S-VolSDF: Sparse Multi-View Stereo Regularization of Neural Implicit Surfaces

Haoyu Wu    Alexandros Graikos    Dimitris Samaras
Stony Brook University
{haoyuwu,agraikos,samaras}@cs.stonybrook.edu

Abstract

Neural rendering of implicit surfaces performs well in 3D vision applications. However, it requires dense input views as supervision. When only sparse input images are available, output quality drops significantly due to the shape-radiance ambiguity problem. We note that this ambiguity can be constrained when a 3D point is visible in multiple views, as is the case in multi-view stereo (MVS). We thus propose to regularize neural rendering optimization with an MVS solution. The use of an MVS probability volume and a generalized cross entropy loss leads to a noise-tolerant optimization process. In addition, neural rendering provides global consistency constraints that guide the MVS depth hypothesis sampling and thus improves MVS performance. Given only three sparse input views, experiments show that our method not only outperforms generic neural rendering models by a large margin but also significantly increases the reconstruction quality of MVS models.

1. Introduction

Neural surface reconstruction techniques, coupled with coordinate-based neural network models, have become increasingly popular in the field of 3D vision [74, 61, 75]. Although these methods perform very well, they require dense input views as supervision. This is limiting for many real-world applications where sparse input images are the only source of information, such as robotics, augmented reality, autonomous driving, and scene reconstruction in-the-wild. As shown in Fig. 1, the reconstruction quality of a scene using VolSDF [74] (a state-of-the-art technique) drops significantly when only 3 views are used. This is due to the shape-radiance ambiguity problem [82].

The shape-radiance ambiguity [82] means that there is a high probability an incorrect geometry reconstruction satisfies the photometric constraint when it is visible from a single view only, as is in the case of sparse views. In that scenario, the photometric loss alone cannot guide the model toward a correct solution. To regularize this, we need to constrain surface points to be visible from multiple views, hence, we need correspondences, as in multi-view stereo (MVS) [7, 38, 68, 71, 72, 9, 77, 21, 70, 80, 66, 69, 63, 84, 86, 16]. Thus, we propose to guide neural rendering optimization with information from MVS. The challenge is how to effectively incorporate the noisy MVS predictions into the neural rendering pipeline.

Many modern MVS methods [71, 21, 16, 9] integrate the evidence for each possible 3D point into a probability volume and regress depth from it. In order to avoid possible errors in MVS 3D point reconstruction, we do not use point estimates, but the whole probability volume. We also note that the rendering weights in neural rendering methods and the probability volume in MVS actually have the same meaning: the probability that a point at a particular location is visible by multiple views. Based on recent MVS literature [46], we can think of all possible 3D points on a ray as interior or exterior to the object (i.e. a binary classification problem). Thus, we can treat the MVS probability
volume as a set of noisy labels for the rendering weights (i.e. occupancy values). Posing neural rendering as a classification problem allows the use of cross entropy loss to optimize neural rendering methods. However, as shown by the classification literature [85, 54], the cross entropy loss is sensitive to noisy labels. Instead, we adopt a generalized cross entropy loss [85] to reduce the penalty on false positive MVS predictions and thus increase the optimization's tolerance to noise.

In order to produce our final geometry, we want to take advantage of global consistency constraints including photometric consistency and surface smoothness imposed by neural rendering. Thus, we propose to incorporate neural surface reconstruction into coarse-to-fine MVS models. Specifically, we use the coarse stage MVS predictions to regularize neural surface optimization. Then, we use the rendered depth maps to guide the next stage’s depth hypothesis sampling in MVS. Moreover, neural surface optimization only requires 10-15 minutes in current hardware to obtain good results because of strong geometry cues from MVS. As a result, we obtain much better surface reconstruction than either MVS or neural rendering alone, at a relatively fast speed.

In this paper, we propose $S$-VolSDF, a novel approach that leverages multi-view stereo priors to optimize neural surface reconstruction with sparse input views. Our main contributions are as follows:

- We propose a simple but effective noise-tolerant cost function that combines multi-view stereo with neural volumetric surface reconstruction methods, so their optimization is regularized by the probability volumes of MVS methods.
- We integrate neural surface reconstruction into multiple coarse-to-fine MVS models. Our method consistently improves depth estimation for better MVS performance at a faster speed.
- We evaluate our method on surface reconstruction and novel view synthesis on the DTU [1] and BlendedMVS [73] datasets. Our reconstruction is significantly better than both neural rendering and MVS models.

