VolRecon: Volume Rendering of Signed Ray Distance Functions for Generalizable Multi-View Reconstruction Supplemental Material

Yufan Ren[1]* Fangjinhua Wang[2]* Tong Zhang[1] Marc Pollefeys[2] Sabine Süsstrunk[1]
[1]IVRL IC EPFL [2]Department of Computer Science, ETH Zurich

1. Volume Rendering of SRDF and SDF

We volume render the Signed Ray Distance Functions (SRDF) based on the volume rendering theory for Signed Distance Functions (SDF) from NeuS [7]. As shown in Fig. 1, we consider multiple surface intersections (shadowed lines) along the ray with several intersection points. For SRDF, we set the surface normal vectors to be random at the intersection points (pink) since the distribution of SRDF is irrelevant to the surface normal. For SDF, we set the surfaces to be perpendicular (blue) to the ray at the intersection points. In this case, the distributions of SRDF and SDF values are the same. Therefore, the distribution of SRDF along the ray is the same as that of SDF with the surfaces perpendicular to the ray direction at the same surface intersection points. Thus we can adopt the way of volume rendering SDF [7] to volume render SRDF.

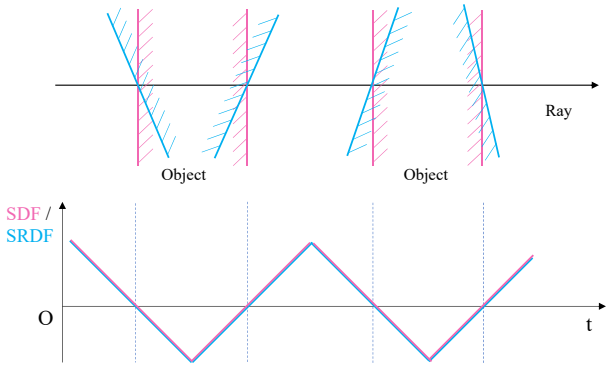

Figure 1. A horizontal ray penetrating surfaces (shadowed lines), in the case of vertical (pink) and random (blue) angle, top. SRDF (blue) is irrelevant to the incidence angle and is equal to the SDF where the surface is vertical to the ray (pink), bottom.

2. Global Feature Volume

Recall that we construct a global feature volume $\mathbf{F}_v$ to get global information. After dividing the bounding volume of the scene into K^3 voxels, we project the center point of each voxel onto the feature map of each source view and obtain the feature with bilinear interpolation. Then we compute the mean and variance of N features for each voxel and concatenate them as the voxel feature. Finally, we use a 3D U-Net [5] for regularization and get the global feature volume $\mathbf{F}_v$. The pipeline is shown in Fig. 2.

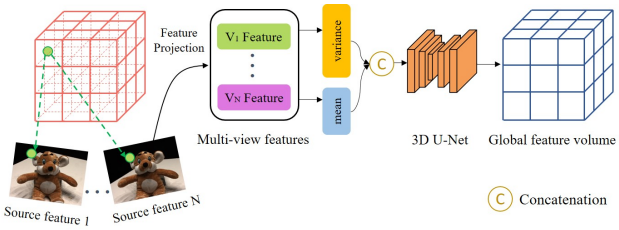

Figure 2. Pipeline of constructing global feature volume. Best viewed on a screen when zoomed in.

3. Point Cloud Reconstruction

Recall that we reconstruct the scene with both TSDF fusion [2] and point cloud fusion. For point cloud reconstruction, we follow the MVS method [9]. Before fusing the depth maps, we filter out unreliable depth estimates with geometric consistency filtering, which measures the consistency of depth estimates among multiple views. For each pixel $\mathbf{p}$ in the reference view, we project it with its depth d_0, to a pixel $\mathbf{p}_i$ in the i-th source view. After retrieving its depth d_i in the source view with interpolation, we re-project $\mathbf{p}_i$ back into the reference view, and retrieve the depth d_{reproj} at this location, $\mathbf{p}_{\text{reproj}}$. We consider pixel $\mathbf{p}$ and its depth d_0 as consistent to the i-th source view, if the distances, in image space and depth, between the original estimate and its re-projection satisfy:

$$|\mathbf{p}_{\text{reproj}} - \mathbf{p}| < \delta, |d_{\text{reproj}} - d_0|/d_0 < \varepsilon, \quad (1)$$

where δ and ε are two thresholds. We set $\delta = 1$ and $\varepsilon = 0.01$, which are the default parameters from MVSNet [9]. Finally,

*Equal contribution

we accept the estimations as reliable if they are consistent in at least N_{geo} source views.

