the samples \(\{x_i\}\) are drawn closer around the estimated surface. As the surface can be estimated directly from the occupancy field \(o_\theta\) via root-finding [38], this eliminates the need for hierarchical two-stage sampling as in NeRF. Our experiments demonstrate that this procedure is particularly effective for estimating accurate geometry, while it allows for resolving ambiguities during early iterations.

More formally, let \(x_s = o + t_s d\). We obtain samples \(x_i = o + t_i d\) by drawing \(N\) depth values \(t_i\) using stratified sampling within the interval \([t_s - \Delta, t_s + \Delta]\) centered at \(t_s\):

\[
t_i \sim U \left[ t_s + \left( \frac{2i - 2}{N} - 1 \right) \Delta, t_s + \left( \frac{2i}{N} - 1 \right) \Delta \right] \tag{13}
\]
During training, we start with a large interval $\Delta_{\text{max}}$ and gradually decrease $\Delta$ for more accurate sampling and optimization of the surface using the following decay schedule

$$\Delta_k = \max(\Delta_{\text{max}} \exp(-k \beta), \Delta_{\text{min}})$$

where $k$ denotes the iteration number and $\beta$ is a hyperparameter. In fact, it can be shown that for $\Delta \to 0$ and $N \to \infty$, volume rendering (7) indeed approaches surface rendering (8): $C_v(\mathbf{r}) \to C_s(\mathbf{r})$. A formal proof of this limit is provided in the supplementary material.

As evidenced by our experiments, the decay schedule in (14) is critical for capturing detailed geometry as it combines volume rendering of large and uncertain volumes in the beginning of training with surface rendering towards the end of training. To reduce free space artifacts, we combine these samples with points sampled randomly between the camera and the surface. For rays without surface intersection, we use stratified sampling on the entire ray.

### 4.4. Implementation Details

**Architecture:** Similar to Yariv et al. [61], we use an 8-layer MLP with a Softplus activation function and a hidden dimension of 256 for the occupancy field $o_\theta$. We initialize the network such that the decision boundary is a sphere [13]. In contrast, the radiance field $c_\theta$ is parameterized as a 4-layer ReLU MLP. We encode the 3D location $x$ and the viewing direction $d$ using Fourier features [34] at $k$ octaves. We empirically found $k = 6$ for the 3D location $x$ and $k = 4$ for the viewing direction $d$ to work best.

**Optimization:** In all experiments, we fit our model to multi-view images of a single scene. During optimization of the model parameters, we first randomly sample a view and then $M$ pixels/rays $\mathcal{R}$ from this view based on the camera intrinsics and extrinsics. Next, we render all rays to compute the loss function in Eq. (9). For root-finding, we use 256 uniform sampled points and apply the secant method with 8 steps [31]. For our rendering procedure, we use $N = 64$ query points inside the interval and 32 in the free space between the camera and the lower bound of the interval. The interval decay parameters are $\beta = 1.5e-5$, $\Delta_{\text{min}} = 0.05$ and $\Delta_{\text{max}} = 1.0$. We use Adam with a learning rate of 0.0001 and optimize for $M = 1024$ pixels per iteration with two decay steps after 200k and 400k iterations. In total, we train our models for 450k iterations.

**Inference:** Our method allows for inferring 3D shapes as well as for synthesizing novel view images. For synthesizing images, we can render our representation in two different ways, we can either use volume rendering or surface rendering. In Fig. 5, we show that both rendering ap-

---

4 We used a single NVIDIA V100 GPU to measure the inference time.

### 5. Experimental Evaluation

We conduct experiments on multi-view 3D reconstruction to assess our method. First, we provide a qualitative and quantitative comparison of our approach to existing methods (IDR [61], NeRF [34], COLMAP [48]) on the widely used DTU MVS [17]. Second, we show a qualitative comparison for samples from the BlendedMVS dataset [60] and synthetic renderings of scenes from the SceneNet dataset [29]. Third, we analyze our rendering procedure and loss function in an ablation study. In the supplementary, we provide results on the LLFF dataset [33].

#### 5.1. Baselines

To validate the effectiveness of our method, we compare it to three different baselines.

**COLMAP [48]:** We consider COLMAP [48] as a classical MVS baseline, as it shows strong performance on multi-view reconstruction and is widely used in related works [38, 61]. We reconstruct a mesh from the output of COLMAP using screened Poisson Surface reconstruction (sPSR) [19]. Following [38, 61], we show results for the quantitatively best trim parameter 7 and for the setting that results in watertight meshes (trim parameter 0).