### 2. Related Work

#### 2.1. Multi-View Stereo (MVS)

Traditional multi-view stereo uses representations such as depth maps, point clouds, and volumetric representations [18]. Depth map based methods [5, 20, 56, 50, 67] typically rely on a reference image and additional nearby source images for depth estimation. Point cloud based methods [19, 32, 35] attempt to optimize a collection of patches that best describe a 3D scene. Volumetric methods [29, 27, 31, 51, 11, 58, 22, 52, 17, 79] often aggregate information into a global representation such as a volume or mesh.

Deep-learning MVS methods [7, 38, 68, 71, 72, 9, 77, 21, 70, 80, 66, 69, 63, 84, 86, 16] typically use depth maps as 3D representations and follow the steps below: i) they use a differentiable homography to aggregate features from nearby views and build the cost volume, ii) they use a 3D CNN to regularize the cost volume and regress the depth and finally, iii) by applying a softmax function, they obtain a probability volume from the cost volume. A winner-take-all technique is often used to determine the depth. Cascade cost volumes [9, 21, 70, 80, 16] and recurrent cost volume regularization [66, 69, 72] further reduce memory consumption. The cascade cost volume is constructed in a coarse-to-fine manner that first regresses a coarse depth in low resolution and then predicts finer depth values in higher resolution based on the depth range inferred from the coarse result.

MVS explicitly forces surface points to be visible from multiple views. This property prevents degenerate geometry in the case of sparse input views. However, the correspondence problem is often hard to solve, which introduces significant noise in the predicted geometry. Furthermore, the use of the $\text{argmax}$ operation (winner-takes-all) removes potentially correct predictions in the MVS probability volume and introduces further noise. Thus, we propose to directly use information from the probability volume instead of the noise-prone MVS point estimates.

#### 2.2. Neural Volumetric Representations

Neural volumetric representations are popular in 3D reconstruction [24, 75, 43, 28, 33, 61, 74, 44, 81, 12] and novel view synthesis [53, 36, 41, 48, 57, 2, 39, 45, 75]. NeRF [41], the most well-known method, is based on the volume rendering equation [40, 26] and stores 3D information inside a neural network in the form of a compact Multi-layer Perceptron (MLP). Due to the expressive power of the neural network, it is able to model high-quality details and reconstruct complex 3D structures with a relatively small storage cost. VolSDF builds on NeRF with improved volumetric rendering of implicit surfaces. However, as shown in Fig. 1, in the case of sparse input views, the quality of the VolSDF reconstruction drops significantly because of the radiance-ambiguity problem described in Sec. 1.

Regularization-based approaches are simple, but efficient ways to mitigate this problem, using priors such as smoothness [42, 44], cross-view semantic similarity constraint [23], normal priors [60], and depth priors [47, 65]. DS-NeRF [14] utilizes estimated depth from structure-from-motion [49]. MonoSDF [78] and SparseNeRF [59] utilize monocular depth estimation. Monocular depth estimation is often not accurate, only roughly approximating shapes, and may lead to sub-optimal results. Sensor depth [15, 30] and MVS [81, 87] have also been adopted to reg-
ularize the training of neural rendering models. Although MVS is a strong prior in general, it can be unreliable when the MVS prediction is noisy with sparse input views.

A different approach is to increase the generalization ability of neural rendering by utilizing priors derived from a larger model trained on multi-view image datasets [37, 6, 62, 76, 34, 10, 34, 55]. PixelNeRF [76] is conditioned on features extracted by a CNN. MVSNeRF [6] forms a neural volume from the cost volume obtained by warping image features, and is conditioned on this neural volume. IBRNet [62] aggregates features from nearby views to infer geometry and adopts an image-based rendering approach. GeoNeRF [25] utilizes a cascaded cost volume and an attention-based technique to aggregate information from different views. SparseNeuS [37] proposes cascaded geometry reasoning and consistency-aware fine-tuning. These methods considerably improve reconstruction, but our experiments show that their results still suffer from entanglement of texture with geometry, and inconsistencies between views.

Our method differs from generic neural rendering methods like MVSNeRF and GeoNeRF in that we explicitly utilize the MVS prior through noise-tolerant test-time optimization. In contrast, generic methods implicitly utilize the MVS prior by conditioning the rendering MLP on features derived from the cost volume, which may not work well in challenging sparse-input scenarios. In Sec. 4.3, we show our approach outperforms generic methods to effectively and reliably disentangle texture and geometry.