4. Generalization on ETH3D

We compare the generalization ability of SparseNeuS [4] and our method on the ETH3D [6] benchmark. Recall that we directly test our model, pretrained on DTU [1], on ETH3D. For a fair comparison, we test the DTU pre-trained SparseNeuS [4] with the same dataset settings and point cloud reconstruction process. As shown in Fig. 6, our method reconstructs the scenes with less noise and higher completeness (fewer holes) than SparseNeuS [4]. This further demonstrates that our method has good generalization capability for large-scale scenes.

5. Baselines with Depth Supervision

Due to the ambiguity between appearance and geometry in NeRF [11], recent methods [3, 8, 10] mainly add additional 3D supervision, *e.g.* depth and normal, into baselines (VolSDF, NeuS) to compare with naive baselines (pixel color loss only).

For a fair comparison, we trained a SparseNeuS model while adding the depth loss (denoted SparseNeuS$_d$) with default settings and the same loss coefficient as ours. Besides, we remove depth loss in VolRecon and denote it as VolRecon*. VolRecon* performs slightly worse with SparseNeuS* (2.04[1] v.s. 1.96) in sparse view reconstruction. We conjecture the reason to be we not using a shape initialization as SparseNeuS [4, 7]. However, SparseNeuS$_d$ still reconstructs over-smoothed surfaces and even performs worse (4.22), Fig. 3, while ours performs better with depth supervision. Furthermore, we noticed that the grid-like pattern persists in the rendered normal map due to limited voxel resolution.

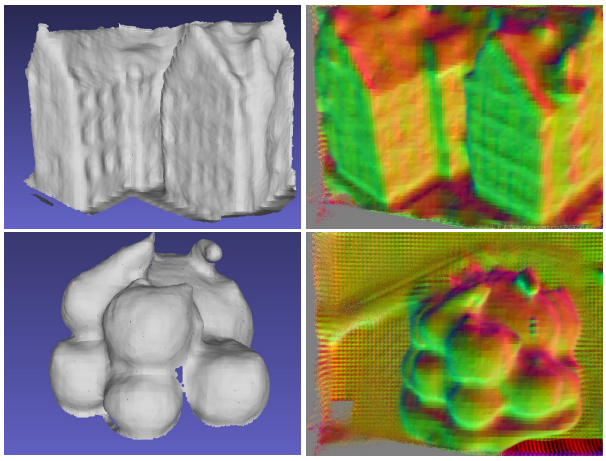

Figure 3. Visualization of reconstructed mesh and rendered normal map for SparseNeuS$_d$. Best viewed when zoomed in.

[1]Chamfer distance, the lower the better, same below

6. Novel View Synthesis of VolRecon

We report novel view synthesis results on DTU dataset [1] in Table 1, where we use the same dataset setting as full view reconstruction and render each view with 4 source views only. Qualitative results are shown in Fig. 4.

Method	PSNR ↑	MSE ↓	SSIM ↑
Ours	15.37	0.04	0.56

Table 1. Quantitative results of novel view synthesis on DTU [1]. Each view is rendered with 4 source views only.

7. Point Cloud Visualization on DTU

We visualize all the reconstructed point clouds of the full view reconstructions on the DTU dataset [1] in Fig. 5.

References

[1] Henrik Aanæs, Rasmus Ramsbøl Jensen, George Vogiatzis, Engin Tola, and Anders Bjorholm Dahl. Large-scale data for multiple-view stereopsis. *IJCV*, 120:153–168, 2016. 2, 3

[2] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Annual Conference on Computer Graphics and Interactive Techniques*, pages 303–312. ACM, 1996. 1

[3] Qiancheng Fu, Qingshan Xu, Yew-Soon Ong, and Wenbing Tao. Geo-neus: Geometry-consistent neural implicit surfaces learning for multi-view reconstruction. *CoRR*, abs/2205.15848, 2022. 2

[4] Xiaoxiao Long, Cheng Lin, Peng Wang, Taku Komura, and Wenping Wang. Sparseneus: Fast generalizable neural surface reconstruction from sparse views. In *ECCV*, pages 210–227. Springer, 2022. 2, 4

[5] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-assisted Intervention*, pages 234–241. Springer, 2015. 1

[6] Thomas Schops, Johannes L Schonberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *CVPR*, pages 3260–3269, 2017. 2, 4

[7] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *NeurIPS*, 34:27171–27183, 2021. 1, 2

[8] Yi Wei, Shaohui Liu, Yongming Rao, Wang Zhao, Jiwen Lu, and Jie Zhou. Nerfingmvs: Guided optimization of neural radiance fields for indoor multi-view stereo. In *ICCV*, pages 5610–5619, 2021. 2

[9] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *ECCV*, pages 767–783. Springer, 2018. 1

Figure 4. Visualization of novel view synthesis. Best viewed when zoomed in.