**NeRF [34]:** Although NeRF targets novel view synthesis, its volume density admits the extraction of geometry. To extract meshes from NeRF, we define a density threshold of 50. We validate this choice in the supplementary.

**IDR [61]:** IDR is the state-of-the-art multi-view reconstruction method for neural implicit surfaces. IDR reconstructs surfaces with an impressively high level of detail and han-
5.2. Datasets

**DTU MVS Dataset [17]:** The DTU MVS dataset contains 49 to 64 images at a resolution of 1200 × 1600 as well as extrinsic and intrinsic camera parameters for all views. The dataset consists of objects with different shapes and appearances. Non-lambertian appearance effects make some of the objects particularly challenging. For each scan, ground truth 3D shapes, as well as the official evaluation procedure, are available. Like previous works [38, 61], we use the “Surface” method of the evaluation script, and evaluate all methods on meshes cleaned with the respective masks. The official evaluation procedure calculates the Chamfer distance between sampled points of the predicted shapes and ground truth shapes provided in the dataset. For evaluating IDR [61], we use the pixel-accurate masks for all images which are provided by the authors of IDR.

**BlendedMVS Dataset [60]:** The BlendedMVS dataset is a large-scale dataset containing multi-view images with respective camera extrinsics and intrinsics. We use examples from the BlendedMVS low-res set with an image resolution of 768 × 576. These examples contain 24 to 64 different views of unmasked images. We define the scene as such that the object is in the center and the closest camera lies near the unit sphere.

**SceneNet Dataset [29]:** For testing our model on complex indoor scenes, we take two scenes from [29] for evaluation. BlenderProc [10] is applied for rendering images of a part of the scenes containing multiple objects. The first scene is a bedroom scene with a bed, a lamp and a bedside table. The other scene is a living room scene with a sofa, a curtain and a round table. We use 83 and 40 images, respectively.

5.3. Comparison on DTU

In Table 1, we quantitatively compare our method to the baselines on the DTU MVS dataset. While the COLMAP baseline with trim parameter \( \zeta = 7 \) shows the best performance on the Chamfer distance, it produces non-watertight meshes with incomplete surfaces. Our method performs nearly on par with the state-of-the-art neural implicit model IDR while not relying on strong mask supervision. NeRF and COLMAP (\( \zeta = 0 \)) also do not use input masks but exhibit worse performance in terms of Chamfer distance.

In Fig. 6, we show qualitative results for our method and the baselines. While COLMAP provides detailed reconstructions, it leads to incomplete geometry due to trimming. For NeRF, holes and noise artifacts can be observed in the reconstructions. In contrast, our approach and IDR (with

![Table 1: Quantitative Comparison on DTU [17].](https://roboimagedata.compute.dtu.dk)

To show our model’s capabilities on more diverse scenes, we use samples of the BlendedMVS dataset and indoor scenes from SceneNet. As there exist no object masks for these scenes, we only consider COLMAP and NeRF as baseline methods. While we have tried running IDR, none of the scenes converged, resulting in degenerate outputs. For scenes with complex backgrounds, we use a background model that learns to capture appearance information outside of our region interest, we refer the reader to the supplementary material for more details.

Our qualitative results in Fig. 8 provide evidence that, unlike existing implicit surface models, our method is able to reconstruct plausible geometry for complex scenes with multiple objects and backgrounds. While COLMAP works well on indoor scenes, it shows artifacts on uniformly colored regions (e.g., round table in the second row). As for the BlendedMVS experiment, NeRF can reason about the overall spatial structure but shows less accurate surfaces with significantly higher levels of noise compared to UNISURF. More results can be found in the supplementary material.
<table>
<thead>
<tr>
<th>GT view</th>
<th>COLMAP [48] $(\zeta = 7)$</th>
<th>NeRF [34]</th>
<th>Ours</th>
<th>IDR [61]</th>
</tr>
</thead>
</table>

Figure 6: **Reconstructed 3D Shapes for DTU.** We show a qualitative comparison of reconstructed surfaces. We use COLMAP [48], NeRF [34] and IDR [61] as baselines. While IDR requires pixel-accurate object masks for all images, the other methods reason about the full scene extent. Our method performs visually on par with IDR while not requiring masks.
Figure 7: **Ablation Study.** We compare our method to different rendering procedures: Surface rendering in isolation (SR), volume rendering with uniformly sampled points (Uniform VR) and hierarchical volume rendering (HVR). This analysis on the Buddha scene (DTU) shows that our full method performs best, both visually and quantitatively. Moreover, our model without surface regularizer (Ours w/o Surf Reg) leads to non-smooth regions, especially at the textureless flat table.