3. Method

![Image](3558)  
**Figure 2.** Our proposed method improves the quality of depth maps obtained from the coarse stage multi-view stereo (MVS) by introducing noise-tolerant optimization techniques. The resulting depth maps then guide depth hypothesis sampling in the finer stage MVS, leading to more accurate and detailed 3D reconstructions.

We propose a novel way to integrate neural volume rendering with multi-view stereo algorithms. Specifically, we adopt VolSDF [74] for the neural surface reconstruction and notice that with sparse input views, VolSDF’s reconstruction quality degrades dramatically. To mitigate this, we propose S-VolSDF that makes use of the correspondence-aware probability volume from MVS algorithms. Fig. 2 and Fig. 3 provide an overall illustration of our method.

### 3.1. Background

**Volume Rendering of Implicit Surfaces.** We use forward volume rendering [40, 26, 41] as our differentiable volumetric representation of the 3D scene and apply VolSDF [74]. VolSDF represents scene geometry as a signed distance function (SDF), which is subsequently transformed into density values for volume rendering. For each pixel, we sample points between the near and far depths along the ray $r$ and approximate the pixel color $\hat{C}$ by:

$$\hat{C}(r) = \sum_{i=1}^{N} w_i \cdot c_i,$$

where $w_i = T_i \left(1 - \exp\left(-\sigma_i \delta_i\right)\right),

T_i = \exp\left(-\sum_{j=1}^{i-1} \sigma_j \delta_j\right).

Here, $w_i$ is the rendering weight, $\sigma_i$ and $c_i$ denote the density and color at the sampled point $i$, respectively and $\delta_i$ is the distance between adjacent samples along the ray. The density value is approximated from the SDF $s$, with learnable parameter $\alpha, \beta$, as follows:

$$\sigma(s) = \begin{cases} \frac{1}{2} \exp \left( -\frac{s}{\beta} \right) \cdot \alpha & \text{if } s \leq 0 \\ \left(1 - \frac{1}{2} \exp \left( -\frac{s}{\beta} \right) \right) \cdot \alpha & \text{if } s > 0 \end{cases} \tag{2}$$

**3.2. S-VolSDF**

Implicit neural 3D representations usually require dense images, since their per-scene optimization can be seen as a trial-and-error process to determine the underlying 3D structures. Therefore, given sparse training views as supervision, neural rendering models often fit the training views flawlessly while the underlying geometry can be vastly incorrect [42]. This can be understood as a local optimum and is known as the shape-radiance ambiguity problem [82]. In Fig. 1, we demonstrate it experimentally by training VolSDF [74] using 3 views only. As shown in Fig. 1, VolSDF completely fails to estimate geometry.

We propose to combine information from MVS to make neural rendering models correspondence-aware. The challenge lies in effectively incorporating noisy MVS predictions into VolSDF [74]. We propose two steps:

**Soft Consistency.** Instead of the hard consistency constraints imposed by estimating the depth of each point, we impose soft consistency constraints by operating directly on the probability volumes: In MVS, depth maps are typically obtained by applying $\arg\max$ on the probability volume along each view direction. Then, photometric and geometric consistency checks [71] are used to filter out depth outliers before fusing the depth maps into a point cloud. $\arg\max$ works well when dense inputs are available, but in the case of sparse inputs, the correct depth is often not assigned the highest probability. As a result, incorrect depths...
Figure 3. **Overview.** We propose to use probability volumes, obtained from multi-view stereo (MVS) models, to supervise the rendering weight estimated by VolSDF [74]. We apply a soft consistency check to refine the volumes. The weight loss function ensures consistency between the probability volume and the rendering weight. This process allows us to use the reconstructed depth information provided by VolSDF to guide the depth hypothesis sampling in the MVS models, as depicted in Fig. 2.

introduced by $\arg\max$ will be filtered out by consistency checks, resulting in an incomplete reconstruction.

Alternatively, we propose directly computing consistency measures on the probability volumes. The reference view is the image, the depth of which we want to determine. The other images are the source views. By applying MVS to these views, we obtain probability volumes. Then, we multiply each probability value $P_{\text{ref}}(x)$ in the reference probability volume with 3D position $x$, with the sum of $P_{\text{src}}^j(x)$ at the same location, to compute a new consistency weighted probability volume. $P_{\text{src}}^j(x)$ is interpolated from the probability volumes of the source views. We demonstrate that this multiplication works adequately in our ablation study in Sec. 4.4. However, significant errors in depth still appear in challenging sparse-input scenarios.