Figure 5. Point cloud visualization of the full view reconstructions on the DTU dataset [1]. Best viewed on a screen when zoomed in.

[10] Jingyang Zhang, Yao Yao, Shiwei Li, Tian Fang, David McKinnon, Yanghai Tsin, and Long Quan. Critical regularizations for neural surface reconstruction in the wild. In *CVPR*, pages 6270–6279, 2022. 2

[11] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *CoRR*, abs/2010.07492, 2020. 2

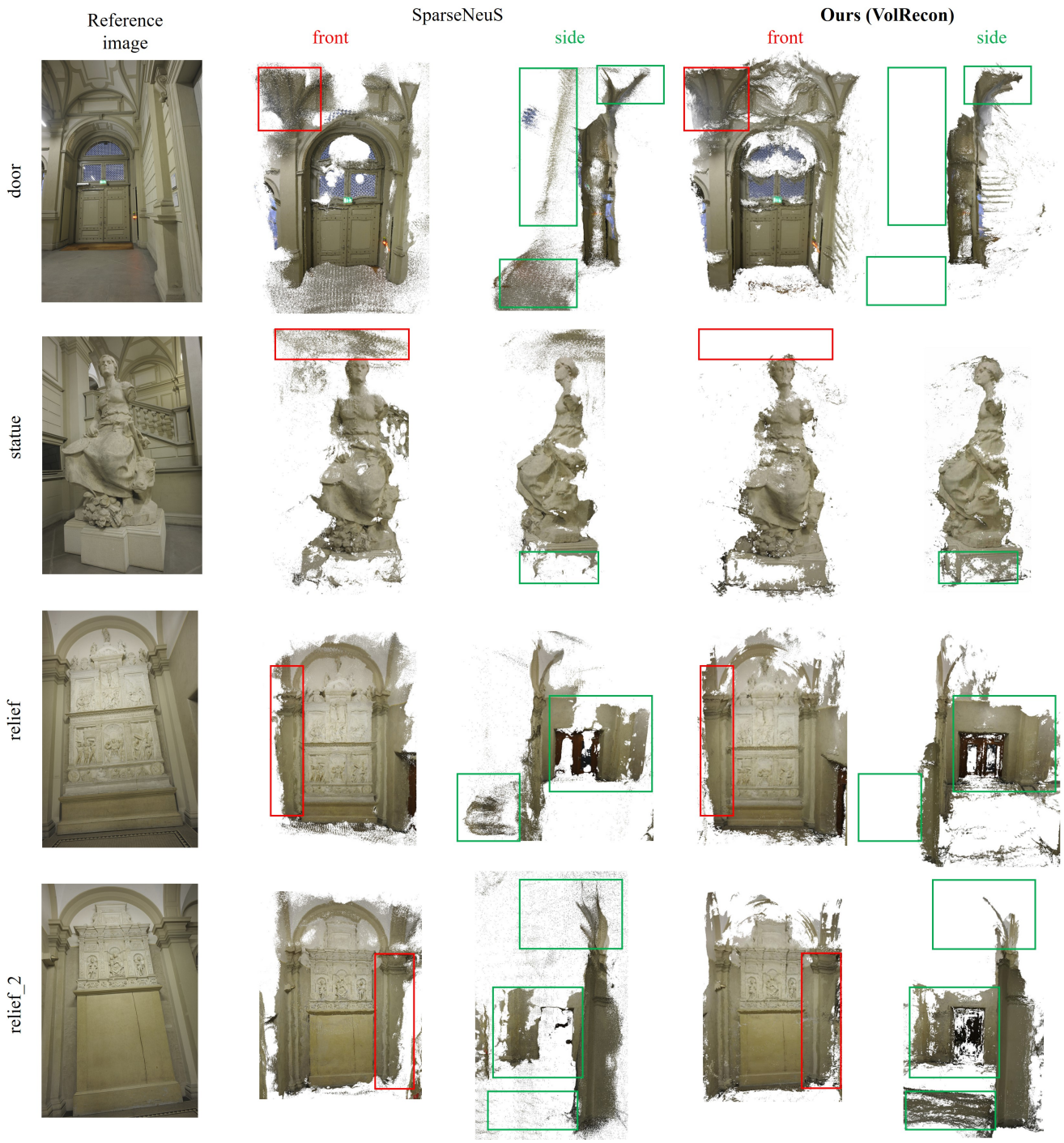

Figure 6. Point cloud visualization of reconstructions on ETH3D [6] benchmark. Compared with SparseNeuS [4], our method produces better reconstruction with less noise (*e.g.*, ground of scene *door* and top of scene *statue*) and higher completeness (fewer holes, *e.g.*, wall of scene *relief* and *relief_2*). Best viewed on a screen when zoomed in.