<table>
<thead>
<tr>
<th>Method</th>
<th>Chamfer Distance (not converging)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR</td>
<td>3.70</td>
</tr>
<tr>
<td>Uniform VR</td>
<td>3.22</td>
</tr>
<tr>
<td>HVR</td>
<td>0.48</td>
</tr>
<tr>
<td>Ours w/o Surf Reg</td>
<td>0.46</td>
</tr>
<tr>
<td>Ours</td>
<td></td>
</tr>
</tbody>
</table>

5.5. Ablation study

We finally investigate the impact of rendering design choices and show ablations of the loss function.

**Rendering Procedure:** We argue for our choices of the rendering procedure by comparing different variations in Fig. 7: First, we consider a baseline which only uses surface rendering during optimization (SR). We use an $\ell_1$ reconstruction loss on the surface color and backpropagate through implicit differentiation following [38]. A second baseline uses uniform volume rendering with 96 query points (Uniform VR). Third, we use NeRF’s hierarchical volume sampling (HVR) [34] with 64+64 sample points for querying the occupancy field. While surface rendering does not converge, Uniform VR and HVR result in overly smooth and bloated shapes with missing details. This provides evidence that the proposed unified model leads to more accurate reconstructions compared to the baselines.

**Losses:** In Fig. 7, we also show an ablation study of our surface regularization term in (9). Without this regularizer surfaces become less smooth in flat and ambiguous areas, e.g., at the table. This regularization term is particularly useful for regions that are observed less frequently, as it incorporates an inductive bias towards smooth surfaces.

6. Discussion and Conclusion

This work presents UNISURF, a unified formulation of implicit surfaces and radiance fields for capturing high-quality implicit surface geometry from multi-view images without input masks. We believe that neural implicit surfaces and advanced differentiable rendering procedures play a key role in future 3D reconstruction methods. Our unified formulation shows a path towards optimizing implicit surfaces in a more general setting than possible before.

**Limitations:** By design, our model is limited to represent solid, non-transparent surfaces. Overexposed and texture-less regions are also limiting factors that lead to inaccuracies and non-smooth surfaces. Furthermore, the reconstructions are less accurate at rarely visible regions in the images. Limitations are discussed in more detail in the supplementary.

In future work, for resolving ambiguities from rarely visible and texture-less regions, a prior is necessary for reconstruction. While we incorporate an explicit smoothness prior during optimization, learning a probabilistic neural surface model which captures regularities and uncertainty across objects would help to resolve ambiguities, leading to more accurate reconstructions.

**Acknowledgement** This work was supported by an NVIDIA research gift, the ERC Starting Grant LEGO-3D (850533) and the DFG EXC number 2064/1 - project number 390727645. Songyou Peng is supported by the Max Planck ETH Center for Learning Systems.
References


[29] John McCormac, Ankur Handa, Stefan Leutenegger, and Andrew J. Davison. Scenenet RGB-D: can 5m synthetic images beat generic imagenet pre-training on indoor segmentation? In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2017. 2, 5, 6, 8
[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. on Graphics, 2019. 5
[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 Proc. of the European Conf. on Computer Vision (ECCV), 2020. 1, 2, 3, 5, 7, 8
potentials. In Proc. of the International Conf. on 3D Vision (3DV), 2015. 2, 4

[56] Benjamin Ummenhofer, Huizhong Zhou, Jonas Uhrig, Niko-
lau Mayer, Eddy Ilg, Alexey Dosovitskiy, and Thomas
Brox. Demon: Depth and motion network for learning
monocular stereo. In Proc. IEEE Conf. on Computer Vision
and Pattern Recognition (CVPR), 2017. 2

[57] Weiyue Wang, Xu Qiangeng, Duygu Ceylan, Radomir
Mech, and Ulrich Neumann. DISN: Deep implicit surface

[58] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan.

[59] Yao Yao, Zixin Luo, Shiwei Li, Tianwei Shen, Tian Fang,
and Long Quan. Recurrent mvsnet for high-resolution multi-
view stereo depth inference. Proc. IEEE Conf. on Computer
Vision and Pattern Recognition (CVPR), 2019. 2

[60] Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren,
Lei Zhou, Tian Fang, and Long Quan. Blendedmvs: A large-