**Noise-Tolerant Loss.** We further propose a noise-tolerant weight loss that utilizes the noisy probability volume to improve the reconstruction of VolSDF [74]. Given points sampled along a viewing ray, we notice that $P$ and $w$ in Eq. (1) actually have the same meaning: their normalized values along the ray/depth both form a depth probability mass function, which can also be seen as the probability that a correspondence exists. The larger the value of $w$, the more likely it is that this point is visible in multiple views (i.e. a correspondence between pixels in different images). Instead of directly checking for consistency between the different probability volumes, we use them (in the form of $w$) during neural rendering optimization. Thus, we leverage the smoothness of neural implicit models and combine the global consistency guaranteed by volumetric rendering.

Specifically, we use our consistency weighted probability volume as supervision to regularize $w$ in the volume rendering Eq. (1). Based on [46], we can think of all possible 3D points on a ray as interior or exterior to the object (i.e. a binary classification problem). Thus, the MVS probability volume becomes a set of noisy positive labels for the rendering weights (i.e. occupancy values) with confidence from soft consistency. Hence, we have a classification problem that allows the use of cross entropy loss to optimize neural rendering methods. However, as shown in [85, 54], the cross entropy loss is sensitive to noisy labels. Based on insights from [85], we adopt a generalized cross entropy loss in Eq. (3) to reduce the penalty on false positive MVS predictions and thus increase optimization tolerance to noise. The noise tolerance level can be controlled by parameter $q$, where the generalized cross entropy loss is equivalent to the cross entropy loss when $q$ approaches 0 [85], and to the Mean Absolute Error (MAE) loss when $q = 1$. Our noise-tolerant weight loss is shown in Eq. (3).

\[
L_{\text{weight}} = \sum_{x \in X} \frac{1 - w(x)^q}{q} \cdot P_{\text{ref}}^r(x),
\]

where $P_{\text{ref}}^r(x) = P_{\text{ref}}(x) \cdot \sum_j P_{\text{src}}^j(x)$.

$w(x)$ is the rendering weight predicted by the neural rendering model at the sampled location $x$ along a ray in the reference view, $P_{\text{ref}}$ is the probability volume in the reference view, and $P_{\text{src}}$ is the probability volume of a source view. In this way, we are essentially optimizing correspondences across images in a globally consistent and noise-
tolerant way. In spirit, this is similar to finding a graph-cut in a volume of correspondence costs described in [58].

Coarse-to-fine MVS Reconstruction. As shown in Fig. 2, we incorporate our method into three coarse to fine MVS models [16, 9, 21]. We use the first coarse stage MVS probability volume to guide VolSDF [74] optimization. After that, we use the depth map obtained from VolSDF and replace the original depth map estimated by the coarse stage MVS model to remove the noise in the coarse stage MVS depth. We then follow the same protocol as in coarse-to-fine MVS models: use the depth map estimated from the coarse stage to guide the sampling range of the depth candidates of the next, finer stage in MVS. Because our coarse guidance depth map is more complete and accurate, the next stage MVS depth estimation is simpler. Therefore, we only use half of the depth search width in the finer stages compared to the default search width used in MVS models. Our surface reconstruction is more complete and still accurate compared to MVS models. Furthermore, as we show in Tab. 1, our method can be effortlessly incorporated into most coarse-to-fine MVS models and achieves considerable improvement compared to standalone MVS models.

Optimization. We use the same loss functions as VolSDF [74], along with our weight loss and sparsity regularization:

$$\mathcal{L}_{\text{sparse}} = \frac{1}{\|Q\|} \sum_{r \in Q} 1/(d_r + \epsilon),$$

(4)

where $d_r$ are predicted depths and $Q$ are rays without MVS supervision ($\sum P'(x) \approx 0$). We encourage sparsity by maximizing depth values. $\mathcal{L}_{\text{sparse}}$ is only used in the first 200 steps, along with heavily Gaussian-smoothed images as photometric supervision to suppress floating surfaces.

Rendering. For coordinate-based MLPs, fitting high-frequency details and maintaining good geometry simultaneously is challenging [74]. Since our method produces reasonably good geometry, we experiment with a simple image-based rendering approach [13, 8, 3] in testing to warp nearby view pixels based on predicted depth maps to synthesize novel views. In areas where there are no valid pixels to warp (i.e., the geometric consistency check between the rendered depths of the novel view and input views fails), we use rendered colors. A 4-level Laplacian pyramid [4] is used to smoothly blend the warped pixels. Our method with image-based rendering is denoted as Ours_{IR}.

4. Experiments

We evaluate our method on complex multi-view 3D surface reconstruction tasks, using two datasets: DTU [1] and BlendedMVS [73], both featuring real objects with diverse materials captured from multiple views. We demonstrate the superiority of our approach over prior work through quantitative and qualitative evaluation (Sec. 4.3). Furthermore, we conduct extensive ablation studies to verify the effectiveness of our design choices (Sec. 4.4).

4.1. Experimental Settings

Datasets. For the DTU dataset [1], we combine the scans used in [75, 74, 76] with the ones used in conventional MVS settings [16, 71], and remove the training scans of common MVS models. Our primary experiments are on three-view 3D reconstruction. Similar to PixelNeRF [76], we use views 25, 22, and 28 for three-view reconstruction. We further test on 6 and 9 input views with consistent improvements in performance. For the BlendedMVS dataset [73], we select 9 challenge scenes, following [74]. For each scene, we select a set of sparse input views (i.e., 3 images) with a relatively wide baseline, similar to the setting in the DTU dataset. The image resolution is set to $768 \times 576$ for both the DTU and BlendedMVS datasets. We use foreground masks from [42, 75] following [42, 37] for evaluation.

Metrics. For surface reconstruction, we follow the standard evaluation protocol in [1, 75, 74] and report the Chamfer distance (in mm) of the output point clouds with ground truth point clouds. For novel view synthesis, we adopt the mean of PSNR, structural similarity index (SSIM) [64], and the LPIPS perceptual metric [83].

Implementation details. We experiment mainly using Cas-MVSNet [21] to obtain the cascade probability volume. We notice that, given only 3 input views, the default plane sweep settings (48, 32, and 8 depth hypotheses with interval widths 4, 2, and 1 respectively) do not retain fine details very well. We change them to 192, 32, and 8 depth hypotheses with intervals 1, 0.5, and 0.5 respectively. We are able to make the finer stage depth search interval widths much smaller because our method produces more complete and accurate coarse depth maps. The batch size is 512 rays. The $q$ in our weight loss Eq. (3) is 0.5 in all experiments. We optimize each scene for 100K steps. Before fusing the depth maps output by the MVS model into a point cloud, standard photometric and geometric consistency [71] checks based on probability values and depth errors are adopted.

4.2. Baselines

Neural Rendering Methods. We compare against state-of-the-art generic neural rendering methods, including IBRNet [62], MVSNeRF [6], GeoNeRF [25], and SparseNeuS [37]. We fine-tune IBRNet and SparseNeuS using three input images for each scene for 20k and 10k iterations, respectively. We only report non-fine-tuned results for MVSNeRF and GeoNeRF because our attempts to fine-tune using 3 images did not succeed due to the inherent difficulty of the task, consistent with [42, 37]. Additionally, we compare our method with per-scene optimization based neural surface reconstruction methods, NeuS [61] and VolSDF [74].

\[^1\]Results for the 6 and 9 image scenarios are in supplementary.
Table 1. Quantitative results on 3D reconstruction for the DTU dataset. “+ Ours” means that we use the cited MVS algorithm as the probability volume builder and optimize using our method. The metric is the Chamfer distance (lower is better).

<table>
<thead>
<tr>
<th>MVS Method</th>
<th>Scan 21</th>
<th>Scan 24</th>
<th>Scan 34</th>
<th>Scan 37</th>
<th>Scan 38</th>
<th>Scan 40</th>
<th>Scan 82</th>
<th>Scan 106</th>
<th>Scan 110</th>
<th>Scan 114</th>
<th>Scan 118</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBRNet_{ft} [62]</td>
<td>3.40</td>
<td>3.54</td>
<td>3.13</td>
<td>6.78</td>
<td>3.32</td>
<td>4.80</td>
<td>3.48</td>
<td>2.59</td>
<td>3.93</td>
<td>1.23</td>
<td>2.74</td>
<td>3.54</td>
</tr>
<tr>
<td>MVSNeRF [6]</td>
<td>2.07</td>
<td>2.35</td>
<td>1.23</td>
<td>3.87</td>
<td>1.36</td>
<td>2.40</td>
<td>2.23</td>
<td>1.64</td>
<td>1.76</td>
<td>0.65</td>
<td>1.86</td>
<td>1.95</td>
</tr>
<tr>
<td>GeoNeRF [25]</td>
<td>2.13</td>
<td>3.04</td>
<td>1.00</td>
<td>3.93</td>
<td>1.20</td>
<td>2.46</td>
<td>2.32</td>
<td>1.82</td>
<td>2.21</td>
<td>0.79</td>
<td>1.67</td>
<td>2.06</td>
</tr>
<tr>
<td>SparseNeuS_{ft} [37]</td>
<td>5.26</td>
<td>4.93</td>
<td>5.59</td>
<td>7.04</td>
<td>5.18</td>
<td>7.38</td>
<td>4.78</td>
<td>4.58</td>
<td>5.61</td>
<td>4.55</td>
<td>5.61</td>
<td>5.51</td>
</tr>
<tr>
<td>NeuS [61]</td>
<td>4.52</td>
<td>3.33</td>
<td>3.03</td>
<td>4.77</td>
<td>1.87</td>
<td>4.35</td>
<td>1.89</td>
<td>4.18</td>
<td>5.46</td>
<td>1.09</td>
<td>2.40</td>
<td>3.36</td>
</tr>
<tr>
<td>VolSDF [74]</td>
<td>4.54</td>
<td>2.61</td>
<td>1.51</td>
<td>4.05</td>
<td>1.27</td>
<td>3.58</td>
<td>3.48</td>
<td>2.62</td>
<td>2.79</td>
<td>0.52</td>
<td>1.10</td>
<td>2.56</td>
</tr>
<tr>
<td>TransMVSNet [16]</td>
<td>3.39</td>
<td>3.61</td>
<td>1.55</td>
<td>4.24</td>
<td>1.95</td>
<td>3.06</td>
<td>3.45</td>
<td>2.94</td>
<td>3.81</td>
<td>1.67</td>
<td>2.34</td>
<td>2.92</td>
</tr>
<tr>
<td>UCSNet [9]</td>
<td>2.57</td>
<td>3.01</td>
<td>1.82</td>
<td>4.07</td>
<td>1.62</td>
<td>3.10</td>
<td>2.49</td>
<td>1.93</td>
<td>1.27</td>
<td>0.68</td>
<td>1.59</td>
<td>2.20</td>
</tr>
<tr>
<td>CasMVSNet [21]</td>
<td>2.40</td>
<td>3.07</td>
<td>1.23</td>
<td>3.27</td>
<td>1.35</td>
<td>2.76</td>
<td>1.82</td>
<td>1.72</td>
<td>1.30</td>
<td>0.70</td>
<td>1.44</td>
<td>1.92</td>
</tr>
</tbody>
</table>

Table 1. Quantitative results on 3D reconstruction for the DTU dataset. “+ Ours” means that we use the cited MVS algorithm as the probability volume builder and optimize using our method. The metric is the Chamfer distance (lower is better).

As shown in Fig. 4, VolSDF [74] and NeuS [61] show suboptimal performance due to the weak photometric constraint in resolving the shape-radiance ambiguity. Fine-tuning SparseNeuS [37] can lead to degenerate results, especially on the BlendedMVS dataset, so we only report its performance on DTU. Fine-tuned IBRNet [62] performs worse than methods using stronger MVS priors such as MVSNeRF [6] and GeoNeRF [25]. Although MVSNeRF and GeoNeRF demonstrate impressive performance, they still fall short compared to our method (see Fig. 8).

As shown in Tab. 1 and Fig. 5, MVS models coupled with our noise-tolerant optimization perform much better than MVS models or VolSDF [74] alone. Thus, our method

with VolSDF being the neural rendering model used to refine MVS predictions in our method. For fair comparison with MVS, we only maintain the foreground depth maps generated by neural rendering techniques by applying standard geometric consistency checks and ground truth masks. We merge the depth maps into a point cloud for evaluation.

MVS Methods. To evaluate the generalizability of our method, we incorporate it into three state-of-the-art coarse to fine MVS models: CasMVSNet [21], UCSNet [9], and TransMVSNet [16]. All MVS networks are pre-trained only on DTU [1] with ground-truth depth as supervision and are frozen during per-scene optimization.

4.3. Comparisons

3D Reconstruction. Our approach surpasses state-of-the-art techniques in 3D reconstruction, as demonstrated by its superior performance on both the DTU [1] and BlendedMVS datasets [73] (Tab. 1 and Tab. 2). We show meshes extracted from neural rendering method outputs in Fig. 4 and Fig. 7.

As shown in Fig. 4, VolSDF [74] and NeuS [61] show suboptimal performance due to the weak photometric constraint in resolving the shape-radiance ambiguity. Fine-tuning SparseNeuS [37] can lead to degenerate results, especially on the BlendedMVS dataset, so we only report its performance on DTU. Fine-tuned IBRNet [62] performs worse than methods using stronger MVS priors such as MVSNeRF [6] and GeoNeRF [25]. Although MVSNeRF and GeoNeRF demonstrate impressive performance, they still fall short compared to our method (see Fig. 8).

As shown in Tab. 1 and Fig. 5, MVS models coupled with our noise-tolerant optimization perform much better than MVS models or VolSDF [74] alone. Thus, our method

3561
can be treated as a general module that can be plugged into other MVS methods and improve their performance.

With the introduction of MVS information, we enable fast per-scene surface optimization. Our output surface reconstruction after 10-15 minutes of training (on an NVIDIA A5000 GPU) is already better than the reconstruction of the fully trained VolSDF \cite{vol sdf} (typically 4-10 hours). More specifically, on DTU, we obtain 39\% better Chamfer distance over the fully-trained VolSDF after 15 minutes of optimization, with our final model achieving a 48\% improvement. Please refer to the supplementary for more details.

**Novel View Synthesis.** Our method excels at improving geometry, yet also demonstrates competitive performance in novel view synthesis (as shown in Tab. \ref{tab:3D Reconstruction} and Tab. \ref{tab:Novel View Synthesis}). Fig. \ref{fig:6} and Fig. \ref{fig:7} illustrate improved view synthesis results compared to other methods, suggesting our method’s capacity to better disentangle geometry and texture. Also, adding the image interpolation to the rendering process greatly enhances LPIPS, while slightly improving PSNR and SSIM, by incorporating more details, as demonstrated in Tab. \ref{tab:3D Reconstruction}, Tab. \ref{tab:Novel View Synthesis}, and Fig. \ref{fig:6}.

![Figure 5. Point cloud visualization on DTU. Results improve in all combinations of our method with different MVS models.](image)

![Figure 6. Our method appears to be more accurate in novel view synthesis on DTU.](image)

<table>
<thead>
<tr>
<th>Scene</th>
<th>Doll</th>
<th>Egg</th>
<th>Head</th>
<th>Angel</th>
<th>Bull</th>
<th>Robot</th>
<th>Dog</th>
<th>Bread</th>
<th>Camera</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>MVSNeRF [6]</td>
<td>5.3</td>
<td>-16.8</td>
<td>-17.7</td>
<td>38.2</td>
<td>13.8</td>
<td>11.9</td>
<td>-0.3</td>
<td>12.3</td>
<td>8.0</td>
<td>6.1</td>
</tr>
<tr>
<td>GeoNeRF [25]</td>
<td>29.3</td>
<td>21.4</td>
<td>11.5</td>
<td>37.6</td>
<td>-0.6</td>
<td>-15.1</td>
<td>13.7</td>
<td>21.4</td>
<td>11.5</td>
<td>14.5</td>
</tr>
<tr>
<td>CasMVSNet [21]</td>
<td>32.9</td>
<td>47.1</td>
<td>17.3</td>
<td>45.9</td>
<td>11.3</td>
<td>11.5</td>
<td>33.3</td>
<td>19.2</td>
<td>30.1</td>
<td>27.6</td>
</tr>
<tr>
<td>Ours</td>
<td>35.0</td>
<td>58.8</td>
<td>38.5</td>
<td>54.7</td>
<td>33.4</td>
<td>23.9</td>
<td>33.7</td>
<td>64.4</td>
<td>43.4</td>
<td>42.9</td>
</tr>
</tbody>
</table>

Table 2. BlendedMVS 3D reconstruction results. Since there are no units in BlendedMVS, we report relative improvement (in \%) over VolSDF \cite{vol sdf} in terms of Chamfer distance.
Figure 7. 3D reconstruction and novel view synthesis comparisons on BlendedMVS. Our results appear more complete and accurate.

<table>
<thead>
<tr>
<th>Method</th>
<th>PSNR ↑</th>
<th>SSIM ↑</th>
<th>LPIPS ↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBRNet [62]</td>
<td>15.71</td>
<td>0.759</td>
<td>0.295</td>
</tr>
<tr>
<td>MVSNeRF [6]</td>
<td>18.37</td>
<td>0.818</td>
<td>0.254</td>
</tr>
<tr>
<td>GeoNeRF [25]</td>
<td>19.45</td>
<td>0.837</td>
<td>0.220</td>
</tr>
<tr>
<td>NeuS [61]</td>
<td>15.34</td>
<td>0.753</td>
<td>0.313</td>
</tr>
<tr>
<td>VolSDF [74]</td>
<td>16.99</td>
<td>0.786</td>
<td>0.332</td>
</tr>
<tr>
<td>Ours</td>
<td>20.21</td>
<td>0.820</td>
<td>0.321</td>
</tr>
<tr>
<td>OursIR</td>
<td>20.58</td>
<td>0.855</td>
<td>0.157</td>
</tr>
</tbody>
</table>

Table 3. Novel view synthesis comparisons on DTU.

<table>
<thead>
<tr>
<th>Method</th>
<th>PSNR ↑</th>
<th>SSIM ↑</th>
<th>LPIPS ↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>MVSNeRF [6]</td>
<td>14.99</td>
<td>0.866</td>
<td>0.164</td>
</tr>
<tr>
<td>GeoNeRF [25]</td>
<td>17.09</td>
<td>0.886</td>
<td>0.139</td>
</tr>
<tr>
<td>VolSDF [74]</td>
<td>14.47</td>
<td>0.860</td>
<td>0.182</td>
</tr>
<tr>
<td>Ours</td>
<td>16.97</td>
<td>0.893</td>
<td>0.154</td>
</tr>
<tr>
<td>OursIR</td>
<td>17.26</td>
<td>0.906</td>
<td>0.105</td>
</tr>
</tbody>
</table>

Table 4. Novel view synthesis comparisons for BlendedMVS.

4.4. Ablation Study

We conduct ablation studies on the DTU dataset (Tab. 5). First, we show that using only the soft consistency constraints without additional optimization still improves the reconstruction result. This supports our assumption that the probability volumes contain more information than lossy depth maps obtained from an \( \text{argmax} \) operation. Second, to evaluate the effectiveness of our weight loss, we replace our loss with the mean squared error (MSE) between the reconstructed depth from VolSDF and the geo-consistency filtered depth map obtained from MVS, similar to DS-NeRF [14]. Third, replacing the probability volumes with the depth maps as input, led to worse performance. Finally, we replace our weight loss with cross entropy loss, showing that generalized cross entropy loss is indeed noise-tolerant. Due to the trade-off between accuracy and completeness in

\[ \text{We set the probability to be 1 only at the depth prediction location.} \]
point cloud filtering, we use Chamfer distance as the metric, following [75, 74]. See supplementary for more details.

<table>
<thead>
<tr>
<th>Method</th>
<th>Chamfer ↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>VolSDF [74]</td>
<td>2.558</td>
</tr>
<tr>
<td>CasMVSNet [21]</td>
<td>1.920</td>
</tr>
<tr>
<td><strong>Ours</strong></td>
<td><strong>1.320</strong></td>
</tr>
<tr>
<td>only soft consistency</td>
<td>1.711</td>
</tr>
<tr>
<td>MSE loss [14]</td>
<td>1.792</td>
</tr>
<tr>
<td>w/o probability volume</td>
<td>1.543</td>
</tr>
<tr>
<td>w/o GCE loss</td>
<td>1.534</td>
</tr>
</tbody>
</table>

Table 5. Ablation studies for the DTU dataset. All rows except the first three are our model with different ablated components.

5. Conclusions

We presented S-VolSDF, a novel approach to recover underlying geometry from sparse input views. Neural rendering optimization mainly relies on dense input images so that it can use trial-and-error mechanisms for reconstruction. Hence, its performance drops considerably with sparse inputs. We regularized the weight distribution with a refined probability volume obtained from MVS algorithms. We further made our method noise-tolerant by applying a generalized cross entropy loss. Our experiments show that our model not only outperforms neural rendering models but also significantly boosts the performance of MVS algorithms.

Discussion and Limitations. While our method is capable of refining the probability volumes of the finer stages of MVS, we notice diminishing improvement because there is not much noise left in these stages. We include an ablation study on this in the supplementary material. While neural rendering models are able to deal with non-opaque, textureless, or glossy surfaces, our introduction of MVS reduces this ability. This is an interesting area of research, particularly in the context of few-view reconstruction.
References


